Stage de fin d'études
Option Mathématiques Appliquées

Inférence Variationnelle et Auto Encodeurs Variationnels pour la réduction de dimension des données single-cell RNA sequencing, application au myélome multiple.

Anthony Ozier-Lafontaine

Superviseurs :
Bertrand Michel
Stéphane Minvielle

Du 1 avril au 27 septembre 2019

Table des matières

Chapitre 1

Contexte Biologique : le Myélome Multiple (MM)

1.1 Description de la maladie

Description générale Le Myélome Multiple (MM) est une hémopathie maligne qui affecte les cellules plasmocytaires dans la moelle osseuse, il est caractérisé par une prolifération aberrante de ces cellules. Aucune cause précise n'a été isolée, et certains facteurs de risques sont suspectés, comme l'exposition aux radiations et aux pesticides, ou aux produits pétroliers tels que le benzène. Le myélome multiple représente 1.7% des cancers, et environ 10% des cancers hématologiques [Kumar et al., 2017]. La moyenne d'age des patients atteints est de soixante-cinq ans au diagnostic. Il n'existe à ce jour aucun traitement curatif et l'espérance de vie médiane est de l'ordre de sept ans.

Les plasmocytes Les cellules tumorales du myélome multiple sont des plasmocytes à longue durée de vie présents dans la moelle et le sang périphérique. Ces cellules immunitaires dont la fonction principale est de produire des immunoglobulines, sont issues des lymphocytes B elles mêmes issues des cellules souches hématopoïetiques à l'origine de toutes les cellules sanguines. Les plasmocytes sont formées à partir de lymphocytes B matures.

Apparition du myélome multiple Les plasmocytes à longue durée de vie sont terminalement différenciées et ne sont pas censés proliférer. Une cellule terminalement différenciée est une cellule spécialisée qui ne se divise plus et ne donne pas naissance à de nouveaux types cellulaires. La prolifération de ces cellules est donc anormale, mais elle n'est pas toujours symptomatique. Elle est par exemple asymptomatique dans le cas du myélome indolent ou SMM (Smoldering Multiple Myeloma) qui est un stade précurseur du myélome multiple. Du point de vue génétique, le myélome multiple regroupe en réalité plusieurs maladies différentes aux symptômes similaires. Il existe plusieurs scénarios cellulaires menant à l'apparition des symptômes du myélome multiple. Même si les événements initiaux sont relativement connus et peu nombreux, le myélome multiple semble être particulièrement chaotique et chaque patient est unique du point de vue cellulaire.

Diagnostic La plupart des myélomes multiples sont découverts fortuitement lors d'un examen de routine ou lié à une autre maladie. Les symptômes du myélome multiple sont résumés par les médecins par l'acronyme CRAB :

- (C) hypercalcémie : taux de calcium dans le sang élevé
- (R) insuffisance rénale
- (A) anémie
- (B) lésions osseuses

Une fois le myélome multiple repéré, des examens approfondis permettent de déterminer les spécificités clinques du patient et le traitement optimal à lui attribuer, parmi ces spécificités, on trouve les anomalies génétiques spécifiques de son myélome. Ces examens approfondis s'effectuent notamment avec un prélèvement de moelle osseuse au niveau du sternum. Les échantillons de cellules que j'ai étudiés pendant mon stage proviennent de prélèvements effectués chez des patients atteints de myélome multiple inclus dans le protocole

MYRACLE [Benaniba et al., 2019]. Les plasmocytes tumoraux prélevés dans le sternum sont des cellules circulantes qui se sont disséminées à partir de foyers. Une cellule circulante est une cellule qui n'est fixée à aucun tissu et peut se déplacer dans le corps. Elles ne sont pas forcément représentatives des plasmocytes des foyers tumoraux qui sont localisés à l'intérieur des os. Avec les plasmocytes sont aussi prélevés d'autres cellules de la moelle qui ne sont pas tumorales, ces cellules d'environnement peuvent nous apprendre comment l'environnement cellulaire réagit au traitement et peut modifier le phénotype de la cellule tumorale.

