Université de Montréal

Traitement des données scRNA-seq issues de la technologie Drop-Seq
Application à l'étude des réseaux transcriptionnels dans le cancer du sein

Par
Marjolaine David

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de Maître ès sciences (M. Sc.)
en informatique, option intelligence artificielle

Janvier 2022

Université de Montréal
Département d'informatique et de recherche opérationnelle,
Faculté des arts et des sciences

Ce mémoire (ou cette thèse) intitulé(e)

Traitement des données scRNA-seq issues de la technologie Drop-Seq
Application à l'étude des réseaux transcriptionnels dans le cancer du sein

Présenté par
Marjolaine David

A été évalué(e) par un jury composé des personnes suivantes

François Major
Président-rapporteur

Sébastien Lemieux
Directeur de recherche

Sylvie Mader
Codirecteur

Miklós Csűrös
Membre du jury

RÉSUMÉ

Les technologies récentes de séquençage de l'ARN de cellules uniques (scRNA-seq, pour *single cell* RNA-seq) ont permis de quantifier le niveau d'expression des gènes au niveau de la cellules, alors que les technologies standards de séquençage de l'ARN (RNA-seq, ou *bulk* RNA-seq) ne permettaient de quantifier que l'expression moyenne des gènes dans un échantillon de cellules. Cette résolution supérieure a permis des avancées majeures dans le domaine de la recherche biomédicale, mais a également posé de nouveaux défis, notamment computationnels.

Les données qui découlent des technologies de scRNA-seq sont en effet complexes et plus bruitées que les données de *bulk* RNA-seq. En outre, les technologies sont nombreuses et leur nombre explose, nécessitant chacune un prétraitement plus ou moins différent. De plus en plus de méthodes sont ainsi développées, mais il n'existe pas encore de norme établie (*gold standard*) pour le prétraitement et l'analyse de ces données.

Le laboratoire du Dr. Mader a récemment fait l'acquisition de la technologie Drop-Seq (une technologie haut débit de scRNA-seq), nécessitant une expertise nouvelle pour le traitement des données qui en découlent. Dans ce mémoire, différentes étapes du prétraitement des données issues de la technologie Drop-Seq sont donc passées en revue et le fonctionnement de certains outils dédiés à cet effet est étudié, permettant d'établir des lignes directrices pour de futures expériences au sein du laboratoire du Dr. Mader.

Cette étude est menée sur les premiers jeux de données générés avec la technologie Drop-Seq du laboratoire, issus de lignées cellulaires du cancer du sein. Les méthodes d'analyses, moins spécifiques à la technologie, ne sont pas étudiées dans ce mémoire, mais une analyse exploratoire des jeux de données du laboratoire pose les bases pour une analyse plus poussée.

Mots-clés : Drop-Seq, scRNA-seq, bio-informatique, cancer du sein.

ABSTRACT

Recent single cell RNA sequencing technologies (scRNA-seq) have enabled the quantification of gene expression levels at the cellular level, while standard RNA sequencing technologies (RNA-seq, or bulk RNA-seq) have only been able to quantify the average gene expression in a sample of cells. This higher resolution has allowed major advances in biomedical research, but has also raised new challenges, in particular computational ones.

The data derived from scRNA-seq technologies are indeed complex and noisier than bulk RNA-seq data. In addition, the number of scRNA-seq technologies is exploding, each of them requiring a rather different pre-processing. More and more methods are thus being developed, but there is still no gold standard for the preprocessing and analysis of these data.

Dr. Mader's laboratory has recently invested in the Drop-Seq technology (a high-throughput scRNA-seq technology), requiring new expertise for the processing of the resulting data. In this thesis, different steps for the pre-processing of Drop-Seq data are reviewed and the behavior of some of the dedicated tools are studied, allowing to establish guidelines for future experiments in Dr. Mader's laboratory.

This study is conducted on the first data sets generated with the Drop-Seq technology of the laboratory, derived from breast cancer cell lines. Analytical methods, less specific to the technology, are not investigated in this thesis, but an exploratory analysis of the lab's datasets lays the foundation for further analysis.

Keywords: Drop-Seq, scRNA-seq, bioinformatics, breast cancer.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

CHAPITRE 1 INTRODUCTION

Le séquençage de l'ARN de cellules uniques (scRNA-seq, pour *single cell* RNA-seq), décrit pour la première fois en 2009 [1], a peu à peu gagné en popularité à mesure que les protocoles sont devenus plus accessibles. Il est aujourd'hui en plein essor : les progrès réalisés au cours des 10 dernières années ont permis le profilage des transcriptomes de cellules avec une résolution et un débit sans précédent. Ce développement récent des techniques de séquençage a révolutionné la recherche biomédicale, contribuant à une meilleure compréhension des systèmes cellulaires, notamment par l'étude approfondie des mécanismes de différenciation, ou encore par l'identification de sous-types de cellules au sein de tissus complexes.

L'approche standard du séquençage de l'ARN (dit *bulk* RNA-seq) ne peut en effet mesurer que l'expression moyenne des gènes dans un échantillon contenant des centaines de milliers de cellules, tandis que le scRNA-seq permet d'explorer les processus biologiques à l'échelle cellulaire en quantifiant l'expression des gènes dans une seule cellule.

Ces avancées dans le domaine de la transcriptomique ouvrent ainsi la porte à de nombreuses opportunités et sont d'intérêt majeur pour la recherche sur le cancer du sein menée dans le laboratoire du Dr. Mader à l'Institut de recherche en immunologie et en cancérologie (IRIC) de l'Université de Montréal, qui a récemment fait l'acquisition de la technologie Drop-Seq dans le cadre d'une collaboration avec le laboratoire du Dr. Hallet.

1.1 L'hétérogénéité transcriptionnelle du cancer du sein

Le cancer du sein, qui touche essentiellement les femmes, est une tumeur (amas cellulaire) du tissu mammaire formée par la prolifération excessive de cellules anormales au niveau des canaux galactophores et des lobules sécrétant le lait (Figure 1). Certaines tumeurs sont indolentes et se développent lentement, tandis que les plus agressives se propagent rapidement dans des tissus voisins, produisant des métastases avant même d'être détectées.

En effet, le terme générique de cancer du sein recouvre différents types de tumeurs, dont les caractéristiques histologiques et moléculaires déterminent le pronostic et appellent un traitement différent. Diverses classifications ont tenté de décrire l'hétérogénéité moléculaire du cancer du sein, dans le but de mettre en place une médecine de précision robuste basée sur des traitements ciblant les molécules produites par les différents types de tumeurs. L'une des premières classifications proposées, depuis longtemps utilisée en clinique pour orienter le choix du traitement, est basée sur la détection immunohistochimique (IHC) du récepteur à l'œstrogène alpha (ER, pour *estrogen receptor*), du récepteur à la progestérone (PR, *progesteron receptor*) et du récepteur membranaire HER2. Une tumeur sera dite « ER positive » (ER+) en présence de ER, « PR négative » (PR-) en l'absence de PR ou encore « HER2 positive » (HER2+) dans le cas d'une surexpression de HER2 (due à une amplification sur le chromosome 17 du gène codant pour HER2). Selon leur statut IHC, les cancers pourront être traités avec une thérapie ciblée ou bien une chimiothérapie : par exemple, pour les cancers de type ER/PR+ (présence de récepteurs hormonaux), dits « luminaux », une hormonothérapie pourrait être proposée au patient comme traitement adjuvant (en complément de la chirurgie). Les cancers luminaux ont en effet un meilleur pronostic que les cancers de type HER2+ ou de type ER-/PR-/HER2- dits « triple négatifs » (TNBC, pour *triple negative breast cancer*), considérés comme plus agressifs.

Avec l'arrivée des techniques de profilage du transcriptome (d'abord les micro-puces, puis le RNA--seq), de nouvelles classifications moléculaires basées sur les profils d'expression de tumeurs ont émergé. Parmi les nombreuses taxonomies proposées, 5 sous-types majeurs dits « sous-types intrinsèques » (caractérisés pour la première fois par Sorlie et al. [2]), recoupant assez bien les statuts IHC, se sont dégagés : *luminal A, luminal B, HER2+, basal-like* et *normal-like*. Le sous-type *normal-like* correspond comme les sous-types luminaux à des tumeurs ER/PR+, tandis que le sous-type *basal-like* correspond à des TNBC. D'autres sous-types plus rares ont été également été identifiés, tels que le sous-type *molecular apocrine* (mApo) ou le sous-type *claudin-low*. Le sous-type mApo correspond à des TNBC ou des tumeurs HER2+ et se caractérise par l'expression des récepteurs des androgènes (AR, *androgen receptor*) [3]. Le sous-type *claudin-low* correspond à des TNBC et se caractérise par l'expression de marqueurs de la transition épithélio-mésenchymateuse (EMT) et de marqueurs de cellules souches [4, 5].

Les différents sous-types moléculaires sont souvent identifiés au moyen d'un *clustering* hiérarchique puis des classificateurs sont développés dans l'objectif de mettre en œuvre des tests d'expression génique, capables de prédire le sous-type de nouveaux cas cliniques. En général, seule une sélection de gènes particulièrement informatifs, appelée « signature génique », est utilisée, afin de rendre les tests plus abordables et donc plus accessibles en clinique (la quantification d'un nombre limité de gènes réduisant

les coûts expérimentaux). Par exemple, le classificateur PAM50 [6] détermine, à partir d'une signature de 50 gènes, le sous-type intrinsèque – parmi *luminal A, luminal B, HER2+, basal-like* – dont le centroïde est le plus proche (corrélation de Pearson) du cas clinique testé. Ce classificateur, qui prédit également un score de risque de récidive (ROR, *risk of recidive*), est surtout utilisé à titre pronostique en clinique (test PROSIGNA®) pour prédire la récidive des tumeurs luminales et ainsi décider si une chimiothérapie doit être prescrite en complément de l'hormonothérapie. Ces classifications moléculaires, qui visent à affiner les sous-groupes IHC assez hétérogènes avec de nouveaux sous-types cliniquement pertinents (prédictifs pour le pronostic et/ou la réponse à un traitement), semblent ainsi offrir des perspectives thérapeutiques prometteuses.

Toutefois, des critiques ont souligné l'instabilité des sous-types définis [7, 8] et leur dépendance à l'égard de la cohorte étudiée ou de la signature de gènes utilisée. Ainsi, ces classificateurs sont encore peu exploités en clinique en raison de leur manque de fiabilité et de robustesse. Bien qu'également critiquée pour l'hétérogénéité biologique des différents sous-types qu'elle identifie, la classification IHC est toujours la stratégie la plus répandue en clinique.

Les travaux de ces 20 dernières années ont tout de même apporté des éléments intéressants et ouvert de nouvelles pistes de recherche pour la taxonomie du cancer du sein. Notamment, une concordance entre plusieurs sous-types identifiés dans la littérature et les stades de différenciation des cellules épithéliales mammaires a été suggérée par Prat et Perou [9], en accord avec le modèle hiérarchique de la tumorigenèse (processus de formation d'une tumeur), selon lequel les cellules cancéreuses seraient organisées hiérarchiquement comme dans les tissus normaux.

D'ordinaire, les tissus se forment par un mécanisme de différenciation cellulaire, au cours duquel une cellule souche va engendrer, par divisions cellulaires successives, une descendance de cellules filles de plus en plus spécialisées grâce à l'activation et à la désactivation de gènes régulée par des facteurs de transcription (TF). Les cellules complètement différenciées auront une fonction spécifique (qui définit le « type » cellulaire), telle que la réponse à des stimulations hormonales provoquant la sécrétion de lait pour les cellules de type luminal ou l'action contractile permettant d'éjecter le lait pour les cellules de type basal (cf. Figure 1). Une cellule souche donnée peut ainsi engendrer différents types cellulaires par différenciation, permettant la formation d'un tissu hétérogène, mais elle peut également s'auto-renouveler pour maintenir un « réservoir » de cellules souches. Le modèle hiérarchique prévoit donc qu'une petite population de cellules appelées « cellules souches cancéreuses » (CSC), capables comme les cellules souches normales d'auto-renouvellement et de différenciation, serait à l'origine de la masse tumorale. Cette dernière serait quant à elle constituée de cellules ne pouvant pas former de nouvelle tumeur mais

proliférant à un rythme plus élevé que les CSC qui l'alimentent. Les CSC, du fait de leurs caractéristiques biologiques uniques [10], pourraient contribuer à la résistance aux traitements ne ciblant que la masse tumorale.

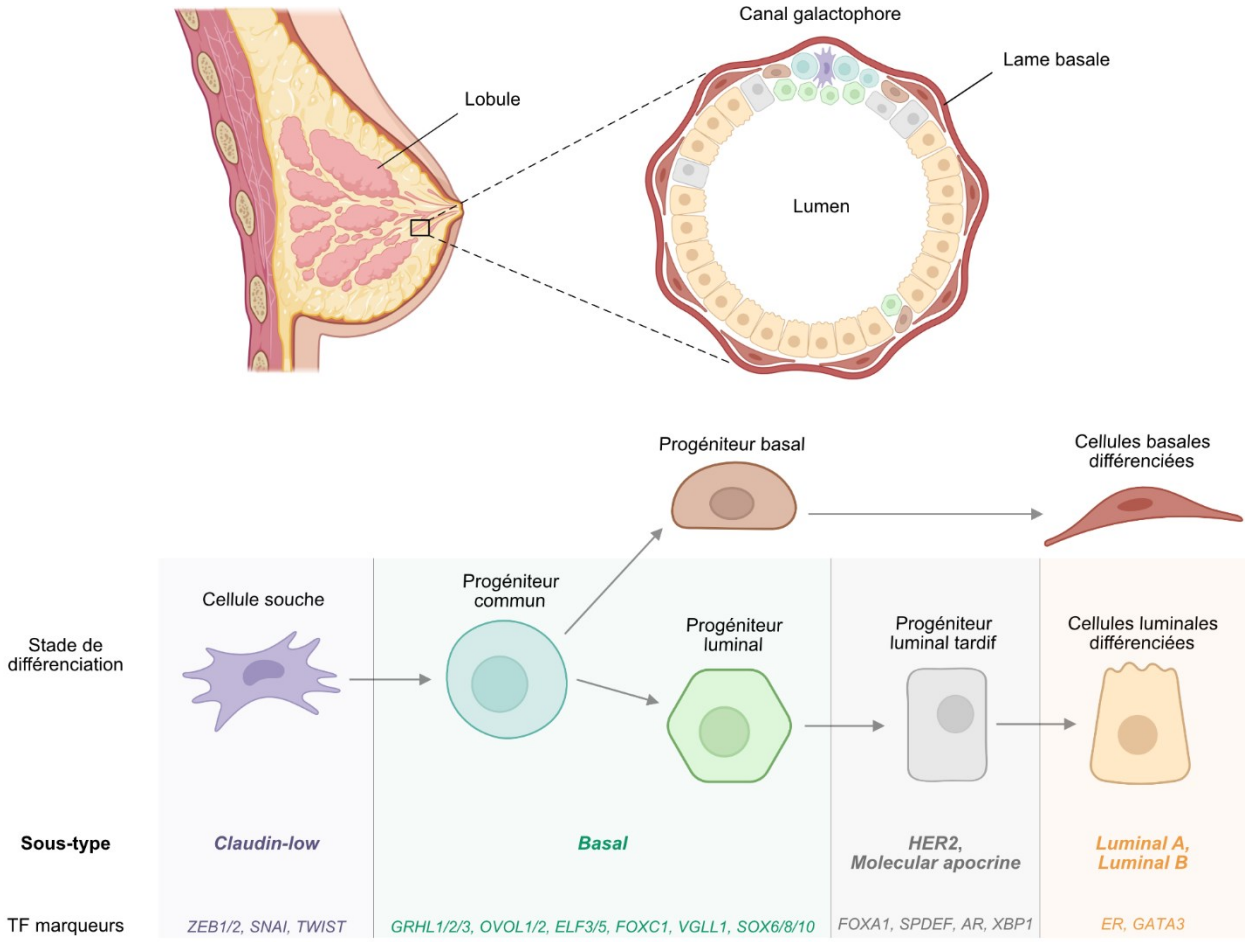


Figure 1 : Anatomie du sein et concordance entre les stades de différenciation du tissu mammaire et différents sous-types de cancer du sein.
*TF = facteur de transcription (*transcription factor*)

Le modèle hiérarchique permet ainsi d'expliquer l'hétérogénéité intra tumorale, mais Perou et al. [9] s'en servent également pour décrire l'hétérogénéité inter tumorale. Ces derniers suggèrent que les différents sous-types du cancer du sein pourraient refléter l'arrêt de la différenciation de CSC mammaires à différents stades, du fait de mutations ou autres altérations génétiques modifiant la balance prolifération/différenciation (cf. Figure 1). Ainsi, certaines tumeurs luminales correspondraient à des cellules relativement bien différenciées (e.g. cancer de sous-type luminal) tandis que d'autres, au pronostic plus sévère (avec des capacités migratoires plus grandes), seraient composées de cellules bloquées à un

stade précoce de la différenciation (e.g. cancer de sous-type basal), voire à l'état de cellule souche (e.g. cancer de sous-type *claudin-low*).

La recherche sur les CSC mammaires et sur le lien entre la différenciation mammaire, les différents sous-types et l'hétérogénéité tumorale peut avoir des impacts majeurs, ouvrant la voie à de nouvelles thérapies telles que des thérapies de différenciation cherchant à cibler les cellules souches cancéreuses en forçant la différenciation de ces dernières.

1.2 Les technologies de scRNA-seq

Le séquençage nouvelle génération (NGS, pour *next-generation sequencing*), également connu sous le nom de séquençage à haut débit, désigne les technologies de séquençage modernes comme Illumina®, Roche 454 ou encore Ion Torrent™, qui permettent de séquencer l'ADN et l'ARN beaucoup plus rapidement que les méthodes antérieures (telles que le séquençage de Sanger). La technologie NGS la plus utilisée est la technologie Illumina®, qui comporte 3 étapes : la préparation des librairies (*libraries* en anglais, aussi appelées banques en français), la génération des *clusters* d'amplification clonale par réaction de polymérase en chaîne (PCR, pour *polymerase chain reaction*) puis le séquençage par synthèse (SBS, pour *sequencing by synthesis*). La préparation de la librairie, commune à toutes les technologies NGS, consiste en la fragmentation aléatoire de l'ADN suivie de la ligation d'amorces et d'adaptateurs (oligonucléotides spécifiques permettant d'enclencher les réactions en aval), ainsi que de la ligation d'un index de multiplexage sur les fragments obtenus. L'index de multiplexage, qui est un code-barres distinct pour chaque échantillon (sauf en cas de collision suite à des erreurs de séquençage), permet d'effectuer le séquençage en parallèle (on parle aussi de séquençage en *pool*) des librairies des différents échantillons.

Une fois préparés et dénaturés, les fragments simple brin sont déposés sur une lame de verre appelée *flow cell*, servant de support à l'amplification qui suit. La *flow cell* est constituée de 8 canaux (*lanes*), eux-mêmes subdivisés en centaines de tuiles (*tiles*), au fond desquelles sont attachés des dizaines de milliers d'oligonucléotides. Ces oligonucléotides sont complémentaires des adaptateurs ajoutés aux extrémités des fragments d'ADN pendant la préparation de la librairie et permettent donc d'hybrider les fragments sur les tuiles. La même quantité de fragments de chaque échantillon séquencé est déposée dans chacun des 8 canaux de la lame, de manière à éviter dans l'analyse en aval des biais liés au canal utilisé ou à la taille des librairies. Les fragments hybridés sont alors amplifiés par PCR en pont (*bridge* PCR), opération à l'issue de laquelle un fragment et ses amplicons, groupés sur la *flow cell*, forment un *cluster*. Pendant le SBS, les brins complémentaires des fragments hybridés et amplifiés sur la *flow cell* sont synthétisés et

chaque nucléotide incorporé (correspondant à un cycle de séquençage) émet un signal de fluorescence. Le signal émis étant différent pour chacun des 4 nucléotides, il est alors possible de déterminer la séquence d'ADN du fragment synthétisé. L'ensemble des signaux d'un *cluster* donné, dont les fragments sont synthétisés simultanément, constituent un signal suffisamment fort pour être détecté. Dans le séquençage dit *single end*, la synthèse s'effectue à partir d'une seule extrémité du fragment tandis que pour le séquençage dit *paired end*, la synthèse est initiée sur une première extrémité (*read* 1), puis elle est interrompue et initiée sur l'extrémité opposée (*read* 2). Le séquençage *paired end* est plus coûteux, mais largement utilisé aujourd'hui car il permet un alignement plus précis des *reads* sur le génome. Le nombre de cycles de séquençage, i.e. de nucléotides incorporés, dépendant du budget expérimental, est réparti sur les deux extrémités pour le séquençage *paired end*.

Un ensemble varié de *designs* expérimentaux peuvent être compatibles avec cette technologie de séquençage, simplement en adaptant la préparation de la librairie. Dans le cas du séquençage d'ARN (RNA-seq) notamment, ces derniers sont convertis en ADN complémentaire (ADNc) par transcription inverse (RT, pour *reverse transcription*) avant d'être fragmentés. Dans le RNA-seq standard (ou *bulk* RNA-seq), les cellules d'un échantillon sont broyées toutes ensemble et leurs ARN messagers (ARNm) totaux, purifiés avec des oligo-dT, sont préparés tous ensemble pour le séquençage. L'information sur la cellule d'origine de ces fragments sera donc perdue : un profil transcriptomique unique sera obtenu pour chaque échantillon, correspondant à la somme des profils des centaines de milliers de cellules de cet échantillon. L'objectif des technologies de scRNA-seq est donc d'adapter la préparation des librairies afin d'obtenir les profils transcriptomiques de chaque cellule au lieu d'un profil moyen pour un échantillon donné.

Les technologies de scRNA-seq les plus utilisées actuellement, 10X Chromium [11], Drop-Seq [12] et In-Drop [13] – développées respectivement en 2017, 2015 et 2015 –, reposent sur un dispositif microfluidique qui envoie des flux aqueux et huileux dans des micro-canaux pour créer une émulsion isolant les cellules les unes des autres dans des gouttelettes aqueuses (*droplets*) séparées par de l'huile. Les cellules sont encapsulées dans les gouttelettes avec de petites billes recouvertes de fragments d'ADN synthétique, qui vont capter et étiqueter les ARNm de la cellule avec des codes-barres cellulaires (CB, pour *cellular barcode*) et des identifiants moléculaires uniques (UMI, *pour unique molecular identifier*), afin de retrouver par la suite l'identité cellulaire et moléculaire des fragments amplifiés et séquencés tous ensemble (cf. Figure 2).

Ce système de gouttelettes et de billes pour isoler les cellules et étiqueter leurs ARNm est particulièrement populaire car il permet un très haut débit (10 000 à 100 000 librairies cellulaires séquencées) grâce à l'isolation et à la préparation rapides des librairies d'un grand nombre de cellules et au multiplexage précoce (ajout du CB dès la capture des ARNm) réduisant les coûts expérimentaux. Effectivement, plus le multiplexage des librairies cellulaires est effectué tôt dans le protocole, plus le nombre de réactions chimiques (RT, PCR, séquençage…) effectuées en *pool* sera grand, et plus les coûts seront réduits. En outre, les UMI ajoutés en même temps que les CB sur les molécules d'ARNm – donc avant que la PCR n'ait lieu – permettent l'élimination des biais liés à l'amplification PCR et ainsi une meilleure reproductibilité des quantifications (on parle de la précision d'une technologie). Certains gènes sont en effet plus favorablement amplifiés que d'autre, selon par exemple leur longueur ou leur pourcentage en GC (%GC).

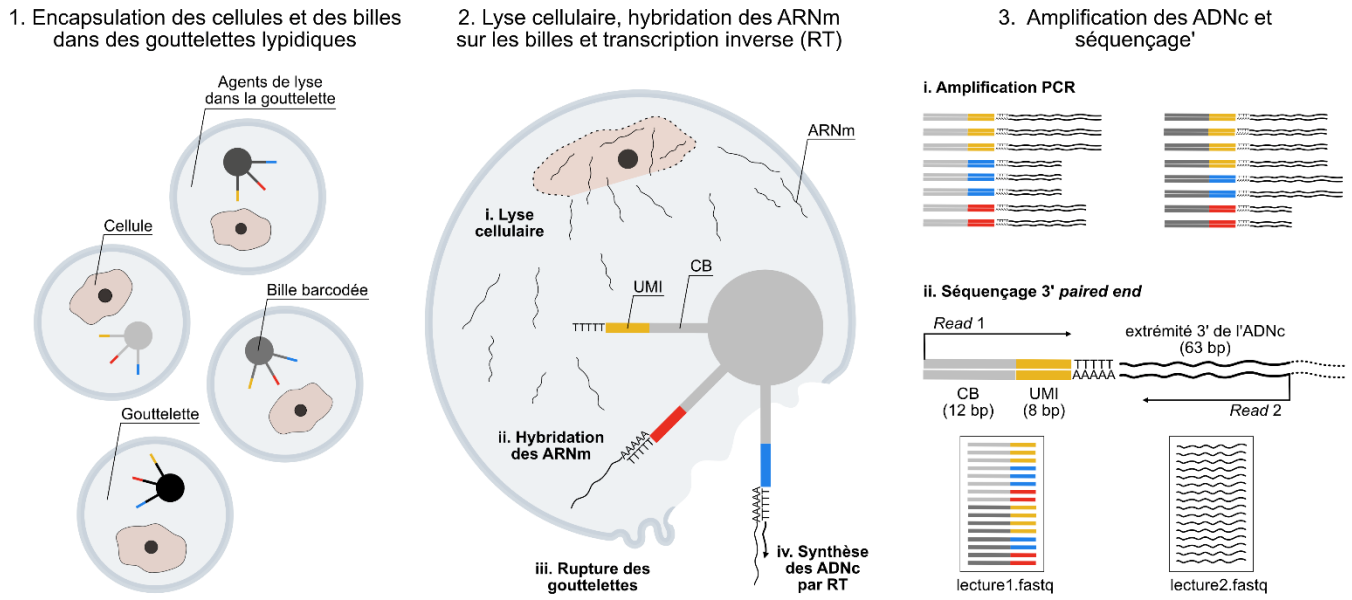


Figure 2 : Aperçu de la technologie Drop-Seq.

Grâce à un dispositif microfluidique composé de micro-canaux, les cellules seront tout d'abord encapsulées dans des gouttelettes aqueuses avec des microparticules, appelées « billes » et des agents de lyse, permettant de casser la cellule et ainsi d'en extraire les ARN messagers (ARNm). Sur chaque bille est fixée une multitude (> 10^8) de fragments d'ADN synthétique, dont les extrémités 3' libres sont des oligo-dT permettant de capturer les ARNm sur la bille suite à la lyse cellulaire. Les 20 nucléotides en amont des oligo-dT permettent d'étiqueter les ARNm et sont composés de deux types séquences : les codes-barres cellulaires (CB), constitués de 12 nucléotides (représentés par des nuances de gris sur le schéma) et les identifiants moléculaires uniques (UMI), formés de 8 nucléotides (représentés par du jaune, rouge ou bleu sur le schéma). Tous les CB d'une même bille sont identiques et diffèrent d'une bille à une autre, tandis que l'ensemble des UMI possibles (4^8) est présent sur chaque bille. Tous les ARNm d'une cellule recevront donc le même CB, mais recevront en général un UMI différent. Après l'hybridation des ARNm et la rupture des gouttelettes, les ARNm seront « convertis » en ADN complémentaire (ADNc) par transcription inverse (RT). Les billes sont traitées à l'exonucléase pour éliminer les amorces/codes-barres/oligo-dT n'ayant pas capturé d'ARNm. Les librairies cellulaires, i.e. les séquences attachées sur chacune des billes, seront amplifiées toutes ensemble par PCR, amorcée par des séquences en amont des codes-barres (non représentées sur le schéma), afin d'obtenir suffisamment d'ADN pour le séquençage qui suit. A l'issue du séquençage 3' *paired-end* en *pool* des librairies cellulaires, les *reads* (séquences lues) seront enregistrés dans deux fichiers pour chacun des échantillons : l'un contenant les extrémités 3' des ADNc (63 derniers nucléotides) et l'autre contenant les CB et UMI associés. Rectangles dans des tons gris : CB ; rectangles colorés : UMI ; lignes ondulées : ARNm ou séquences d'ADNc.

Néanmoins, ce multiplexage précoce présente également des inconvénients. Notamment, il sacrifie la couverture de séquençage, ne permettant de séquencer que l'extrémité 3' des transcrits (« séquençage 3' tag »). En effet, à la suite de la fragmentation, seuls les fragments incluant les codes-barres et l'extrémité 3' seront amplifiés et séquencés – tous les autres seront sans intérêt puisqu'ils n'incluent pas les codes-barres (cf. Figure 5). Ainsi, bien que le séquençage 3' tag soit efficace pour estimer le niveau d'expression des gènes, il ne permet pas d'estimer celui des isoformes[1], ni d'étudier de façon exhaustive les variations alléliques, contrairement aux technologies de séquençage *full-length* effectuant un séquençage des transcrits sur toute leur longueur. Il a par ailleurs été montré que les technologies *full-length*, plus sensibles, permettent la détection d'un plus grand nombre de gènes [14].

Outre les limitations en termes de couverture et de sensibilité, les technologies utilisant des gouttelettes (et en particulier la technologie Drop-Seq [12]) présentent quelques autres inconvénients. Notamment, contrairement au scénario idéal où chaque bille serait encapsulée individuellement avec exactement une cellule, il arrive par exemple que plusieurs billes se retrouvent encapsulées avec une cellule. Ces « multiplets de billes » sont surtout problématiques dans la technologie Drop-Seq, qui utilise des billes dures en polystyrène, dont l'encapsulation suit une loi de Poisson. Les technologies 10X Chromium [11] ou InDrops [13] sont moins affectées par ce problème car elles utilisent des billes en hydrogel, qui permettent de réduire la variance du nombre de billes encapsulées par gouttelette – l'encapsulation de ces dernières est dite « sub-poissonienne » –, la plupart des gouttelettes contenant alors exactement une bille. Les multiplets de billes mènent à la même situation que des erreurs d'incorporation de nucléotides au niveau des CB (qui peuvent survenir tant au cours de la synthèse des CB que du séquençage), provoquant tous deux l'association de fragments d'une seule et même cellule à différents CB. Ces erreurs d'incorporation sont en revanche plus faciles à corriger lors du prétraitement des données, car les CB utilisés dans l'expérience sont souvent référencés. C'est pour éviter les multiplets de billes, problème auquel il est difficile de remédier, que de faibles concentrations de billes sont utilisées dans la technologie Drop-seq, menant par conséquent à une quantité importante de gouttelettes sans billes et donc de cellules perdues (seulement 5% de cellules capturées).

Il arrive également que plusieurs cellules soient encapsulées avec une même bille, on parle alors de « multiplets de cellules », ce qui peut conduire à des biais importants dans les analyses en aval. Par exemple, si les cellules étudiées sont en cours de différenciation, les multiplets cellulaires risquent de

[1] Les isoformes d'un gène sont les différents ARNm distincts produits à partir de ce dernier pendant l'épissage alternatif.

regrouper des cellules arrêtées à différents stades. Ces multiplets pourraient alors être confondus avec des cellules transitant d'un stade de différenciation à un autre, puisque le profil d'expression qui en découlera sera un profil moyen de différents stades. Afin de minimiser les multiplets de cellules, qui compromettent l'analyse, de faibles concentrations de cellules sont en général utilisées dans le protocole expérimental. Ces faibles concentrations vont cependant aboutir à un nombre élevé de « gouttelettes vides », qui désignent les gouttelettes encapsulant des billes sans aucune cellule – et non pas à proprement parler de simples gouttelettes sans billes ni cellules[2]. En théorie, ces gouttelettes vides ne devraient pas poser de problème, car les codes-barres sans ADNc à la suite de la RT sont généralement retirés des billes – e.g. avec un traitement à l'exonucléase dans la technologie Drop-Seq. Cependant, des débris cellulaires contaminent la plupart des gouttelettes, de sorte que les codes-barres des billes issues de gouttelettes vides seront tout de même amplifiés et séquencés. Ces débris incluent des organites intacts, tels que des mitochondries ou des noyaux, ainsi que des ARNm. Les ARNm (dits « ambiants ») peuvent être sécrétés par d'autres cellules, mais la plupart des débris cellulaires sont libérés par la lyse d'autres cellules du même échantillon [15], qui peuvent avoir été endommagées ou stressées par le dispositif microfluidique et qui seront elles-mêmes encapsulées.

Ainsi, chaque technologie ayant ses avantages et ses inconvénients, le choix de la technologie se fait non seulement en fonction du budget, mais aussi du problème considéré. Par exemple, si l'on souhaite étudier les changements d'expression de gènes faiblement exprimés ou d'isoformes, il est préférable d'utiliser une technologie *full-length*. Si l'on souhaite caractériser les sous-types d'un tissu hétérogène, on va plutôt privilégier le haut débit des technologies à gouttelettes 3' tag pour capturer plus de cellules et ainsi détecter plus de sous-populations, certaines pouvant être assez rares. Si le budget est limité ou qu'on souhaite mettre en place des expériences sur mesure, la technologie Drop-Seq est plus intéressante, de par son protocole *open source* et flexible et son coût (0.44$/cellule, 30 000$ pour la technologie), plus abordable que celui de ses homologues 10X Chromium (0.87$/cellule, 50 000$ pour la technologie) ou InDrops (0.47$/cellule, 50 000$ pour la technologie) [16]. Cependant, si l'échantillon à étudier comporte une quantité limitée de cellules, la technologie Drop-Seq est à éviter en raison de son faible taux de capture des cellules.

Il est également possible, dans une certaine mesure, de réduire les biais liés aux différentes technologies lors du prétraitement des données, en utilisant par exemple des métriques de qualité pour éliminer les

[2] Les gouttelettes sans billes sont en effet anecdotiques dans les analyses en aval car leur contenu ne sera pas séquencé, les amorces de RT et de PCR étant situées sur les billes – en amont des codes-barres.

multiplets de cellules ou les gouttelettes ou encore, des méthodes d'imputation pour inférer les valeurs d'expression de gènes non détectés dans certaines cellules. Le prétraitement adéquat sera donc fonction de la technologie et du *design* expérimental. Par exemple, dans le cas d'une technologie *full-length* permettant d'étudier les variations alléliques, il est profitable d'utiliser un outil d'alignement base par base (tel que STAR [17]) qui fournit la position de chaque nucléotide de chaque *read*, contrairement aux outils de pseudo-alignement (tel que Salmon [18] ou Kallisto [19]), qui permettent seulement l'estimation du niveau d'expression des gènes. Ces derniers, plus rapides et moins gourmands en mémoire, seraient en revanche adaptés aux technologies 3' tag haut débit où le nombre élevé de cellules séquencées peut conduire à de larges jeux de données et ne permettant de toute façon qu'une étude très limitée des variations alléliques. Pour les technologies *full-length*, il est également nécessaire de prendre en compte dans la normalisation la longueur des transcrits, puisqu'un transcrit plus long va générer plus de fragments. Ce n'est à l'inverse pas nécessaire pour les technologies 3' tag, où un seul fragment par transcrit – celui correspondant à l'extrémité 3' – est amplifié et séquencé. Même pour les technologies à gouttelettes, le prétraitement approprié diffère d'une technologie à une autre. Par exemple, la correction des CB ne peut pas être effectuée de la même manière pour la technologie 10X Chromium que pour la technologie Drop-Seq, pour laquelle les CB utilisés dans l'expérience sont générés aléatoirement – ces dernières utilisent en outre des CB de tailles différentes.

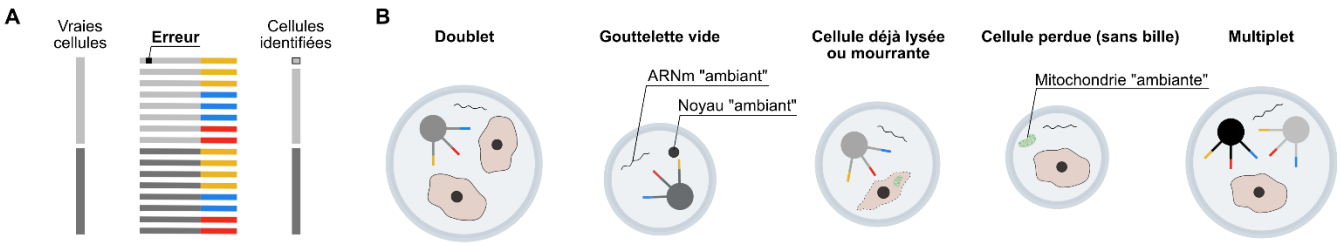


Figure 3 : Erreurs techniques pouvant survenir au cours d'une expérience Drop-Seq.

[A] Erreurs de codes-barres. Au cours de la préparation de la librairie ou encore du séquençage, des erreurs d'incorporation de nucléotides peuvent se produire. Si une erreur survient au niveau des codes-barres cellulaires (CB) ou des UMI, cela peut mener à une surestimation du nombre de cellules ou du nombre de molécules. **[B]** Problèmes d'encapsulation. Dans le scénario idéal, une bille est encapsulée avec une cellule viable dans une gouttelette. Cependant, avec le stress provoqué par le système microfluidique, certaines cellules seront lysées ou mourantes au moment de l'encapsulation. L'encapsulation se faisant de manière aléatoire, certaines gouttelettes contiendront plusieurs cellules ou plusieurs billes, ou bien seulement une bille. De l'ARNm ambiant, sécrété par les autres cellules ou relâché par des cellules lysées sera également encapsulé dans la plupart des gouttelettes. Il arrive même que des compartiments cellulaires entiers contenant des ARNm, comme la mitochondrie ou le noyau cellulaire, restés intacts après la lyse, soient encapsulés dans certaines gouttelettes. Une cellule encapsulée sans bille sera perdue, car son contenu ne sera pas séquencé – les séquences qui permettent la capture des ARNm ou encore la RT étant attachées aux billes. Rectangles dans des tons gris : CB ; ronds dans des tons gris : billes ; rectangles colorés : UMI ; petit carré noir sur un CB : erreur dans la séquence d'un code-barres ; lignes ondulées : séquences d'ARNm ambiant.