1.2 Problématique du projet SIRIC

Mon stage s'intègre dans le projet SIRIC ILIAD. Le label SIRIC (Site de Recherche Intégrée sur le Cancer) est distribué par l'*Institut National du Cancer* (INCa) dans le cadre d'un appel à candidature. Le consortium *Imaging and Longitudinal Investigations to Ameliorate Decision Making* (ILIAD) rassemble 3 établissements de santé, 27 plateformes technologiques de haut niveau et 9 laboratoires dont le *Centre de Recherche en Cancerologie et Immunologie Nantes-Angers* (CRCINA) où j'ai fait ce stage et le *laboratoire de mathématiques Jean Leray* (LMJL) où je continuerai à travailler sur ce projet en thèse. L'un des objectifs d'ILIAD est d'étudier la résistance des cancers aux traitements et la variabilité d'expression génique chez les cellules tumorales. L'équipe dont je fais partie étudie les échantillons de moelle prélevés au diagnostic dans deux conditions expérimentales : un échantillon contrôle et un échantillon traité *ex vivo*, i.e. après prélèvement, avec un traitement expérimental. À l'aide de la technologie *single cell RNA sequencing* (scRNAseq), on peut mesurer le transcriptome de chaque cellule de chaque condition. On obtient ainsi une information précise sur chaque cellule de l'échantillon prélevé et on peut alors étudier l'effet du traitement.

Chapitre 2

Contexte Expérimental : Single Cell RNA-sequencing

"L'homme ne peut observer les phénomènes qui l'entourent que dans des limites très restreintes ; le plus grand nombre échappe naturellement à ses sens, et l'observation simple ne lui suffit pas. Pour étendre ses connaissances, il a dû amplifier, à l'aide d'appareils spéciaux, la puissance de ces organes, en même temps qu'il s'est armé d'instruments divers qui lui ont servi à pénétrer dans l'intérieur des corps pour les décomposer et en étudier les parties cachées."

Introduction à la médecine expérimentale, Claude Bernard, 1865.

2.1 Fondamentaux biologiques

La cellule. L'objet d'étude de l'approche single-cell RNA-sequencing (scRNAseq) est la cellule. Dans le cadre de la théorie cellulaire, la cellule est l'unité de base de tous les organismes biologiques. [Shleiden, 1838] [Schwann, 1839]. L'Homo-Sapiens est un organisme pluricellulaire constitué de cellules eucaryotes, des cellules composées d'un noyau ainsi que de plusieurs organites, isolées du milieu extérieur par une membrane plasmique. Parmi les organites qui peuplent la cellule, les mitochondries nous intéressent particulièrement car à l'instar du noyau, elles possèdent aussi un acide désoxyribo-nucléique (ADN). Toutes les cellules d'un même organisme partagent la même séquence d'ADN à quelques mutations aléatoires près.

Le transcriptome. L'ADN est un polymère constitué de chaines de nucléotides. Les 4 nucléotides de l'ADN sont la thymine (T), la guanine (G), la cytosine (C) et l'adénine (A). L'information génétique contenue dans l'ADN sert en partie à la production de protéines, la constitution de chaque protéine sécrétée par une cellule est codée sur une séquence particulière de l'ADN qu'on appelle un gène. Les gènes codant pour des protéines représentent 2% du génome humain. Depuis les premiers séquençages complets du génome dans les années 2000, on sait que l'ADN nucléique contient environ 30 000 gènes codants, et l'ADN mitochondrial en contient exactement 37. Les gènes codant pour des protéines sont lus par les ARN polymérase pour créer des ARN messagers, des polymères orientés 3' - 5', ces derniers sont relâchés dans le cytoplasme et traduits en protéines par des ribosomes. Les protéines sont des unités fonctionnelles de la cellule, elles remplissent diverses fonctions, certaines sont expulsées de la cellule et participent à la communication intercellulaire, d'autres se fixent à la membrane plasmique et participent aux interactions entre le milieu extracellulaire et l'intérieur de la cellule, et les protéines restantes remplissent un large panel de fonctions nécessaires au fonctionnement de la cellule. On appelle transcriptome d'une cellule l'ensemble des ARN messagers de son cytoplasme à un instant donné.

La technologie scRNAseq permet de quantifier le transcriptome d'une cellule. Dans cette étude, le transcriptome quantifié est la seule information globale disponible pour chaque cellule. Ce transcriptome est un reflet quantitatif de l'ensemble des protéines sécrétées par la cellule, et ces protéines participent à l'activité fonctionnelle des cellules.