1.3 Organisation et objectifs de ce mémoire

Grâce à l'acquisition récente de la technologie Drop-Seq, des projets novateurs ont été enclenchés, tels que l'étude de la relation entre l'expression d'un facteur de transcription et celle de ses gènes cibles, ou encore l'étude de l'hétérogénéité transcriptionnelle dans des lignées cellulaires de cancer du sein. Des expériences de perturbations combinées de facteurs de transcription sont même en cours, le haut débit de la technologie et le protocole *open-source* et flexible permettant l'accès à un nombre élevé de réplicats.

Cette acquisition nécessite cependant une expertise nouvelle, les données qui découlent de la technologie Drop-Seq étant intrinsèquement différentes de celles issues d'une technologie classique de *bulk* RNA-seq. L'objectif premier de ce mémoire est donc d'apporter une meilleure compréhension de ce type de données, et de mettre en place une méthodologie computationnelle et éventuellement des recommandations expérimentales applicables à des expériences futures dans le laboratoire. Le deuxième objectif est d'explorer les régulations transcriptionnelles et l'hétérogénéité tumorale qui en découle dans le cadre des projets enclenchés mentionnés plus haut, et qui seront plus amplement décrits dans le chapitre 2.

Le chapitre 2 présentera également les différentes étapes du prétraitement des données Drop-Seq et les outils et stratégies que j'ai employés afin de générer une matrice d'expression cellules x gènes à partir de fichiers FASTQ contenant les séquences brutes.

Le chapitre 3 sera plus descriptif, correspondant à une étude des erreurs dans les séquences des codes-barres cellulaires (CB) pouvant survenir à différentes étapes du protocole expérimental.

Le chapitre 4 présentera les étapes complémentaires de prétraitement visant à « nettoyer » la matrice d'expression obtenue et les méthodes mises en œuvre avant d'effectuer une analyse des données, dont certains résultats préliminaires seront discutés dans le chapitre final.

Le chapitre final inclura également une critique de la littérature et de méthodologie choisie dans les chapitres précédents, ainsi que des lignes directrices pour guider les projets futurs du laboratoire impliquant la technologie Drop-Seq.

Dans son ensemble, ce mémoire apporte une contribution dans le domaine de la bio-informatique grâce à l'établissement d'un *workflow* computationnel pour le traitement des données Drop-Seq. Ce *workflow*, récapitulé sous forme de schéma explicatif (Figure 4), reprend les étapes et les méthodes les plus couramment mises en œuvre dans la littérature. Chacune d'entre elles est passée en revue afin de fournir une compréhension approfondie du but de chaque étape et du fonctionnement de chaque méthode, mais aussi de la nature des données. Cela a également permis d'identifier une faille dans l'une de ces méthodes, appelée Alevin, et mené à la mise en place d'une stratégie pour y remédier. En outre un ensemble de métriques pouvant être utilisées pour le contrôle de qualité des codes-barres cellulaires ont été explorées. Certaines ont même permis de caractériser des erreurs de séquençage dans les codes-barres cellulaires. Enfin, une stratégie a été proposée pour le contrôle de qualité des échantillons. Ce dernier n'est en général pas effectué dans les études présentées dans la littérature, mais peut s'avérer utile pour détecter une erreur expérimentale.

L'ensemble du code qui a permis de générer les résultats présentés dans ce mémoire est disponible sur un répertoire GitHub privé (https://github.com/Marjolaine28/BC-scRNAseq).

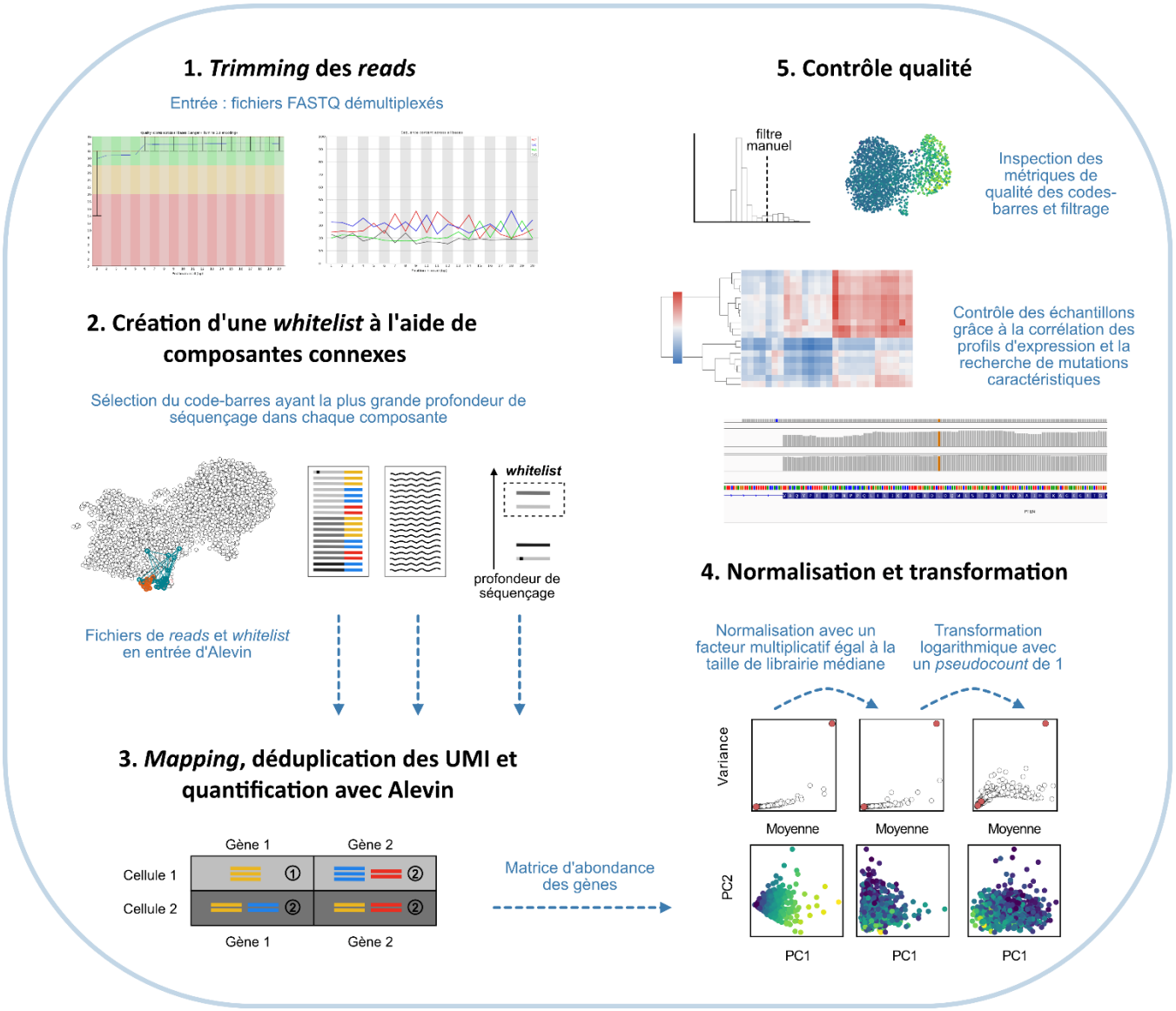


Figure 4 : *Workflow* pour le traitement des données issues de la technologie Drop-Seq

Étapes pour le traitement des données Drop-Seq décrites dans les chapitres 2 et 4. Après avoir éliminé les adaptateurs des *reads,* une *whitelist* (liste de CB exempts d'erreurs) est ensuite générée grâce à la méthode décrite dans le chapitre 2. Alevin utilise ensuite cette *whitelist* pour corriger les autres CB, puis effectue une déduplication des UMI (élimination des code-barres correspondant à des duplicatas PCR), ainsi que le *mapping* des *reads* associés aux codes-barres inclus dans la *whitelist*. La matrice d'abondance générée par Alevin est ensuite normalisée en divisant chaque valeur d'abondance par la taille de librairie (somme des valeurs d'abondance d'un code-barres) puis en les multipliant par la taille de librairie médiane. Les valeurs sont ensuite transformées avec une fonction logarithmique et un *pseudocount* de 1. Pour finir, un contrôle de qualité est effectué afin d'éliminer les codes-barres pouvant correspondre à des gouttelettes vides (pas de cellule) ou encore à des cellules cassées. Les CB indésirables sont identifiés grâce à des métriques de qualité. Un contrôle des échantillons peut également être effectué, en recherchant des mutations caractéristiques ou en calculant la corrélation de leurs profils d'expression avec d'autres échantillons de référence.

CHAPITRE 2 GÉNÉRATION D'UNE MATRICE D'EXPRESSION

Ce chapitre présentera les différents jeux de données utilisés dans le cadre de ce mémoire, ainsi que la stratégie que j'ai mise en place pour générer une matrice d'expression cellules × gènes à partir de fichiers FASTQ de données Drop-Seq.

2.1 Présentation des jeux de données

2.1.1 Données scRNA-seq issues de la technologie Drop-Seq

Objectifs expérimentaux

Dans le cadre de 3 projets menés en collaboration avec le Dr. Hallet, les librairies cellulaires de 19 échantillons ont été séquencées, réparties en 4 lots de séquençage (Tableau 1). Le projet 1, élaboré par le Dr. Hallet, a pour but d'étudier l'hétérogénéité de différentes lignées cellulaires du cancer du sein, afin d'identifier d'éventuelles sous-populations cellulaires arrêtées à différents stades de différenciation. Les projets 2 et 3, élaborés par le Dr. Mader, visent quant à eux à caractériser les régulations transcriptionnelles impliquées dans le cancer du sein, grâce à des expériences de perturbation de facteurs de transcription (TF, pour *transcription factor*) sur différentes lignées de cancer du sein.

Le projet 1 est une expérience de séquençage de 8 lignées cellulaires représentant différents sous-types du cancer du sein : MCF7, T47D, ZR751 (sous-type luminal), SKBR3 (sous-type *molecular apocrine*), BT20, HCC70 (sous- type basal), MDAMB436 et MDAMB231 (sous-type *claudin-low*). Les échantillons de la lignée MCF7 a ont été reséquencés pour ce projet car leur milieu de culture était différent de celui des autres lignées. En effet, les différentes lignées étant susceptibles de comporter des sous-populations communes (telles que des cellules souches cancéreuses), il est nécessaire de les cultiver toutes dans le même milieu pour éviter des effets de *batch.*

Le projet 2 est une expérience pilote de *knock-down* du gène ESR1, qui encode le récepteur à œstrogène alpha (ERα). ERα est un facteur de transcription dont l'activation par la liaison de l'hormone œstrogène induit la prolifération cellulaire dans les tissus mammaires. Dans cette expérience, la lignée MCF7 a été transfectée avec des petits ARN interférents (siRNA, *small interfering RNA*), qui sont des ARN pouvant empêcher l'expression du gène en clivant des ARNm auxquels ils se lient spécifiquement.

L'extinction du gène sera plus ou moins forte en fonction du nombre de siRNAs reçus, c'est pourquoi on parle dans ce cas-là de *knock-down*, et non de *knock-out* (extinction complète du gène). Les siRNA utilisés pour cette expérience ciblent donc des séquences du gène ESR1, qui correspondent à différents domaines protéiques (domaine E pour le siER#1 et domaine D pour le siER#2). Les cellules MCF7 ont également été transfectées avec un siRNA contrôle (*non-targeting* siRNA) qui n'est complémentaire à aucune séquence du génome. De ce fait, le siRNA contrôle permet de décrire les effets non-spécifiques (*off-target*) de la transfection, c'est-à-dire les éventuels changements de niveaux d'expression des gènes qui ne sont pas liés au *knock-down* de ESR1 – provenant par exemple d'une complémentarité partielle des siRNA ou encore de la procédure de transfection des cellules en tant que telle qui peut avoir un impact sur le transcriptome. Ce projet pilote a été développé avec la perspective de mettre en place des expériences de perturbations combinées de TF avec une technologie de type Perturb-Seq. Son objectif principal est donc de rechercher les paramètres expérimentaux (nombre de cellules, profondeur de séquençage, etc.) adéquats pour détecter le changement d'expression d'un TF et de ses gènes cibles dans des échantillons de cellules uniques. En effet, le Drop-Seq – et le scRNA-seq de manière générale – est nouveau dans le laboratoire du Dr. Mader, et n'a été implanté dans aucun autre laboratoire de l'IRIC. Il est donc nécessaire d'établir des lignes directrices pour assurer la robustesse des expériences à venir.

Le projet 3 est une expérience de surexpression des TF VGLL1 et GRHL2 sur la lignée MDAMB231, qui favorisent la transition mésenchymo-épithéliale (*mesenchymal-epithelial transition,* MET). Les cellules ont pour cela été transfectées avec des vecteurs viraux (pMIG), certains exprimant seulement la GFP (*empty* pMIG : cas contrôles permettant de prendre en compte les effets non-spécifiques), et d'autres exprimant VGLL1 (pMIG-VGLL1) ou GRHL2 (pMIG-GRHL2). Après transduction, les cellules ont été triées par FACS (*fluorescence-activated cell sorting*) avec des anti-GFP pour récupérer uniquement celles exprimant la GFP (i.e. ayant reçu un vecteur viral). Chaque cellule va sur-exprimer différemment GRHL2/VGLL1, en fonction du nombre de vecteurs reçus lors de la transduction et de l'endroit où s'intègre GRHL2/VGLL1 (dépendant du statut de la chromatine). L'endroit où GRHL2/VGLL1 aura également un impact plus ou moins grand sur le transcriptome : il peut par exemple s'insérer dans un gène, ou encore près d'un gène qu'il va réguler. De plus, un TF se fixant aux promoteurs de ses gènes cibles avec une affinité plus ou moins forte, il doit s'exprimer plus ou moins fortement pour activer ces derniers. Le but de ce projet est donc d'étudier la relation entre l'expression de TF – qui ne sont normalement pas ou peu exprimés dans les cellules MDAMB231 – et celle de leurs gènes cibles, qui n'est pas forcément linéaire. *A posteriori*, il a même semblé envisageable d'appliquer cette étude à toutes les lignées cellulaires séquencées, en explorant les TF intrinsèquement exprimés dans chacune d'entre elles.

Protocole expérimental et nombre de cellules attendu

Pour la technologie Drop-Seq, l'encapsulation d'une cellule et celle d'une bille sont indépendantes et suivent chacune une loi de Poisson. On peut donc estimer l'encapsulation conjointe des cellules et des billes grâce à la double loi de Poisson, qui correspond au produit de deux lois de Poisson indépendantes. On a donc $C \sim Poisson(\lambda_c)$ et $B \sim Poisson(\lambda_b)$ avec C le nombre de cellules encapsulées dans une gouttelette et B le nombre de billes encapsulées dans une gouttelette. λ_c est égal au nombre moyen de cellules par gouttelette, et λ_b est égal au nombre moyen de billes par gouttelette. Il est donc possible de contrôler λ_c et λ_b en jouant sur la dilution des cellules et des billes. Dans leur article décrivant la technologie Drop-Seq [12], Macosko et al. recommandent notamment d'utiliser de faibles concentrations de cellules et de billes afin de minimiser le nombre de doublets (plusieurs cellules ou plusieurs billes encapsulées). Pour les 19 échantillons, les concentrations recommandées par Macosko et al. ont donc été utilisées, soit 1.2 ml de solution contenant les cellules en suspension concentrées à 10^5 cellules/ml et 1.2 ml de solution contenant le tampon de lyse et les billes concentrées à 1.2×10^5 billes/ml ont donc été introduites dans deux des canaux microfluidiques. Le nombre total de cellules utilisées pour chaque échantillon, noté N_c est donc égal à 1.2×10^5 ; le nombre total de billes utilisées pour chaque échantillon, noté N_b est quant à lui égal à $1.2^2 \times 10^5$. L'huile introduite dans le troisième canal a permis la formation de gouttelettes aqueuses d'un volume ~ 1 nl (125 µm de diamètre)[3]. On peut donc estimer le nombre N_g de gouttelettes à 2 x 1.2 ml$/1$ nl, soit 2.4×10^6. On a alors $\lambda_c = N_c/\ N_g$ et $\lambda_b = N_b/\ N_g$. On peut finalement calculer le nombre attendu de gouttelettes contenant c cellules et b billes, avec $N_g \times \mathbb{P}(C = c, B = b) = N_g \times Poisson(C = c) \times Poisson(B = b) = N_g \times \frac{e^{-\lambda_c}\ {\lambda_c}^c}{c!} \times \frac{e^{-\lambda_b}\ {\lambda_b}^b}{b!}$. Ainsi, on s'attend à obtenir ~ 130 000 gouttelettes avec au moins une bille mais aucune cellule (dites « gouttelettes vides », le contenu des gouttelettes sans billes n'étant pas séquencé), ~ 7 600 gouttelettes contenant exactement une bille et une cellule, ~ 230 gouttelettes avec deux cellules et une bille (doublets cellulaires) et ~ 230 gouttelettes avec deux cellules et une bille (doublets de billes). Les autres multiplets (par exemple les triplets, ou les gouttelettes contenant à la fois deux billes et deux cellules) représentent un nombre minime de gouttelettes (~ 15). Les taux de doublets de billes et de cellules seraient donc de ~ 3% chacun, concordant avec les chiffres rapportés par Macosko et al. [12]. Ces derniers signalent

[3] Pour obtenir des gouttelettes de tailles uniformes, les membres du laboratoire du Dr. Hallet ont dû ajuster les débits des différents flux. En effet les gouttelettes obtenues initialement étaient de taille trop variable, soit trop petites pour contenir une cellule et une bille – menant à un grand nombre de gouttelettes inutiles – soit trop grandes, les rendant plus susceptibles de contenir plusieurs billes ou cellules – beaucoup de gouttelettes contenaient effectivement des doublets voire des triplets.

également qu'après les différentes étapes de rinçage et de traitement enzymatique des billes, seulement 20 à 40% d'entre elles sont récupérées. Par conséquent, ~ 3 000 codes-barres cellulaires (CB) représentant des cellules seront en théorie séquencés, ainsi que ~ 200 CB issus de doublets de billes et ~ 100 CB correspondant à des doublets cellulaires. En pratique, les membres du laboratoire du Dr. Hallet ont indiqué qu'ils récupéraient en général seulement ~ 2 000 CB représentant des cellules avec les concentrations utilisées.

Qualité des échantillons

Pour les échantillons ER1 et ER2 du projet 2 (lot de séquençage DSP779), le nombre de cellules (comptées avec un hématimètre) était plus faible ($80k$ cellules au lieu de $100k$), on s'attend donc à récupérer moins de cellules pour ces derniers.

En ce qui concerne le lot de séquençage DSP1090, les librairies cellulaires ont dû être préparées à plusieurs reprises, car trop peu d'ADNc était récupéré. Le nombre de *reads* récupérés pour ce lot suggère que le problème a persisté : en effet la somme des profondeurs de séquençage de l'ensemble des échantillons d'un lot devrait être égale à ~800 M (*flow-cell high output* de 400 M en *paired-end*), ce qui n'est pas le cas pour ce lot. Les membres du laboratoire du Dr. Hallet soupçonnent un problème lié à l'enzyme Nextera (transposase Tn5, cf. Figure 5) ou à l'amplification PCR.

2.1.2 Données de *bulk* RNA-seq

Dans mes analyses, j'ai à plusieurs reprises utilisé des jeux de données de *bulk* RNA-seq, essentiellement pour comparer les profils d'expression d'échantillons de *bulk* RNA-seq et de scRNA-seq provenant d'une même ? lignée. J'ai me suis notamment servie un jeu de données public, regroupant les données d'expression de 56 lignées cellulaires [20] du cancer du sein. J'ai également exploité différents jeux de données du laboratoire, datant de 2015 à 2021, dont les lots de séquençage sont les suivants : DSP280 (cellules de la lignée T47D, transfectées avec de petits ARN en épingle à cheveux (shRNA, *small hairpin RNA*) bloquant différents facteurs de transcription et coactivateurs transcriptionnels ou traitées avec de l'œstrogène), DSP356 (cellules de la lignée MCF7 traitées avec différents ligands, incluant de l'œstrogène et des anti-œstrogènes) , DSP550 (cellules des lignées ZR75, T47D et MDAMB436 transductées avec des vecteurs viraux contenant le gène GRHL2), DSP589 (cellules de la lignée HCC70

transfectées avec des shRNA bloquant le facteur de transcription GRHL2) et DSP1111 (cellules des lignées MCF7, T47D et ZR75 transductées avec des vecteurs viraux contenant le gène CBP).

Lot de séquençage	Projet	Nom de l'échantillon	Index de multiplexage	Lignée cellulaire	Condition	Milieu de culture	Lot de billes	Profondeur de séquençage (M)
DSP762	1	MCF7-atcc1	N702	MCF7	*wt*	DMEM	12819	272
	1	MCF7-labo1	N705	MCF7	*wt*	DMEM	12819	326
	1	T47D	N707	T47D	*wt*	RPMI	12819	271
	1	ZR75	N701	ZR751	*wt*	RPMI	12819	272
DSP779	2	NT	N701	MCF7	*wt*	DMEM	12819	241
	2	CTRL	N702	MCF7	*non-targeting* siRNA	DMEM	12819	303
	2	ER1	N705	MCF7	siER#1	DMEM	12819	334
	2	ER2	N707	MCF7	siER#2	DMEM	12819	270
DSP992	1	MDAMB231	N702	MDAMB231	*wt*	RPMI	030619	229
	1	BT20	N701	BT20	*wt*	RPMI	030619	150
	1	MDAMB436	N705	MDAMB436	*wt*	RPMI	030619	201
	1	SKBR3	N712	SKBR3	*wt*	RPMI	030619	177
	1	HCC70	N703	HCC70	*wt*	RPMI	030619	218
DSP1090	3	PMIG2	N701	MDAMB231	*empty* pMIG	RPMI	071219	42
	3	VGLL1	N705	MDAMB231	pMIG-VGLL1	RPMI	071219	42
	3	GRHL2	N712	MDAMB231	pMIG-GRHL2	RPMI	101609	47
	3	PMIG1	N707	MDAMB231	*empty* pMIG	RPMI	071219	48
	1	MCF7-labo2	N702	MCF7	*wt*	RPMI	071219	65
	1	MCF7-atcc2	N703	MCF7	*wt*	RPMI	071219	83

Tableau 1 : Récapitulatif des 19 échantillons préparés et séquencés suivant le protocole Drop-Seq.

2.2 Démultiplexage des échantillons

La première étape dans le traitement des données RNA-seq consiste à convertir les fichiers de données brutes obtenus en sortie du séquenceur Illumina du format binaire BCL (*binary base call*) au format texte FASTQ, qui récapitule l'ensemble des séquences lues et leurs scores de qualité, appelés Q-scores. Les Q-scores sont générés pendant le séquençage par le logiciel d'analyse en temps réel (RTA, pour *real time*

analysis) intégré dans le séquenceur, qui se base pour ce faire sur la qualité du signal fluorescent. Les Q-scores sont en général basés sur le système *Phred*, où un score de $10x$ indique une probabilité d'identification incorrecte du nucléotide de $1/10^x$. Le démultiplexage est souvent effectué en même temps que la conversion en FASTQ, où les séquences associées à un index de multiplexage donné, i.e. issues du même échantillon, seront écrites dans le même fichier FASTQ en sortie. L'index lu pour chaque séquence est comparé avec les index utilisés pour l'expérience, en permettant un certain nombre de *mismatches*, pour prendre en compte les erreurs d'incorporation de nucléotides qui peuvent survenir lors de l'amplification ou du séquençage de l'index.

Tous les échantillons du laboratoire ont été démultiplexés par les membres de la plateforme informatique avec l'outil bcl2fastq distribué par Illumina. Selon les index présents sur une même *flow-cell* (connus), $n = 0$, 1 ou 2 *mismatche*s ont été tolérés tant qu'il n'y avait pas d'ambiguïté possible – i.e. tant que les index de chaque paire d'échantillons avaient moins de $2n + 1$ bases différentes.

2.3 Contrôle qualité et correction des données brutes

2.3.1 Évaluation de la qualité des *reads* avec FASTQC

Tout comme dans le *bulk* RNA-seq, les *reads* générés en scRNA-seq sont soumis à un contrôle qualité avant tout autre traitement. L'outil FASTQC [2] génère dans un rapport un ensemble de visualisations permettant de contrôler différentes métriques et de détecter ainsi d'éventuels problèmes dus à la préparation de la librairie ou au séquençage.

Il donne notamment une information sur la distribution par position des Q-scores (scores de qualité) des séquences d'un fichier de *reads*, qui estiment la probabilité d'erreur d'identification d'un nucléotide (plus le Q-score est bas, plus la probabilité d'erreur est élevée). Cette information peut par exemple révéler qu'un canal, voire un lot de séquençage entier, est inutilisable. Elle peut aussi être utile pour identifier des sous-séquences de basse qualité, qui pourront ensuite être éliminées par une méthode de *trimming* (cf. section 2.3.2). En examinant les rapports de qualité générés par FASTQC, je n'ai pas noté de Q-scores critiques nécessitant d'exclure un canal ou un échantillon. J'ai cependant observé des scores plus bas pour certaines positions des *reads forward* – en particulier la première position –, ce qui pourrait s'avérer problématique puisqu'ils correspondent aux séquences des codes-barres. J'ai également pu observer de manière systématique – pour tous les *reads forward* et *reverse* de tous les échantillons – des Q-scores légèrement plus bas en début de séquence, précisément pour les 5 premières positions. En examinant les

rapports FASTQC d'autres expériences effectuées à l'IRIC – de *bulk* RNA-seq notamment, j'ai pu constater que toutes les données de séquençage obtenues avec l'instrument NextSeq 500 étaient caractérisées par des scores de qualité plus bas pour les 5 premiers nucléotides. Ces derniers sont probablement un artéfact du logiciel intégré qui assigne toujours des Q-scores plus bas pour les 5 premières positions des séquences [3, 4], du fait que la calibration du signal (*template generation*) s'effectue durant les 5 premiers cycles de séquençage pour NextSeq 500 [5]. D'autre part, j'ai également remarqué une baisse systématique des Q-scores en fin de séquence. Ce phénomène est bien connu [6] et caractéristique des technologies Illumina qui reposent sur un système de séquençage par synthèse. Il est en effet fréquent que la qualité du séquençage diminue dans les derniers cycles en raison d'une désynchronisation croissante du processus de synthèse (incorporation de nouveaux nucléotides) au sein des *clusters*. Il arrive par exemple que la synthèse d'une séquence reste « bloquée » au niveau d'un nucléotide (*phasing)*, qui sera alors lu à nouveau au cycle suivant. Le signal envoyé pour cette séquence ne sera donc plus synchronisé avec le reste du *cluster* dont elle fait partie. De telles anomalies sont rares, mais s'accumulent, de nouvelles séquences étant touchées à chaque nouveau cycle et polluant de plus en plus les signaux des *clusters*. Ainsi, j'ai pu noter que les *reads* du second fichier (extrémités 3' des ADNc), qui sont plus longs, sont plus touchés par cette baisse de qualité en fin de séquence.

D'autres métriques générées par FASTQC peuvent être utiles pour détecter des taux de nucléotides anormaux ou des sous-séquences sur-représentées, qui sont souvent le signe de contaminations ou de biais dans la librairie. Un pourcentage en GC (%GC) différent du taux attendu pour les ARNm humains peut par exemple indiquer la présence d'un autre transcriptome dans la librairie (e.g. contamination bactérienne) ou encore d'ARN ribosomaux (qui ne sont pas censés être quantifiés). Mais la plupart du temps la contamination provient des adaptateurs qui sont ajoutés aux extrémités des fragments d'ADNc pour amorcer les réactions chimiques de séquençage en 5' sur chaque brin.

Après la RT, les librairies d'ADNc vont en effet subir une fragmentation aléatoire par la transposase Tn5, et les fragments en 3' – comprenant les codes-barres et l'extrémité 3' de l'ADNc – seront amplifiés (Figure 5). Cependant, il arrive que la transposase « coupe trop », et que la longueur des fragments 3' d'ADNc soit alors inférieure au nombre de nucléotides qui seront lus pendant le séquençage. Dans cette situation, le *read reverse* peut alors inclure en 3' la queue poly-A de l'ADNc, voire les codes-barres ou l'adaptateur de l'extrémité opposée. Si la transposase coupe jusqu'au niveau des codes-barres, aucune portion de l'ADNc ne sera incluse dans le *read reverse* et le *read forward* lui-même contiendra l'adaptateur de l'extrémité opposée. De telles paires de *reads*, composées uniquement d'adaptateurs et ne contenant aucun insert sont communément appelées des dimères d'adaptateurs. Il y a plusieurs façons de

repérer ce type de contamination dans les *reads*, notamment avec leur %GC ou leur pourcentage nucléotidique (%nt) par position, qui seront influencés par les séquences contaminantes récurrentes. Ces séquences peuvent même être identifiées par FASTQC, qui va rechercher des séquences sur-représentées (d'au moins 20 bp et avec au maximum un *mismatch*) dans les *reads* et les comparer à un ensemble d'adaptateurs couramment utilisés.

J'ai ainsi pu détecter des contaminations par des adaptateurs et des queues poly-A grâce aux différentes métriques de qualité calculées par FASTQC. Notamment, certains échantillons présentaient un %A plus élevé et croissant à chaque position, provenant vraisemblablement des queues poly-A des ADNc séquencées lorsque l'insert est trop court. J'ai également pu observer pour un échantillon une composition nucléotidique prédominante à chaque position, ainsi qu'un pic marqué dans la distribution des %GC, qui sont typiques d'une contamination par des dimères d'adaptateurs (Figure 6). Aucun adaptateur connu n'a été identifié par FASTQC parmi les *reads* des 19 échantillons que j'ai contrôlés, mais des séquences sur-représentées – correspondant à l'adaptateur Nextera ME (Mosaic End) – ont été retrouvées dans la plupart des fichiers de *reads forward* et de *reads reverse*, correspondant respectivement à l'adaptateur Nextera ME et à l'amorce PCR 1 (*PCR primer 1*). Les efforts consentis par le laboratoire Teichmann pour récapituler la structure de la librairie Drop-Seq [7] – et de diverses autres librairies de *single cell* [8] – m'ont beaucoup aidée à identifier les adaptateurs contaminants et à mieux comprendre les différentes étapes de la construction de la librairie, que j'ai à mon tour récapitulées dans la Figure 5.

Le pourcentage nucléotidique (%nt) par position a également révélé certains biais dans les librairies. J'ai pu noter tout d'abord des compositions de nucléotides prédominantes pour les 14 premières positions des *reads reverse*, correspondant à un biais présent dans la plupart des librairies de RNA-seq. Ce biais vient de la fragmentation avec la transposase Tn5, qui devrait couper l'ADNc en 3' de manière aléatoire mais coupe semble-t-il préférentiellement certains motifs [9].

J'ai également noté que dans les *reads forward,* la composition nucléotidique des codes-barres cellulaires (12 premières positions) était légèrement différente à chaque position, tandis que la composition des UMI (8 dernières positions) était plus constante (lignes de %nt parallèles). Cette différence provient vraisemblablement du protocole de fabrication des code-barres, quelque peu différent pour les code-barres cellulaires et les UMI. En effet, les codes-barres cellulaires sont générés avec une méthode de *split-pool*, où les billes sont divisées aléatoirement en 4 groupes égaux recevant chacun un nucléotide différent, puis regroupées, et ce 12 fois. Ainsi, le pourcentage de billes recevant un certain nucléotide va varier légèrement à chaque série de *split-pool*, selon la répartition des billes. En revanche pour les UMI, toutes les billes sont incubées ensemble avec les 4 nucléotides en concentrations égales et

soumises à huit cycles de synthèse, ce qui explique la composition plus régulière des différents nucléotides à chaque position.

De plus, j'ai constaté des biais dans la composition nucléotidique des *reads forward* au niveau des codes-barres cellulaires et des UMI. Notamment pour un lot de billes (12819, utilisé pour les lots de séquençage DSP762 et DSP779), la première position des codes-barres cellulaires présentait un %T plus faible. Ce biais pourrait provenir d'une division inégale des billes lors de la première série de *split-pool*. Cependant, ce biais n'était pas présent dans la composition nucléotidique calculée non plus sur l'ensemble des *reads*, mais sur l'ensemble des codes-barres cellulaires (un seul *read* par code-barre cellulaire). Une autre explication pourrait être que la ligation pendant la première série de *split-pool* était moins efficace pour le groupe de billes recevant un T, ces dernières se retrouvant *in fine* avec moins de codes-barres (donc moins de *reads*). J'ai aussi remarqué que le %G des *reads forward* au niveau de l'UMI (8 dernières positions) était nettement plus grand et croissant à chaque position. Une étude a en effet démontré un biais de queue poly-G dans les UMI de certains lots de billes Drop-Seq [10].

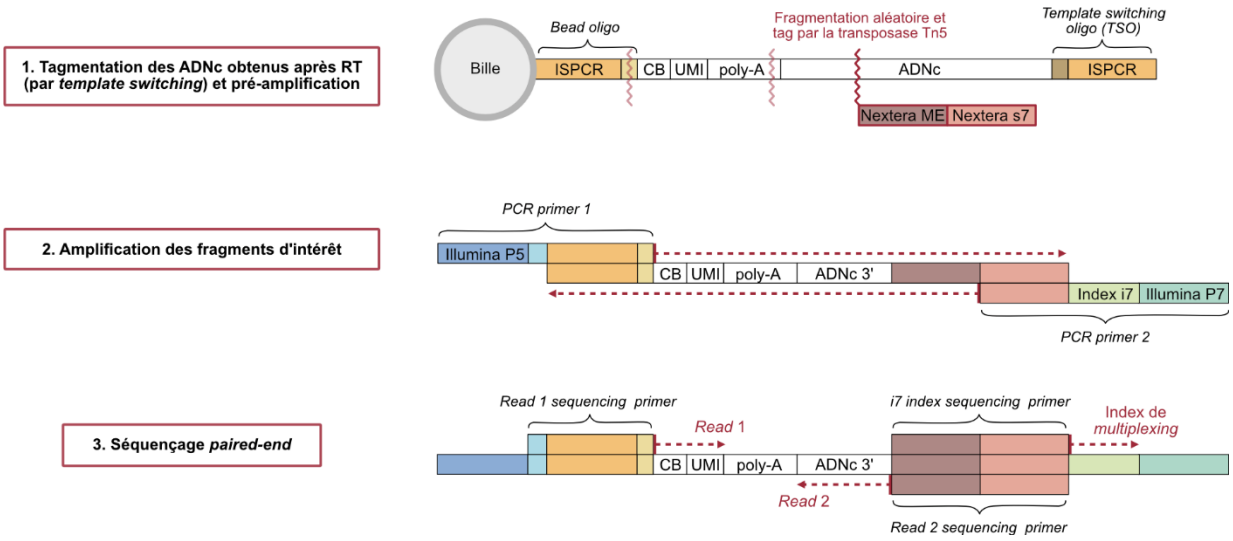


Figure 5 : Structure des librairies Drop-Seq.

Dans la technologie Drop-Seq, les librairies d'ADNc sont préparées pour un séquençage *paired-end*, de manière à séquencer les codes-barres pendant le séquençage *forward* et les ADNc pendant le séquençage *reverse*. Pendant la préparation de la librairie, la fragmentation des ADNc est effectuée avec la transposase Tn5, qui va couper aléatoirement les séquences constituées d'adaptateurs flanquant les ADNc obtenus après RT par *template switching* et pré-amplification par *in situ* PCR (ISPCR). Cette réaction est appelée « tagmentation », car la transposase va fragmenter les ADNc et les « tagger » en ajoutant des adaptateurs aux extrémités coupées. Après la tagmentation, seuls les fragments en amont de la séquence initiale (i.e. ceux qui étaient directement attachés aux billes) sont amplifiés. Ces fragments incluent, pour la plupart, les séquences des codes-barres, qui doivent permettre de retrouver leur identité cellulaire et moléculaire après le séquençage en *pool*, et l'extrémité des ADNc qui correspondait initialement à l'extrémité 3' des ARNm. Cependant certains d'entre n'incluent pas ou peu d'ADNc, car la transposase a « trop » coupé. La queue poly-A, les code-barres ou encore l'amorce PCR pourraient alors être lus pendant le séquençage *reverse*. Il arrive même que les codes-barres soient tronqués pendant la tagmentation : les adaptateurs ajoutés par la transposase, accolés aux codes-barres, pourraient alors être lus pendant le séquençage *forward*. Les autres fragments générés sont ignorés, car ils n'incluent pas de séquences de codes-barres. Pendant l'amplification, une séquence appelée *index* est également ajoutée, différente pour chaque échantillon, pour permettre le *de*multiplexage, qui vise à retrouver l'échantillon d'origine de chaque fragment après le séquençage en *pool* des librairies de tous les échantillons (appelé multiplexage). Lignes rouges en zigzag : fragmentations possibles par la Tn5 ; rectangles de couleur : adaptateurs.

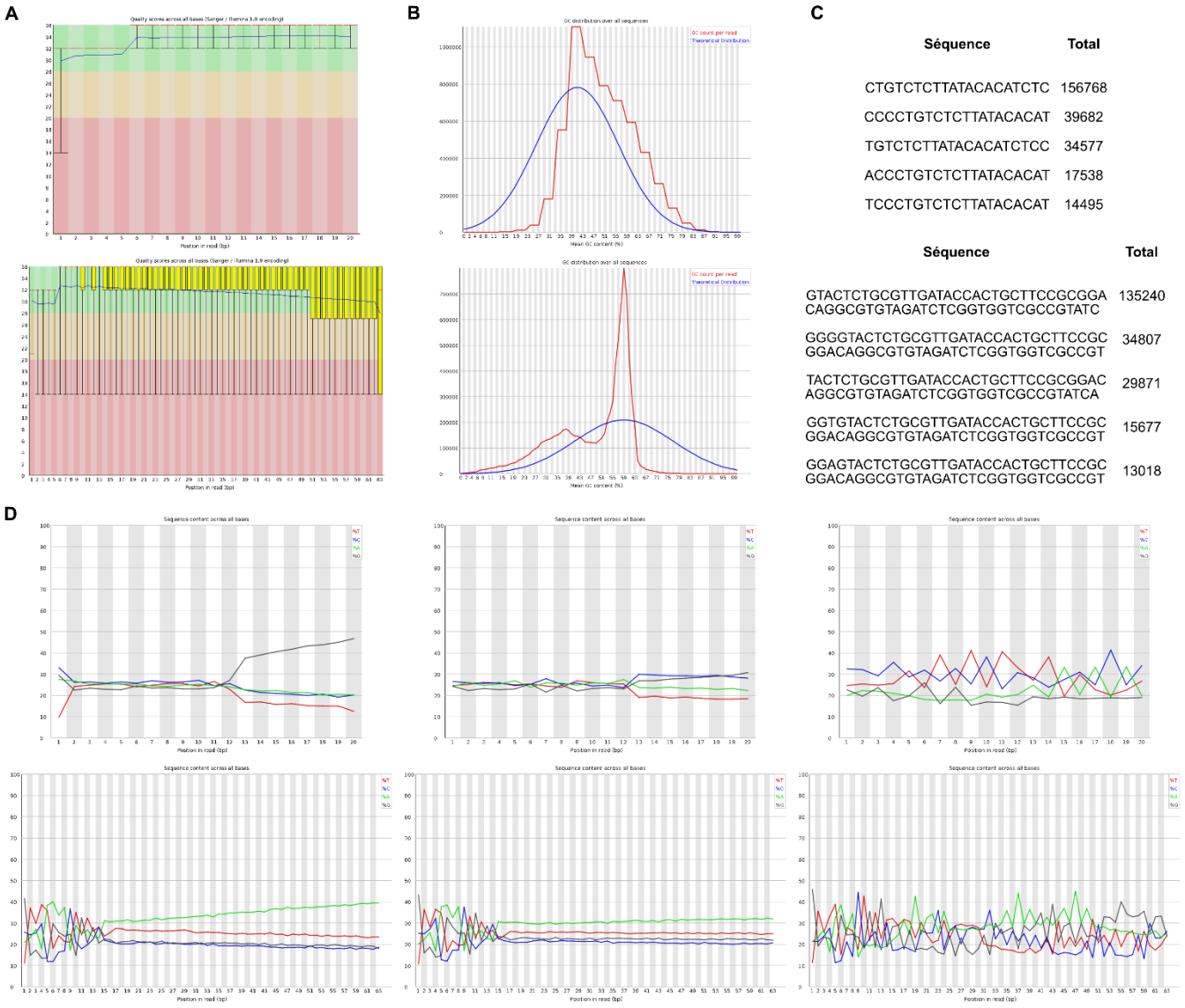


Figure 6 : Métriques rapportées par FASTQC.

[A] Q-scores par position pour les *reads forward* (haut) et *reverse* (bas) de l'échantillon NT (lot de séquençage DSP779). Tous les échantillons de l'ensemble des lots de séquençage présentent un profil de qualité semblable. Rouge : q-scores entre 0 et20 ; orange : q-scores entre 20 et26 ; vert : q-scores entre 26 et 36. **[B]** Distribution des %GC des *reads forward* (haut) et *reverse* (bas) de l'échantillon GRHL2 (lot de séquençage DSP1090), représentée par la courbe rouge. Les librairies de ce lot de séquençage sont particulièrement contaminées en concatémères d'adaptateurs, ce qui affecte la forme de la distribution des %GC. La courbe bleue représente la distribution théorique correspondant à une loi normale centrée sur le %GC moyen des *reads* (courbe bleue). **[C]** Top 5 des séquences les plus sur-représentées. Pour les *reads forward* (haut), ces séquences proviennent de l'adaptateur Nextera ME et pour les *reads reverse* (bas), elles proviennent de l'amorce PCR 1. **[D]** Composition nucléotidique par position des *reads forward* (haut) et *reverse* (bas) des échantillons NT (gauche), BT20 (milieu) et GRHL2 (droite). L'échantillon NT présente des biais au niveau des UMI dans les *reads forward* (c'est le cas pour tous les échantillons préparés avec le même lot de billes) et une contamination par des queues poly-A dans les *reads reverse*. Pour l'échantillon GRHL2 (ainsi que tous les autres échantillons du même lot de séquençage), la composition nucléotidique est typique d'une forte contamination en dimères d'adaptateurs. Vert : A ; rouge : T ; noir : G ; bleu : C.

2.3.2 *Trimming* des reads avec Cutadapt

Au vu des résultats du contrôle qualité des *reads*, j'ai donc décidé d'effectuer un *trimming* des *reads*, c'est-à-dire d'en extraire les séquences contaminantes (adaptateurs et queues poly-A). Il m'a paru préférable d'éliminer les séquences d'adaptateurs retrouvées dans les *reads forward,* supposés correspondre aux codes-barres. Certaines d'entre elles, présentes dans de nombreux *reads* (Figure 6), pourraient en effet induire de faux codes-barres et compliquer les analyses subséquentes. Il est également possible que la présence d'adaptateurs ou de queues poly-A parmi les *reads reverse* incluant les séquences d'ADNc réduise les scores d'alignement de ces derniers, j'ai donc décidé de les éliminer. Il aurait aussi été possible d'utiliser l'option *–softclip* de Salmon, qui permet d'ignorer dans le score d'alignement les extrémités des *reads* qui n'ont pas été alignées.

Pour effectuer le *trimming* des *reads*, j'ai utilisé l'outil Cutadapt [21], qui va rechercher une séquence d'adaptateur spécifiée par l'utilisateur et la supprimer le cas échéant. L'utilisateur peut également spécifier des paramètres qui vont permettre par exemple d'ajuster la longueur des *matchs*, ou encore de filtrer des *reads* devenus trop courts après le *trimming*. J'ai notamment filtré les *reads* (*forward* ou *reverse*) dont la longueur était < 20 après le *trimming* – tout code-barres tronqué a ainsi été exclu.

2.4 Estimation de l'abondance des gènes

2.4.1 Présentation d'Alevin

Alevin [22] est un outil pour le prétraitement des données scRNA-seq issues de technologies utilisant des gouttelettes, incorporé dans le logiciel Salmon [18] initialement développé pour le *bulk* RNA-seq. Il est dit « de bout en bout » car il intègre l'ensemble des étapes (ii et iii, schématisées dans la Figure 7 et habituellement effectuées séparément) permettant d'estimer l'abondance des gènes dans les cellules à partir de deux fichiers FASTQ d'échantillons démultiplexés.

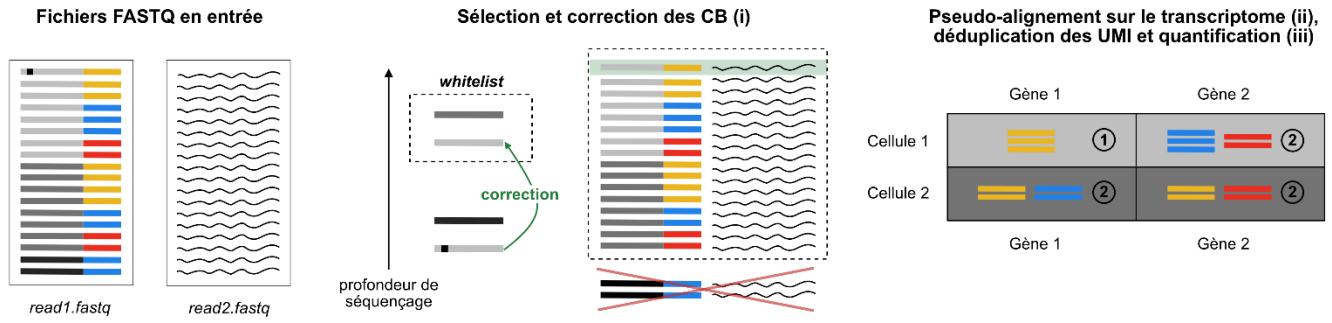


Figure 7 : Aperçu du logiciel Alevin.

À partir des deux fichiers FASTQ en entrée, Alevin estime l'abondance de chaque gène dans chaque cellule récapitulée sous forme de matrice *cellules* x *gènes* en sortie. Pour cela, Alevin crée dans un premier temps une liste de codes-barres cellulaires (CB), appelée *whitelist*, constituée des CB ayant les plus grandes profondeur de séquençage. Les CB sélectionnés dans la *whitelist* sont ensuite utilisés comme référence pour corriger les autres CB qui pourraient comporter une erreur survenue par exemple lors du séquençage. Si un CB hors de la *whitelist* est voisin (distance d'édition de 1) d'un CB de la *whitelist*, il sera alors corrigé, i.e. remplacé par le CB de la *whitelist* en question. Ainsi, les séquences d'ADNc associées au CB qui a été corrigé pourront être considérés lors du *mapping*, qui n'est effectuée que pour les séquences d'ADNc associées à un CB de la *whitelist*. Après le *mapping* des séquences d'ADNc sur le transcriptome, Alevin effectue une déduplication des UMI, en utilisant les séquences des UMI. La position des séquences d'ADNc sur le transcriptome est également utilisée afin de distinguer les séquences d'ADNc ayant le même UMI mais issues de différentes molécules – puisqu'une cellule peut contenir jusqu'à un million d'ARNm et que seulement 4^8 UMI peuvent être encodés. Ainsi, toutes les séquences associées au même CB et au même UMI, et placées sur le même gène compteront pour 1 dans l'estimation de l'abondance de ce gène. Une étape optionnelle, appelée *final whitelisting*, classifie ensuite, à partir de la matrice obtenue, les CB en CB de bonne ou mauvaise qualité ; cette étape n'est pas représentée dans le schéma. Rectangles dans des tons gris : CB ; rectangles colorés : UMI ; petit carré noir sur un CB : erreur dans la séquence d'un CB ; lignes ondulées : séquences d'ADNc ; rectangles en pointillés : *reads*/CB pour lesquels le *mapping* sera effectué ; croix rouge : CB ignorés pendant le *mapping* même après correction des CB ; rectangle vert clair : reads récupérés grâce à la correction des CB ; chiffres entourés : nombre de molécules uniques comptées pour chaque gène dans chaque cellule, rapporté dans la matrice en sortie.

La première étape effectuée dans Alevin consiste à créer une *whitelist*, que les auteurs définissent dans leur article comme une liste de « CB représentant des cellules correctement capturées et étiquetées » [22]. En d'autres termes les auteurs cherchent, grâce à une procédure qu'ils appellent *whitelisting*, à se débarrasser des CB représentant des gouttelettes vides. Pour cela, Alevin retient dans la *whitelist* les CB avec les plus grandes profondeurs de séquençage (nombres de *reads* associés à chaque CB), étant donné que les CB étiquetant des cellules sont exposés à un plus grand nombre d'ARNm que ceux issus de gouttelettes vides, exposés à seulement quelques ARNm ambiants. Il est fréquent, lors du prétraitement des données de scRNA-seq issues d'une technologie utilisant des gouttelettes, d'appliquer un tel seuil afin d'éliminer les gouttelettes vides. Cependant, ce seuil est plus souvent appliqué sur les tailles de librairie (nombre d'UMI) des CB que sur les profondeurs de séquençage.

Idéalement, le seuil serait déterminé selon le nombre de cellules qui ont été isolées au cours de l'expérience : par exemple, si l'on sait qu'exactement 3 000 cellules ont été isolées, les 3 000 CB avec le plus d'UMI (ou de *reads*) pourraient être sélectionnés. Il est possible d'estimer le nombre de cellules isolées avec une bille, l'encapsulation de ces dernières suivant une loi de Poisson dépendante des concentrations de billes et de cellules utilisées (cf. section 2.1). Mais l'estimation obtenue reste approximative car elle ne prend pas en compte les aléas du protocole, tels que les billes perdues lors des étapes de lavage. Afin de déterminer un seuil approprié, une pratique courante est donc de rechercher un changement marqué dans la distribution des tailles de librairie (ou des profondeurs de séquençage), appelé « *knee point* ». Cette méthode a d'abord été décrite par Macosko et al. [12], qui définissent le *knee point* comme le seuil à compter duquel la somme cumulée des tailles de librairie triées par ordre croissant atteint un plateau. Dans leur article décrivant également la technologie Drop-Seq qu'ils ont développée [12], ils constatent que le seuil identifié arbitrairement en se basant sur la visualisation de cette somme cumulée (« *knee plot* ») est proche de l'estimation basée sur la loi de Poisson, démontrant la significativité de ce seuil.

Pour déterminer le seuil au-dessus duquel un CB est ajouté à la *whitelist*, le *knee point* est recherché automatiquement dans l'implémentation d'Alevin. Pour cela, un noyau gaussien est utilisé afin d'estimer la distribution des profondeurs de séquençage, dans laquelle est ensuite recherché un minimum local. La recherche du *knee point* est la méthode par défaut pour générer la *whitelist* dans Alevin, mais l'utilisateur peut également spécifier un seuil lui-même avec le paramètre *--forceCells*, ou encore fournir une *whitelist* prête à l'emploi sous la forme d'un fichier contenant une liste de CB avec le paramètre *--whitelist*.

D'autres algorithmes ont également été implémentés afin d'identifier automatiquement le *knee point*. Par exemple EmptyDrops [15] calcule grâce à une *spline* d'interpolation (fonction par morceaux constituée d'un polynôme sur chaque intervalle entre les points des données) une approximation de la fonction f définie par $log(r) = f(log(nUMI))$. Pour chaque CB, r représente le rang du CB dans la liste des CB triés par $nUMI$ croissant, $nUMI$ dénotant la taille de librairie (nombre d'UMI). EmptyDrops identifie ensuite la valeur r pour laquelle la courbure négative de f, calculée à partir de la dérivée seconde, est minimale – i.e. le point où la courbure est la plus marquée. Bien que la définition du *knee point* ou du moins la méthode pour l'identifier ou le visualiser varient (Figure 8), chaque implémentation cherche à

détecter, que ce soit avant ou après la déduplication des UMI, un seuil au-dessus duquel les CB ont une librairie nettement plus complexe que les autres.

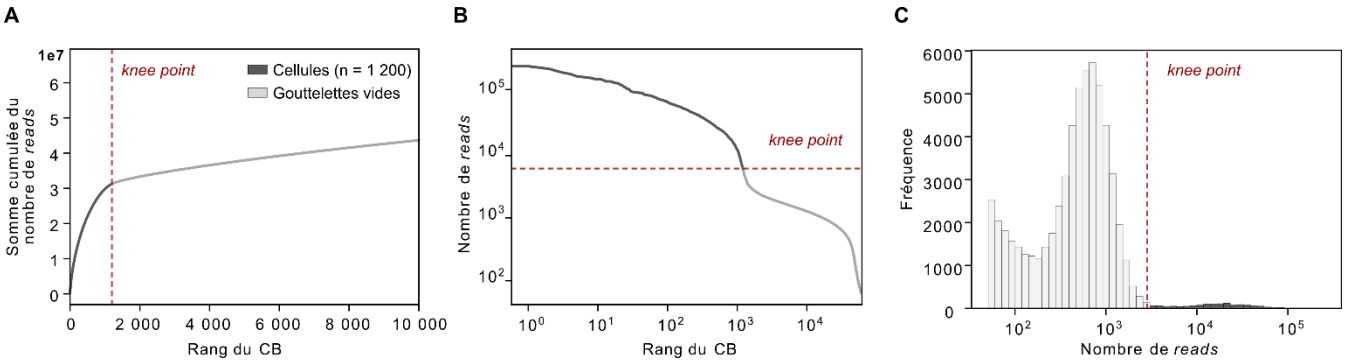


Figure 8 : Différentes versions du « *knee plot* ».

[A] *Knee plot* selon Makosko et al. [12] Le *knee point* correspond au point, identifié visuellement, où la somme cumulée du nombre de *reads* des CB triés par profondeurs de séquençage croissante se met subitement à augmenter plus lentement. **[B]** *Knee plot* illustrant la méthode implémentée dans EmptyDrops [12], qui cherchent à identifier le point où la courbure négative des profondeurs de séquençage des CB triées par ordre croissant, calculée à partir de la dérivée seconde, est minimale – i.e. la plus marquée. **[C]** *Knee plot* illustrant la méthode implémentée dans UMI-tools [23] et Alevin [22], qui cherchent respectivement à identifier un minimum local dans la distribution du nombre de UMI ou de *reads*. Dans [A], [B] et [C], l'échantillon BT20 (DSP992) a été utilisé à titre d'exemple car le *knee point* était assez marqué. Ce dernier, représenté par une ligne rouge en pointillés, a été identifié visuellement.

La *whitelist* générée est ensuite utilisée pour la correction des CB. Au cours du protocole expérimental, des erreurs (substitutions, délétions ou insertions de nucléotides) dans les séquences des CB peuvent survenir, menant à l'éclatement d'une librairie cellulaire en plusieurs sous-librairies. Il est donc important de corriger ces erreurs afin de fusionner les sous-librairies issues d'une même cellule et d'éviter des biais dans l'analyse en aval. Pour chaque CB du premier fichier de *reads* qui n'est pas retrouvé dans la *whitelist*, Alevin va donc tenter de le corriger en effectuant un seul changement dans la séquence (substitution, délétion ou insertion). Si un changement mène à un CB de la *whitelist*, alors il sera conservé (Alevin privilégiera les substitutions, qui sont les erreurs les plus courantes). Les CB pour lesquels aucun changement n'a été conservé à la suite de cette procédure, toujours hors de la *whitelist*, seront ignorés en aval. La *whitelist* sert donc de référence pour la correction des CB, alors qu'elle est définie par les auteurs d'Alevin comme une liste de CB associés à des cellules, et qu'elle est générée avec la méthode du *knee point* – méthode habituellement utilisée pour filtrer les CB issus de gouttelettes vides. Il semble donc y avoir une certaine incohérence quant à la définition de la *whitelist* et à son utilisation, qui sera discutée plus amplement dans la section 5.1 du chapitre de discussion.

Une fois la procédure de correction achevée, l'étape suivante consiste à identifier les transcrits dont sont issus les *reads* d'ADNc correspondant à chaque CB de la *whitelist*. Pour cela, Alevin utilise un algorithme déjà implémenté dans Salmon pour le traitement des *reads* de *bulk* RNA-seq, appelé *selective alignment*. Cet algorithme génère dans un premier temps un alignement partiel des *reads* sur le transcriptome (dit *lightweight alignment* ou *mapping*), afin d'identifier pour chaque *read* des transcrits candidats sur lesquels il sera aligné. Cet alignement partiel est obtenu par une méthode dite de « pseudo-alignement », permettant de générer un alignement sans avoir à effectuer une comparaison base par base, grâce à l'identification de matchs exacts entre des sous-séquences d'un *read* (mini-*reads*) et la référence. Cette tâche est plus simple que celle classiquement réalisée par les outils d'alignement tels que STAR, qui consiste à identifier une région du transcriptome sur laquelle le *read* entier s'aligne le mieux possible (moyennant quelques erreurs, ou *mismatches*). En effet, elle peut être facilement accomplie en utilisant une table de hachage qui va permettre, pour chacun des *k-mers* lus dans un *read* (sous-séquences nucléotidiques de longueur k, qui correspondent ici aux mini-*reads*), de retrouver rapidement les transcrits dans lesquels apparaît ce *k-mer*, sans effectuer d'alignement à proprement parler. Les *matches* des différents mini-*reads* sont ensuite recoupés pour déterminer le transcrit candidat dont est probablement issu le *read* (ou les transcrits candidats, en cas de *multimapping*).

Les méthodes de pseudo-alignement reposent sur l'idée qu'il n'est pas nécessaire de connaître la position précise de chaque *read* dans le transcriptome et qu'on peut se contenter de déterminer quels sont les transcrits dont ils sont le plus probablement issus si l'on s'intéresse exclusivement à la quantification des transcrits. Celles implémentées dans Salmon/Alevin et Kallisto sont très similaires, impliquant toutes deux la création préalable d'un index du transcriptome. Cet index est une structure de données permettant une recherche efficace dans le transcriptome, basée notamment sur un graphe de De Bruijn dont chaque arc représente un k-mer présent dans le transcriptome, et chaque nœud un (k-1)mer. Ce graphe est combiné à une table de hachage, qui permet de retrouver rapidement dans le graphe l'arc correspondant à un *k-mer* donné. Une fois l'index généré, les *reads* seront scannés de gauche à droite avec des fenêtres de *k* nucléotides pour effectuer le pseudo-alignement. Les *k-mers* lus seront recherchés dans le graphe grâce à la table de hachage, *k* va ainsi représenter la taille minimale du *match* exact, qui pourra être "étendu" aux *k-mers* voisins dans le graphe. Après avoir lu tous les *k-mers* du *read*, les transcrits candidats sont alors identifiés à partir de tous les chemins extraits du graphe, correspondant à des séquences retrouvées dans différents transcrits de la référence.

Dans le *selective alignment* de Salmon, les transcrits candidats et les *reads* seront finalement alignés base par base pour générer des scores d'alignement, permettant éventuellement de distinguer des transcrits

candidats trouvés à partir d'un même chemin du graphe. L'algorithme de *selective alignment* déployé dans la nouvelle version de Salmon est donc dit « hybride » : il utilise à la fois une méthode de pseudo-alignement pour effectuer dans un premier temps un *mapping* et une « sélection » des transcrits candidats, ainsi qu'une méthode d'extension d'alignement base par base pour générer ensuite des scores précis et filtrer ces derniers. Une autre particularité de Salmon est d'offrir la possibilité d'intégrer dans l'index du transcriptome des *decoys*, correspondant à des séquences génomiques « trompeuses » – voire au génome entier – dont certains *reads* peuvent être issus (par exemple des introns). Il arrive en effet que des régions non annotées du génome, telles que des régions intergéniques ou des introns, soient très similaires aux transcrits annotés, pouvant engendrer un biais de quantification (une surestimation de l'abondance de certains transcrits). Les *reads* ayant un score d'alignement plus élevé pour de telles séquences ne seront ainsi pas comptabilisés lors de la quantification, qui sera effectuée seulement pour les transcrits annotés.

La troisième étape mise en œuvre dans Alevin correspond à la correction et déduplication des UMI, ainsi qu'à la quantification des transcrits, qui se font conjointement. Pour chaque CB de la *whitelist*, Alevin va générer un graphe à partir de l'ensemble des classes d'équivalences et des UMI associés. Les nœuds du graphe correspondent aux couples UMI-classes d'équivalences uniques, reliés entre eux si les classes d'équivalence possèdent au moins un transcrit en commun et si les UMI ont une distance de Hamming inférieure ou égale à 1. Les arêtes du graphe peuvent être directionnelles, de A vers B si A correspond à un couple observé deux fois plus souvent que B, ou bidirectionnelles sinon. Alevin cherche ensuite un ensemble minimal de chemins qui couvrent le graphe entièrement. Chacun de ces chemins compte comme une molécule d'ARNm unique (pré-PCR), dont l'identité correspond au transcrit partagé par toutes les classes d'équivalence des nœuds du chemin. Si un seul transcrit est commun à tous les nœuds d'un chemin, ce dernier contribuera à hauteur de 1 *count* dans l'expression du transcrit. Sinon (cas de *multimapping*), le *count* sera distribué entre les différents transcrits identifiés pour le chemin, grâce à un algorithme *d'expectation-maximization* (EM) – les transcrits avec le plus de *counts* recevront une plus grande part, car ils ont une plus grande probabilité d'être exprimés. Il est important de noter qu'ainsi, Alevin utilise non seulement les séquences des UMI pour la déduplication de ces derniers, mais également les positions des *reads* sur le transcriptome. Un UMI de 8 nucléotides comme dans la technologie Drop-Seq ne permet effectivement d'étiqueter que $4^8 = 65\,536$ molécules d'ARNm, alors que sur chaque bille sont attachées $\sim 10^8$ séquences de codes-barres, et qu'une cellule peut contenir jusqu'à 1 million d'ARNm. Conséquemment, plusieurs molécules d'ARNm uniques pourront recevoir le même UMI, mais cela n'est pas un problème si l'on tire également parti du *mapping* pour le processus de déduplication : deux *reads* avec le même UMI, mais attribués à un transcrit de référence diffèrent, correspondent nécessairement à

deux molécules d'ARNm distinctes, et non à deux duplicatas PCR (cf. UMI jaune dans la Figure 7). La position des *reads* dans un même transcrit de référence n'est cependant pas prise en compte, puisqu'après tagmentation (cf. Figure 5), deux duplicatas issus de la pré-amplification par ISPCR mèneront à deux fragments différents (et seront donc alignés à un endroit différent dans le transcrit de référence). À l'issue de l'étape de déduplication des UMI et de quantification, tous les *counts* attribués aux transcrits d'un même gène sont finalement regroupés pour obtenir une expression des gènes au lieu des transcrits (on parle de quantification au niveau des gènes).

L'étape finale, appelée *final whitelisting*, est optionnelle. Elle permet de raffiner la première *whitelist* grâce à un algorithme de classification inspiré de Petukhov et al. [24]. Les CB de la *whitelist* sont répartis en deux groupes de dimensions égales : le premier, regroupant les plus grandes tailles de librairie, sera dit de « haute qualité », et le second sera dit « ambigu ». Parmi les CB hors de la *whitelist*, les 1 000 ayant les plus grandes tailles de librairie constitueront le groupe de « basse qualité ». Un classificateur bayésien naïf est ensuite entraîné sur un ensemble de métriques de qualité (telles que le nombre de gènes détectés, ou encore l'expression du génome mitochondrial ; voir la section 4.1 du Chapitre 4) calculées pour les CB des zones de haute et basse qualité. Le classificateur est ensuite utilisé pour prédire la qualité (haute ou basse) des CB de la zone ambiguë.

Cette étape finale est motivée par le fait qu'un simple seuil sur les tailles de librairies ne permet sans doute pas de séparer parfaitement les CB étiquetant des cellules viables des autres CB indésirables. Les auteurs mentionnent que des CB erronés pourraient subsister au-dessus du seuil et qu'inversement, des CB associés à une cellule pourraient tomber en dessous du seuil. La méthode qu'ils proposent ne permet cependant pas de récupérer ces derniers dans la *whitelist* finale, puisque les CB en dessous du seuil assignés au groupe de basse qualité servent uniquement à la phase d'entrainement (seuls les CB ambigus au-dessus du seul seront classifiés).

2.4.2 Choix des paramètres d'Alevin

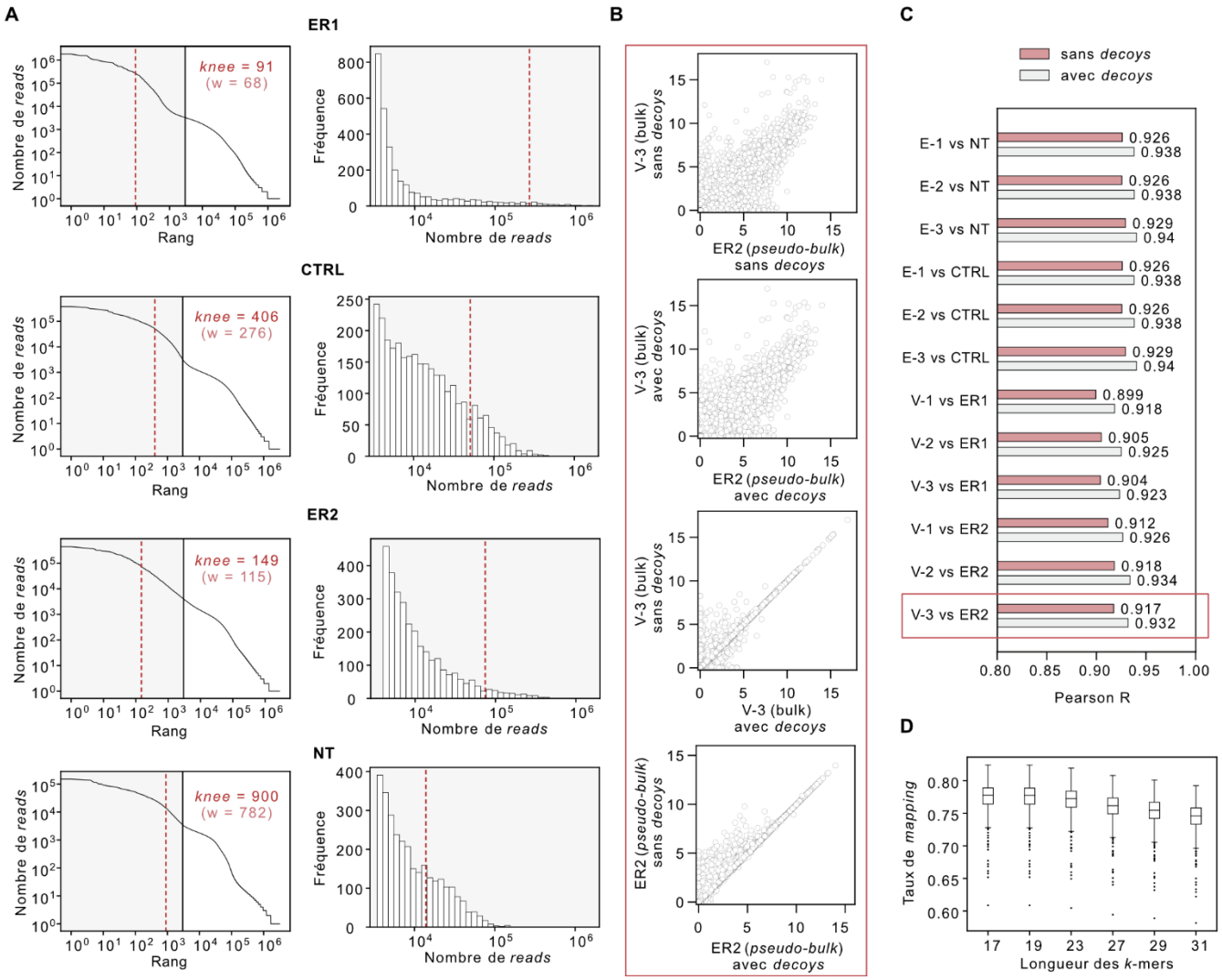


Figure 9 : Sélection des paramètres pour le logiciel Alevin

[A] *Knee point* identifié par Alevin pour les 4 échantillons de l'expérience pilote (lot de séquençage DSP779). Le *knee point* permet de déterminer le nombre de CB qui seront retenus dans la matrice d'expression finale. L'*index* du transcriptome fourni à Alevin a été construit avec une longueur de *k*-mer égale à 31 (k = 31) et l'annotation *gencode* 34 du génome GRh38. La zone grisée correspond aux 3 000 CB avec les plus grandes profondeurs de séquençage (nombre de *reads*), qui est le nombre estimé, selon la loi de Poisson, de cellules encapsulées. Les histogrammes à droite n'incluent que ces 3 000 CB. Le chiffre *w* en rouge entre parenthèses correspond au nombre de CB inclus dans la *whitelist* finale d'Alevin (après l'étape de *final whitelisting*). Très peu de CB sont sélectionnés par la méthode du *knee point* par rapport à l'estimation du nombre de cellules encapsulées (3 000). Pour éviter de perdre des CB pertinents, il est donc préférable de sélectionner les 3 000 CB avec les plus grandes profondeurs de séquençage avec le paramètre *--forceCells*, et d'éliminer en aval les CB indésirables restants. **[B]** Corrélations croisées entre 4 profils d'expression. Deux profils d'expression sont des *pseudo-bulk*s – somme des profils d'expression de tous les CB – de l'échantillon ER2 (DSP779, scRNA-seq). Ils sont obtenus en effectuant le *mapping* avec Alevin en utilisant pour le premier un *index* contenant des *decoys*, et pour le second un *index* sans *decoys*. Les deux autres profils sont des profils d'expression de l'échantillon V-3 (DSP356, *bulk* RNA-seq), obtenus en effectuant le *mapping* avec Salmon en utilisant également des *decoys* pour l'un mais pas pour l'autre. Les *index* des

pseudo-bulks et des *bulks*, qu'ils contiennent ou non des *decoys*, ont été construits avec k = 31, *gencode* 34, GRh38. Avant de calculer les corrélations, les *pseudo-bulks* et les *bulks* ont au préalable été normalisés, respectivement par CPM et TPM, et finalement log2(+1) transformés. L'ajout de *decoys* dans l'index semble avoir surtout un impact sur les gènes faiblement exprimés. **[C]** Corrélations entre les *pseudo-bulks* et *bulks* de différentes paires d'échantillons. Deux corrélations sont calculées pour chaque paire d'échantillons. L'une est calculée entre le *pseudo-bulk* et le *bulk* obtenus en utilisant Salmon avec un *index* contenant des *decoys*. L'autre est calculée entre le *pseudo-bulk* et le *bulk* obtenus en utilisant Salmon avec un *index* sans *decoys*. Les deux *index* sont construits avec k = 31, *gencode* 34, GRh38. Avant de calculer les corrélations, les *pseudo-bulks* et les *bulks* sont au préalable normalisés et transformés comme décrit dans [B]. Tous les échantillons dont sont issus les *bulks* ou les *pseudo-bulks* proviennent de la lignée MCF7 du laboratoire. Les échantillons de *bulk* RNA-seq E-1, E-2, E-3 sont trois réplicats de cellules traitées avec de l'œstrogène (E pour *estrogen*) et V-1, V-2, V-3 sont trois réplicats de cellules cultivées sans œstrogène (V pour *vehicle*). Les échantillons de scRNA-seq ER1 et ER2, CTRL et NT correspondent respectivement à des cellules transfectées avec des siRNA bloquant ESR1 (gène codant pour le récepteur des œstrogènes), à des cellules transfectées avec un siRNA contrôle et à des cellules non transfectées. Les échantillons d'une même paire peuvent donc être considérés comme des réplicats biologiques, les uns correspondant à des cellules pour lesquelles la voie de signalisation œstrogénique est active (E-1, E-2, E-3, CTRL, NT) et les autres à des cellules pour lesquelles elle est inactive (V-1, V-2, V-3, ER1, ER2). La corrélation entre le *pseudo-bulk* et le *bulk* d'une paire d'échantillons devrait donc être plus élevée lorsque le prétraitement est plus adapté. Ici, la corrélation est légèrement plus grande lorsque des *decoys* sont inclus dans *l'index* du transcriptome. **[D]** Distributions des taux de *mapping* pour les CB de l'échantillon NT selon la longueur de *k*-mer utilisée pour construire l'*index* du transcriptome fourni à Alevin. Les *index* construits pour chaque longueur de *k*-mer ont été obtenus avec GRh38, *gencode* 34 et en utilisant des *decoys*. Seuls les 500 CB ayant les plus grandes profondeurs de séquençage ont été considérés, afin d'exclure la plupart des CB issus de gouttelettes vides ou associés à des cellules cassées, pour lesquels le taux de *mapping* est toujours bien plus bas (cf. section 4.1 du Chapitre 4). La longueur des *reads* variant de 20 à 63 bp – un *trimming* des adaptateurs a été effectué au préalable –, une longueur de *k*-mer plus petite que celle définie par défaut (k = 31, optimale pour des reads de taille > 75 bp), permet donc le *mapping* d'une fraction légèrement plus grande de *reads*

Le comportement par défaut d'Alevin pour traiter les fichiers FASTQ en entrée n'est pas toujours adapté au jeu de données : l'utilisateur peut donc ajuster la valeur de certains paramètres afin d'optimiser la qualité de la matrice d'expression en sortie.

Par exemple, la méthode du *knee point*, qui est la procédure par défaut pour générer la *whitelist* initiale dans Alevin, dépend beaucoup de la qualité de l'échantillon. Pour les échantillons de mauvaise qualité – e.g. comportant beaucoup de cellules cassées –, il arrive que le *knee point* dans la distribution des profondeurs de séquençage des CB ne soit pas suffisamment net et soit surestimé par Alevin, qui identifie alors le mauvais minimum local. Plusieurs utilisateurs ont ainsi signalé sur GitHub (cf. https://github.com/COMBINE-lab/salmon/issues/, *issues* 340, 362, 374, 396, 625) un nombre très petit de CB retenus dans la matrice d'expression finale comparé à leurs attentes en utilisant Alevin avec les paramètres par défaut – i.e. avec la méthode du *knee point* – sur leur jeu de données. Ayant rencontré le même problème avec les 4 échantillons ER1 et ER2, CTRL et NT de l'expérience pilote (Figure 8A), j'ai donc utilisé pour ces échantillons le paramètre *--forceCells* avec la valeur 3 000, qui applique un seuil strict en sélectionnant les 3 000 CB avec les profondeurs de séquençage les plus grandes. Cette valeur

correspond au nombre attendu de cellules encapsulées, calculé avec la loi de Poisson à partir des concentrations de billes et de cellules utilisées dans l'expérience, qui est en pratique supérieur au nombre de cellules réellement encapsulées (cf. section 2.1.1). La *whitelist* de 3 000 CB pourra donc inclure un nombre plus ou moins conséquent de CB indésirables, selon le nombre de cellules viables récupérées pour chaque échantillon.

Comme il sera discuté dans la section 4.1 du chapitre 4, la quantité de CB indésirables inclus dans la *whitelist* a un impact sur la procédure de *final whitelisting* effectuée par Alevin en aval à partir de la matrice d'expression. Si la *whitelist* inclue trop de CB indésirables, la procédure de *final whitelisting* ne parviendra pas à tous les éliminer. Si la *whitelist* est déjà très stricte et ne contient que des CB pertinents, un bon nombre de CB pertinents seront éliminés à tort durant la procédure de *final whitelisting.* Par exemple pour l'échantillon ER1, seulement 91 cellules tombent au-dessus du *knee point* ce qui semble déjà très sévère ; après la procédure de *final whitelisting*, 25 cellules supplémentaires sont perdues. Cette procédure étant optionnelle – *whitelist* finale écrite dans un fichier à part –, je propose donc dans le chapitre 4 de choisir manuellement des seuils sur certaines métriques de qualité afin d'éliminer les CB indésirables restants dans la *whitelist* de 3 000 CB.