2.2 L'épopée des échantillons : Du prélèvement à la donnée

La technologie scRNAseq [Macosko et al., 2015] nous offre un degré de précision que les biologistes n'osaient même pas imaginer il n'y a pas si longtemps. Elle permet de mesurer l'expression génique des cellules individuellement à travers la quantification de leur transcriptome. Avant le single-cell, la quantification du transcriptome existait déjà à l'échelle de la population de cellules, on était alors incapables d'attribuer chaque brin d'ARN messager capturé à une cellule particulière. On avait donc accès à une quantification du transcriptome moyen de la population de l'échantillon.

Contrôle/Traité. Suite au prélèvement de moelle osseuse au diagnostic, les cellules sont séparées en deux échantillons, l'échantillon contrôle C et l'échantillon traité T. L'échantillon C est plongé dans un milieu neutre, et l'échantillon T est mis en présence du traitement qu'on souhaite étudier. Après un temps suffisant pour que le traitement ait agi, on passe à l'étape qu'on appelle préparation de la librairie d'ADN. La préparation de la librairie correspond à l'ensemble des traitements nécessaires pour passer d'un échantillon de cellules vivantes à la librairie. La librairie est une solution contenant uniquement des brins d'ADN complémentaires des ARN messagers capturés dans les cellules sélectionnées par l'étape de purification et prêts à être séquencés. Le séquençage est la lecture et la numérisation des brins d'ADN de cette librairie, ce sont ces données que j'ai étudiés.

Purification. Les cellules qui nous intéressent dans la moelle osseuse sont les plasmocytes, ce sont des cellules mononucléées, et elles sont présentes en faible proportion dans les échantillons. La purification consiste à isoler les cellules mononucléées de l'échantillon, et parfois même à isoler les plasmocytes.

- Isolation des cellules mono-nucléées. les cellules sont mélangées à une solution à gradient de densité, le Ficoll, avant d'être centrifugées. Cette manipulation permet de réunir toutes les cellules mono-nucléées de l'échantillon, en suspension dans un anneau. Cet anneau est prélevé à l'aide d'une pipette.
- Isolation des plasmocytes. Des billes couvertes de marqueurs protéiques spécifiques des plasmocytes sont fixées aux parois d'une colonne dans laquelle on verse les cellules mononucléées. Les plasmocytes s'accrochent aux billes tandis que les autres cellules descendent au fond du tube.

Aspect Single-Cell. La spécificité du single-cell consiste à isoler les cellules dans des bulles à l'aide d'une méthode découverte dans le domaine de la micro-fluidique. Cette méthode permet de créer une émulsion de bulles de milieu cellulaire. Chaque cellule est encapsulée dans une bulle contenant un agent de lyse et une bille. La fonction de l'agent de lyse est de détruire la membrane de la cellule, et la bille est recouverte de brins d'ADN censés s'hybrider avec les ARN messagers. En effet, l'ADN et l'ARN sont des structures moléculaires très semblables et capables de s'hybrider l'une à l'autre. L'ADN est une structure moléculaire plus robuste que l'ARN. Chaque brin d'ADN de la bille sert à marquer les ARN messagers capturés, pour cela, il est divisé en trois parties :

- une chaine de peptides spécifiques de la bille, le Code Barre de l'encapsulation.
- une chaine de peptides unique appelée *unique molecular identifiant* (UMI).
- une chaine de thymines pour s'hybrider avec l'ARN messager qui possède une queue d'adénines appelée queue poly-A.

L'agent de lyse libère les ARN messagers du cytoplasme, et les fragments d'ADN se décrochent de la bille et s'hybrident aux ARN messagers à proximité. Chaque brin d'ARN messager hybridé avec un brin d'ADN de la bille est considéré comme capturé et sera potentiellement présent dans la librairie finale. Pour assurer l'encapsulation de toutes les cellules, les billes sont en excès dans cette manipulation et moins de 10% des billes seront encapsulées avec une cellule. La figure 2.1 illustre le processus d'encapsulation.