J'ai également voulu étudier l'impact des *decoys* sur la qualité du *mapping*. Pour cela j'ai tiré parti d'une expérience de *bulk* RNA-seq (DSP356) effectuée sur la même lignée – MCF7 cultivée dans le laboratoire – il y a plusieurs années. J'ai utilisé Salmon (équivalent d'Alevin pour le *bulk* RNA-seq) avec et sans *decoys* sur les réplicats des différents échantillons du jeu de données associé. J'ai ensuite créé pour chacun des échantillons de l'expérience pilote (DSP779) un *pseudo-bulk*, correspondant à la somme des profils d'expression des cellules uniques. Les *pseudo-bulks* ont été créés à partir des profils d'expression générés par Alevin avec et sans *decoys*. Jugeant que certains échantillons des deux expériences pouvaient être considérés comme des réplicats biologiques, j'ai calculé la corrélation entre leurs profils d'expression respectifs (*bulk* pour DSP356 et *pseudo-bulk* pour DSP779), dans l'idée qu'un traitement optimal des données devrait maximiser ces corrélations. Comme les auteurs recommandent l'utilisation des *decoys*, les profils générés avec des *decoys* devraient corréler plus fortement entre eux que les profils analogues générés sans *decoys.* L'utilisation des *decoys* semble améliorer légèrement la qualité du *mapping* (Figure 8C) : en effet, l'augmentation des coefficients de corrélation est certes faible, mais elle est systématique. L'augmentation est probablement faible car ce sont les gènes les moins exprimés qui sont le plus impactés (Figure 8B).

Le rôle des *decoys* (qui incluent le génome au complet) est d'exclure de la quantification les *reads* ayant un score d'alignement avec ces derniers plus grand qu'avec le transcriptome de référence.

L'utilisation des *decoys* devrait donc permettre de réduire l'expression de certains gènes, surestimée à cause d'alignements douteux sur le transcriptome de séquences issues de régions non annotées du génome. Cependant, en comparant les deux profils d'expression d'un des réplicats de l'expérience DSP356 (profils *bulks* du réplicat V-3) générés en effectuant le *mapping* avec ou sans *decoys*, j'ai pu noter que certains gènes étaient plus exprimés dans le profil d'expression généré avec des *decoys* (Figure 8B). En examinant le biotype de ces gènes, j'ai constaté qu'ils correspondaient presque tous à des pseudogènes. J'ai effectivement utilisé pour le *mapping* un transcriptome de référence complet, incluant également les transcrits ne codant pas pour des protéines. Bien que les ARN soient capturés par leur queue poly-A, caractérisant les ARNm qui sont traduits en protéines, certains ARN sans queue poly-A sont également capturés au passage et cette information peut par la suite s'avérer utile, comme discuté dans le chapitre 4. Toutefois l'expression des pseudogènes est souvent le résultat d'une erreur de l'outil d'alignement, qui distingue difficilement les gènes de leurs pseudogènes, dont les séquences sont presque identiques du fait de leur homologie – origine évolutive commune. Il arrive souvent que le score d'alignement d'un *read* soit aussi bon avec un gène qu'avec le pseudogène de ce gène, et le *read* aura alors plusieurs gènes « candidats », ce qui est un cas d'alignement multiple, ou *multimapping* – on parle aussi de *multimapping read* pour le *read* en question. Une explication potentielle à l'expression accrue de pseudogènes lors de l'utilisation de *decoys* pourrait être que les *decoys* interfèrent avec l'algorithme d'*expectation-maximisation* (EM) mis en œuvre pour traiter les cas de *multimapping* lors de la quantification. Pour chaque *read* aligné sans ambiguïté sur un gène, le gène en question reçoit pendant la quantification une valeur d'expression égale à 1, appelée *count*. En cas de *multimapping*, l'algorithme d'EM « divise » le *count* entre les gènes candidats, attribuant une plus grande part aux candidats avec lesquels le plus de *reads* ont déjà été alignés sans ambiguïté (cf. section 2.4.1). Il est possible que les *decoys* éliminent certains alignements douteux sur des gènes ayant un pseudogène et que lors de la quantification, le pseudogène reçoive alors une plus grande part du *count* pour les *multimapping reads* ayant comme candidats un gène et son pseudogène. Ainsi, l'expression accrue des pseudogènes correspond plus vraisemblablement à un artéfact qu'à un ajustement pertinent du profil d'expression permis par les *decoys*. Cet artéfact ne devrait néanmoins pas avoir d'impact sur les analyses en aval, qui n'impliquent généralement que les gènes codant pour des protéines.

L'indexation avec les derniers est plus gourmande en mémoire (3 GB de RAM sans *decoys* versus 32 GB avec *decoys* pour k = 31) et légèrement plus lente (10 min sans *decoys* versus 1h30 avec *decoys* pour k = 31) et, comme mentionné dans un *benchmark* récent [25], et suggéré par les résultats de la Figure 9, l'amélioration apportée par ces derniers semble négligeable. Plusieurs analyses en aval étant

centrées sur des facteurs de transcription, qui sont eux-mêmes des gènes faiblement exprimés, il n'est cependant pas exclu que cette légère amélioration puisse tout de même jouer un rôle. J'ai donc tout de même décidé d'utiliser les *decoys* sur l'ensemble des échantillons, les ressources utilisées n'étant pas excessives. De plus, une fois généré, l'index peut être utilisé autant de fois que nécessaire pour le *mapping* de différents jeux de données.

Un autre paramètre à considérer lors de l'indexation est la longueur des *k-mers*, qui est par défaut de 31, correspondant d'après les auteurs à la valeur optimale pour des *reads* de longueur de taille supérieure ou égale à 75. Pour certains jeux de données, les *reads* seront plus courts : il pourrait donc être profitable d'utiliser une longueur de *k-mers* plus petite afin de maximiser le taux de *mapping* (fraction des *reads* alignés sur le transcriptome). La longueur des *k-mers* représente en effet la taille minimale d'un *match* exact qu'un *read* doit avoir avec un transcrit de référence pour que ce transcrit soit un « candidat ». Pour les jeux de données Drop-Seq explorés dans le cadre de ce mémoire, les *reads* étaient de longueur 63 et après le *trimming* (cf section. 2.3.2), les *reads* les plus courts étaient de longueur 20. En utilisant k = 31, tous les *reads* de longueur < 31 après le *trimming* seraient alors ignorés. J'ai donc essayé plusieurs valeurs de *k* (Figure 8D) plus petites que 31 et impaires, l'imparité étant requise par Alevin – et par la plupart des algorithmes faisant usage des *k-mers*. En effet contrairement à un *k-mer* pair, un *k-mer* impair ne peut jamais former de palindrome, c'est-à-dire correspondre à son *reverse* complémentaire, ce qui est préférable pour la construction d'un graphe de De Bruijn. Les taux de *mapping* obtenus étaient légèrement plus élevés pour les longueurs des *k-mers* plus petites. Pour k = 17 et k = 19, les taux semblaient presque identiques, j'ai donc décidé d'utiliser l'index construit avec k = 19 pour le *mapping* de l'ensemble des échantillons. Ici encore, l'amélioration apportée semble minime et les index construits avec des longueurs de *k-mers* plus petites nécessitent davantage de mémoire (31 GB de RAM pour k = 31 versus 51 GB pour k = 19 avec les *decoys*). Les temps d'exécution étant en revanche équivalents (1h30 pour k = 31 et pour k = 19 avec les *decoys)*. Mais les ressources restaient accessibles et comme mentionné plus haut, l'index une fois construit peut être ensuite utilisé pour le *mapping* de nombreux jeux de données.

Bien-sûr, le *mapping* aussi est plus gourmand en mémoire et s'effectue plus lentement lorsque l'index est construit avec des *decoys* et une longueur de *k-mers* plus petite. Mais la mémoire allouée restait de l'ordre de 50 GB de RAM et le temps d'exécution dépassait rarement les 2 heures. Que ce soit pour le *trimming* ou pour la construction de l'index du transcriptome permettant le *mapping*, j'ai donc toujours opté pour la stratégie la plus « prudente », tant que les ressources utilisées restaient raisonnables.

2.4.3 Mise en évidence de CB erronés dans la *whitelist* d'Alevin

Après avoir généré les matrices d'expression pour l'ensemble des 19 échantillons en utilisant Alevin avec les paramètres sélectionnés dans la section 2.4.2, j'ai tout d'abord mené une première analyse exploratoire dans le but de contrôler la qualité des cellules. Certaines métriques de qualité (cf. section 3.1.1 du Chapitre 3) ont fait ressortir des paires de cellules dont la proximité des CB et des profils d'expression suggère qu'elles représentent en réalité une seule et même librairie cellulaire, scindée suite à une erreur de CB « récurrente ». Une erreur récurrente désigne ici une substitution spécifique don le taux est élevé, c'est-à-dire touchant un nombre particulièrement élevé de *reads*. Un CB erroné issu d'une telle erreur aura donc une grande profondeur de séquençage et pourra par conséquent se retrouver sélectionné dans la *whitelist* d'Alevin (Figure 10 A).

Afin de vérifier cette hypothèse, j'ai comparé le nombre observé de CB « voisins » au nombre attendu pour un tirage aléatoire de 3 000 CB (taille des *whitelists*). Deux CB sont dits voisins si la distance de Hamming entre leurs séquences est égale à 1, i.e. si l'une peut être obtenue à partir de l'autre en effectuant une seule substitution (remplacement d'un nucléotide par un autre). Effectivement, si des CB erronés sont sélectionnés avec leur CB d'origine dans la *whitelist,* le nombre observé sera plus grand que le nombre attendu. *A contrario,* en l'absence de CB erronés dans la *whitelist,* ces deux nombres devraient être proches. J'ai donc calculé pour chaque échantillon de l'expérience pilote (DSP779) le nombre de CB ayant au moins un voisin parmi les 2 999 autres CB de la *whitelist* C_1. Des valeurs excessivement élevées ont été obtenues, allant de 907 CB pour l'échantillon ER2 à 1 557 CB (soit la moitié des CB de la *whitelist*) pour l'échantillon CTRL (Figure 10 B). De plus, les distributions du nombre de voisins des CB ont révélé des différences entre les échantillons, certains CB de l'échantillon ER1 ayant jusqu'à 14 voisins, contre seulement 4 voisins au maximum pour l'échantillon NT (Figure 10 B). Ces différences, liées à la distribution des profondeurs de séquençage des CB de chaque échantillon, sont discutées dans la section 2.4.4 ainsi que dans la section 4.1.1 du chapitre 4.

J'ai ensuite estimé la valeur attendue de C_1 selon la loi Binomiale, notée $\mathbb{E}(C_1)$, pour un tirage de 3 000 CB de longueur 12 encodés avec 4 nucléotides, de la manière suivante :

Soit Ω *l'ensemble des CB de longueur 12 que l'on peut encoder avec 4 nucléotides, de taille* $|\Omega| = 4^{12} = 16\,777\,216$. *Soit* $c_a \in \Omega$. *Une première expérience de Bernoulli consiste à tirer* $c_b \in \Omega \backslash \{c_a\}$, *dont l'issue est un succès si* c_a *et* c_b *sont voisins. Sachant qu'un CB a 36 voisins (3 substitutions possibles pour chacune des 12 positions), la probabilité* p *d'un succès est de* $\frac{36}{|\Omega|-1}$. *Le nombre de succès obtenu après la répétition de* n *expériences de* $\mathrm{Bernoulli}$ *indépendantes peut être modélisé par une loi* $\mathrm{Binomiale}$ *de*

paramètres n et p, avec p la probabilité d'un succès. La variable $C \sim Binomiale(n, p)$, avec $n = 2\,999$, va alors décrire le nombre de voisins de c_a dans un tirage (avec remise car les expériences sont indépendantes) de $2\,999$ CB dans Ω. On peut ensuite définir la variable $C_1 \sim Binomiale(3\,000, p_1)$, avec $p_1 = P(C \geq 1) = 1 - P(C = 0) = 1 - (1 - p)^n$, qui représente le nombre de CB ayant au moins un voisin parmi un tirage de $3\,000$ CB. Enfin, on calcule $\mathbb{E}(C_1) = 3\,000\, p_1$.

Dans l'énoncé ci-dessus, j'ai présumé l'indépendance des expériences de Bernoulli afin de pouvoir appliquer la loi Binomiale, ce qui n'est pas tout à fait adéquat pour le problème posé. Pour valider mon calcul, j'ai donc également estimé la valeur attendue de C_1 de manière empirique, en effectuant une simulation *in silico* du problème (Figure 10 C). Les deux estimations, concordantes, ont ainsi prédit un nombre C_1 d'environ 20 CB, valeur bien en dessous de celles observées dans l'expérience pilote.

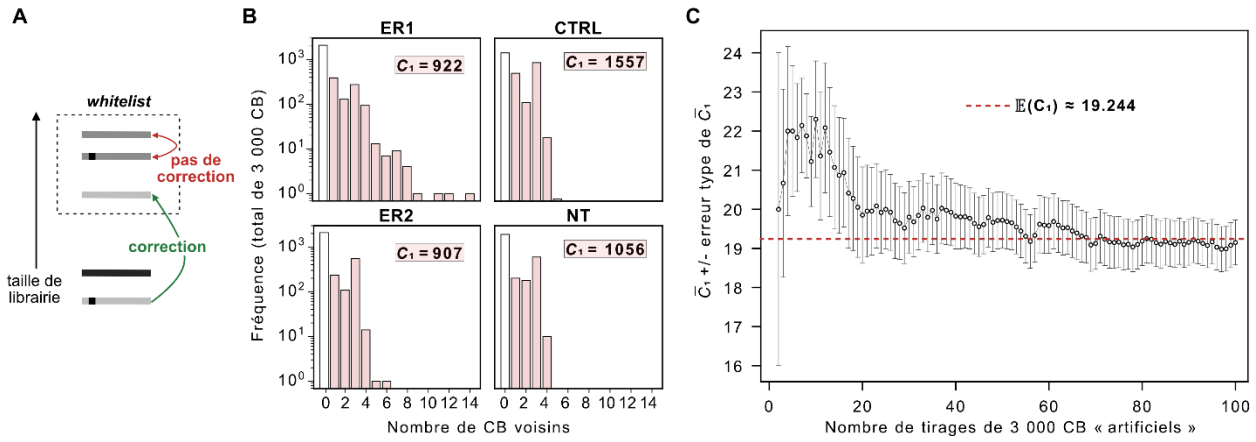


Figure 10 : Problématique soulevée par la méthode de correction des CB implémentée dans Alevin.
[A] Scénario dans lequel une erreur de CB n'est pas corrigée par Alevin. Si une erreur est suffisamment fréquente, elle pourra être sélectionnée dans la *whitelist* et échappera alors à la correction. Le rectangle en pointillé représente la *whitelist*, les rectangles avec différentes nuances de gris représentent différents CB, et un petit carré noir sur un CB représente une erreur. **[B]** Distribution du nombre de CB voisins pour les 4 échantillons de l'expérience pilote (DSP779). Pour chacun des échantillons, les CB étudiés sont ceux sélectionnés dans la *whitelist*, soit les 3 000 les plus fréquents. C_1 dénote le nombre de CB avec un voisin ou plus parmi les 2 999 autres CB, correspondant à la somme des fréquences en rouge clair. **[C]** Nombre attendu de CB avec au moins un voisin dans un tirage de 3 000 CB. $\overline{C_1}$ représente la moyenne empirique de C_1, basée sur la répétition d'une simulation où l'on génère de manière computationnelle et aléatoire 3 000 CB « artificiels » de longueur 12 et encodés avec 4 lettres. Chaque CB artificiel est généré en ajoutant aléatoirement un des 4 nucléotides suivant une loi uniforme (chaque nucléotide est équiprobable), jusqu'à obtenir une séquence de longueur 12. À chaque nouveau tirage *in silico*, on évalue C_1 pour les 3 000 CB tirés et on recalcule la moyenne. La moyenne empirique converge vers l'espérance théorique calculée à partir de la loi binomiale, qui est dénotée $\mathbb{E}(C_1)$ et représentée par des pointillés rouges. Ainsi, C_1 observé dans les échantillons de l'expérience pilote ($\approx$ 1 000) est bien plus élevé que C_1 attendu ($\approx$ 20), ce qui suggère la présence d'erreurs non corrigées par Alevin dans les données. Ceci montre que la méthode de correction des CB implémentée dans Alevin ne permet pas de corriger les erreurs les plus récurrentes, car les CB erronés qui en découlent se retrouvent sélectionnés dans la *whitelist*.

J'ai ainsi démontré que la procédure de correction des CB implémentée dans Alevin n'est pas adaptée, car elle ne permet pas de corriger les erreurs les plus récurrentes, les CB erronés qui en découlent se retrouvant sélectionnés dans la *whitelist*. Pour les 4 échantillons de l'expérience pilote, un nombre élevé de CB erronés ne pouvaient effectivement pas être corrigés par Alevin, car ils faisaient partie des 3 000 CB avec les plus grandes profondeurs de séquençage sélectionnés dans la *whitelist*. Pour éviter de sélectionner des CB erronés dans la *whitelist*, il aurait fallu choisir un seuil bien plus strict, éliminant par conséquent un grand nombre de CB pertinents étiquetant des cellules. Dans la Figure 12, on peut en effet observer que la distribution des profondeurs de séquençage des CB erronés se superpose fortement à celle des CB valides (sans erreur).

2.4.4 Création d'une nouvelle *whitelist*

Après avoir identifié les erreurs non corrigées par Alevin, j'ai tout d'abord songé à me servir d'un autre logiciel n'utilisant pas la même procédure de correction des CB qu'Alevin pour générer la matrice d'expression à partir des fichiers FASTQ. L'équipe qui a développé la technologie Drop-Seq a notamment implémenté en Java une suite logicielle (dite « Drop-Seq tools ») permettant d'effectuer une à une les étapes intégrées dans Alevin. Cette suite inclut un outil de correction des CB, dont la procédure ne repose pas sur une *whitelist* contrairement à celle mise en œuvre dans Alevin. Pour l'ensemble des CB ayant une profondeur de séquençage supérieure à 20, toutes les paires de CB voisins (distance de Hamming égale à 1) sont générées. Pour chaque paire, le CB avec la plus grande profondeur de séquençage est désigné comme le CB d'origine. À partir de ces paires, les substitutions les plus récurrentes, dites « systématiques », sont identifiées, et seuls les CB issus de telles substitutions et associés à un seul CB d'origine sont corrigés. Cette procédure de correction semble intéressante et pour la mettre en œuvre, on pourrait modifier le code source d'Alevin pour l'y intégrer, ou bien utiliser les outils DetectBeadSubstitutionErrors et DetectBeadSynthesisErrors de la suite Drop-Seq tools. Ces derniers produisent un fichier BAM à partir d'un fichier BAM sans alignements, traité au préalable avec les outils tagCells et tagUMIs pour ajouter des « tags » sur les codes-barres. Une fois la correction effectuée, les fichiers BAM en sortie pourraient être convertis à nouveau en fichiers FASTQ, qui est le format que supporte Alevin en entrée. Avec cette solution cependant, une « double correction » des CB serait effectuée, puisqu'Alevin va alors à son tour effectuer une correction des CB à partir des fichiers FASTQ : certains CB qui n'auraient pas été corrigés par la méthode de Drop-Seq tools pourraient alors se retrouver corrigés par Alevin en aval.

J'ai finalement opté pour une stratégie n'exploitant pas la procédure de correction implémentée dans Drop-Seq tools, et permettant d'apporter des améliorations à Alevin sans avoir à modifier son code source. Cette stratégie repose sur la définition même d'une *whitelist*, qui est selon moi « une liste de CB exempte de CB erronés ». Alevin acceptant également une *whitelist* externe pour la correction, j'ai imaginé de modifier la *whitelist* initiale de 3 000 CB en excluant les CB erronés pour qu'en aval, ils puissent bénéficier de la correction effectuée par Alevin (cf. section 2.4.1).

Pour créer la nouvelle *whitelist* « filtrée », j'ai tout d'abord défini un graphe non connexe, dont les nœuds correspondent aux 3 000 CB ayant les plus grandes profondeurs de séquençage – i.e. ceux de la *whitelist* initiale. Dans ce graphe, deux CB sont dits voisins et sont alors connectés si leur distance d'édition est égale à 1, i.e. si une seule substitution, insertion ou délétion dans la séquence de l'un permet

de retomber sur la séquence de l'autre. Dans la section 2.4.3, la distance de Hamming m'a permis de démontrer la présence de CB erronés issus de substitutions. Il est moins évident de démontrer la présence de CB erronés issus d'insertions ou de délétions, car pour un certain CB, le nombre de CB voisins découlant d'une seule insertion ou d'une seule délétion dépend de la séquence du CB en question. Par exemple la séquence « AA » aurait 10 voisines possibles tandis que la séquence « AT » en aurait 13. J'ai donc dans un premier temps utilisé la distance de Hamming pour construire le graphe, mais les métriques de qualité qui avaient permis la découverte des CB erronés (cf. section 3.1.1 du chapitre 3) ont révélé des CB erronés résiduels issus d'insertions ou de délétions. Aussi ai-je finalement décidé de construire le graphe en utilisant la distance d'édition prenant également en compte les insertions et délétions, et correspondant de surcroît à la distance utilisée par Alevin pour corriger les CB. Pour appliquer la distance d'édition, j'ai suivi la stratégie implémentée dans Alevin, où tous les voisins potentiels de chacun des 3 000 CB sont générés. Pour savoir si deux CB sont voisins, il suffit ainsi de regarder si l'un est dans la liste des voisins potentiels de l'autre.

Chaque composante connexe du graphe est donc supposée représenter une seule et même cellule, dont la librairie s'est retrouvée scindée en plusieurs sous-librairies, résultant chacune d'une substitution différente sur le CB d'origine (voisins de niveau 1). Ces sous-librairies peuvent à leur tour être scindées à la suite de nouvelles substitutions (voisins de niveau 2 et plus). Une composante connexe est donc constituée d'un CB d'origine et d'un ensemble de CB erronés découlant de substitutions successives sur ce CB d'origine.

Dans la Figure 11A, on peut voir que les CB formant une composante connexe (correspondant vraisemblablement à une cellule unique) apparaissent regroupés dans la représentation 2D générée par l'algorithme UMAP, qui rapproche les CB associés à des cellules ayant des profils d'expression semblables. Cet algorithme est utilisé pour guider l'identification de sous-populations cellulaires dans les analyses en aval, sur la base des *clusters* de CB qu'il met en évidence, censés représenter des groupes de cellules biologiquement proches. Il apparaît donc que les CB erronés de la *whitelist* peuvent fausser ces analyses, en déformant la structure sous-jacente (parfois appelée *manifold*) décrivant la biologie des cellules, que l'algorithme de UMAP tente d'identifier et de projeter dans deux dimensions. Dans le pire des cas, cela pourrait conduire à l'identification fautive de *clusters* de cellules – une composante connexe de 21 voisins comme celle qui apparaît dans la Figure 11A est toutefois un cas extrême.

Idéalement, on voudrait assigner le même CB à toutes les sous-librairies d'une même composante connexe afin de fusionner ces dernières, provenant vraisemblablement de la même cellule. Il ne serait en théorie même pas nécessaire d'identifier précisément le CB d'origine – celui synthétisé sans erreur sur la bille –, la séquence en tant que telle important peu. Il serait par exemple possible d'utiliser pour chaque

composante un CB choisi au hasard dans celle-ci. Cependant, avec la méthodologie mise en œuvre, il n'est pas possible de corriger tous les CB d'une composante. Pour une composante connexe donnée, si l'on ne garde dans la *whitelist* qu'un seul CB, Alevin ne corrigera que les CB voisins de ce dernier – les *reads* de tous les autres CB de la composante seront perdus. Si l'on garde en revanche plusieurs CB, plus de CB de la composante seront corrigés, mais cela ramènerait au problème de départ où plusieurs sous-librairies issues d'une même cellule sont présentes dans la *whitelist*. Au mieux, on peut donc conserver dans la *whitelist* le CB de chaque composante permettant de récupérer le maximum d'information. Notamment, en conservant celui avec la profondeur de séquençage la plus grande ou avec le plus de CB voisins – capable d'« absorber » le plus grand nombre de sous-librairies de la composante –, on peut maximiser le nombre de *reads* récupérés dans les librairies cellulaires finales. Étant donné que le nombre de voisins d'un CB et sa profondeur de séquençage corrélaient fortement (Figure 11B), j'ai finalement décidé de n'utiliser que la profondeur de séquençage. Cette corrélation entre le nombre de voisins d'un CB et sa profondeur de séquençage explique par ailleurs les différences entre les distributions des nombres de CB voisins des 4 échantillons, mentionnées dans la section 2.4.3 à propos de la Figure 10. La nouvelle *whitelist* inclue donc tous les CB n'ayant aucun voisin (composantes connexes de taille 1) ainsi qu'un CB de chaque composante, correspondant à celui avec la plus grande profondeur de séquençage – lequel pourrait de surcroît correspondre au véritable CB d'origine. La méthode développée pour la création de la nouvelle *whitelist* est décrite plus bas (voir Algorithmes 1 et 2) sous forme de pseudo-code.

J'ai finalement voulu vérifier que les *reads* des CB non corrigés par Alevin – ceux qui n'étaient pas des voisins directs du CB sélectionné dans la nouvelle *whitelist* – ne représentaient pas une perte d'information trop importante. Pour les échantillons de l'expérience pilote (DSP779), après avoir constaté que ces CB étaient peu nombreux, j'ai examiné quel pourcentage leurs *reads* représentaient par rapport aux *reads* des CB sélectionnés dans la nouvelle *whitelist* et de ceux corrigés par Alevin – voisins directs des CB sélectionnés. J'ai constaté qu'en ajoutant aux *reads* des CB sélectionnés dans la nouvelle *whitelist* ceux de leurs CB voisins directs, leurs profondeurs de séquençage respectives se voyaient doublées. Cependant, l'ajout complémentaire des *reads* des quelques CB restants dans leurs composantes connexes respectives ne rapportait qu'une quantité négligeable de *reads* (Figure 11C).

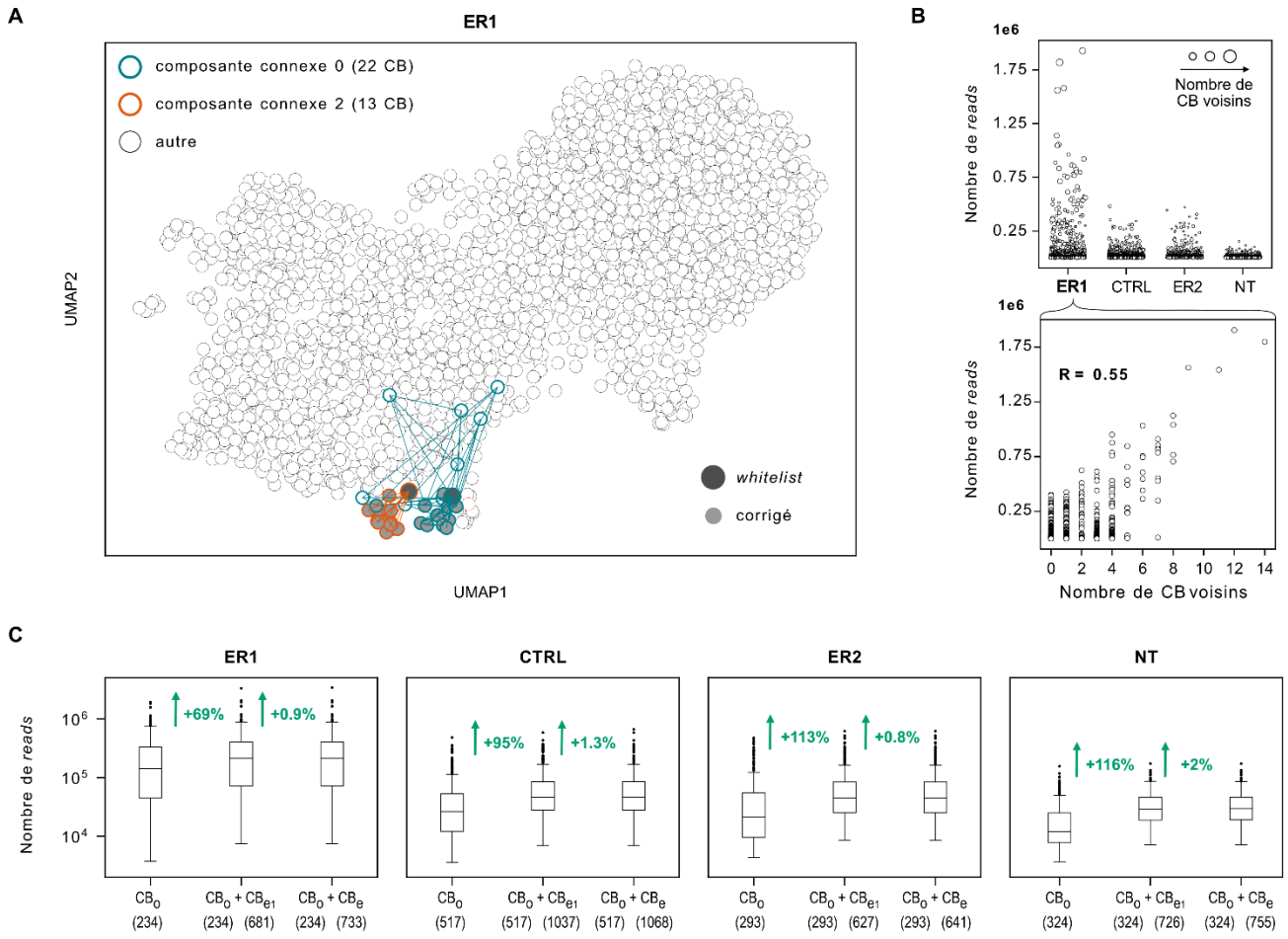


Figure 11 : Arguments en faveurs de la nouvelle méthode de *whitelisting* des codes-barres cellulaires (CB).

[A] Représentation UMAP générée à partir des 20 premières composantes principales (PC, pour *principal component*), calculées sur les profils d'expression de l'échantillon ER1 obtenus avec Alevin (k = 19, *--forceCells* 3 000, GRh38, *gencode* 34) normalisés par CPMedian (division par la taille de librairie et multiplication par la taille de librairie médiane, cf. 4.2) et transformés avec un logarithme de base 2 et un *pseudocount* égal à 1. Les paramètres utilisés – autres que le nombre de PC – sont ceux assignés par défaut dans l'implémentation de Scanpy [26]. Deux exemples de composantes connexes sont mis en évidence, avec les lignes connectant les CB voisins, i.e. ceux dont la distance de Hamming est égale à 1. Chaque cercle représente un CB (i.e. une potentielle cellule) et le cercle gris foncé dans chaque composante connexe correspond au CB sélectionné dans la nouvelle *whitelist* (CB de la composante ayant la plus grande profondeur de séquençage), censé représenter le CB « d'origine » (sans erreur). Les cercles gris clair correspondent aux CB qui seront corrigés par Alevin avec la nouvelle *whitelist,* i.e. les CB voisins du CB d'origine. Les autres CB des composantes, qui ne sont pas des voisins directs du CB d'origine, ne seront pas corrigés par Alevin. **[B]** Corrélation entre le nombre de voisins d'un CB et sa profondeur de séquençage. La corrélation est plus marquée pour l'échantillon ER1, pour lequel certains CB ont une très grande profondeur de séquençage. R est le coefficient de corrélation de Pearson. **[C]** Nombre de *reads* par CB d'origine sans correction (CB_o), ou après avoir corrigé les CB erronés voisins ($CB_o + CB_{e1}$), ou après avoir corrigé l'ensemble des CB erronés de la composante connexe ($CB_o + CB_e$). Les CB erronés voisins, notés CB_{e1} (pour CB erronés voisins de niveau 1), correspondent aux CB qui sont des voisins directs du CB d'origine et qui seront donc corrigés par Alevin avec la nouvelle *whitelist*. L'ensemble des CB erronés d'une composante connexe, notés CB_e (pour CB erronés) inclut tous les CB de la composante sauf le CB d'origine. Le chiffre en vert indique le pourcentage de *reads* gagnés après avoir corrigé les voisins directs, puis après avoir corrigé les CB résiduels dans chaque composante.

Algorithme 1 : Parcours en profondeur (DSF, pour *Depth-First Search*)

Entrée

MA : matrice d'adjacence des code-barres

$c = [\,]$: liste vide destinée à stocker les code-barres inclus dans la composante

$B = [b_1, \ldots, b_n]$: liste des n codes-barres

i : index d'un code-barres à partir duquel la recherche en profondeur est initiée

$V = \{b_1 : 0, \ldots, b_n : 0\}$: dictionnaire répertoriant le statut (visité ou non) des n codes-barres pour la recherche en profondeur des composantes connexes

Sortie

c : liste des code-barres inclus dans la composante

1. Initialisation

$b = B[i]$: code-barres à partir duquel on commence la recherche en profondeur

BV = codes-barres voisins du code-barres b

2. Mise à jour du dictionnaire V et de la composante en cours c

$V[b] = 1$

$c \mathrel{+}= b$

3. Extraction des voisins du code-barres b

Pour j de 1 à n :

    Si $MA[i, j] = 1$:

        $bv = B[j]$

        Si $V[bv] = 0$:

            $c = DFS(MA, c, B\,j, V)$

4. Retourne c

Algorithme 2 : Création de la nouvelle *whitelist*

Entrée

$B = [b_1, \dots, b_n]$: liste des n codes-barres

$P = \{b_1 : p_1, \dots, b_n : p_n\}$: dictionnaire des profondeurs de séquençages des n codes-barres

Sortie

W : *whitelist*

1. Initialisation

$W = [\,]$: *whitelist*

$C = [\,]$: composantes connexes

$V = \{b_1 : 0, \dots, b_n : 0\}$: dictionnaire répertoriant le statut (visité ou non) des n codes-barres pour la recherche en profondeur des composantes connexes

2. Génération de la matrice d'adjacence MA

Pour i allant de 1 à n :
  Pour j de 1 à n :
    Si $B[i] \neq B[j]$ ET
      $(distance\_hamming(B[i], B[j]) = 1$ OU
      $distance\_levenstein(B[i][:-1], B[j]) = 1$ OU
      $distance\_levenstein(B[i], B[j][:-1]))$:

        Alors $MA[i, j] = 1$
    Sinon :

        $MA[i, j] = 0$

4. Recherche des composantes connexes

Pour i allant de 1 à n :
  Si $V[B[i]] = 0$:
    $C \mathrel{+}= DFS(MA, [], B, i, V)$

4. Extraction du code-barres ayant la plus grande profondeur de séquençage dans chaque composante

Pour chaque composante c dans C :
  Si $taille(c) = 1$:
    $W \mathrel{+}= c$
  Sinon :
    $p = [\,]$
    Pour chaque code-barres b dans c :
      $p \mathrel{+}= P[b]$
    $W \mathrel{+}= max(p)$

5. Retourne W

Après avoir traité les échantillons de l'expérience pilote (DSP779) avec Alevin en utilisant la nouvelle *whitelist*, j'ai pu observer comme prévu que les distributions des métriques de qualité ayant permis de détecter les CB erronés (cf. section 3.1.1 du Chapitre 3) étaient plus homogènes qu'en utilisant la *whitelist* initiale de 3 000 CB (Figure 13). J'ai en outre constaté que le *knee point*, plus marqué dans la distribution des tailles de librairie que dans la distribution des profondeurs de séquençage, et apparaissait plus clairement avec la nouvelle *whitelist* qu'avec la *whitelist* initiale de 3 000 CB – en particulier pour les échantillons NT et CTRL – où il n'est quasiment pas visible (Figure 12). En effet, la distribution des tailles de librairies des CB erronés (présents dans la *whitelist* de 3 000 CB mais exclus de la nouvelle *whitelist*) se superpose à celle des CB valides, et semble « cacher » le *knee point*. En outre, avec la nouvelle *whitelist,* les CB valides vont récupérer pendant la correction les *reads* des CB erronés voisins qui étaient inclus dans la *whitelist* initiale. La taille de librairie accrue de ces CB va alors créer un décalage dans la distribution, laissant apparaître plus clairement le *knee point*.

Ces observations confirment que le *knee point* n'est pas une méthode adaptée pour la création d'une *whitelist* servant à la correction des CB. Non seulement, le *knee point* est plus marqué dans la distribution des tailles de librairie (donc après la correction des CB). De surcroît il ne semble pas possible, en utilisant un simple seuil sur les profondeurs de séquençage, de sélectionner uniquement des CB valides (sans erreur) associés à des cellules sans perdre un grand nombre de ceux-ci. En effet, la distribution des profondeurs de séquençage des CB valides et celle des CB erronés se superposent fortement, masquant d'autant plus le *knee point*, même dans la distribution des tailles de librairies.

Pour certains échantillons, comme l'échantillon ER2, le *knee point* n'est toujours pas visible, même en utilisant la nouvelle *whitelist* et la distribution des tailles de librairie. Il a été démontré que pour des échantillons de mauvaise qualité, comportant un nombre élevé de cellules endommagées, le *knee point* est moins marqué [27]. En effet, si l'échantillon comportait beaucoup de cellules lysées, les tailles de librairie « intermédiaires » de ces dernières ou des gouttelettes contenant alors beaucoup d'ARNm ambiant pourraient masquer le *knee point*. Par exemple pour ER2, peu de cellules ont été comptées avec l'hématimètre : il est donc possible qu'à cause de la transfection, ce dernier inclue beaucoup de cellules cassées masquant le *knee point*. Ce pourrait également être dû à des cellules mourantes ou quiescentes, qui contiennent peu d'ARNm dû à leur métabolisme respectivement en déclin et ralenti. Les cellules de l'échantillon ER2 sont en effet transfectées avec un siRNA bloquant ESR1, donc inhibant la prolifération cellulaire et pouvant ainsi mener à la mort ou la quiescence de ces dernières ; la procédure de transfection en elle-même peut également être un facteur de stress cellulaire qui déclenche la mort cellulaire.

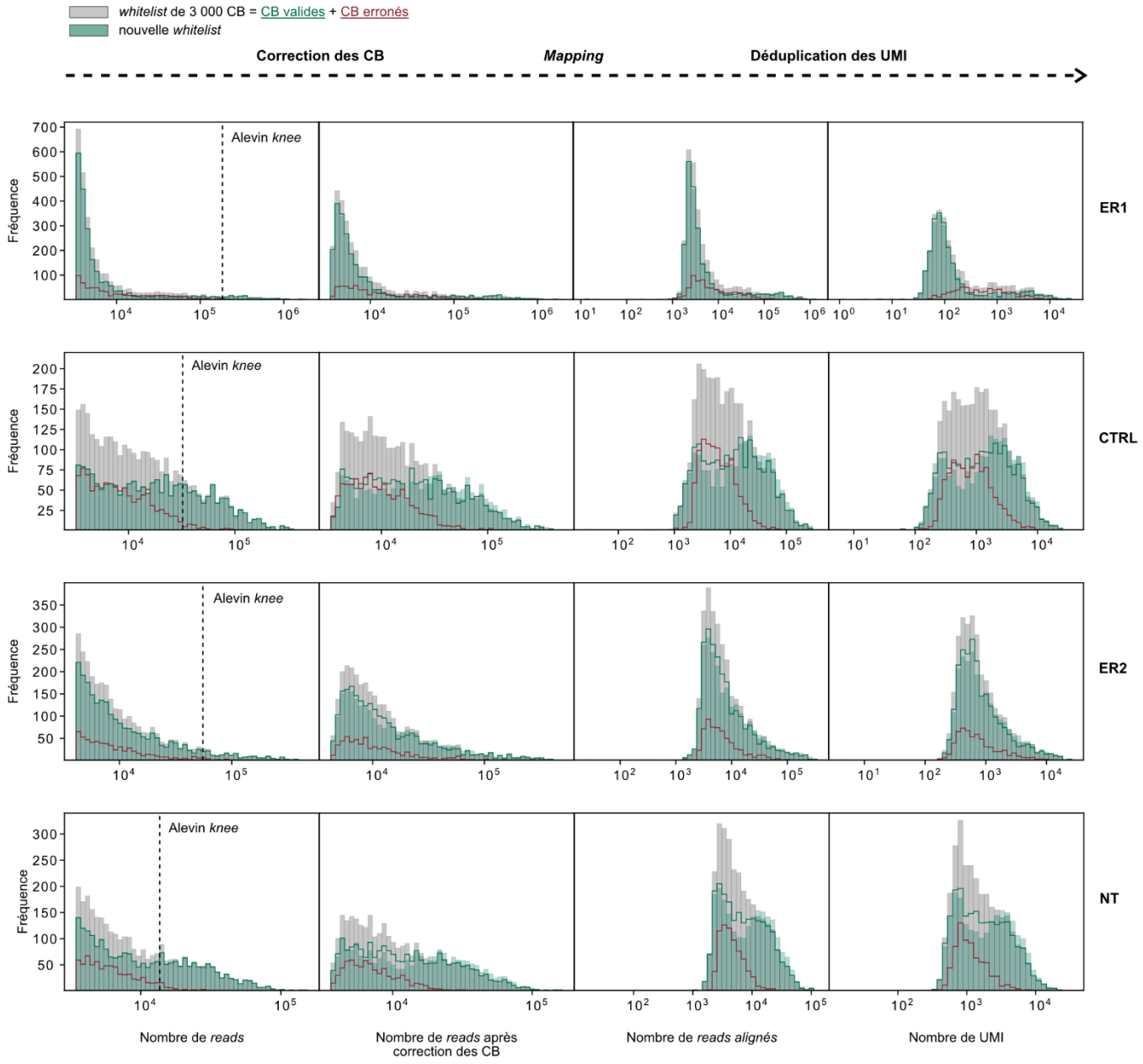


Figure 12 : Impact de la nouvelle *whitelist* sur la distribution des fréquences de CB

Pour chacun des 4 échantillons (ER1, CTRL, ER2, NT) de l'expérience pilote (lot de séquençage DSP779), les distributions suivantes sont représentées : celle de l'occurrence des 3 000 CB de la *whitelist* initiale après différentes étapes du prétraitement effectué par Alevin utilisant cette *whitelist* initiale (d_1, histogramme plein gris) ; celle de l'occurrence des CB de la nouvelle *whitelist* – générée grâce aux composantes connexes – après différentes étapes du prétraitement effectué par Alevin utilisant cette nouvelle *whitelist* (d_2, histogramme plein vert). La distribution d_1 est décomposée en deux distributions : celle des CB erronés, correspondant aux CB exclus de la nouvelle *whitelist* (d_{1e}, histogramme ligne rouge) ; celle des CB valides, correspondant aux CB gardés dans la nouvelle *whitelist* (d_{1v}, histogramme ligne verte). Avant tout prétraitement, les distributions correspondent aux distributions des profondeurs de séquençage des CB, on a alors $d_{1v} = d_2$. Cependant, dès lors que la correction des CB a été effectuée, $d_{1v} \neq d_2$. On peut voir que d_{1e} se superpose à d_{1v} et « cache » le *knee point*, qui semble être surestimé par Alevin. Le *knee point* dans d_{1v} apparaît plus marqué après les différentes étapes de prétraitement, ce qui pourrait expliquer pourquoi Alevin surestime celui-ci. Il se pourrait donc que le *knee point* soit plus facilement identifié après déduplication des UMI (i.e.

dans la distribution des tailles de librairie). Dans d_2, le *knee point* semble également plus marqué que dans d_{1v} après déduplication des UMI, suggérant que la nouvelle *whitelist* facilite l'identification de ce dernier.

Une étude comparant différentes technologies de scRNA-seq utilisant des gouttelettes [16] a montré que le *knee point* était peu marqué dans les données issues de la technologie Drop-Seq. Les auteurs l'ont justifié par la taille des billes plus variable dans la technologie Drop-Seq que dans les autres. Une autre explication pourrait être que le dispositif microfluidique dans la technologie Drop-Seq est plus agressif, menant à un nombre plus conséquent de cellules endommagées. L'utilisation du *knee point* ainsi que des métriques de qualité permettant d'éliminer les gouttelettes vides ou les cellules endommagées sera plus amplement discutée dans le chapitre 4.

Pour tous les échantillons des 3 projets et 4 lots de séquençage (cf. section 2.1.1), la matrice d'expression a donc été obtenue en effectuant tout d'abord un *trimming* des *reads* comme décrit dans la section 2.3.2, puis en utilisant Alevin avec une nouvelle *whitelist* générée à partir des composantes connexes, ainsi qu'une longueur de *k-mers* égale à 19 et que des *decoys*. Ces derniers ont été générés à partir de l'annotation du transcriptome de référence *gencode* v34 et du génome de référence GRh38. Pour les échantillons du projet 3, la séquence nucléotidique de la GFP a également été incluse.

CHAPITRE 3 ORIGINE EXPÉRIMENTALE DES ERREURS DE CODES-BARRES CELLULAIRES

Le chapitre 2 présentait la procédure que j'ai mise en place pour acquérir une matrice d'expression à partir de fichiers FASTQ démultiplexés, en utilisant notamment le logiciel Alevin. Ayant observé dans certains échantillons, en particulier ceux de l'expérience pilote (DSP779), une quantité non-négligeable de codes-barres cellulaires (CB) erronés inclus dans les matrices d'expression générées par Alevin – qui intègre pourtant une procédure de correction des CB –, j'ai mis en place une stratégie y remédier.

Dans le présent chapitre, j'étudie de manière plus approfondie ces CB erronés, vraisemblablement issus de substitutions « systématiques », en caractérisant différents types d'erreurs/de substitutions.

3.1 Mise en évidence de différents groupes d'erreurs spécifiques d'un lot de billes

3.1.1 Découverte de CB erronés

C'est en analysant les 4 échantillons de l'expérience pilote (DSP779) avec certaines métriques de qualité que j'ai initialement découvert, parmi les 3 000 CB constituant la *whitelist* et retenus dans la matrice d'expression, des CB erronés qui avaient échappé à la procédure de correction d'Alevin.

J'ai tout d'abord constaté que pour chaque échantillon, une partie des CB avaient un taux de déduplication (td) [4] plus bas que les autres CB (Figure 13), c'est-à-dire qu'ils étaient associés à un nombre moindre de duplicatas PCR par transcrit. Le td mesure en effet le pourcentage de *reads* éliminés par la déduplication des UMI et est ainsi représentatif du taux de duplication, i.e. du pourcentage de duplicatas PCR séquencés. J'ai ensuite calculé pour chaque CB de chaque échantillon le coefficient de Pearson maximal (R_{max}) entre son profil d'expression et ceux des 2 999 autres CB (cette métrique de qualité est parfois utilisée pour éliminer les CB indésirables issus par exemple de gouttelettes vides, qui auront des R_{max} plus bas puisque leurs profils d'expression sont principalement constitués de bruit provenant des ARNm ambiants). Ce faisant, il m'est apparu que l'ensemble des CB présentant un td moindre avaient de

[4] $td = 1 - nUMI/nreads$, avec $nUMI$ le nombre de UMI et $nreads$ le nombre de *reads*.

surcroit un R_{max} plus élevé que celui des autres CB du même échantillon avec une profondeur de séquençage ou une taille de librairie similaire (cf. Figure 13). Pour chacun de ces CB anormaux (CB_i), j'ai donc recherché quel était l'autre CB (CB_j) de l'échantillon pour lequel le R_{max} avait été obtenu. J'ai alors découvert que la distance d'édition (cf. section 2.4.4) entre un CB_i et son CB_j était systématiquement égale à 1 (Figure 13), suggérant que l'ensemble des CB anormaux corresponde en réalité à des CB erronés – ayant un profil d'expression fortement corrélé à celui de leur CB d'origine du fait qu'ils représentent la même cellule. La présence de CB erronés dans les *whitelists* de 3 000 CB utilisées pour générer les matrices d'expression des différents échantillons a ensuite été confirmée par ma démarche décrite dans la section 2.4.3.

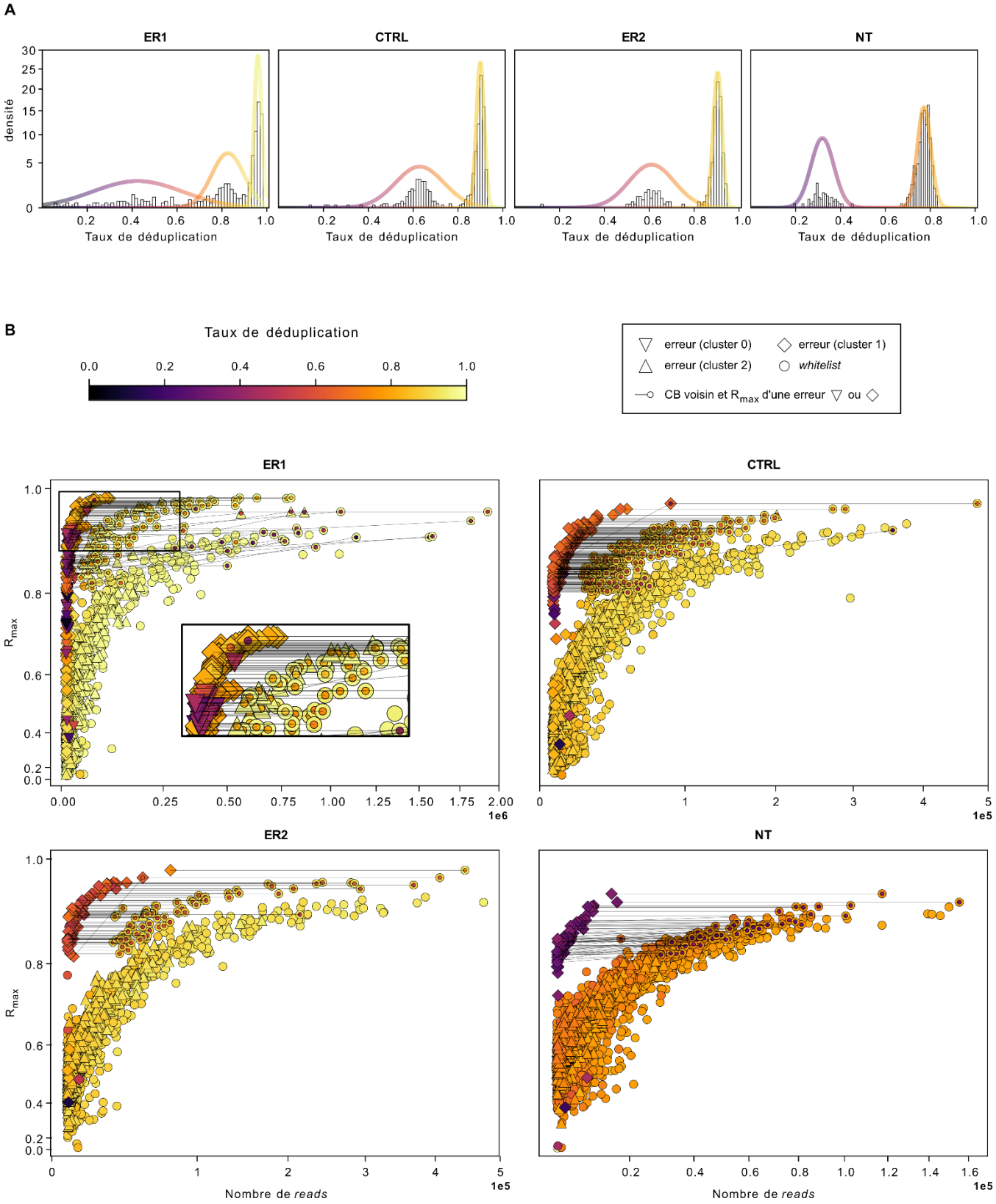
3.1.2 Répartition des CB erronés en différents sous-groupes

Cependant, après avoir suivi la procédure décrite dans la section 2.4.4 pour identifier tous les CB erronés et leurs CB d'origine respectifs grâce à des composantes connexes, je me suis aperçue que la majeure partie des CB erronés avaient en fait un td et un R_{max} normaux (Figure 13). J'ai donc cherché à comprendre l'origine de ces différences observées parmi les CB erronés.

Pour cela j'ai tout d'abord réparti les CB erronés en différents groupes selon leur td en utilisant un modèle de mixture de gaussiennes. J'ai défini le nombre de *clusters* pour chaque échantillon en me basant sur une inspection visuelle des distributions de leurs td. J'ai donc spécifié 3 *clusters* pour l'échantillon ER1 (*clusters* 0, 1 et 2), et 2 *clusters* pour les autres échantillons (*clusters* 1 et 2).

3.1.3 Analyse des substitutions dont les CB erronés sont issus

Les CB erronés étant majoritairement engendrés par des substitutions, j'ai ensuite cherché à identifier la nature de ces dernières. Pour chaque CB erroné exclu de la *whitelist*, j'ai donc recherché le CB dont il dérive, qui ne correspond pas exactement à la tache où l'on cherche à déterminer le CB d'origine identifiant une cellule. Pour mieux comprendre la différence entre ces derniers, on peut imaginer chaque composante connexe comme un arbre, dont la racine serait le CB d'origine identifiant la cellule, et la descendance l'ensemble des CB erronés. Les substitutions subies par le CB d'origine génèrent des CB « enfants », qui peuvent à leur tour avoir des enfants. Ici, on ne cherche donc pas à identifier la racine de l'arbre mais plutôt à retrouver toutes les paires parent⟶enfant.


A
ER1
CTRL
ER2
NT
densité
Taux de déduplication
B
Taux de déduplication
0.0
0.2
0.4
0.6
0.8
1.0
erreur (cluster 0)
erreur (cluster 1)
erreur (cluster 2)
whitelist
CB voisin et Rmax d'une erreur ▽ ou ◇
Rmax
1e6
1e5
Nombre de reads

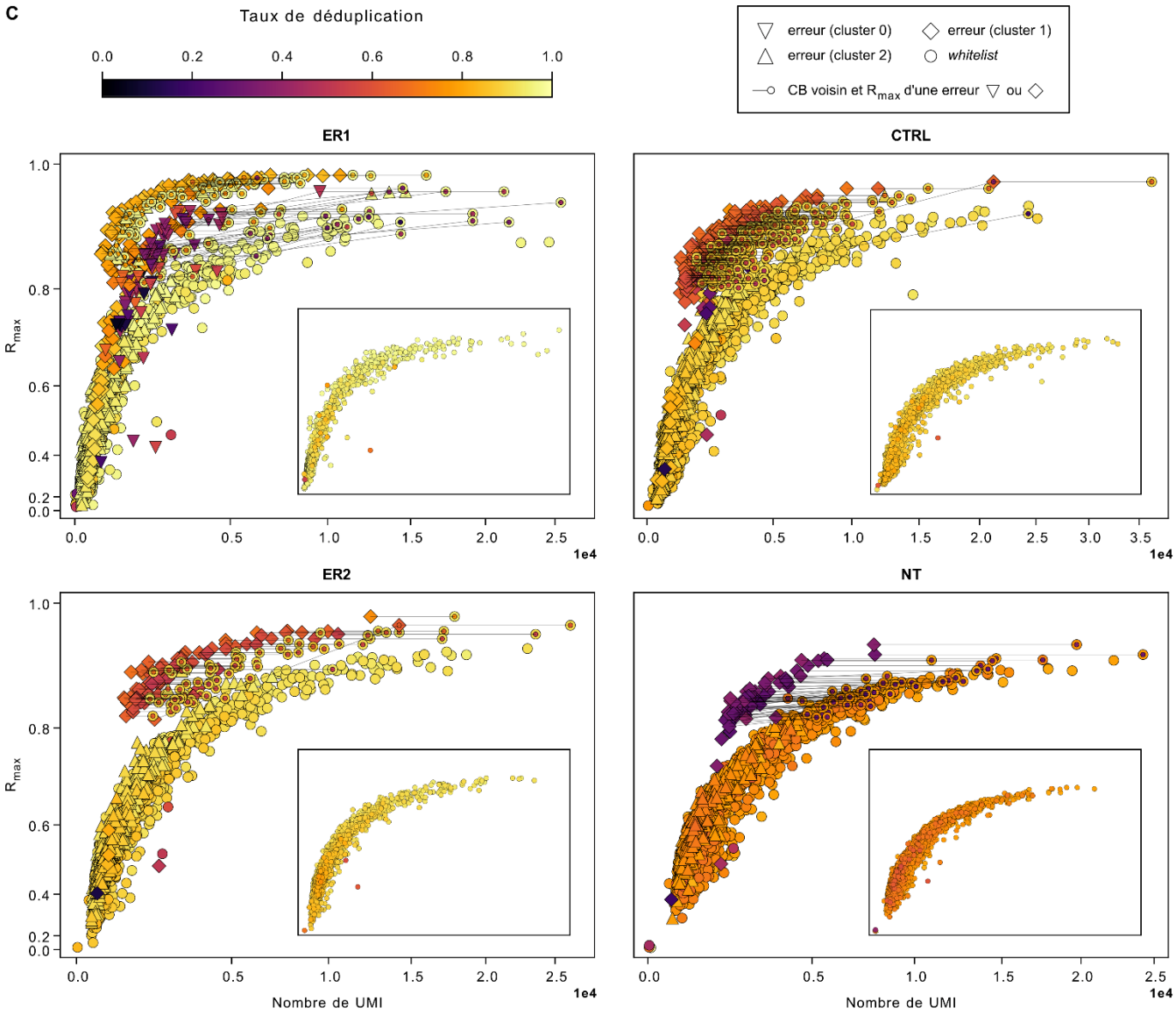


Figure 13 : Observation de différents groupes d'erreurs de CB avec des propriétés distinctes.

[A] Groupes d'erreurs de CB identifiés dans la whitelist de 3 000 CB de chaque échantillon. Les groupes sont identifiés grâce à un clustering utilisant un mélange gaussien pour modéliser la distribution des taux de déduplication des CB erronés présents dans la whitelist. 3 composantes gaussiennes ont été utilisées pour l'échantillon ER1 et 2 pour les autres échantillons, en se basant sur l'inspection visuelle des distributions des taux de déduplication. Les clusters de CB erronés 0, 1 et 2 correspondent respectivement aux composantes gaussiennes avec une moyenne de la plus basse à la plus élevée. Chaque CB est attribué à la composante normale pour laquelle la probabilité prédite est la plus grande. **[B]** Relation entre la profondeur de séquençage (nombre de reads) d'un CB et le R_{max}.de ce dernier Le R_{max}d'un CB dénote ici la corrélation maximale entre son profil d'expression et ceux des autres CB de la whitelist. Un cercle avec un point de couleur au milieu correspond à un CB voisin (distance d'édition égale à 1) d'un CB erroné du cluster 0 ou 1 et avec lequel le R_{max} de ce même CB erroné a été obtenu. Un tel CB correspond vraisemblablement au CB parent du CB erroné en question (i.e. au CB initial sur lequel l'erreur est survenue, pouvant correspondre au CB d'origine ou à un CB déjà issu d'une erreur), auquel il est relié par une ligne – tracée seulement si $R_{max} > 0.8$ pour ne pas surcharger la visualisation. La couleur du point au milieu du cercle représentant le CB parent correspond au taux de déduplication du CB erroné. **[C]** Relation entre la taille de librairie (nombre de UMI) d'un CB et son R_{max}. Les miniatures correspondent à la même visualisation après avoir utilisé Alevin avec la nouvelle whitelist. On peut constater qu'Alevin a bien corrigé les erreurs, les clusters de CB erronés n'apparaissant plus dans les visualisations.

Ainsi, pour chaque CB erroné (hors de la nouvelle *whitelist*), j'ai identifié son voisin (distance de Hamming égale à 1) avec la plus grande profondeur de séquençage. J'ai ensuite analysé les substitutions parent⟶enfant, (Figure 14) et j'ai pu observer des substitutions systématiques caractéristiques de chaque *cluster*. Pour tous les échantillons, le *cluster* de CB erronés avec un td « normal » (*cluster* 2) compte beaucoup de substitutions G⟶N en dernière position. Le *cluster* 1, avec un td plus faible, correspond principalement à des substitutions de A⟶T en première position. Pour l'échantillon ER1, les substitutions caractérisant le *cluster* 0 (avec un td encore plus faible que le *cluster* 1) sont plus diversifiées et moins fréquentes. Ces *patterns* de substitutions se répètent pour les échantillons d'une même expérience mais également pour deux expériences utilisant le même lot de billes, comme mentionné par l'équipe Drop-Seq dans leur « *cookbook* computationnel » [28] qui évoque des erreurs de synthèse de CB systématiques. Ce dernier ne mentionne cependant pas la présence de différentes catégories d'erreurs systématiques engendrant chacune des caractéristiques distinctes dans le profil d'expression.

Sur la base de ces observations, j'ai émis l'hypothèse que les CB des différents groupes que j'observais, avec un td normal ou faible, provenaient respectivement d'erreurs systématiques de synthèse et de séquençage. En effet, j'ai pu observer dans le rapport FASTQC (Figure 6) des scores de qualité plus bas pour la première position, sur laquelle les erreurs du *cluster* 1 comptent un nombre élevé de substitutions. De plus, en resituant les deux types d'erreurs dans le fil du protocole expérimental, on peut constater que leurs caractéristiques (en termes de td et R_{max}) recoupent celles des groupes observés.

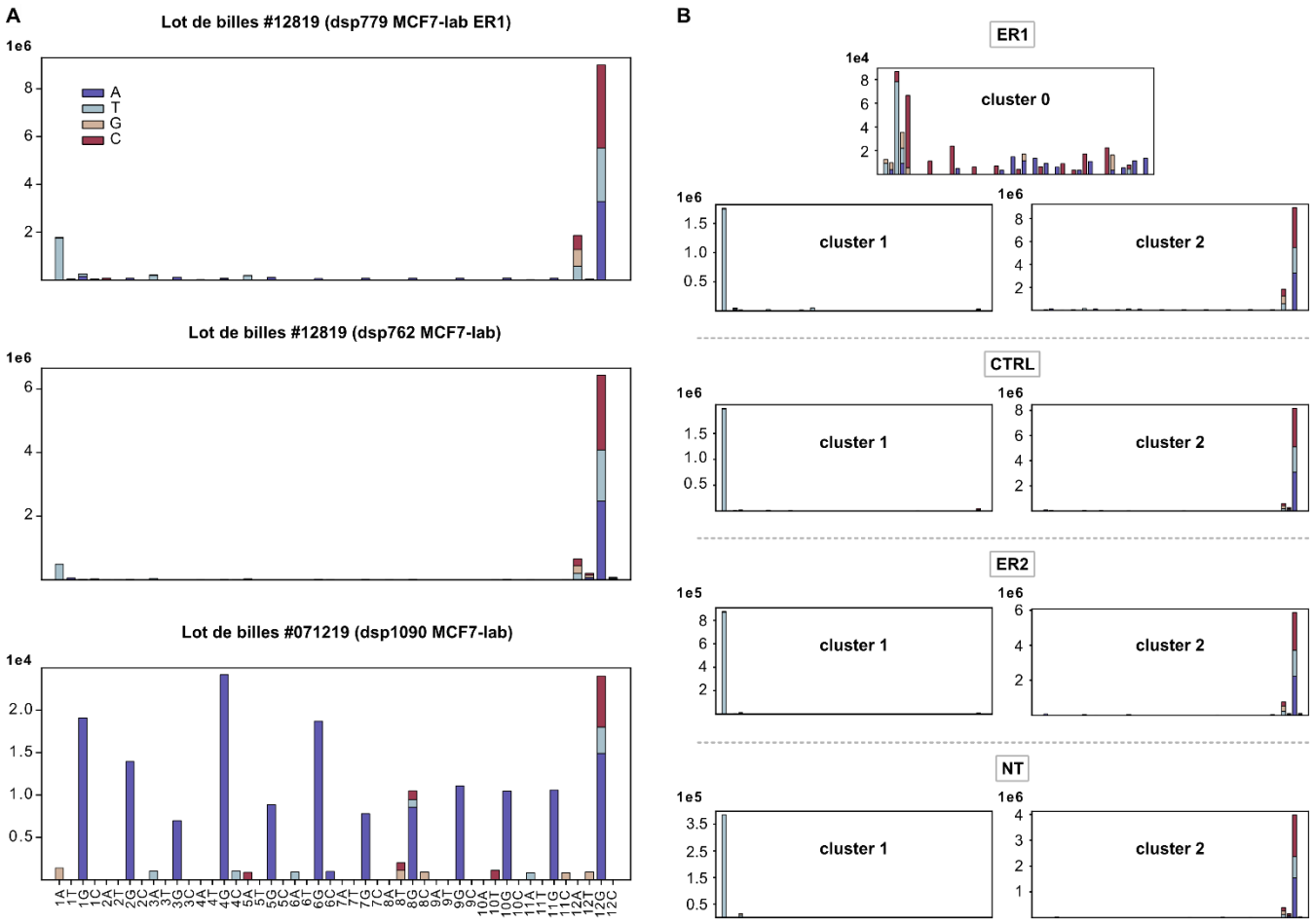


Figure 14 : Substitutions observées pour chaque paire de CB voisins.

[A] Fréquence des substitutions par position pour les échantillons ER1 (lot de séquençage DSP779, lot de billes 12819), MCF7-labo-1 (lot de séquençage DSP762, lot de billes 12819) et MCF7-labo-2 (lot de séquençage DSP1090, lot de billes 071219). Les substitutions sont basées sur les paires de CB voisins (distance de Hamming égale à 1), comportant chacune un CB erroné (exclu de la *whitelist*) et un CB parent – CB initial sur lequel l'erreur est survenue pouvant correspondre au CB d'origine ou à un CB déjà issu d'une erreur. Pour chaque CB erroné, le CB parent est désigné comme le CB voisin (distance de Hamming égale à 1) avec la plus grande profondeur de séquençage (nombre de *reads*). La fréquence des substitutions est déterminée en comptabilisant le nombre de *reads* du CB erroné pour chaque paire de CB voisins. **[B]** Fréquence des substitutions par position pour les différents *clusters* de CB erronés identifiés dans les échantillons du lot de séquençage DSP779. La fréquence des substitutions pour un *cluster* donné est obtenue en considérant seulement les paires de voisins dans lesquelles le CB erroné appartient à ce *cluster*.

3.2 Identification du type d'erreur expérimentale à l'origine de chaque groupe de CB erronés

3.2.1 Caractérisation des erreurs de synthèse et de séquençage

On peut voir les erreurs de synthèse et de séquençage comme des échantillonnages des séquences survenant à différents moments de la préparation de la librairie (Figure 15). Pour les erreurs de synthèse, présentes sur les billes au moment de la capture des molécules d'ARNm, l'échantillonnage sera effectué sur les ARNm, tandis que pour les erreurs de séquençage, il sera effectué sur les ADNc amplifiés. Les erreurs de synthèse vont scinder les cellules en « sous-cellules », chacune d'entre elles comportant un sous-ensemble différent de molécules d'ARNm et donnant lieu à une sous-librairie qui sera amplifiée. Les erreurs de séquençage, quant à elles, vont directement scinder les librairies cellulaires amplifiées en sous-librairies, comportant chacune un sous-ensemble des duplicatas PCR de chaque molécule. En effet, toutes les molécules d'une librairie cellulaire ont la même probabilité d'être touchées par une erreur de séquençage spécifique sur leur CB, puisqu'elles comptent un nombre quasiment identique – à quelques biais PCR près – de duplicatas. Ainsi, les erreurs de séquençage vont toucher un certain nombre de duplicatas de chaque molécule, tandis que les erreurs de synthèse vont toucher tous les duplicatas de certaines molécules.

Pour les erreurs de séquençage, une même molécule peut être retrouvée à la fois dans la librairie d'un CB erroné et dans celle de son CB d'origine, si chacune reçoit au moins un duplicata de cette molécule – i.e. si l'erreur touche au moins un et au plus $n - 1$ duplicatas, où n est le nombre de duplicatas de la molécule. Par exemple, pour une erreur de séquençage sur un CB dont le taux serait égal à 0,5 la moitié des duplicatas de chaque molécule serait retenue par le CB d'origine, et l'autre moitié serait récupérée par le CB erroné – cet exemple fait abstraction des autres erreurs qui pourraient toucher ce CB. Toutes les molécules étant retrouvées dans les deux sous-librairies respectives, les deux profils d'expression estimés après la déduplication des UMI seraient identiques – et seraient même identiques à celui qui aurait été estimé à partir de la librairie initiale –, leur corrélation serait donc égale à 1. Leur td serait également identique, mais plus petit que celui des autres CB (valides) de l'échantillon – et que celui qui aurait été obtenu avec la librairie initiale. En revanche, si l'on considère une erreur dont le taux est plus petit, les sous-librairies des CB erronés récupéreront moins de duplicatas par molécule et auront donc un td plus faible que celle du CB d'origine, qui aura conservé la majorité des duplicatas. De surcroît, la corrélation entre les deux profils d'expression sera plus faible, car certaines molécules ne seront plus prises en compte

dans la librairie du CB erroné. Néanmoins, si la profondeur de séquençage augmente, il y aura davantage de duplicatas lus par molécule et ainsi, même une erreur dont le taux est faible pourra toucher un nombre conséquent de molécules distinctes. La corrélation entre les profils d'expression estimés pour les CB erronés et ceux estimés pour leurs CB d'origine va donc augmenter avec la profondeur de séquençage – tout comme les td de l'ensemble des CB, qu'ils soient valides ou erronés.

Les différentes sous-librairies issues d'une erreur de synthèse – que ce soit celles associées aux CB erronés ou au CB d'origine –, retenant chacune tous les duplicatas des molécules échantillonnées, devraient quant à elles avoir un td semblable à celui d'un CB valide. De plus, puisque ces dernières ne comptent aucune molécule en commun, les profils d'expression qui en découleraient seraient moins similaires que ceux estimés à partir des sous-librairies issues d'une erreur de séquençage.

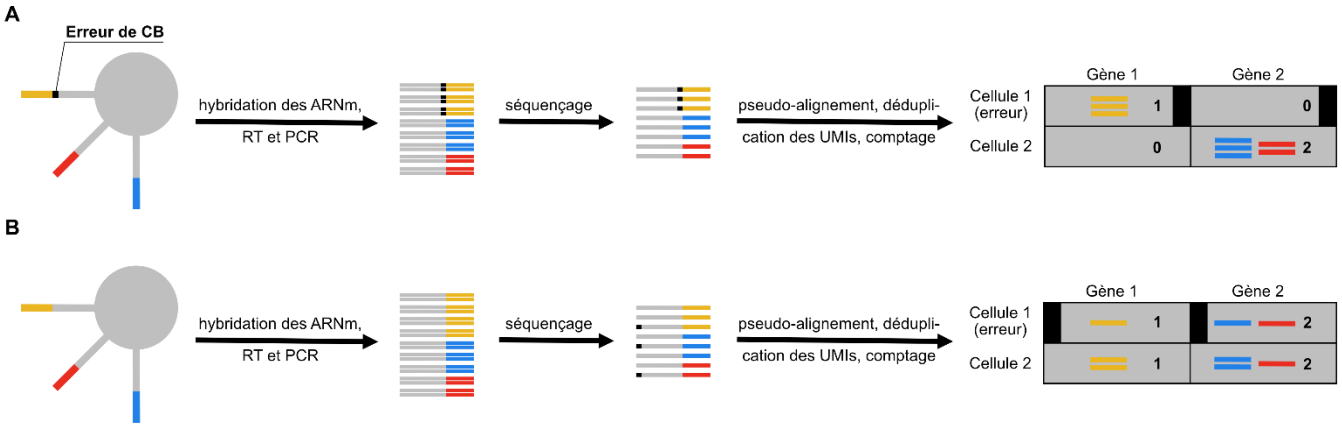


Figure 15 : Caractérisation de différents types d'erreurs de CB.

[A] Erreurs de synthèse. Lors d'une erreur de synthèse, qui survient en amont de l'amplification PCR, tous les duplicatas d'une même molécule seront impactés : les molécules seront partagées entre le CB d'origine et le CB erroné. **[B]** Erreurs de séquençage. Lors d'une erreur de séquençage, qui survient en aval de l'amplification PCR, quelques duplicatas de chaque molécule seront impactés : les duplicatas seront partagés entre le CB d'origine et le CB erroné. Rectangles dans des tons gris : CB ; rectangles colorés : UMI ; petit carré noir sur un CB : erreur dans la séquence d'un CB.

1.1.1 Recoupement des différents groupes de CB erronés avec les erreurs de séquençage et de synthèse

Les erreurs de séquençage pourraient donc être à l'origine des R_{max} et des td aberrants observés pour certains CB (*clusters* 0 et 1) des échantillons de l'expérience pilote. On peut également noter que les R_{max} des CB du *cluster* 1 sont plus élevés dans ER1, en raison de la profondeur de séquençage plus élevée des

CB de cet échantillon (différence discutée dans la section 4.1.1 du chapitre 4). Cela explique également pourquoi le *cluster* 0 est identifié uniquement dans l'échantillon ER1. En effet les CB qui le composent, avec des td et des R_{max} plus petits que ceux du *cluster* 1, proviennent probablement d'erreurs de séquençage dont les taux d'erreur sont plus faibles – n'apparaissant ainsi qu'avec des profondeurs de séquençage plus grandes. Les CB du *cluster* 2, caractérisés par un td et un R_{max} normaux pourraient quant à eux être la conséquence d'erreurs de synthèse.

Pour confirmer cette hypothèse j'ai donc tout d'abord comparé les Q-scores moyens des CB valides à ceux des CB erronés de chacun des 3 *clusters* (les Q-scores estiment la qualité du séquençage pour chaque position d'un *read)*. En accord avec mon hypothèse, j'ai pu observer que la distribution des Q-scores moyens des CB du *cluster* 2 était similaire à celle des CB valides tandis que les Q-scores des *clusters* 0 et 1 étaient plus bas (Figure 16).

J'ai ensuite évalué pour chaque CB erroné de chaque *cluster* la fraction des UMI qu'il partageait avec son CB « parent ». À titre de comparaison, j'ai également calculé des fractions « contrôles », permettant d'estimer le nombre d'UMI partagés (fortuitement) par deux CB indépendants (Figure 16).

Pour cela j'ai calculé pour chaque paire $CB_{parent} \longrightarrow CB_{erroné}$, la fraction des UMI d'un $CB_{valide} \overset{lib}{\sim} CB_{erroné}$ partagée avec un autre $CB_{valide} \overset{lib}{\sim} CB_{parent}$. Ici, $CB_i \overset{lib}{\sim} CB_j$ signifie que le nombre d'UMI associés au CB_i est proche du nombre d'UMI associés au CB_i. En effet, la fraction calculée va dépendre de la taille de librairie des deux CB considérés. Par exemple, si l'on considère un CB parent associé à un très grand nombre d'UMI, proche de 4^8 (nombre maximal d'UMI distincts), n'importe quel autre CB partagera une fraction importante de ses UMI avec ce CB parent.

Pour calculer ces fractions contrôle, j'ai également exploré la possibilité de générer *in silico* des UMI et d'effectuer, pour chaque paire de $CB_{parent} \longrightarrow CB_{erroné}$, deux tirages aléatoires de n_e et de n_p UMI artificiels, n_e et n_p dénotant respectivement le nombre d'UMI associés au CB erroné et au CB parent. Cependant, les fractions obtenues étaient bien plus petites que celles obtenues avec la première démarche. J'ai pu expliquer cela par le fait que les différents nucléotides n'étaient pas présents en quantités égales dans les UMI des échantillons (avec notamment un %G sensiblement plus grand à chaque position, cf. Figure 6), alors que les UMI que j'avais générés *in silico* étaient basés sur une distribution uniforme des différents nucléotides à chaque position. Cela illustre bien la limitation des simulations computationnelles, qui ne sont pas capables d'intégrer tous les paramètres et les aléas d'une expérience effectuée en laboratoire. J'ai donc préféré la première démarche, utilisant les CB valides des différents échantillons pour calculer les fractions contrôles.