RT-PCR. La solution résultant de la manipulation single-cell est nettoyée pour ne garder que les ARN messagers hybridés avec des brins d'ADN de bille. Ensuite, une *reverse transcriptase* (RT) est appliquée pour transformer chaque brin d'ARN messager capturé en ADN complémentaire. A ce stade, la solution contient des brins d'ADN complémentaires des ARN messagers prolongés par le code barre spécifique de l'encapsulation et l'UMI. Mais chacun de ces fragments est unique, or, le séquençage est sujet à des erreurs. On amplifie le signal pour le rendre plus robuste en dupliquant un grand nombre de fois chaque fragment d'ADN à l'aide

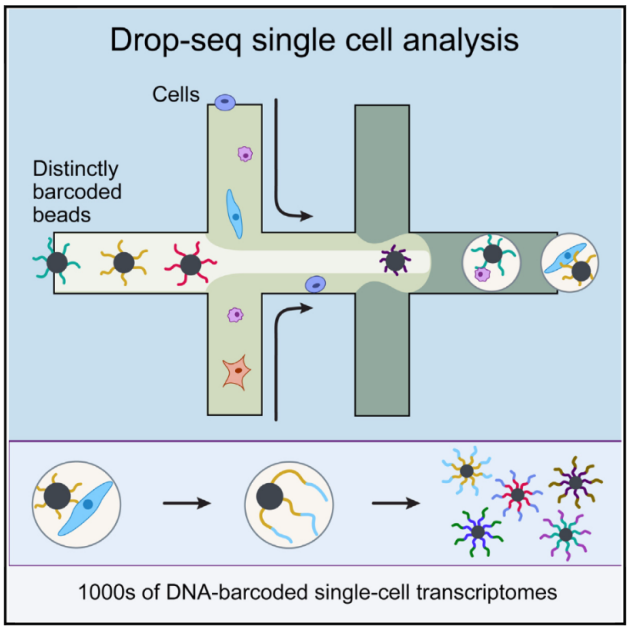

FIGURE 2.1 – Schéma de l'encapsulation cellulaire lors de la manipulation single-cell RNA sequencing [Macosko et al., 2015]

d'une *polymerase chain reaction* (PCR). La librairie est la solution obtenue après ces deux manipulations qu'on appelle RT-PCR.

Séquençage. Le séquençage est effectué sur une machine fabriquée par l'entreprise Illumina. La librairie est insérée dans un compartiment qu'on appelle un LANE. Chaque LANE est contenu dans un Flowcell. La LANE déverse la librairie sur une bande parsemée d'îlots d'accroches où les fragments d'ADN de la librairie se fixent. Un processus permet de forcer chaque îlot d'accroches à ne contenir que des copies du même fragment d'ADN. Une fois que chaque îlot est peuplé, des nucléotides fluorescents sont relachés pour s'hybrider aux brins fixés, et une photographie de haute qualité est prise à chaque hybridation de nucléotide. On sait quel nucléotide s'est hybridé aux brins d'un îlot grâce à la couleur de fluorescence de l'îlot sur la photographie. La succession des images permet de reconstituer la suite de nucléotide du brin d'ADN de chaque îlot.

Alignement sur un génome de référence et matrice de comptage. Un fichier contenant toutes ces images est généré, c'est le fichier BCL. A partir de cette étape, il n'y a plus aucune manipulation biologique et tous les traitements ultérieurs sont numériques. Un algorithme d'analyse d'images fourni par Illumina génère un fichier au format *.txt* avec la suite de nucléotides constituant chaque brin séquencé, c'est le fichier fastQ. Dans ce fichier, on a accès au code barre spécifique de l'encapsulation, à l'UMI et au 98 premièrs nucléotides de l'ADN complémentaire de l'ARN messager. Les gènes sont alignés sur une annotation du génome de référence hg38 / GRCh38 [Schneider et al., 2016] pour identifier le gène correspondant à chaque fragment d'ADN séquencé, à l'aide de l'outil *CellRanger* fourni par l'entreprise *10X Genomics*. Le code barre spécifique de l'encapsulation et l'UMI permettent d'identifier les multiples copies des fragments d'ADN complémentaire amplifiés par la PCR. Le fichier BAM synthétise les résultats de ces deux traitements. A partir de ce fichier, on construit la matrice de comptage avec une ligne par code barre spécifique d'encapsulation et une colonne par gène défini dans le génome de référence, la valeur d'une case correspond au nombre d'UMI différents correspondants à la ligne et la colonne, il s'agit du comptage d'ARN messagers.