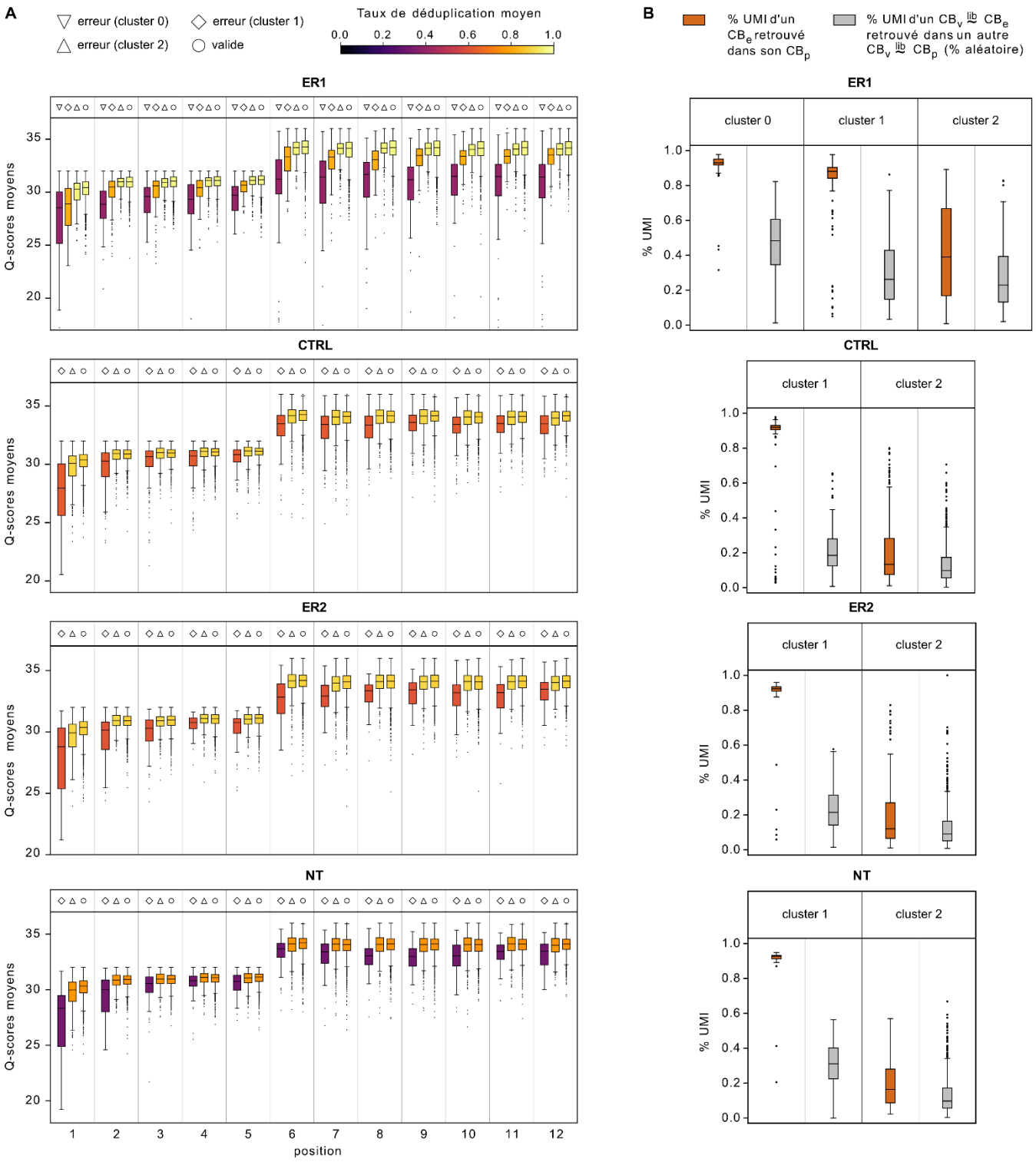


Figure 16 : Validation des différents types d'erreurs de CB.

[A] Distribution des Q-scores moyens par position des CB erronés de chaque *cluster* et des CB valides – inclus dans la nouvelle *whitelist*. Le Q-score moyen pour un CB donné est calculé à partir de l'ensemble des *reads* de ce CB. **[B]** Distribution de la fraction des UMI associés aux CB erronés de chaque *cluster* qui correspondent également à des UMI associés à leur CB parent. Pour chacun des CB erronés d'un *cluster* donné, deux CB valides sont également

sélectionnés selon leurs profondeurs de séquençage respectives, l'une étant la plus proche de celle du CB erroné (premier CB valide) et l'autre étant la plus proche de celle du CB parent (second CB valide). La fraction des UMI du premier CB valide retrouvée parmi les UMI du second CB valide est ensuite calculée. Cela permet de déterminer la fraction d'UMI partagée par hasard entre deux CB indépendants et de profondeur de séquençage proches de celles du CB erroné et du CB parent considérés. $CB_i \sim^{lib} CB_j$: CB_i proche de CB_j en termes de profondeur de séquençage ; CB_e : CB erroné ; CB_p : CB parent ; CB_v : CB valide.

J'ai ainsi pu constater que les CB erronés des *clusters* 0 et 1 partageaient la quasi-totalité de leurs UMI (fraction proche de 1) avec leur CB parent, ce qui est caractéristique des erreurs de séquençage (comme expliqué dans la section 3.2.1). En revanche il s'est avéré que les CB erronés du *cluster* 2 semblaient partager avec leur CB parent des fractions aléatoires de leurs UMI – proches des fractions contrôles.

La mise en évidence des différences entre les CB erronés des *clusters* 0 et 1, d'une part, et ceux du *cluster* 2, d'autre part, a donc permis d'identifier les deux types d'erreurs expérimentales dont ils proviennent respectivement : des erreurs de séquençage versus des erreurs de synthèse.

L'étude présentée dans ce chapitre a permis d'apporter une meilleure compréhension des données issues de la technologie Drop-Seq. Elle a également permis d'identifier des métriques qui peuvent être utilisées pour détecter un CB provenant d'une erreur de séquençage, et qui pourraient donc être exploitées pour l'étape de correction des CB. Ces métriques en revanche ne permettent pas de détecter les CB issus d'erreurs de synthèse, et ne seront donc pas suffisantes à elles seules pour détecter l'ensemble des CB erronés.

CHAPITRE 4 PRÉPARATION DE LA MATRICE D'EXPRESSION

Le chapitre 2 présentait la procédure que j'ai mise en place pour acquérir une matrice d'expression à partir de fichiers FASTQ démultiplexés en utilisant le logiciel Alevin. Le chapitre 3 constituait une parenthèse, décrivant les différents types d'erreurs pouvant survenir sur les codes-barres cellulaires (CB) et les caractéristiques qui en découlent. Ce chapitre quant à lui présente les différentes étapes permettant de préparer, ou « nettoyer » la matrice d'expression avant d'effectuer des analyses en aval.

4.1 Contrôle qualité des cellules

4.1.1 Métriques de qualité

Le contrôle qualité (QC) des cellules, ou plutôt des codes-barres cellulaires (CB), consiste à filtrer tous les CB qui ne sont pas d'intérêt pour les analyses en aval, comme ceux issus de gouttelettes vides ou encore ceux correspondant à des doublets cellulaires (la plupart des multiplets étant des doublets). Les CB associés à des cellules cassées, mourantes ou stressées peuvent être également exclus lors du QC mais ils sont parfois retenus dans certaines études, telles que celles menées sur des tissus nerveux, difficiles à dissocier.

Les gouttelettes vides auront en général une librairie peu complexe, constituée uniquement d'ARNm ambiants, tandis que les doublets cellulaires auront une librairie particulièrement complexe, puisque regroupant deux librairies cellulaires. Les gouttelettes vides et les doublets cellulaires sont donc généralement éliminés en appliquant un seuil sur une métrique de qualité reflétant la complexité de la librairie, comme le nombre de *reads*, le nombre d'UMI, ou encore le nombre de gènes détectés (i.e. dont l'expression est non nulle). Le coefficient de corrélation de Pearson maximal entre le profil d'expression d'un CB et ceux des autres CB, que je note R_{max}, est une autre métrique parfois utilisée. L'assomption sous-jacente est que le profil d'expression associé à un CB issu d'une gouttelette vide aura un faible R_{max} puisqu'il représente du « bruit ambiant » (débris cellulaires présents dans la solution, qui ont été encapsulés). A l'inverse, les profils d'expression de deux CB associés à des cellules d'une même sous-population corréleront fortement. Cette métrique pourrait éliminer à tort les cellules uniques

représentantes de leur population dans le jeu de données. Cependant cela n'est pas réellement problématique, puisqu'il faut au moins deux cellules appartenant à une population pour la caractériser. Afin d'identifier le seuil approprié pour éliminer les gouttelettes vides, il est courant de rechercher un *knee point* dans la distribution de l'une de ces métriques, le plus souvent dans celle du nombre d'UMI (cf. sections 2.4.1 et 2.4.2 du Chapitre 2). Ce dernier est censé représenter la démarcation entre les CB dont le profil d'expression correspond aux ARNm ambiants encapsulés dans des gouttelettes vides et ceux dont le profil d'expression correspond à de véritables cellules. Si le *knee point* n'est pas visible ou peu marqué, cela peut être révélateur d'une mauvaise qualité d'échantillon, comme expliqué dans la section 2.4.4.

Un bon nombre de cellules cassées ou stressées peuvent être éliminées en même temps que les gouttelettes vides, puisqu'elles contiennent ou expriment moins d'ARN, mais certaines seront conservées car leurs CB passeront au travers du filtre défini par le *knee point*. Une ou plusieurs métriques reflétant la « santé » cellulaire sont donc utilisées en complément de celle employée pour les gouttelettes vides. La métrique la plus commune pour éliminer les cellules cassées ou stressées résiduelles, appelée fraction mitochondriale, est pour un CB donné la proportion des *reads* (dédupliqués) associés à ce dernier qui ont été alignés sur un gène du génome mitochondrial – il s'agit en d'autres termes de l'expression mitochondriale. En effet, les gènes mitochondriaux sont fortement exprimés par les cellules stressées et sont impliqués dans les signaux déclenchant l'apoptose (mort cellulaire). De plus, lorsque les cellules sont cassées, elle se « vident » de leur contenu cytoplasmique qui sera perdu, mais les ARN mitochondriaux restent quant à eux bloqués dans la membrane mitochondriale. Les ARN mitochondriaux représenteront alors une plus grande proportion des ARN hybridés sur la bille et l'expression du génome mitochondrial apparaîtra plus grande. Inversement, les gènes codant pour des protéines ribosomales (RP, *ribosomal protein*), qui sont parmi les gènes les plus exprimés et dont les ARN sont localisés dans le cytoplasme donc perdus lors de la perforation de la membrane, apparaîtront moins exprimés.

Les ARNm, purifiés et capturés au moyen d'oligo-dT, sont censés être les seuls séquencés mais un certain nombre d'autres ARN (dits contaminants) peuvent être également capturés s'ils possèdent de petits tronçons de poly-A au sein de leur séquence [29]. C'est en particulier le cas lorsque la quantité d'ARNm est limitée – par exemple, quand la cellule est cassée – ou que les ARN contaminants sont abondants. Aisni, la fraction d'ARN ribosomaux (ARNr) – proportion des *reads* dédupliqués alignés sur un gène ribosomal – qui représentent au moins 80% des ARN d'une cellule, est parfois utilisée dans le QC. Dans l'échantillon ER1, la fraction d'ARNr du génome nucléaire était quasi nulle dans toutes les cellules, mais celle des ARNr mitochondriaux était significative et révélatrice de cellules cassées (Figure 18).

La fraction d'ARN long non-codant (ARNlnc) peut elle aussi permettre de détecter les cellules cassées. Tout comme les ARN mitochondriaux, les ARNlnc, qui sont eux localisés dans le noyau, seront conservés lors d'une lyse cellulaire. La quantité d'ARNm d'une cellule cassée étant limitée, il y aura davantage d'ARNlnc hybridés sur la bille – qu'ils possèdent ou non une queue poly-A. L'expression du gène MALAT-1 codant pour un ARNlnc est ainsi parfois utilisée pour détecter les cellules cassées [30]. Des fractions mitochondriale et d'ARNlnc toutes deux très élevées pourrait être associées à des cellules très endommagées, complètement vidées de leur contenu cytoplasmique ou presque et n'ayant conservé que leurs organites – parmi lesquels les mitochondries et le noyau, contenant des ARN qui seront capturés.

Le taux de *mapping* est une autre métrique utilisée dans la littérature pour le QC des cellules [22, 31]. Un taux de *mapping* bas peut effectivement être l'indice d'une forte dégradation des ARNm, laquelle engendre des fragments d'ADNc plus courts dans la librairie. Les *reads* séquencés seront alors eux-mêmes plus courts (ou pollués par des séquences d'adaptateurs si le *trimming* n'a pas été effectué) et seront donc plus compliqués à aligner sur le génome ou le transcriptome de référence. Dans l'échantillon ER1, les cellules ayant un taux de *mapping* particulièrement bas présentent également une fraction d'ARNlnc élevée, laissant supposer qu'elles correspondent à des cellules très endommagées (pour lesquelles la fraction d'ARN nucléaire est plus grande). Le taux de *mapping* bas dans ces cellules pourrait donc être dû au fait qu'une grande proportion de *reads* provient d'ARN non matures, dits ARN pré-messagers, qui sont uniquement présents dans le noyau et incluent dans leurs séquences des introns – non intégrés dans le transcriptome de référence. La proportion élevée de *reads* alignés sur des *decoys* pour ces mêmes cellules corrobore cette hypothèse.

Le taux de déduplication est une métrique utilisée seulement dans Alevin pour la classification finale des CB (voir section suivante). Un taux de déduplication élevé peut refléter le séquençage excessif d'une librairie peu complexe, i.e. comportant peu d'ADNc distincts au départ. On dit que des *reads* ont été « gâchés », car les *reads* « en trop » ont alors seulement servi à séquencer des duplicatas PCR de molécules déjà séquencées, plutôt que de nouvelles molécules. Si les taux de déduplication sont élevés pour tous les échantillons d'une *flow cell*, cela indique donc qu'une profondeur de séquençage plus petite, permettant de détecter quasiment le même nombre de molécules distinctes, aurait pu être utilisée (on parle de saturation du séquençage). Des taux de déduplication inégaux entre les différents échantillons d'une même *flow cell* peuvent révéler un nombre de cellules encapsulées diffèrent d'un échantillon à un autre. En effet, la profondeur de séquençage étant répartie également entre les librairies des différents échantillons d'une même *flow cell* (puisque la même quantité d'ADNc de chaque librairie est déposée sur la *flow cell*), il y aura davantage de duplicatas PCR séquencés pour un échantillon comportant peu de

cellules au départ, dont la librairie est peu complexe (car constituée de moins de librairies cellulaires), et la profondeur de séquençage par CB sera plus grande. Les différents échantillons de l'expérience DSP779 reflètent bien ce phénomène : les taux de déduplication et les profondeurs de séquençage par CB sont particulièrement élevés dans l'échantillon ER1, pour lequel peu de cellules sont récupérées à l'issue du QC, tandis qu'ils sont plus bas dans l'échantillon NT, pour lequel un plus grand nombre de cellules ont pu être récupérées.

Toutefois, des taux de déduplication différents pour des cellules d'un même échantillon sont plus difficiles à expliquer, la raison pour laquelle Alevin utilise cette métrique n'est donc pas claire. Comme discuté dans le chapitre 3, de telles disparités pourraient être liées à des erreurs de séquençage ayant éclaté les libraires cellulaires. L'étape de correction des CB est censée y remédier et rassembler les sous-librairies éclatées. Cependant, la correction n'opère que sur les CB situés à une distance d'édition égale à 1 d'un CB de la *whitelist* (censée représenter les CB d'origine) : elle ne prend donc pas en compte les erreurs successives (i.e. les erreurs survenant sur des CB déjà erronés), de sorte que les *reads* des CB issus d'erreurs successives seront perdus. Des disparités dans les taux de déduplication des CB d'un échantillon pourraient donc provenir de telles erreurs.

4.1.2 Contrôle qualité automatique avec Alevin

Alevin intègre dans son implémentation un QC automatique basé sur différentes métriques de qualité, évitant la détermination arbitraire des différents seuils. Le QC est généralement effectué après la génération de la matrice d'expression, à partir de laquelle les métriques de qualité sont calculées. Alevin effectue toutefois le QC en deux étapes, l'une avant et l'autre après la génération de la matrice.

La première étape (détaillée dans la section 2.4.1 du Chapitre 2), appelée *whitelisting* par les auteurs d'Alevin, consiste en l'identification automatique du *knee point*. Comme Alevin procède à cette étape avant la génération de la matrice d'expression – donc avant le *mapping* et la déduplication des UMI –, le *knee point* est identifié dans la distribution du nombre de *reads*. Comme expliqué dans le chapitre 2, le *knee point* est moins marqué dans la distribution du nombre de *reads* que dans celle du nombre d'UMI, ce qui pourrait participer au fait qu'Alevin surestime souvent le *knee point*.

La surestimation du *knee point* semble avoir des conséquences sur la seconde étape de QC effectuée en aval par Alevin, appelée *final whitelisting*, visant à éliminer les CB indésirables qui ont échappé à un simple seuil sur les profondeurs de séquençage. Cette dernière répartit dans un premier temps les CB de la *whitelist* initiale en deux groupes de même dimension, selon leurs profondeurs de séquençage. Les CB

du groupe avec les plus grandes profondeurs de séquençage sont dits de « haute qualité » et ceux de l'autre groupe « ambigus ». Un troisième groupe de même taille que les deux autres, rassemblant les CB dits de « basse qualité », est formé à partir des CB ayant une profondeur de séquençage juste en dessous du *knee point* – ou du seuil choisi manuellement, ici : 3 000. Les CB ambigus sont ensuite classés en haute ou basse qualité avec un classificateur bayésien. Ce dernier, basé sur différentes métriques de qualité calculées pour chaque CB, est entraîné avec les CB des groupes de haute et basse qualité.

Avec la procédure par défaut d'Alevin qui génère la *whitelist* initiale en utilisant la méthode du *knee point*, l'étape de *final whitelisting* apparaît très stricte : très peu de CB sont retenus dans la *whitelist* finale (i.e. la liste de CB du groupe de haute qualité et ceux du groupe ambigu classés en haute qualité par la procédure de *final whitelisting*), ce qui est au moins en partie lié au fait que le *knee point* a été surestimé. Avec une *whitelist* de 3 000 CB générée en utilisant le paramètre *--forceCells* 3 000, la procédure semble en revanche trop permissive. Il est très probable que les 1 500 CB qui seraient placés dans le groupe « haute qualité » incluent d'ores et déjà des CB indésirables.

Bien qu'intéressante, la méthodologie d'Alevin semble donc très instable. En outre, l'implémentation ne permet pas d'effectuer la seconde étape de QC lorsque l'utilisateur fournit une *whitelist* externe – ce qui est mon cas avec la nouvelle *whitelist* créée en utilisant des composantes connexes pour éliminer les CB erronés. J'ai également pu remarquer une erreur dans l'implémentation de cette procédure : j'ai constaté que certains CB inclus dans la *whitelist* finale n'étaient pas au-dessus du *knee point* (ou du seuil utilisé pour générer la première *whitelist*, e.g. 3 000 si le paramètre *--forceCells* 3 000 est utilisé), ce qui est incohérent avec la méthode décrite. Comme je l'ai signalé sur Github, je soupçonne que l'ordre des CB utilisé pour générer les différents groupes (haute qualité, basse qualité et ambiguë) soit celui des CB dans la matrice finale, lequel ne correspond pas à un tri selon les profondeurs de séquençage (cf. https://github.com/COMBINE-lab/salmon/issues/739). Une inspection rapide du code source ne m'a pas permis de repérer le problème, cela demanderait un examen plus poussé. Il semble toutefois que ce *bug* n'ait pas d'impact majeur, puisque l'ordre des CB dans la matrice d'expression correspond « presque » à un tri des CB selon leurs profondeurs de séquençage, et que le comportement de la procédure selon le seuil – trop stricte avec le *knee point* identifié par défaut ou trop permissive avec *--forceCells* 3 000 – était celui attendu.

4.1.3 QC « manuel » des différents échantillons

Pour éliminer les CB indésirables restants dans la nouvelle *whitelist* (exempte de CB erronés), j'ai finalement appliqué des seuils « manuels » sur la taille de librairie et la fraction mitochondriale, ces deux

métriques s'avérant suffisantes pour éliminer à la fois les gouttelettes vides, les doublets cellulaires, et les cellules cassées. Elles sont en effet toutes deux très utilisées dans la littérature pour le QC, conjointement avec le nombre de gènes détectés. Cependant le nombre de gènes détectés corrélant fortement avec la taille de librairie, j'ai considéré qu'il était suffisant d'utiliser seulement la taille de librairie (Figure 17). De manière générale, les métriques reflétant la complexité de la librairie corrèlent fortement entre elles et en utiliser plusieurs pour le QC serait redondant. Pour être certaine de pas éliminer de cellules pertinentes, j'ai pour chaque métrique testé plusieurs seuils plus ou moins permissifs. J'ai à chaque fois inspecté les représentations UMAP, et ajusté les seuils si la représentation révélait un groupement de CB selon une métrique de qualité.

Après avoir appliqué un seuil sur les tailles de librairie des CB de l'échantillon ER1 (DSP779) permettant d'éliminer à la fois les gouttelettes vides et cellules cassées, j'ai pu observer dans la représentation UMAP deux groupes de cellules, caractérisés par des taux de déduplication distincts. En inspectant l'expression de différents gènes d'intérêt, j'ai cependant observé que chacun des deux *clusters* présentait un sous-groupe de cellules exprimant plus faiblement TFF1, qui est un gène cible de ER dont l'expression est inhibée par des siRNA. Ces sous-groupes pourraient donc représenter des cellules pour lesquelles la transfection a fonctionné, ou du moins a été plus efficace. J'ai donc considéré que les deux *clusters* étaient pertinents et qu'il ne fallait pas appliquer de seuil sur le taux de déduplication. Dans de tels, cas où la variation d'une métrique de qualité semble masquer la variation biologique d'intérêt, il pourrait être également intéressant d'explorer les méthodes permettant d'éliminer les variations techniques ou biologiques « indésirables ».

Le *cluster* de cellules ayant un taux de déduplication plus bas dans l'échantillon ER1 pourrait correspondre à des cellules pour lesquels des *reads* ont été perdus du fait d'erreurs successives sur leur CB, comme suggéré plus haut. Il est en effet possible que de telles erreurs soient rares et se produisent donc surtout lorsque la profondeur de séquençage par CB est grande, ce qui est le cas de l'échantillon ER1 pour lequel moins de cellules ont été isolées (chacune cellule recevant alors plus de *reads*). L'expression distincte des pseudogènes et des gènes codant pour des RP qu'on peut également observer dans les deux *clusters* est cependant difficile à recouper avec les taux de déduplication hétérogènes de ces derniers. L'explication des taux de déduplication différents pourrait donc être à rechercher ailleurs que dans des erreurs successives sur les CB.

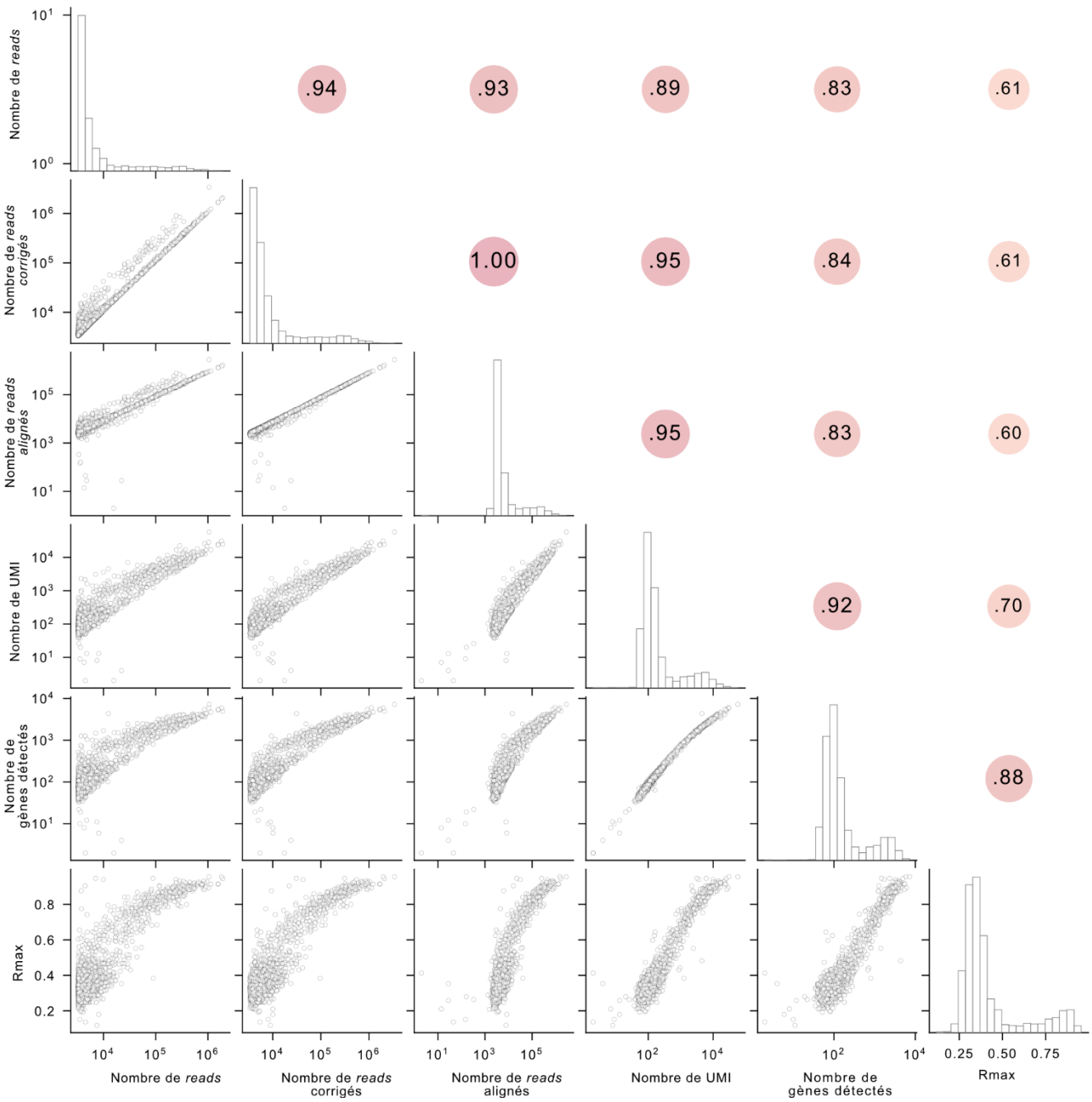


Figure 17 : Corrélation entre les différentes métriques reflétant la complexité des librairies

Les différentes métriques ont été calculées pour l'échantillon ER1 de l'expérience pilote (DSP779), dont les profils d'expression ont été générés avec Alevin (k=19, GRh38, *gencode*34) en lui fournissant la nouvelle *whitelist* exempte de CB erronés. R_{max} dénote la corrélation maximale entre le profil d'expression d'un CB et ceux des autres CB de la *whitelist*.

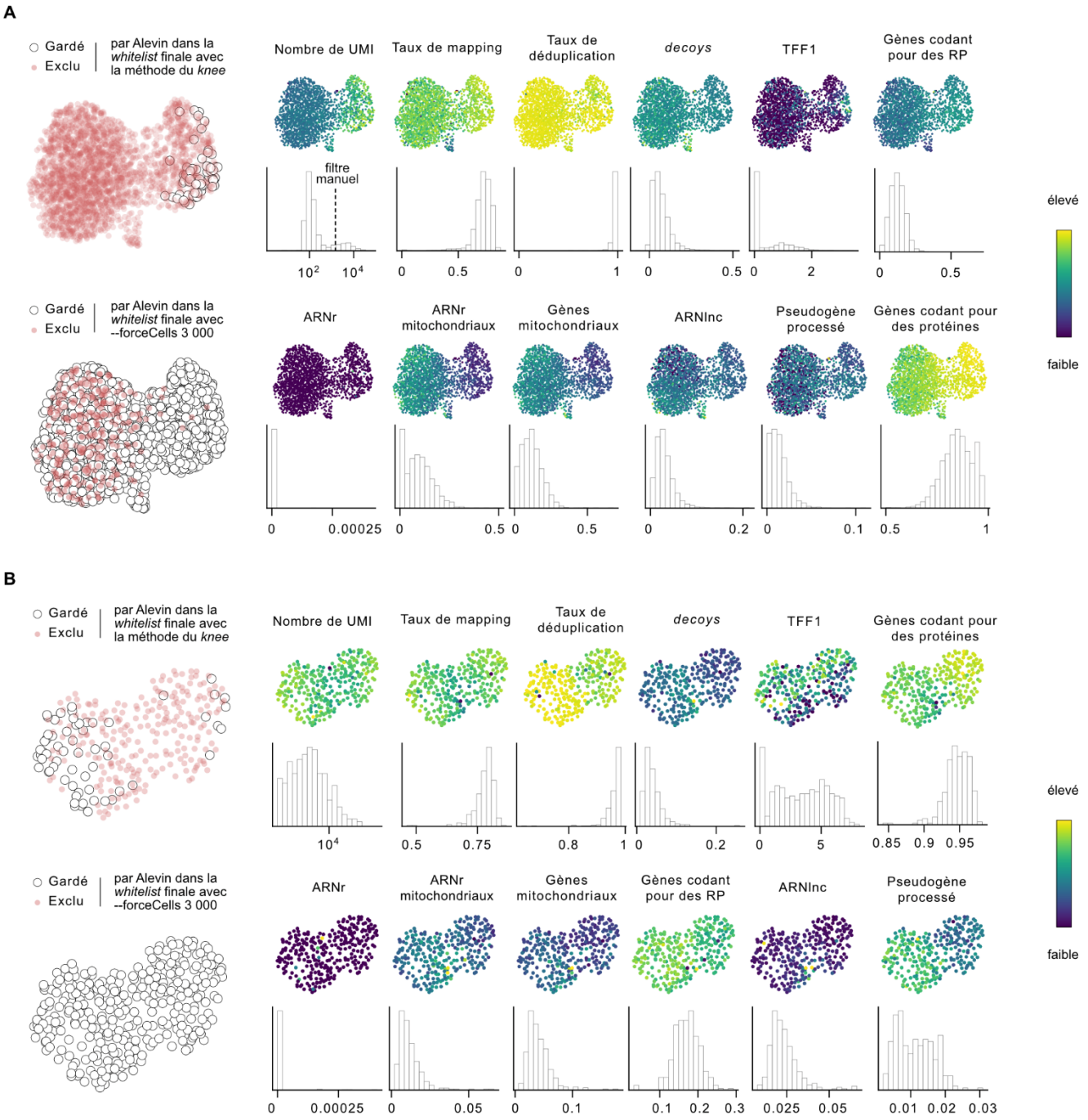


Figure 18 : Contrôle qualité des cellules

Représentations UMAP générées à partir des 20 premières PC, calculées sur les profils d'expression de l'échantillon ER1 générés avec Alevin (k=19, GRh38, *gencode* 34), en lui fournissant la nouvelle *whitelist* exempte de CB erronés. Dans **[A]**, les profils d'expression de l'ensemble des CB de la nouvelle *whitelist* sont considérés ; dans **[B]**, seuls les profils d'expression des CB ayant une taille de librairie supérieure à 1 500 correspondant au seuil indiqué dans [A]– qui est la stratégie retenue pour le QC de ER1 – sont considérés. Les profils d'expression sont également normalisés par CPMedian (cf. section 4.2) et transformés avec un logarithme de base 2 et un *pseudocount* égal à 1 avant de générer la représentation UMAP. Les paramètres utilisés pour cette dernière – autres que le nombre de PC – sont ceux assignés par défaut dans l'implémentation de Scanpy [26].

4.2 Normalisation et transformation des données

4.2.1 Données de *bulk* RNA-seq

En *bulk* RNA-seq, les différents échantillons sont généralement normalisés afin d'éliminer ou du moins de réduire les disparités dues à leurs profondeurs de séquençage respectives, et de pouvoir ainsi les comparer. En effet, si l'on calculait les *fold changes* (ratios de l'expression moyenne des gènes entre deux échantillons) sur les données brutes, on verrait que tous les gènes sont surexprimés dans l'échantillon avec la plus grande profondeur de séquençage. Les méthodes de normalisation les plus populaires consistent à diviser le nombre de fragments comptés pour chaque gène et chaque échantillon par la profondeur de séquençage de l'échantillon, ainsi que par la longueur « exonique » du gène si nécessaire. En effet, la plupart des technologies de *bulk* RNA-seq sont *full-length*, c'est-à-dire que tous les fragments issus d'un transcrit donné seront séquencés – permettant le séquençage de ce dernier sur toute sa longueur. L'expression des gènes plus longs, générant plus de fragments, sera donc exagérée. La valeur obtenue après division est généralement multipliée par une puissance de 10, le plus souvent par un million. Par exemple la normalisation TPM (pour *transcript per million*) est définie comme suit : $TPM_{e,g} = 10^6 \frac{RPK_{e,g}}{\sum_g RPK_e}$, avec $RPK_{e,g} = \frac{count_{e,g}}{longueur_g}$ et où e représente un échantillon et g un gène. Ce facteur multiplicatif permet de placer les valeurs d'expression dans un intervalle non seulement plus « raisonnable » (des valeurs d'expression de l'ordre de 10^{-4} étant peu intuitives), mais surtout déterminant pour la transformation logarithmique qui suit.

En effet, après avoir normalisé les données, une transformation est effectuée afin de réduire la dépendance de la variance par rapport à la moyenne – appelée hétéroscédasticité. Le logarithme est la transformation la plus standard, qui permet en outre de réduire l'asymétrie de la distribution et ainsi de rapprocher cette dernière d'une gaussienne, la distribution normale des données étant un prérequis pour de nombreuses méthodes, telles que l'analyse en composantes principales (PCA, pour *principal component analysis*). Une base 2 est utilisée pour le logarithme afin de calculer des *fold changes* facilement interprétables : un certain *fold change* signifiera alors que l'expression a été multipliée par $2^{fold\ change}$, par exemple un *fold change* de 1 signifiera que l'expression a doublé. Avant de pouvoir appliquer la transformation logarithmique, il faut cependant ajouter à chaque terme de la matrice d'expression une valeur positive communément appelée *pseudocount*, afin d'éviter l'opération $log(0)$ – indéfinie. Le choix du *pseudocount* est crucial et a un impact majeur sur la variance des données. Avec un *pseudocount* trop grand, prenant le pas sur les termes de la matrice d'expression qui deviennent

négligeables par rapport au *pseudocount*, la transformation logarithmique sera appliquée à des valeurs trop proches les unes des autres. Cela équivaudra presque à effectuer une transformation linéaire avec une pente très faible. La variance sera alors très petite et l'hétéroscédasticité toujours présente. À l'inverse, un *pseudocount* trop petit étirera trop les valeurs nulles par rapport au reste de la distribution. L'effet de la transformation sera dans ce cas-là trop important, avec la variance dépendant fortement du nombre de valeurs nulles ainsi que de l'écart entre ces valeurs nulles et le reste de la distribution. Le facteur multiplicatif mentionné plus haut, aura sur la variance l'effet inverse du *pseudocount*, car $log(mx + c) = log(m) + log(x + c/m)$ et $var(log(m)) = 0$ (m est une constante). Par exemple, $log(10^6 x + 1)$ aura la même variance que $log(x + 1/10^6)$. Ainsi, un *pseudocount* de 1 est habituellement choisi, afin de garder des valeurs nulles pour les gènes non exprimés ($x = 0$), puisque $log(1) = 0$, et la variance est alors ajustée grâce au facteur multiplicatif, un million étant une valeur adéquate pour les données de *bulk* RNA-seq.

4.2.2 Données de scRNA-seq

Les données de scRNA-seq sont également normalisées en divisant le nombre de fragments comptés pour chaque gène par la profondeur de séquençage de chaque cellule, ou par leur taille de librairie si des UMI sont utilisés. En divisant par la taille de librairie, on élimine non seulement les disparités dues à la taille des cellules – les cellules plus grosses contenant plus de molécules –, mais également les biais relatifs à la capture des ARNm ou à la RT – l'efficacité variable de ces dernières faisant que certaines cellules, quoique de taille égale (i.e. avec le même nombre de molécules), n'auront pas le même nombre d'UMI.

Dans la technologie Drop-Seq, où seul un fragment – celui en 3' – est séquencé pour chaque transcrit, il n'est pas nécessaire de prendre en compte la longueur du gène : la normalisation appliquée est donc analogue à la normalisation CPM (*counts per million*) utilisée en *bulk* RNA-seq, où $CPM_{e,g} = 10^6 \frac{count_{e,g}}{\sum_g count_e}$, où e représente un échantillon et g un gène. La seule différence est que le facteur multiplicatif appliqué est souvent inférieur à un million, afin de compenser les valeurs d'expression nulles, qui représentent une proportion élevée dans les données scRNA-seq. Le facteur utilisé est généralement la taille de librairie médiane (j'appelle cette normalisation « CPMedian »), qui semble raisonnable – le biais relatif à la taille de librairie étant alors moins prononcé – bien que cela ne corresponde pas forcément à la valeur optimale (Figure 19).