Chapitre 3

Contexte statistique : modélisation et analyse des données scRNAseq

3.1 Pré-traitements et bonnes pratiques

Filtrage. La matrice de comptage est l'objet sur lequel sont effectuées toutes les analyses statistiques. Dans un premier temps, on réduit grossièrement sa taille. On réduit le nombre de lignes en discriminant les codes barres spécifiques d'encapsulation qui correspondent à une cellule encapsulée par rapport à celles qui n'ont pas encapsulé de cellules. La quantité d'ARN messagers distincts, c'est à dire la quantité d'UMI, est bien plus importante quand une cellule a été encapsulée. On définit alors un seuil spécifique à chaque expérimentation pour distinguer les deux. On réduit le nombre de colonnes de la matrice en identifiant les gènes non-exprimés ou très peu exprimés dans l'échantillon de cellules. Une condition arbitraire est choisie, par exemple, ne garder que les gènes exprimés dans au moins 3 cellules distinctes. Ces pré-traitements préliminaires ne rendent pas les données exploitables directement et un travail minutieux d'exploration des données est souvent nécessaire pour détecter les cellules et gènes aberrants.

Biais spécifiques. Il y a deux biais connus spécifiques à cette technologie et dont les méthodes de correction sont robustes.

- **Les multiplets** : on trouve parfois plusieurs cellules encapsulées avec une seule bille, on parle de multiplet. Les multiplets les plus fréquents sont des doublets. Ces observations sont faussées car il n'est pas possible d'attribuer chaque gène transcrit à une cellule d'origine. Il existe des méthodes statistiques pour détecter les multiplets [DePasquale et al., 2018] [McGinnis et al., 2019] [Wolock et al., 2018]
- **L'ARN de la soupe** : la solution dans laquelle les cellules sont en suspension pendant l'étape d'encapsulation est appelée la soupe. Elle contient parfois les gènes transcrits relâchés par les cellules fragilisées lors des étapes précédentes et lysées avant l'encapsulation. Ces gènes transcrits vont dégrader le signal de chaque cellule. La méthode de référence pour détecter les transcrits de la soupe utilise les ARN messagers détectés dans les encapsulations sans cellules [Young and Behjati, 2018].

Exception faite des approches astucieuses qui utilisent les données brutes [La Manno et al., 2018], la matrice de comptage filtrée et pré-traitée ainsi obtenue est l'objet d'étude de la plupart des analyses statistiques scRNAseq. Dans notre cas, ces données nous offrent une vision d'ensemble sur les cellules tumorales et d'environnement, mais elles ne peuvent être étudiées à la main et l'usage d'outils statistiques est indispensable. En effet, nous sommes dans un cas typique de BIG DATA avec plusieurs gigaoctets par patient, et en même temps dans celui des statistiques en grande dimension avec plusieurs dizaines de milliers de variables (gènes) par cellule observée. Il existe donc beaucoup d'approches statistiques en fonction de la problématique biologique.

3.2 Objectifs des analyses

Le bestiaire des méthodes existantes pour étudier ces données est énorme et de nouvelles approches sont publiées chaque jour. Derrière cette diversité d'approches, les objectifs sont souvent les mêmes :

Normaliser. Les cellules d'un même échantillon présentent souvent des différence d'expression très importantes. Afin de rendre ces données comparables, une étape de normalisation est presque indispensable. Il existe de nombreuses façons de normaliser les données, la difficulté étant de rendre ces données comparables sans pour autant gommer cette différence informative d'expression. La plupart du temps, la normalisation consiste en un facteur de taille de la cellule et en la définition d'une fonction de normalisation dépendant de ce facteur, appliquée à chaque comptage de gènes dans chaque cellule. La détermination de ce facteur de taille peut prendre en compte le type cellulaire ou bien la spécificité de chaque gène.