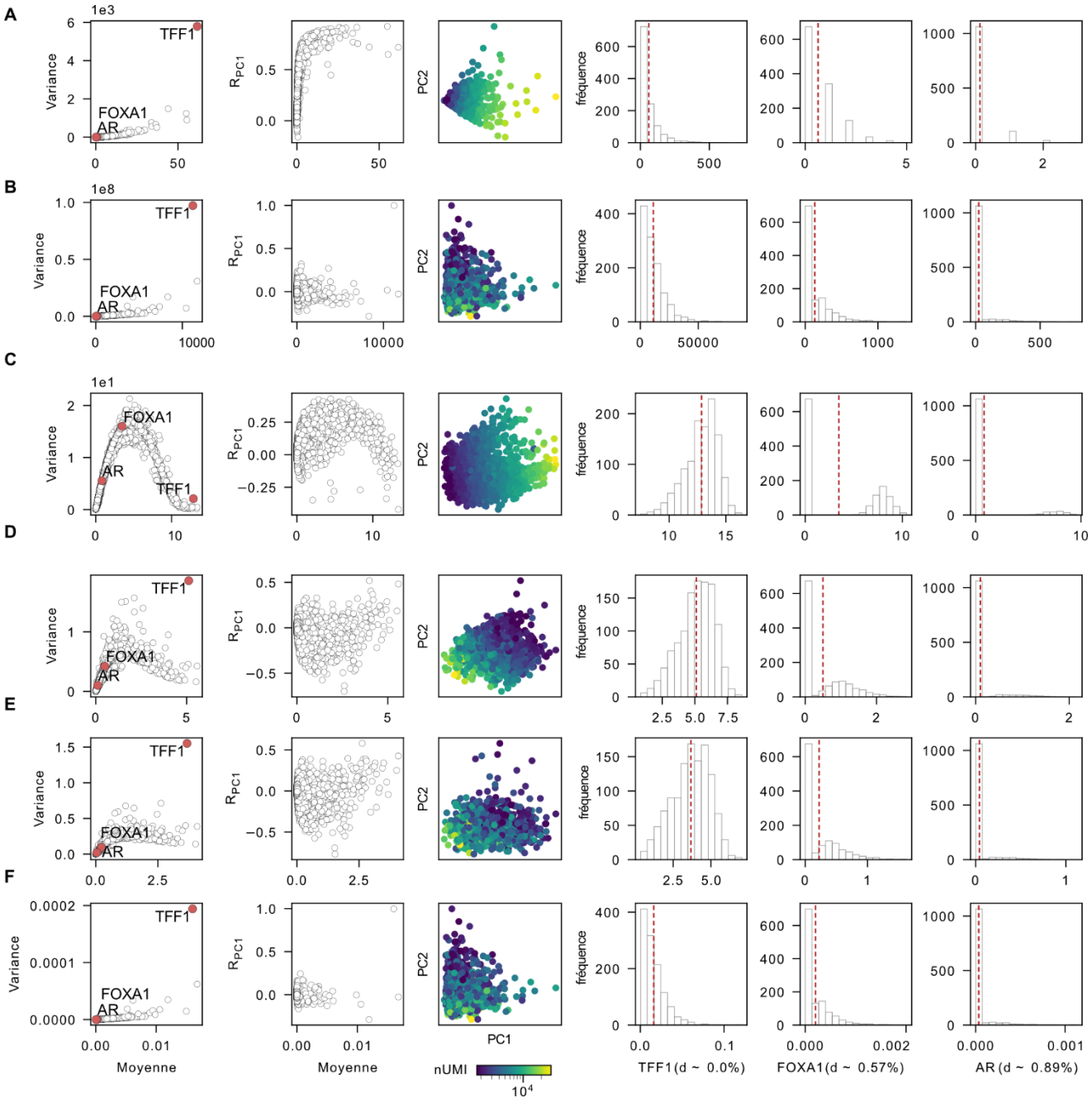


Figure 19 : Impact de la normalisation et de la transformation logarithmique sur les données scRNA-seq

Hétéroscédasticité mise en évidence par l'expression moyenne des gènes en fonction de leur variance ou de leur corrélation avec la PC1 (R_{PC1}), biais dû à la taille de librairie (nUMI), expression de TFF1 (fortement exprimé, 0% de *dropouts*), de FOXA1 (moyennement exprimé, 57% de *dropouts*) et d'AR (peu exprimé, 89% de *dropouts*) pour l'échantillon NT (DSP779), dont les profils d'expression sont **[A]** bruts ; **[B]** normalisés par CPM ; **[C]** normalisés par CPM puis transformés avec un logarithme de base 2 et *pseudocount* de 1, noté log2(+1) ; **[D]** normalisés par CPMedian (analogue à CPM mais avec un facteur multiplicatif égal à la taille de librairie médiane, ici : 4 430) et log2(+1) transformés ; **[E]** normalisés par CP1500 (analogue à CPM mais avec un facteur multiplicatif égal à 1500) et log2(+1) transformés ; **[F]** normalisés par CP1 (analogue à CPM mais avec un facteur multiplicatif égal à 1) et log2(+1) transformés. Dans les données brutes, la taille de librairie est fortement corrélée à la première composante principale (PC1), indiquant qu'elle est la plus grande source de variation dans les données. La normalisation CPM

élimine bien le biais dû à la taille de libraire, mais après une transformation log2(+1), utilisée pour réduire l'hétéroscédasticité – dont la conséquence est que la PC1 est majoritairement gouvernée par un gène fortement exprimé –, le biais est à nouveau présent. En utilisant un facteur multiplicatif inférieur à un million, l'écart entre les valeurs nulles et non nulles, accru par la transformation logarithmique et à l'origine du biais, sera plus petit. On peut donc atténuer le biais dû à la taille de librairie qui revient après la log2(+1) transformation en diminuant le facteur multiplicatif. La taille de libraire médiane est souvent utilisée, permettant d'obtenir un résultat satisfaisant bien que non optimal (ici un facteur de 1500 semble éliminer totalement le biais de taille dû à la librairie tout en atténuant l'hétéroscédasticité). Cependant, si l'on diminue trop ce facteur, les valeurs seront trop proches les unes des autres, et la transformation sera alors proche d'une transformation linéaire (i.e. elle n'aura aucun effet sur l'hétéroscédasticité).

Ces valeurs nulles peuvent découler d'une expression réellement nulle d'un gène dans une cellule. Elles peuvent par exemple refléter un gène inactif dans un certain type cellulaire, ou alors du bruit biologique résultant de la nature stochastique de l'expression des gènes, qui n'est pas linéaire dans le temps mais survient plutôt par pulsations (phénomène appelé *transcriptional bursting* [32, 33]). Cependant, la plupart des valeurs nulles dans les données scRNA-seq sont souvent liées aux limites des protocoles expérimentaux, comme la sensibilité des technologies (capacité à détecter des gènes faiblement exprimés, fonction de l'efficacité de la capture des ARNm et de la RT) ou la profondeur de séquençage (nombre total de fragments qui ont pu être séquencés, fonction du budget). Les valeurs nulles représentant du bruit, que ce soit du bruit biologique ou technique, sont souvent appelées « *dropouts* » dans les études de scRNA-seq.

Si l'on utilisait pour les données scRNA-seq un facteur multiplicatif d'un million comme en *bulk* RNA-seq, les valeurs nulles seraient trop étirées, et la variance des gènes serait fortement influencée par la proportion de valeurs nulles. Les gènes détectés dans la moitié des cellules (celles ayant les plus grandes profondeurs de séquençage ou celles pour lesquelles la capture des ARNm ou la RT ont été plus efficaces), donc moyennement exprimés, auraient ainsi la plus grande variance. Non seulement le problème initial où les gènes les plus exprimés ont la plus grande variance serait transformé en un autre problème, où les gènes moyennement exprimés auraient la plus grande variance, mais de surcroît, le biais dû à la taille de librairie serait réintroduit (Figure 19).

4.3 Contrôle des lignées cellulaires

4.3.1 Exploration des marqueurs de chaque échantillon au moyen d'une analyse différentielle

Avant de mener une analyse approfondie des différents échantillons, j'ai effectué une analyse différentielle préliminaire pour vérifier que j'étais bien à même de retrouver les gènes marqueurs de

chaque lignée dont étaient issus les échantillons. De nombreuses méthodes ont été développées spécifiquement pour l'analyse différentielle des données scRNA-seq, tentant par exemple de modéliser une inflation de « zéros » (valeurs d'expression nulles) [34], les données de scRNA-seq étant très éparses. Plusieurs études [35, 36] ont cependant démontré que les distributions résultant du comptage des UMI, contrairement à celles résultant du comptage des *reads* bruts incluant les duplicatas PCR, ne présentaient pas d'inflation de zéros. En outre, une comparaison de différentes méthodes d'analyse différentielle [37] a permis d'établir que les méthodes spécifiquement dédiées aux données de scRNA-seq avaient une performance comparable à celle des méthodes développées pour les données de *bulk* RNA-seq, et que même un simple *t-test* de Welch – test statistique général comparant les moyennes de deux groupes de variances inégales, en supposant une distribution normale – avait une performance raisonnable. Le *t-test* de Welch est en effet utilisé dans de nombreuses publications pour l'analyse différentielle de données de scRNA-seq, alors qu'il est peu utilisé dans les études de *bulk* RNA-seq. Ceci vient probablement du fait que les tailles des groupes testés en scRNA-seq (nombre de cellules) sont bien plus grandes qu'en *bulk* RNA-seq (nombre de réplicats), permettant ainsi une estimation de la variance plus précise et donc une puissance statistique supérieure (moins de faux négatifs).

J'ai donc utilisé un simple *t-test* de Welch pour comparer l'expression de chaque gène entre les 19 échantillons : après avoir exclu les gènes exprimés dans moins de 5 cellules, il restait encore 26 316 gènes à tester. Le nombre élevé de tests peut s'avérer problématique, menant à un grand nombre de faux positifs, c'est-à-dire de gènes identifiés à tort comme des gènes différentiellement exprimés (DEG, pour *differentially expressed genes*) d'un échantillon à un autre. La probabilité d'observer une différence fortuite entre les valeurs moyennes d'expression d'un gène dans différents échantillons est dite *p-value* du gène. Ainsi, en sélectionnant les gènes avec une *p-value* < 0.05 (seuil de significativité standard), on tolère une probabilité de sélectionner jusqu'à 5% de faux positifs. En valeur absolue, lorsqu'on effectue un grand nombre de tests, cela représente un nombre important de faux positifs : dans le cas présent, plus de 1 000 gènes seraient considérés à tort comme des DEG. Afin de contrôler le nombre de faux positifs, j'ai donc utilisé une correction de Benjamini-Hochberg qui, étant peu conservative (i.e. peu sévère), est souvent utilisée dans les analyses de données RNA-seq. Cette procédure ajuste les *p-values*, c'est-à-dire augmente leur valeur en fonction du nombre de tests effectués. Elle suppose que les faux positifs ont des *p-values* plus grandes que les vrais positifs. Ainsi moins de faux positifs passeront le seuil habituel de 0.05 si celui-ci est appliqué sur des *p-value* ajustées (aussi appelées des FDR, pour *false discovery rates*). Précisément, 5% des gènes différentiellement exprimés seront des faux positifs, au lieu de 5% de l'ensemble des gènes.

J'ai dans un premier temps exploré les principaux marqueurs pour chacun des 19 échantillons, que j'ai défini comme les 6 gènes ayant les plus petites *p-values* ajustées, afin de vérifier qu'ils correspondaient bien aux marqueurs attendus pour les différentes lignées séquencées. La lignée HCC70 représente un sous-type basal du cancer du sein, composé de cellules épithéliales (différenciées) ; la lignée MDAMB436, quant à elle, est représentative du sous-type *claudin-low,* composé de cellules mésenchymateuses (peu différenciées). L'exploration des 6 marqueurs principaux a cependant révélé que la lignée HCC70 exprimait fortement certains marqueurs mésenchymateux (e.g. VIM), tandis que la lignée MDAMB436 exprimait certains marqueurs épithéliaux (e.g. EPCAM, CD24, KRT19), suggérant un échange entre les deux lignées (Figure 20).

4.3.2 Analyse d'enrichissement

J'ai alors effectué une analyse d'enrichissement avec EnrichR [38], qui effectue un test exact de Fisher pour déterminer la significativité de l'intersection entre une liste de DEG fournie par l'utilisateur et une liste de référence. La *p-value* retournée, représentant la probabilité pour que cette intersection soit fortuite, est ensuite ajustée avec la correction de Benjamini-Hochberg. EnrichR donne accès à une collection de répertoires indexant chacun un ensemble de gènes avec des attributs (appelés *enrichment terms*) appartenant à une certaine catégorie, telle que la voie métabolique ou le type cellulaire. EnrichR va donc tester la significativité de l'intersection entre chacune des listes de référence d'un répertoire, regroupant les gènes partageant un même attribut – e.g. impliqués dans une même voie métabolique – et la liste de gènes fournie par l'utilisateur.

L'un des répertoires disponibles recense notamment les gènes les plus exprimés identifiés pour chacune des 1 035 lignées cellulaires cancéreuses du jeu de données public Cancer Cell Line Encyclopedia (CCLE) [39]. J'ai donc utilisé ce répertoire pour déterminer si les gènes surexprimés dans chacun des 19 échantillons incluaient une proportion significative de gènes marqueurs d'une certaine lignée, me permettant ainsi de vérifier de quelle lignée cellulaire chaque échantillon était issu.

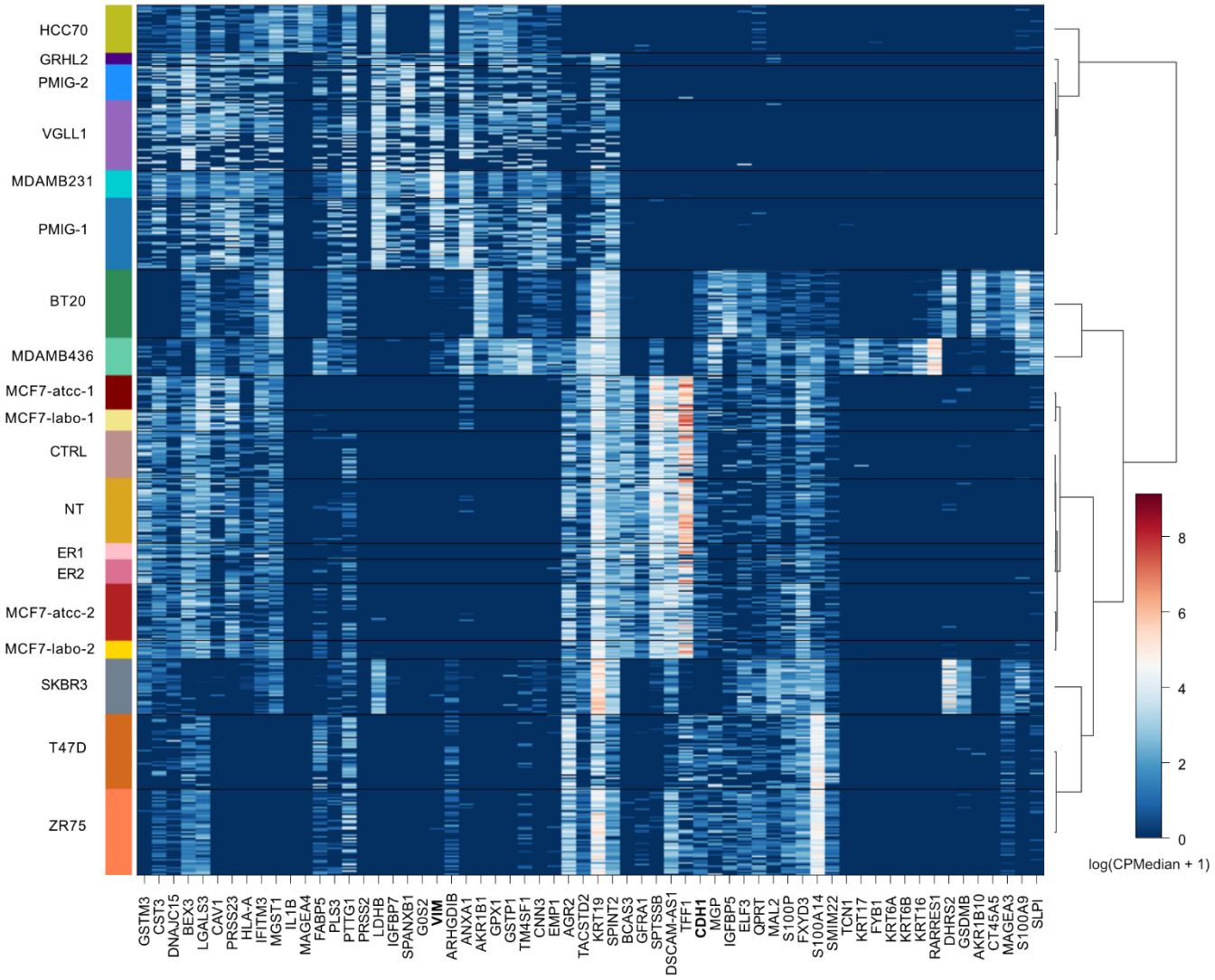


Figure 20 : Expression des gènes marqueurs des différents échantillons.

Pour chaque échantillon, les gènes marqueurs ont été définis comme les 10 gènes avec les plus grands *fold changes* absolus, calculés en effectuant la différence entre le *pseudo-bulk* (somme des profils d'expression bruts de chaque cellule) de l'échantillon considéré et celui du reste des échantillons, tous deux normalisés par CPM et log2(+1) transformés. En ne gardant qu'une seule occurrence des gènes redondants (identifiés pour plusieurs échantillons), 51 gènes ont été obtenus. La *heatmap* récapitule l'expression normalisée par CPMedian et log2(+1) transformée de ces 51 gènes dans chaque cellule de chaque échantillon. Un *clustering* hiérarchique généré avec la méthode de Ward a également été utilisé afin d'ordonner les gènes et les cellules pour la visualisation. Le *clustering* a été effectué sur les matrices des corrélations de Pearson – corrélations entre chaque paire de gènes pour le *clustering* sur les gènes ou entre chaque paire d'échantillons pour le *clustering* sur les échantillons – calculées à partir des 19 *pseudo-bulks* des échantillons. Seul le dendrogramme représentant le *clustering* sur les échantillons est montré. Les gènes VIM et CDH1, dont l'expression suggère un échange entre les échantillons HCC70 et MDAMB231, sont en gras.

Pour identifier les gènes surexprimés dans chaque échantillon, j'ai sélectionné les gènes dont la *p-value* ajustée obtenue avec le *t-test* était < 0.05 et le *fold change* > 1. Le *fold change* d'un gène d'un échantillon à un autre est défini comme le ratio des expressions moyennes du gène pour ces deux

échantillons. Il permet de quantifier la taille d'effet (*effect size*) du gène en question, c'est-à-dire son changement d'expression entre les deux échantillons. Il a été ici obtenu en calculant la différence entre le profil d'expression agrégé, ou *pseudo-bulk* (somme des profils d'expression de chaque cellule) d'un échantillon et celui du reste des échantillons, respectivement normalisés par CPM et log2(+1) transformés – le logarithme d'un quotient étant égal à la différence des logarithmes du dividende et du diviseur. La base de 2 utilisée pour la transformation logarithmique permet d'obtenir des *fold changes* facilement interprétables : un certain *fold change* signifiera alors que l'expression a été multipliée par $2^{fold\ change}$ (cf. section 4.2.1). Il est nécessaire d'appliquer un seuil à la taille d'effet en complément du seuil sur les *p-values ajustées*, car lorsque la taille des échantillons comparés (*sample size*) est grande, la plupart des *p-values* seront très petites : même des différences négligeables, ayant peu d'intérêt, seront alors significatives au regard du seuil habituellement utilisé (0.05).

L'analyse d'enrichissement a ainsi permis de confirmer l'échange entre les échantillons HCC70 et MDAMB436, chacun exprimant les gènes marqueurs de la lignée étiquetée pour l'autre échantillon. Elle a également révélé que l'échantillon T47D ne présentait pas d'enrichissement significatif pour les marqueurs de la lignée T47D. De plus les résultats d'enrichissement obtenus pour cet échantillon étaient similaires à ceux obtenus pour l'échantillon ZR75, avec un enrichissement significatif pour les marqueurs de la lignée ZR75.

Ces observations suggèrent que l'échantillon T47D ait été accidentellement préparé à partir d'une lignée ZR75 du laboratoire. Cependant cet échange – ainsi que l'échange entre les échantillons HCC70 et MDAMB436 – pourrait aussi provenir d'une interversion entre les cultures cellulaires du laboratoire. Un membre du laboratoire du Dr. Mader m'a effectivement signalé qu'elle avait noté lors d'une inspection au microscope des deux lignées une morphologie étonnamment similaire (formant toutes deux des « paquets », cf. Figure 22) entre ces dernières. Ce cas serait plus problématique qu'une erreur survenue lors de la préparation des suspensions cellulaires pour le Drop-Seq ou de l'annotation des échantillons, car d'autres expériences pourraient alors être compromises. Les résultats obtenus pour tous les autres échantillons étaient ceux attendus, avec un enrichissement significatif – et en général maximal – pour les marqueurs de la lignée étiquetée. Certains échantillons présentaient également un fort enrichissement pour les marqueurs d'une lignée provenant du même patient que la lignée étiquetée – e.g. AU565 pour l'échantillon SKBR3 – ou pour les marqueurs d'une lignée contaminée par la lignée étiquetée – e.g. YMB1 pour l'échantillon ZR75 ou encore KPL1 pour l'échantillon MCF7.

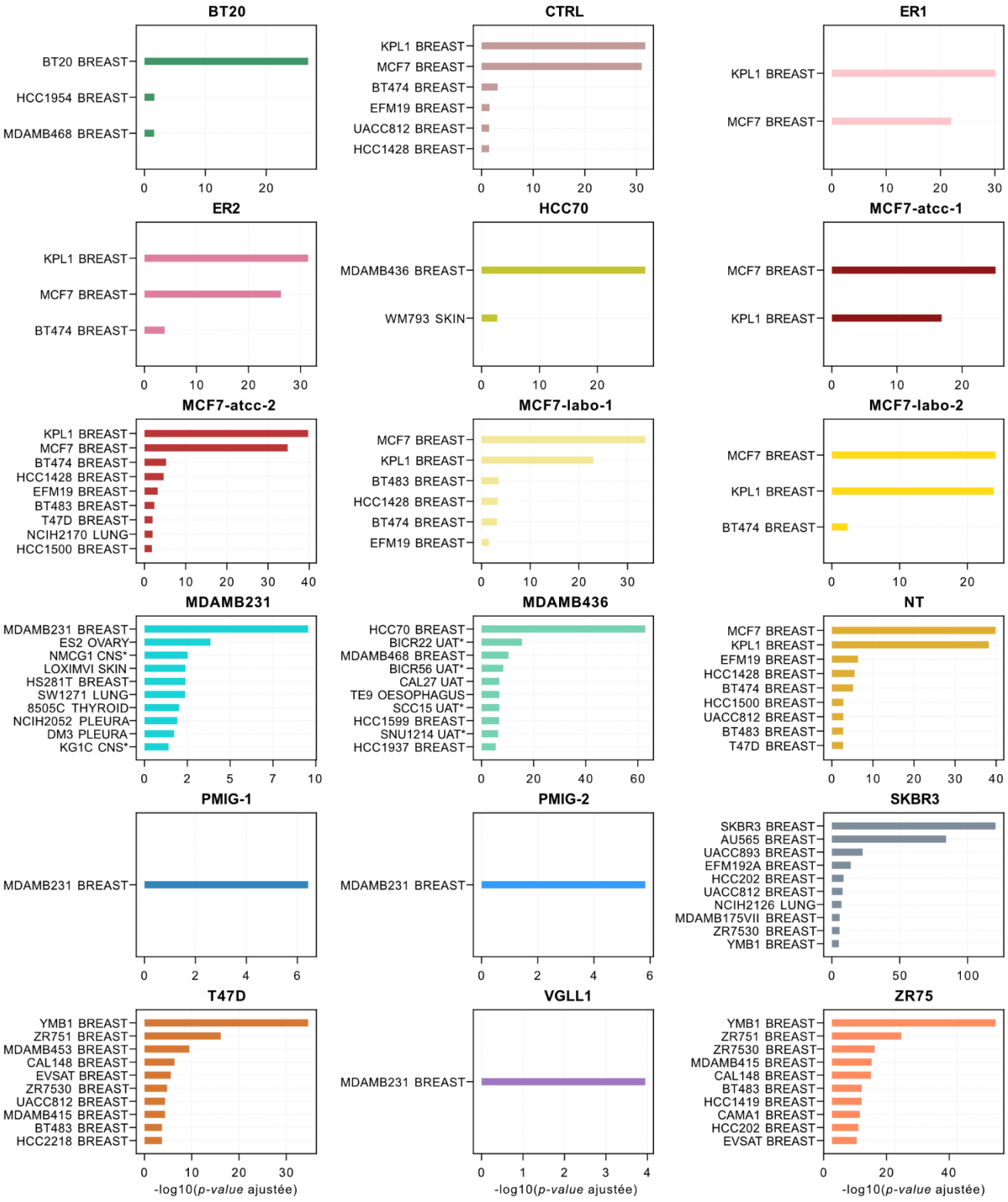


Figure 21 : Analyse d'enrichissement des gènes surexprimés dans les différents échantillons.

Les gènes surexprimés ont été identifiés en sélectionnant ceux dont la *p-value* était < 0.05 et le *fold change* > 1, après avoir filtré au préalable les gènes exprimés dans moins de 5 cellules ainsi que les gènes mitochondriaux et ceux codant pour des protéines ribosomales. Les *p-values* ont été obtenues en comparant les profils d'expression normalisés par

CPMedian et log2(+1) transformés de chaque échantillon à ceux du reste des échantillons avec des *t-tests* de Welch, suivis d'un ajustement avec la méthode de Benjamini-Hochberg. Les *fold changes* ont été calculés en effectuant la différence entre le *pseudo-bulk* (somme des profils d'expression bruts de chaque cellule) de l'échantillon considéré et celui du reste des échantillons, tous deux normalisés par CPM et log2(+1) transformés. L'analyse d'enrichissement a ensuite été effectuée avec EnrichR – qui effectue un test de Fisher exact suivi d'un ajustement avec la méthode de Benjamini-Hochberg – pour tester la significativité de l'intersection entre les gènes surexprimés dans chaque échantillon et des gènes marqueurs de différentes lignées cellulaires cancéreuses répertoriées dans le Cancer Cell Line Encyclopedia (CCLE). *CNS : *central nervous system* ; *UAT : *upper aerodigestive tract*.

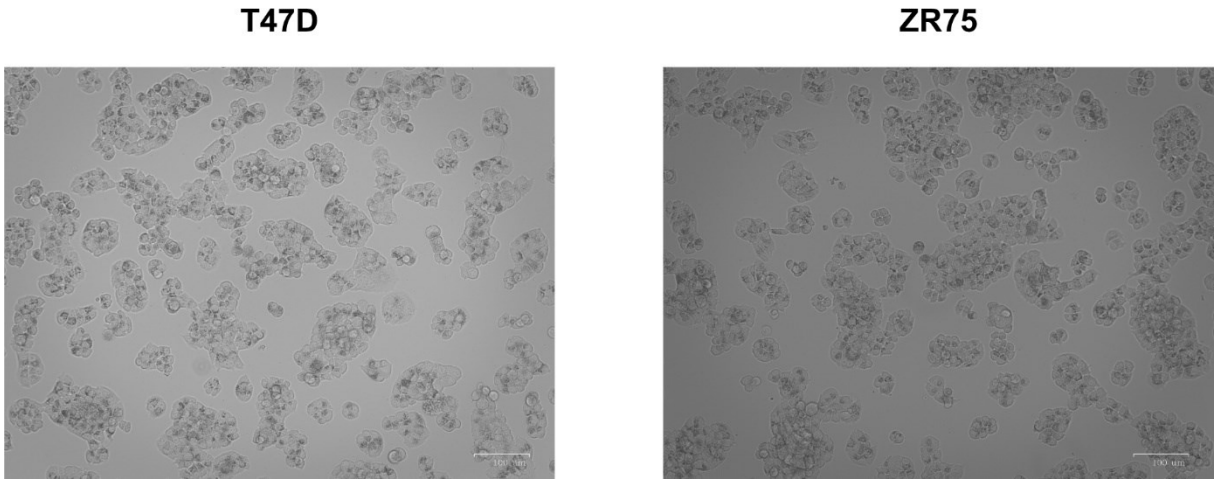


Figure 22 : Morphologie cellulaire des lignées du laboratoire étiquetées en tant que T47D et ZR75.

4.3.3 Corrélation des échantillons du laboratoire avec des *bulks* publics

Pour confirmer mes soupçons sur la lignée d'origine des échantillons HCC70, MDAMB436 et T47D, j'ai ensuite calculé le coefficient de corrélation de Pearson entre les *pseudo-bulks* des 19 échantillons et les profils d'expression de lignées du cancer du sein séquencées en bulk RNA-seq. Pour cela j'ai utilisé un jeu de données public comportant 56 lignées cellulaires du cancer du sein [20], incluant ZR75, T47D, MCF7, HCC70, SKBR3 et MDAMB231, séquencées en *bulk* RNA-seq, dont j'ai « mappé » les *reads* avec Salmon (k = 31, *decoys*, GRh38, *gencode* 34) pour obtenir l'abondance de chaque gène. Après avoir effectué une normalisation TPM pour les *bulk* et CPM pour les *pseudo-bulks*, tous les profils ont été log2(+ 1) transformés (cf. section 4.2.1). Pour chaque profil de *pseudo-bulk*, la corrélation était la plus forte avec le profil *bulk* de la lignée cellulaire correspondante (si présente), sauf pour HCC70, MDAMB436 et T47D. Le *pseudo-bulk* de la lignée MDAMB436 corrélait fortement avec le bulk de la lignée HCC70, tandis que le *pseudo-bulk* étiqueté comme HCC70 corrélait plutôt avec les mêmes profils *bulk* que les *pseudo-bulks* de la lignée MDAMB231 (la lignée MDAMB436 ne figurant pas parmi les 56 lignées séquencées dans le jeu de données public), indiquant une lignée plutôt mésenchymateuse (Figure 23). Ces deux observations confirment à nouveau l'échange entre HCC70 et MDAMB436.

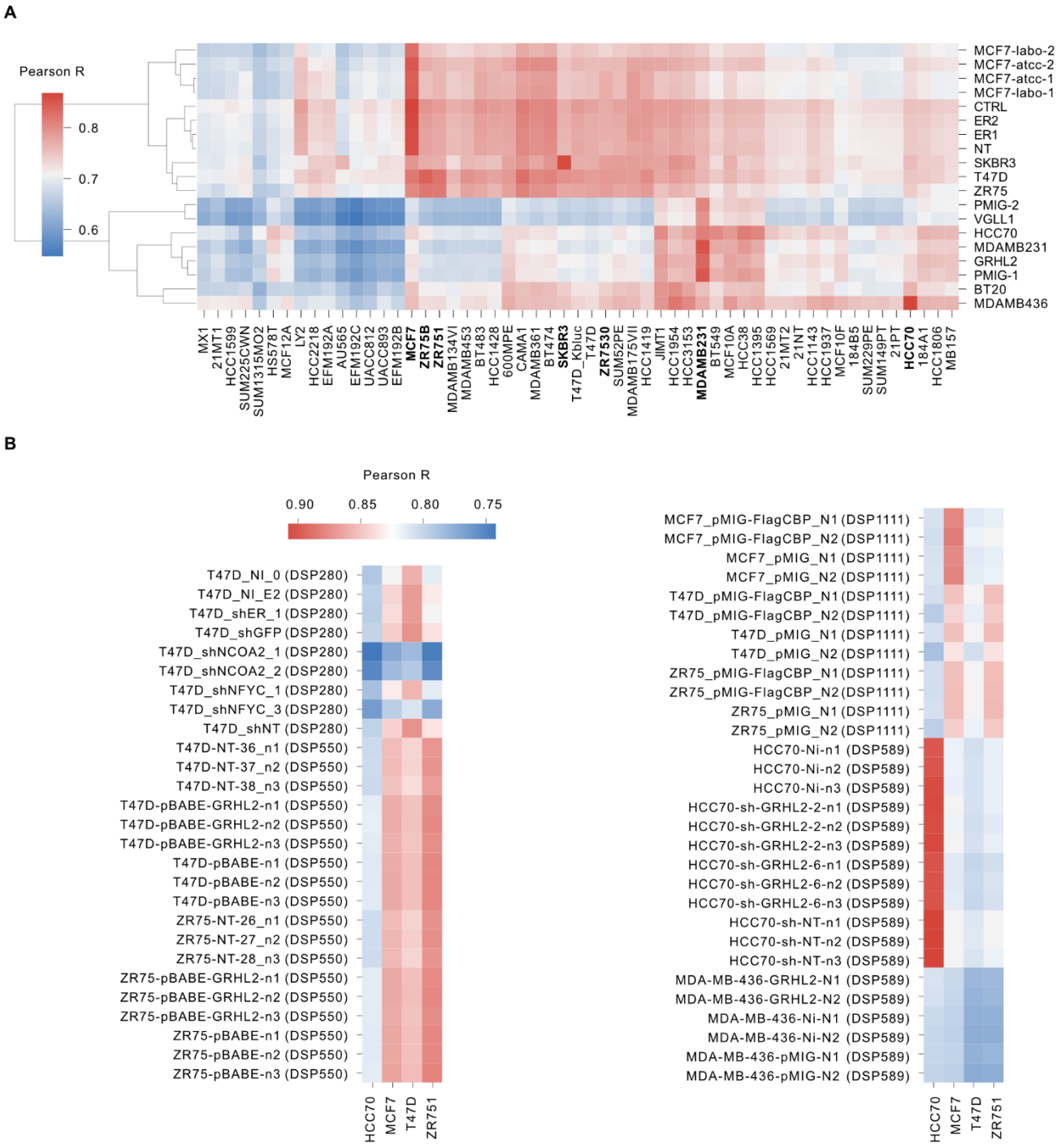


Figure 23 : Corrélation des profils d'expression de RNA-seq issus du laboratoire avec ceux de 55 lignées cellulaires du cancer du sein.

[A] Corrélation de Pearson entre les profils d'expression de *pseudo-bulks* du laboratoire (lignes) et ceux de 55 *bulks* du jeu de données public (colonnes) de Daemen et al. [20]. Le jeu de données comprend initialement 56 lignées, mais la lignée HCC202 a été exclue car les corrélations obtenues avec les *pseudo-bulks* étaient aberrantes. Les données brutes SRA (fichiers FASTQ) du jeu de données de Daemen et al. ont été obtenues via le Geo Portal avec le code d'accès GSE48213. Le *mapping*, la quantification et la normalisation ont ensuite été effectués avec Salmon (k=31, *decoys*, GRh38, *gencode*34), et les profils normalisés par TPM (valeurs TPM retournées par Salmon) ont

finalement été transformés avec un logarithme de base 2 et un *pseudocount* de 1, noté log2(+ 1). Les *pseudo-bulks* ont quant à eux été obtenus en agrégeant les profils de scRNA-seq générés avec Salmon (k=19, *decoys*, GRh38, *gencode*34), puis en normalisant par CPM les profils agrégés et en effectuant une transformation log2(+1). **[B]** Corrélation de Pearson entre les profils d'expression de *bulks* du laboratoire (lignes) et ceux de 4 *bulks* du jeu de données public (colonnes) de Daemen et al. Les profils d'expression des *bulk* du jeu de données de Daeman et al. ont été obtenus avec le prétraitement décrit plus haut. Ceux des *bulks* du laboratoire ont quant à eux été obtenus en effectuant l'alignement des *reads* avec STAR [17] (GRh38) puis quantifiés et normalisés par TPM avec RSEM [40] (*gencode*37) et finalement log2(+1) transformés.

Pour le *pseudo-bulk* de l'échantillon T47D, la corrélation la plus forte était avec le *bulk* de la lignée ZR75B. Il présentait globalement des corrélations proches de celles obtenues pour le *pseudo-bulk* ZR75. Cela étaie l'hypothèse que l'échantillon étiqueté comme T47D dans le lot de séquençage DSP762 provient vraisemblablement d'une lignée ZR75.

Pour savoir si les échanges avaient eu lieu pendant la préparation des échantillons ou si les cultures elles-mêmes avaient été interverties, j'ai ensuite effectué la même analyse de corrélation avec plusieurs jeux de données de *bulk* RNA-seq du laboratoire. Si ce sont les cultures qui ont été interverties, le problème peut effectivement toucher d'autres expériences. Tous les jeux de données ont été alignés avec STAR [17] (GRh38) et quantifiés avec RSEM [40] (*gencode* 37) par la plateforme informatique de l'IRIC (DSP550, DSP589, DSP1111) ou par moi-même (DSP280). Après les avoir normalisés par TPM (valeurs retournées par RSEM) et log2(+ 1) transformés, j'ai ensuite comparé leurs profils d'expression avec ceux des lignées MCF7, HCC70, T47D et ZR751 du jeu de données public de 56 lignées, que j'avais obtenus avec Salmon (cf. plus haut). Cette comparaison n'est pas idéale, car il est toujours plus prudent de comparer des échantillons traités avec les mêmes outils computationnels et les mêmes paramètres. Cependant, cela ne devrait pas être problématique dans le cas présent, le but étant de déterminer, pour chaque profil d'expression obtenu avec STAR (*bulks* du laboratoire), avec quel profil d'expression obtenu avec Salmon (*bulks* publics) il corrèle le plus. J'ai pu constater que pour le lot de séquençage DSP589, l'étiquetage des échantillons en tant que HCC70 ou MDAMB436 semblait correct : leurs profils d'expression corrélaient respectivement fortement et faiblement avec le profil d'expression de la lignée HCC70 du jeu de données public. Pour les échantillons du lot de séquençage DSP280 étiquetés en tant que T47D, les profils d'expression corrélaient plus fortement avec le profil d'expression de la lignée T47D du jeu de données public. L'étiquetage en tant que T47D semblait donc correct pour ce lot, en revanche il semblait erroné pour les lots de séquençage DSP550 et DSP1111, les profils d'expression des échantillons étiquetés comme des T47D corrélant plus fortement avec le profil d'expression de la lignée ZR75 du jeu de données public.

4.3.4 Recherche de mutations caractéristiques

J'ai finalement effectué un dernier contrôle afin de confirmer que les cultures cellulaires du laboratoire que l'on croyait être des T47D étaient en réalité des ZR75. Pour cela, j'ai recherché certaines mutations caractéristiques des lignées T47D et ZR75 dans les *reads* de différents échantillons étiquetés en tant que T47D ou ZR75, au moyen de l'outil IGV [41], qui permet de visualiser les alignements des *reads* sur un génome de référence.

J'ai utilisé les alignements des *reads* des échantillons de *bulk* RNA-seq du laboratoire générés avec STAR (cf. section 4.3.3) qui contrairement à Salmon, effectue un alignement nucléotide par nucléotide et dont les fichiers en sortie – au format BAM – sont compatibles avec IGV. J'ai également aligné avec STAR tous les *reads reverse* de l'échantillon T47D de scRNA-seq du laboratoire (lot de séquençage DSP762), comme si c'était un échantillon de *bulk* RNA-seq issu d'un séquençage *single-end*.

J'ai dans un premier temps consulté les alignements pour les positions du génome correspondant à la mutation L194F du gène TP53 (C => T, locus chr17:7,674,951) caractéristique de la lignée T47D. Cependant, la couverture était peu profonde pour cette mutation (seulement un *read* aligné sur le locus correspondant) à cause de la couverture inégale des transcrits liée au séquençage 3' (Figure 24).

J'ai donc recherché une autre mutation spécifique de la lignée T47D pour laquelle la couverture était plus profonde, en l'occurrence la mutation H1047R du gène PIK3CA (A => G, locus chr3:179,234,297). J'ai également recherché la mutation L108R du gène PTEN (T => G, locus chr3:179,234,297), spécifique de la lignée ZR75. Parmi les échantillons étiquetés comme T47D ou ZR75 que j'ai consultés, celui du lot de séquençage DSP280 était le seul dont les *reads* présentaient la mutation H1047R caractéristique de la lignée T47D. À l'inverse, j'ai pu retrouver la mutation L108R caractéristique de la lignée ZR75 dans les *reads* de tous les échantillons – y compris ceux étiquetés comme T47D – sauf celui du lot de séquençage DSP280 étiqueté comme T47D (Figure 24).

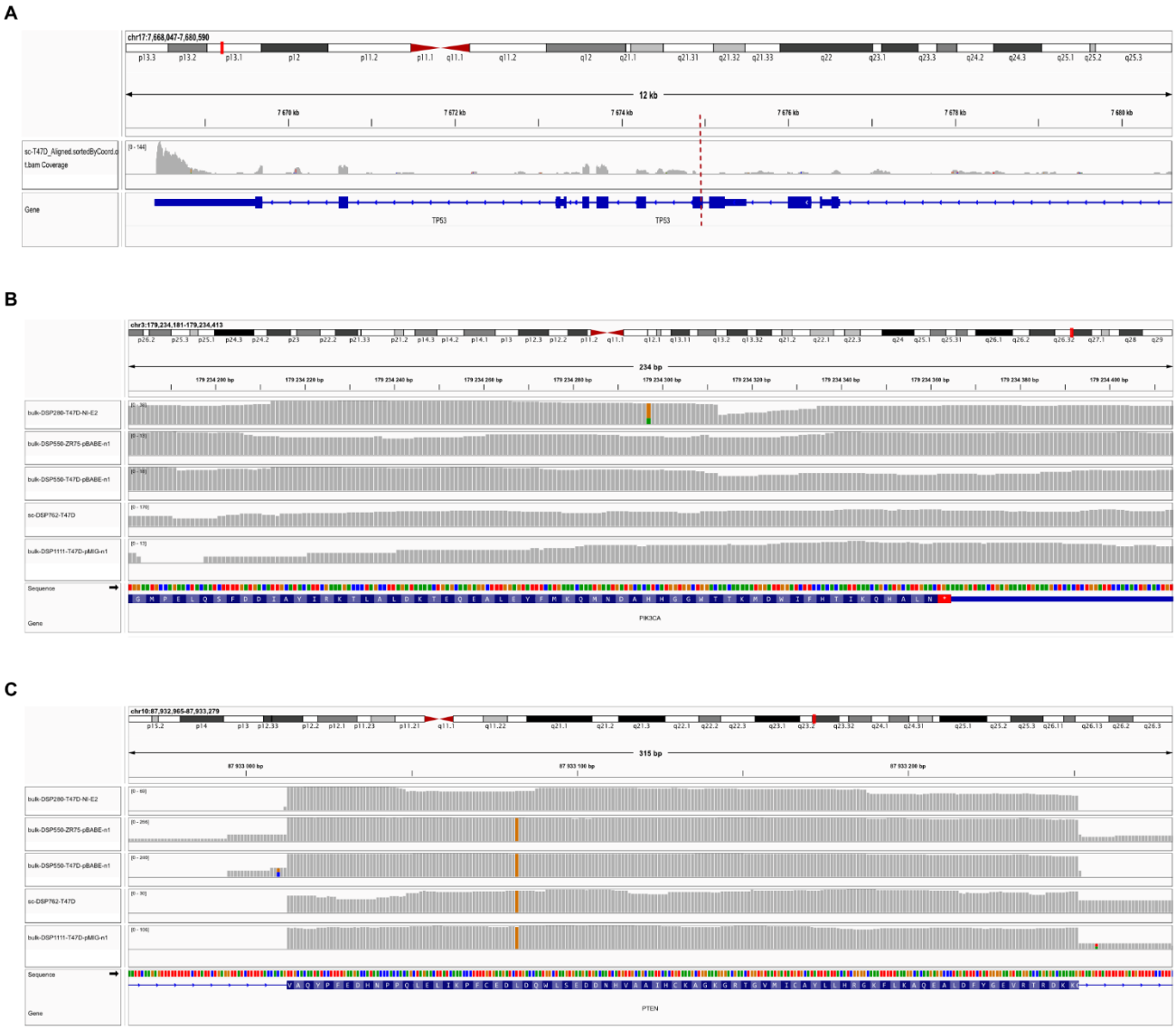


Figure 24 : Visualisation avec IGV des alignements obtenus avec STAR pour des échantillons du laboratoire étiquetés en tant que ZR75 ou T47D.

[A] Couverture de la mutation L194F du gène TP53 (C => T, locus chr17:7,674,951 indiqué par la ligne rouge en pointillés), caractéristique de la lignée T47D, par les *reads* de l'échantillon T47D du lot de séquençage DSP762 (expérience de scRNA-seq). L'alignement a été obtenu comme décrit en [B]. Seulement un *read* couvre cette mutation, probablement à cause du biais de couverture en 3' lié à la technologie de séquençage 3' tag. **[B]** Recherche de la mutation H1047R du gène PIK3CA (A => G, locus chr3:179,234,297), caractéristique de la lignée T47D, au sein des *reads* des échantillons du laboratoire T47D NI-E2 (DSP280, *bulk* RNA-seq), ZR75-pBABE-n1 (DSP550, *bulk* RNA-seq), T47D-pBABE-n1 (DSP550, *bulk* RNA-seq), T47D (DSP762, scRNA-seq) et T47D-pMIG-n1 (DSP1111, *bulk* RNA-seq) – présentés dans cet ordre de haut en bas dans la figure. Les alignements ont été obtenus avec STAR (GRh38), en ignorant pour l'échantillon de scRNA-seq le fichier de codes-barres – i.e. seuls les *reads reverse* ont été alignés, comme les *reads* d'une expérience de *bulk single-end*. Seul l'échantillon T47D NI-E2 (DSP280, *bulk* RNA-seq) présente cette mutation. **[C]** Recherche de la mutation L108R du gène PTEN (T => G, locus chr3:179,234,297), caractéristique de la lignée ZR75, au sein des *reads* des échantillons du laboratoire – mêmes échantillons et même procédure pour l'alignement que dans [B]. Tous les échantillons sauf T47D NI-E2 (DSP280, *bulk* RNA-seq) présentent cette mutation.

CHAPITRE 5 DISCUSSION

Les chapitres précédents présentaient divers étapes du prétraitement et outils dédiés à cet effet, ainsi différentes caractéristiques des données issues de la technologie Drop-Seq.

Le présent chapitre recoupera différentes observations faites dans les chapitres précédents avec la littérature, et présentera des résultats préliminaires ainsi que des lignes directrices et des perspectives en vue d'une analyse plus poussée des jeux de données utilisés dans ce mémoire ou de nouvelles expériences à venir.

5.1 Définition de la « *whitelist* » dans la littérature

Alevin [22], ainsi que Kallisto-BUStools [42] ou encore STAR-solo [43], développés plus récemment, sont tous des outils de bout en bout destinés au prétraitement de données scRNA-seq issues de diverses technologies. Concernant la correction des CB, ces outils semblent s'être inspirés de la procédure implémentée dans CellRanger [11] – leur équivalent commercial développé par 10X pour la technologie Chromium. Dans CellRanger, la *whitelist* désigne l'ensemble des CB du kit Chromium utilisé pour l'expérience, peu importe qu'ils caractérisent des gouttelettes vides ou des cellules. Cette *whitelist* sert alors uniquement de référence pour la correction des CB, et les CB issus de gouttelettes vides sont éliminés dans un second temps.

La procédure implémentée dans CellRanger n'est cependant pas adaptée aux technologies pour lesquelles, contrairement à la technologie Chromium, les CB utilisés ne sont pas connus d'avance. C'est le cas justement de la technologie Drop-Seq, où les CB sont générés aléatoirement avec la technique du *split-pool* (décrite brièvement dans la section 2.3.1). L'équipe qui a développé la technologie Drop-Seq (Macosko et al.) a ainsi proposé une procédure de correction des CB ne reposant pas sur une *whitelist*, incluse dans la suite logicielle Drop-Seq tools qu'elle a développée en Java (cf. section 2.4.4).

Dans STAR-solo la correction n'est effectuée qu'à partir d'une *whitelist* fournie par l'utilisateur, les CB ne sont donc même pas corrigés pour de telles technologies. Alevin et Kallisto-BUStools intègrent quant à eux une fonctionnalité permettant de générer une *whitelist* à partir des CB séquencés, mais au lieu d'une liste de CB exempte de CB erronés, une liste exempte de CB issus de gouttelettes vides est créée. En effet, la *whitelist* est générée en sélectionnant les CB ayant les plus grandes profondeurs de séquençage,

ce qui est habituellement utilisé pour exclure les CB issus de gouttelettes vides. En appliquant une telle procédure, il est implicitement admis que les erreurs de CB sont peu récurrentes et que leur taux est suffisamment faible pour que les CB erronés, ayant alors des profondeurs de séquençage petites, ne soient pas sélectionnés dans la *whitelist*.

Cette assomption est évoquée par Dr. Tom Smith – dont l'équipe a développé la suite logicielle UMI-tools [23], et qui est l'un des auteurs de l'article décrivant Alevin – dans un billet de son blog [44], où il suggère que plusieurs minimaux locaux caractérisent la distribution. Parmi eux, certains correspondraient à des gouttelettes vides, d'autres à des CB erronés ; le pic avec les tailles de librairies les plus grandes correspondrait aux CB associés à des cellules. Cette assomption n'est toutefois pas mentionnée dans l'article décrivant Alevin, j'ai de surcroît démontré qu'elle était fausse. Comme mentionné dans la section 2.4.1, la définition du terme *whitelist* semble donc confuse dans la littérature, et cela n'est pas sans conséquences (cf. section 2.4.3).

Il est surprenant que les CB erronés susceptibles d'être sélectionnés dans la *whitelist* ne soient pas pris en compte dans Alevin. En effet, Dr. Tom Smith a pourtant mentionné dans un autre billet de son blog [44] – dont j'ai pris connaissance plus tardivement – que des CB erronés pouvaient se retrouver au-dessus du *knee point*. Une solution à ce problème a même été mise en place dans UMI-tools, grâce au paramètre *--ed-above-threshold=[discard/correct]* qui permet d'exclure ou de corriger les CB erronés sélectionnés dans la *whitelist*. De manière similaire à la procédure implémentée dans Drop-Seq tools, pour chaque paire de CB voisins (distance de Hamming égale à 1) identifiée dans la *whitelist*, le CB avec la plus petite profondeur de séquençage est éliminé ou corrigé, i.e. remplacé par l'autre CB de la paire.

5.2 Améliorations suggérées pour le logiciel Alevin

Plusieurs modifications pourraient être suggérées pour améliorer Alevin. Notamment, une procédure de correction des CB différente – telle que celle proposée dans la section 2.4.4 ou celle implémentée dans Drop-Seq tools – pourrait être implémentée, et la recherche du *knee point* pourrait être effectuée en aval de la génération de la matrice d'expression, à partir de la distribution des tailles de librairies. Cela permettrait d'identifier plus précisément le *knee point*, et éventuellement de rendre plus robuste l'étape de *final whitelisting*.

Les différents groupes de CB utilisés pour la classification pourraient également être construits autrement, en incluant parmi les CB ambigus des CB en dessous du *knee point*. Cela permettrait peut-être de récupérer des cellules de petite taille ou quiescentes en dessous du *knee point*.

Il faudrait également revoir les métriques utilisées pour la classification, parmi lesquelles par exemple celle du nombre de gènes détectés n'est pas forcément pertinente. Cette métrique est en effet fortement corrélée avec la profondeur de séquençage et la taille de librairie, utilisées l'une ou l'autre pour identifier le *knee point* et définir les différents groupes de CB pour la classification (Figure 17). Certaines métriques pour lesquelles le biais ciblé n'est pas clair, telles que le taux de déduplication, pourraient également être mises de côté.

Enfin quelques *bugs* seraient à corriger, en particulier celui que j'ai signalé sur GitHub (cf. https://github.com/COMBINE-lab/salmon/issues/739), ou le fait que les CB de basse qualité utilisés pour la classification lors de la *final whitelisting* sont inclus dans la matrice d'expression finale, ce qui m'a troublée et semble avoir également troublé d'autres utilisateurs (cf. https://github.com/COMBINE-lab/salmon/issues/739).

5.3 Vers un contrôle qualité automatique des cellules

La stratégie de contrôle qualité des cellules (QC) pour laquelle j'ai opté dans le chapitre 4, où des seuils sur des métriques de qualité sont ajustés manuellement, est très répandue dans la littérature, appréciée qu'elle est pour sa simplicité et son interprétabilité. Cependant l'utilisation de seuils arbitraires n'est pas idéale, pouvant mener à du *data peaking*, où les seuils sont ajustés par le chercheur pour améliorer les résultats d'une analyse en aval, e.g. d'un test statistique. De plus, si le nombre d'échantillons à traiter est conséquent, la tâche peut vite devenir rébarbative, le QC devant être effectué pour chaque échantillon individuellement. Il pourrait donc être pertinent d'explorer plus amplement les différentes méthodes de QC automatique proposées dans la littérature.

L'une d'entre elles, implémentée dans EmptyDrops [15] et intégrée dans la nouvelle version de CellRanger [11] – équivalent commercial d'Alevin, développé par 10X – effectue un test statistique pour déterminer si le profil d'expression des différents CB dévie significativement du profil d'expression « ambiant ». Ce dernier est défini comme la somme des profils d'expression des CB avec les plus petites tailles de librairie (à savoir, dans l'article, les CB pour lesquels le nombre d'UMI est inférieur à 100). Pour prévenir l'élimination de cellules pertinentes dont le profil est assez proche du profil ambiant – il arrive par exemple qu'un certain type cellulaire soit plus susceptible d'être lysé et contribue alors plus fortement au profil ambiant –, un *knee point*, au-dessus duquel les CB seront systématiquement conservés, est également identifié. À première vue, il semble que cette méthode soit moins instable qu'Alevin, car elle

semble moins dépendante du *knee point*, particulièrement peu marqué dans les données Drop-Seq par rapport aux autres technologies à gouttelettes [16].

En outre elle permet, contrairement à Alevin ou à un simple seuil sur la taille de librairie, de récupérer des CB pouvant correspondre à des cellules de petite taille ou endommagées, qui pourraient passer au travers du *knee point*/seuil et être éliminées à tort. Comme son nom (EmptyDrops) l'indique, cette méthode ne vise néanmoins qu'à éliminer les gouttelettes vides : si l'on ne souhaite pas inclure les cellules endommagées dans les analyses, il faudra donc effectuer un QC supplémentaire.

Dans l'article, les auteurs utilisent en complément un seuil « adaptatif » sur la fraction mitochondriale, défini comme 3 fois l'écart médian absolu (MAD, *median absolute value*). Un seuil est dit adaptatif s'il est calculé à partir de la distribution des données, en vue de s'adapter à n'importe quel échantillon, type cellulaire ou encore espèce – c'est l'équivalent de la généralisation en *machine learning*. Le MAD est une méthode robuste pour détecter les valeurs aberrantes (*outliers*), le nombre de valeurs aberrantes identifiées étant peu sensible au nombre de MAD choisi pour le seuil. Les méthodes de détection des valeurs aberrantes en général supposent cependant que la plupart des cellules de l'échantillon soient viables, ce qui n'est pas toujours le cas : il arrive donc que ce seuil soit trop permissif [45].

Ainsi, bien qu'un QC complètement automatisé fût idéal, il est possible qu'en raison des caractéristiques variables des différents jeux de données, une méthode interactive basée sur des seuils arbitraires déterminés de façon appropriée – i.e. en évitant le *data peaking* – soit plus raisonnable et plus fiable pour éliminer les cellules endommagées.

5.4 Résultats préliminaires

Dans le cadre de ce mémoire, j'ai effectué des analyses des différents échantillons afin d'étudier les régulations transcriptionnelles impliquées dans le cancer du sein et l'hétérogénéité tumorale qui en découle. À défaut d'intégrer les résultats des analyses complètes dans mon mémoire – incluant des analyses différentielles et d'enrichissement –, j'ai regroupé dans la Figure 25 des résultats préliminaires, qui pourraient tout de même guider le *design* expérimental ou l'analyse computationnelle d'expériences futures menées au sein du laboratoire.

Notamment, une représentation UMAP intégrant les différents échantillons du projet 1 (expérience pilote dans laquelle des cellules MCF7 ont été transfectées avec des siRNA bloquant l'expression de ESR1) a révélé un groupe de cellules provenant des échantillons ER1 et ER2, pour lesquelles l'expression de TFF1 était faible. TFF1 étant un gène cible de ESR1, ces cellules correspondent vraisemblablement à

des cellules pour lesquelles la transfection a été efficace – les cellules des échantillons ER1 et ER2 ayant été respectivement transfectées avec un siER#1 et un siER#2. Les autres cellules des échantillons ER1et ER2, plutôt groupées avec celles des échantillons contrôles NT et CTRL, pourraient correspondre quant à elles à des cellules pour lesquelles la transfection n'a pas fonctionné. Ainsi, une analyse différentielle par échantillons (NT, CTRL, ER1, ER2) ne m'a pas permis de détecter les gènes marqueurs de la réponse œstrogénique (analyse non intégrée dans ce mémoire), le signal étant probablement trop « dilué ».

La représentation UMAP des échantillons du projet 1 a également révélé un autre groupe de cellules, qui étaient quant à elles caractérisées par une forte expression du gène HSPA5 (Figure 25). Ce gène étant un marqueur de la réponse au stress du réticulum endoplasmique, le groupe de cellules pourrait représenter un groupe de cellules stressées par le processus de transfection – d'autant que ce groupe de cellules est principalement composé de cellules transfectées.

J'ai par ailleurs constaté que les TF d'intérêt – i.e. impliqués dans la différenciation des tissus mammaires –, faiblement exprimés, étaient peu détectés. Même le TF GRHL2, pourtant surexprimé par un procédé de transduction dans les cellules de l'échantillon GRHL2 (projet 3), n'était pas suffisamment détecté dans ces dernières. Il n'était donc pas possible de tirer de conclusion quant à la relation entre l'expression des TF d'intérêt et celle de leurs gènes cibles.

Enfin, une représentation UMAP générée à partir de l'ensemble des échantillons dédiés au projet 2 – hormis ceux de cellules cultivées dans le milieu DMEM – n'a pas permis de mettre en évidence la présence de sous-populations communes entre les différentes lignées. En revanche, en générant les représentations à partir de chaque lignée individuellement, j'ai pu caractériser une hétérogénéité dans certaines d'entre elles. Par exemple, la plupart des cellules issues des lignées MCF7 du laboratoire et cultivées dans du DMEM se retrouvaient groupées avec celles de la lignée MCF7 d'ATCC (également cultivées dans du DMEM), mais certaines formaient un groupe à part. Cette hétérogénéité pourrait être liée au métabolisme des cellules, comme le suggère l'expression du gène MT-CO2 impliqué dans la respiration cellulaire. Mais elle pourrait également être liée aux régulations transcriptionnelles impliquées dans la différenciation du tissu mammaire, le groupe de cellules à part exprimant plus fortement le TF XBP1. Pour l'échantillon de cellules provenant de la lignée HCC70, j'ai également identifié une hétérogénéité grâce aux représentations UMAP, qui ont mis en évidence un groupe de cellules exprimant plus fortement les gènes ALDHA1 et KRT6A, marqueurs des cellules souches cancéreuses et de la transition épithélio-mésenchymateuse (EMT).

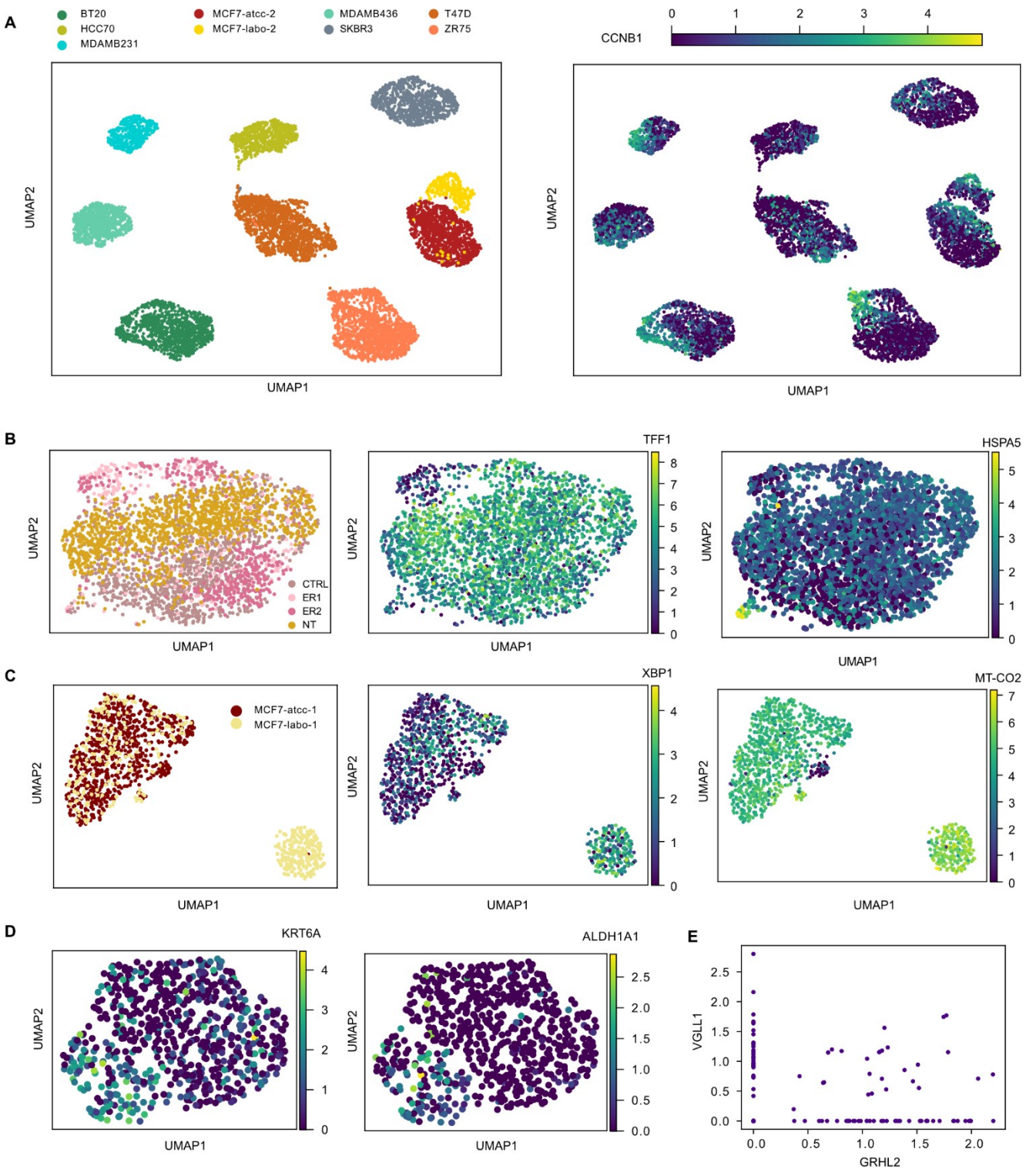


Figure 25 : Analyse exploratoire des régulations transcriptionnelles dans le cancer du sein

Représentations UMAP générée à partir des 20 premières composantes principales (PC, pour *principal component*), calculées sur **[A]** les profils d'expression des échantillons du projet 2, **[B]** les profils d'expression des échantillons du projet 1, **[C]** les profils d'expression des échantillons des lignées MCF7 cultivées dans du DMEM (DSP762) et **[D]** les profils d'expression de l'échantillon HCC70 (projet 2) tous obtenus avec Alevin (k = 19, *decoys*, GRh38,

gencode 34) normalisés par CPMedian (division par la taille de librairie et multiplication par la taille de librairie médiane, cf. section 4.2.2) et transformés avec un logarithme de base 2 et un *pseudocount* égal à 1, noté log2(+1). Les paramètres utilisés – autres que le nombre de PC – sont ceux assignés par défaut dans l'implémentation de Scanpy [26]. **[E]** Relation entre l'expression normalisée par CPMedian et log2(+1) transformée des TF GRHL2 et VGLL1 dans l'échantillon GRHL2 (projet 3).

5.5 Conclusion et perspectives

Dans le chapitre 2, j'ai démontré que la procédure de correction des CB d'Alevin, basée sur une *whitelist* répertoriant les CB valides (sans erreur) utilisés dans l'expérience, n'était pas adaptée aux données issues de la technologie Drop-Seq, pour laquelle les CB sont synthétisés aléatoirement. C'est également le cas d'autres outils de bout en bout permettant de générer une matrice d'expression à partir de fichiers FASTQ, tels que Kallisto-BUStools. En effet, les outils de bout en bout se veulent en général applicables à une diversité de technologies, mais il semblerait qu'ils soient surtout optimisés pour la technologie 10X Chromium (largement plus utilisée que ses analogues Drop-Seq et InDrops). Pour traiter des données issues de la technologie Drop-Seq, il apparaît donc préférable d'utiliser la suite logicielle Drop-Seq tools spécifiquement implémentée pour cette technologie, bien que j'aie proposé dans le chapitre 2 une solution pour malgré tout utiliser Alevin. Une suite logicielle n'est pas aussi pratique qu'un traitement de bout en bout, mais apporte en revanche plus de flexibilité. En effet, comme je l'ai constaté dans le chapitre 2, si une étape intégrée dans un outil de bout en bout nécessite un changement, la seule solution est bien souvent de modifier le code source. L'étape d'alignement dans Drop-Seq tools est effectuée avec STAR, qui fournit un alignement position par position pouvant être utile quoique le séquençage des extrémités 3' des ARNm ne permette pas une étude exhaustive des variations alléliques.

Cet alignement peut par exemple servir au QC des échantillons, que je suggère d'intégrer systématiquement dans la routine de prétraitement des données en suivant une ou plusieurs des stratégies proposées dans le chapitre 4, qui ont permis de révéler une interversion entre certaines cultures cellulaires du laboratoire.

En ce qui concerne le QC des cellules, il serait intéressant d'explorer l'outil EmptyDrops qui, comme discuté plus haut dans la section 5.3, permettrait de récupérer des cellules contenant initialement peu d'ARNm et pouvant être confondues avec des gouttelettes vides, telles que des cellules de petite taille ou des cellules quiescentes. Cela pourrait s'avérer pertinent de détecter de telles cellules, puisque les cellules souches cancéreuses (CSC) sont des cellules quiescentes. Différentes métriques de qualité pourraient ensuite être utilisées conjointement pour éliminer les cellules mortes restantes, en appliquant des seuils définis manuellement sur ces dernières. Il serait préférable que la quantification effectuée en amont avec

Drop-Seq tools soit basée sur une annotation de l'ensemble du transcriptome, puisque les gènes non-codants s'avèrent informatifs quant à la santé des cellules et peuvent constituer des métriques de qualité.

Dans le chapitre 4, j'ai montré que la transformation logarithmique utilisée en *bulk* RNA-seq pour réduire l'hétéroscédasticité n'était pas adaptée aux données Drop-Seq qui contiennent beaucoup de valeurs nulles. Si l'on considère que le domaine scRNA-seq évolue vers un plus haut débit (plus de cellules séquencées) plutôt qu'une plus grande sensibilité (plus de gènes détectés), cela pourrait signifier que la normalisation et la transformation des données ne sont pas la marche à suivre. Au lieu de ça, modéliser directement les distributions des données de comptage brutes pourrait être une stratégie plus robuste. La plupart des méthodes d'analyse actuelles sont bien sûr développées pour des données normalisées et transformées – et supposent par exemple une distribution normale de ces dernières –, permettant ainsi une application plus large, mais quelques méthodes basées sur des modèles de données de comptage brutes, tels que le modèle de Poisson ou le modèle binomial négatif, commencent à émerger.

Les résultats préliminaires du projet 1 présentés dans la section précédente ont quant à eux apporté des éléments d'information pour les expériences de Perturb-Seq prévues au sein du laboratoire. Les résultats de l'analyse différentielle ont en effet suggéré que le signal était trop « dilué » par les cellules pour lesquelles la transfection n'avait pas fonctionné : pour détecter les gènes marqueurs de la réponse œstrogénique, il faudrait ainsi comparer les cellules réellement transfectées à celles non transfectées, plutôt que de simplement comparer les différents échantillons comme il serait fait en *bulk*. Le *design* des expériences de Perturb-Seq à venir, permettant l'ajout de codes-barres indiquant quelle cellule a été transfectée avec quel siRNA, s'avère donc pertinent au vu de ces résultats.

Les résultats préliminaires du projet 3 ont quant à eux suggéré que la sensibilité de la technologie Drop-Seq limitait l'analyse de la relation entre le niveau d'expression des TF, qui sont faiblement exprimés, et celle de leurs gènes cibles. Il serait donc intéressant d'explorer les méthodes d'imputation ou de débruitage, qui tentent respectivement d'inférer les valeurs d'expression « manquantes » (nulles) ou de corriger l'ensemble des valeurs d'expression des différents gènes dans une cellule, en se basant sur les niveaux d'expression des cellules voisines (proches en termes de profil d'expression). Ces méthodes sont cependant à considérer avec précaution, car il a été démontré qu'elles introduisent de fausses corrélations entre les expressions des différents gènes [46]. Une alternative consisterait à modifier le *design* expérimental en intégrant des amorces spécifiques aux TF d'intérêts et leurs gènes cibles connus, afin d'amplifier spécifiquement les ARN issus de ces derniers. Le nombre de réplicats élevé ainsi que la quantification de l'expression des gènes au niveau cellulaire en scRNA-seq sont en effet idéaux pour étudier les corrélations entre les expressions des différents gènes.

Les résultats préliminaires du projet 2 n'ont pas permis de mettre en évidence la présence de sous-populations communes entre les différentes lignées. Il pourrait donc être intéressant d'explorer les méthodes permettant d'éliminer les variations indésirables, par exemple en utilisant un modèle des données de comptage brutes intégrant des covariables. Il est en effet possible que la variation liée à l'échantillon puisse masquer la présence de sous-populations communes – e.g. des CSC – entre les différentes lignées. Ces méthodes sont souvent utilisées pour intégrer des jeux de données de différentes technologies, voire de différentes espèces. Elles peuvent également être utilisées pour éliminer une variation technique, comme le taux de déduplication (cf. section 4.1.3), ou biologique, comme le cycle cellulaire. Les résultats préliminaires ont en effet révélé que le cycle cellulaire constituait l'hétérogénéité majeure pour la plupart des lignées.

Dans le présent mémoire, j'ai ainsi exploré diverses caractéristiques des données qui ont révélé des erreurs ou des biais liés au protocole expérimental de la technologie Drop-Seq et à la qualité des échantillons, contribuant ainsi à une meilleure compréhension des données Drop-seq. J'ai également passé en revue différentes étapes du prétraitement des données issues de cette technologie introduite nouvellement dans le laboratoire du Dr. Mader et étudié le fonctionnement de certains outils et méthodes dédiés à ces étapes. J'ai donc établi des lignes directrices pour le prétraitement des données en vue de futures expériences dans le laboratoire. Enfin, j'ai effectué une analyse exploratoire des données dont les résultats préliminaires ont posé les bases d'analyses plus poussées et apporté des informations qui pourraient s'avérer utiles pour le *design* expérimental d'expériences futures.

RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Tang F, Barbacioru C, Wang Y, et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* 2009; 6: 377–382.

[2] T S, Cm P, R T, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences of the United States of America*; 98. Epub ahead of print 9 November 2001. DOI: 10.1073/pnas.191367098.

[3] Guedj M, Marisa L, de Reynies A, et al. A refined molecular taxonomy of breast cancer. *Oncogene* 2012; 31: 1196–1206.

[4] Herschkowitz JI, Simin K, Weigman VJ, et al. Identification of conserved gene expression features between murine mammary carcinoma models and human breast tumors. *Genome Biol* 2007; 8: 1–17.

[5] R S, P F, A G, et al. Claudin-low breast cancers: clinical, pathological, molecular and prognostic characterization. *Molecular cancer*; 13. Epub ahead of print 10 February 2014. DOI: 10.1186/1476-4598-13-228.

[6] Ohnstad HO, Borgen E, Falk RS, et al. Prognostic value of PAM50 and risk of recurrence score in patients with early-stage breast cancer with long-term follow-up. *Breast Cancer Res* 2017; 19: 120.

[7] Av K, Ss J, A L, et al. Discovery and validation of breast cancer subtypes. *BMC genomics*; 7. Epub ahead of print 9 November 2006. DOI: 10.1186/1471-2164-7-231.

[8] B W, Fl B, Js R-F. The contribution of gene expression profiling to breast cancer classification, prognostication and prediction: a retrospective of the last decade. *The Journal of pathology*; 220. Epub ahead of print January 2010. DOI: 10.1002/path.2648.

[9] Prat A, Perou CM. Mammary development meets cancer genomics. *Nat Med* 2009; 15: 842–844.

[10] Diehn M, Cho RW, Lobo NA, et al. Association of reactive oxygen species levels and radioresistance in cancer stem cells. *Nature* 2009; 458: 780–783.

[11] Gx Z, Jm T, P B, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*; 8. Epub ahead of print 16 January 2017. DOI: 10.1038/ncomms14049.

[12] Macosko EZ, Basu A, Satija R, et al. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* 2015; 161: 1202–1214.

[13] Am K, L M, I A, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*; 161. Epub ahead of print 21 May 2015. DOI: 10.1016/j.cell.2015.04.044.

[14] Ziegenhain C, Vieth B, Parekh S, et al. Comparative Analysis of Single-Cell RNA Sequencing Methods. *Molecular Cell* 2017; 65: 631-643.e4.

[15] Lun ATL, Riesenfeld S, Andrews T, et al. EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol* 2019; 20: 1–9.

[16] Zhang X, Li T, Liu F, et al. Comparative Analysis of Droplet-Based Ultra-High-Throughput Single-Cell RNA-Seq Systems. *Molecular Cell* 2019; 73: 130-142.e5.

[17] Dobin A, Davis CA, Schlesinger F, et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 2013; 29: 15.

[18] Patro R, Duggal G, Love MI, et al. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods* 2017; 14: 417–419.

[19] Bray NL, Pimentel H, Melsted P, et al. Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol* 2016; 34: 525–527.

[20] Daemen A, Griffith OL, Heiser LM, et al. Modeling precision treatment of breast cancer. *Genome Biol* 2013; 14: 1–14.

[21] Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* 2011; 17: 10–12.

[22] Srivastava A, Malik L, Smith T, et al. Alevin efficiently estimates accurate gene abundances from dscRNA-seq data. *Genome Biol* 2019; 20: 1–16.

[23] T S, A H, I S. UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome research*; 27. Epub ahead of print March 2017. DOI: 10.1101/gr.209601.116.

[24] Petukhov V, Guo J, Baryawno N, et al. dropEst: pipeline for accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome Biol* 2018; 19: 1–16.

[25] Booeshaghi AS, Pachter L. Benchmarking of lightweight-mapping based single-cell RNA-seq pre-processing. *bioRxiv* 2021; 2021.01.25.428188.

[26] Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018; 19: 1–5.

[27] 10x_Technical_Note_DeadCell_Removal_RevA, https://assets.ctfassets.net/an68im79xiti/4tVumiyINGgAeoCg8SiWGG/1cf0888200d668142612c8d3f3679cf4/CG000130_10x_Technical_Note_DeadCell_Removal_RevA.pdf.

[28] Drop-Seq computational cookbook, https://github.com/broadinstitute/Drop-seq/blob/master/doc/Drop-seq_Alignment_Cookbook.pdf.

[29] Low mapping rate 2 - Ribosomal RNA. *What do you mean 'heterogeneity'?*, https://www.nxn.se/valent/2017/9/5/atnvo2gidxn1leskaboauiw6pfz0rv (accessed 27 January 2022).

[30] Why do I see high levels of Malat1 in my gene expression data? *10X Genomics*, https://kb.10xgenomics.com/hc/en-us/articles/360004729092-Why-do-I-see-high-levels-of-Malat1-in-my-gene-expression-data- (accessed 28 January 2022).

[31] Jiang P, Thomson JA, Stewart R. Quality control of single-cell RNA-seq by SinQC. *Bioinformatics* 2016; 32: 2514.

[32] Suter DM, Molina N, Gatfield D, et al. Mammalian genes are transcribed with widely different bursting kinetics. *Science* 2011; 332: 472–474.

[33] Marinov GK, Williams BA, McCue K, et al. From single-cell to cell-pool transcriptomes: Stochasticity in gene expression and RNA splicing. *Genome Res* 2014; 24: 496–510.

[34] Lopez R, Regier J, Cole MB, et al. Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018; 15: 1053–1058.

[35] Svensson V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol* 2020; 38: 147–150.

[36] Cao Y, Kitanovski S, Küppers R, et al. UMI or not UMI, that is the question for scRNA-seq zero-inflation. *Nat Biotechnol* 2021; 39: 158–159.

[37] Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. *Nat Methods* 2018; 15: 255–261.

[38] Chen EY, Tan CM, Kou Y, et al. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics* 2013; 14: 128.

[39] J B, G C, N S, et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*; 483. Epub ahead of print 28 March 2012. DOI: 10.1038/nature11003.

[40] Li B, Dewey CN. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* 2011; 12: 1–16.

[41] Robinson JT, Thorvaldsdóttir H, Winckler W, et al. Integrative Genomics Viewer. *Nature biotechnology* 2011; 29: 24.

[42] Melsted P, Booeshaghi AS, Gao F, et al. Modular and efficient pre-processing of single-cell RNA-seq. 2019; 673285.

[43] Kaminow B, Yunusov D, Dobin A. STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data. *bioRxiv* 2021; 2021.05.05.442755.

[44] Estimating the number of true cell barcodes in single cell RNA-Seq (part 2). *CGAT*, https://cgatoxford.wordpress.com/2017/05/23/estimating-the-number-of-true-cell-barcodes-in-single-cell-rna-seq-part-2/ (2017, accessed 31 January 2022).

[45] Hippen AA, Falco MM, Weber LM, et al. miQC: An adaptive probabilistic framework for quality control of single-cell RNA-sequencing data. *PLOS Computational Biology* 2021; 17: e1009290.

[46] Andrews TS, Hemberg M. False signals induced by single-cell imputation. *F1000Res* 2019; 7: 1740.