Réduire la dimension. En grande dimension, les données sont difficilement comparables. De nombreuses méthodes de réduction de dimension ont été publiées pour ces données. La réduction de dimension consiste à projeter les observations dans un espace de dimension inférienre au nombre de gènes. Les réductions de dimensions les plus classiques sont l'analyse en composantes principales (PCA) ou la factorisation en matrices non-négatives (NMF) [Welch et al., 2018]. Certaines méthodes de réduction de dimension ont pour but la visualisation des données, les algorithme tSNE [van der Maaten and Hinton, 2012] et UMAP [McInnes et al., 2018] sont les plus utilisés et permettent de représenter visuellement les cellules en réduisant la dimension des observations à deux ou trois de manière non linéaire. Enfin, il est possible de réduire la dimension des données de manière probabiliste [Lopez et al., 2018][Ding et al., 2018] [Wang and Gu, 2017], ce type d'approche est décrit dans le chapitre suivant.

Identifier les types cellulaires. La technologie scRNAseq contribue massivement au projet "Human Cell Atlas" dont l'objectif est d'identifier, de localiser et de caractériser les types cellulaires qui composent le corps humain. Pour identifier les types cellulaires d'un échantillon à partir de la quantification du transcriptome, on étudie en général les gènes qui caractérisent les clusters obtenus par une méthode non supervisée sur les données en dimension réduite[Xu and Su, 2015] [Levine et al., 2015]. L'un des grands enjeux de ce projet est de permettre la découverte de types cellulaires rares encore non caractérisés. Pour cela, on peut par exemple segmenter l'espace des cellules et identifier les cellules qui se retrouvent souvent isolées à l'issu de la segmentation [Jindal et al., 2018]. Ces méthodes sont appliquées à toutes les études scRNAseq car on passe toujours par une étape d'identification des types cellulaires en présence dans un échantillon.

Reconstruire les trajectoires et la phylogénie cellulaires. Les cellules sont des objets dynamiques dont le transcriptome change au cours du temps et de nombreuses études s'intéressent à l'expression génétique au cours du temps. Or une matrice de comptage est l'observation d'une population de cellules à un instant donné. Pour contourner cette limitation, il existe des méthodes pour construire un pseudo-temps en considérant que toutes les cellules du même type dans l'échantillon sont des observations de la même cellule à des instants différents [La Manno et al., 2018] [Levine et al., 2015] [Ding et al., 2019].

3.3 Nature statistique des données de comptage

Il y a deux familles d'approches pour l'étude des données scRNAseq. La première famille d'approche est basée sur des métriques. Ces approches consistent à considérer les comptages observés sans prendre en compte leur nature probabiliste, et à analyser les cellules ou les gènes à partir de ces observations [Butler and Satija, 2017]. La seconde famille d'approches considère que les comptages observés sont des réalisations d'une variable aléatoire. On modélise en général les comptages par des lois discrètes sur-dispersées [Hafemeister and Satija, 2019], une composante *zero-inflated* (ZI) peut parfois s'avérer utile pour prendre en compte l'aspect épars des données [Pierson and Yau, 2015] [Svensson, 2019]. L'intérêt de cette approche aléatoire est de permettre la modélisation des différentes sources de variance des données avec des modèles hiérarchiques [Vallejos et al., 2015].

Une part importante de mon travail de stage a été de prendre en main l'outil de réduction de dimension probabiliste *single-cell Variational Inference* (scVI) [Lopez et al., 2018] et de l'appliquer aux données de l'équipe dans l'espoir de faire apparaitre plus de structure qu'avec les réductions de dimension linéaires. Cet outil de la seconde famille d'approches est basé sur un modèle hiérarchique qui distingue les sources de variance technologiques et biologiques, et considère que les comptages sont distribués selon une loi *zero-inflated negative binomial* (ZINB). L'implémentation de scVI repose sur les *auto-encodeurs variationnels* (VAE) [Kingma and Welling, 2013] pour estimer les lois des variables latentes du modèle hierarchique à partir des

comptages. Les VAE sont des réseaux de neurones qui implémentent une résolution de l'*inférence variationnelle* (VI) [Blei et al., 2017]. Dans le chapitre suivant, je présenterai scVI après avoir introduit la VI et les VAE.

Chapitre 4

Application de l'inférence variationnelle aux données scRNAseq

4.1 Introduction à l'inférence variationnelle

4.1.1 Présentation du problème.

Dans beaucoup de sciences expérimentales, l'objet que l'on souhaite étudier est abstrait ou difficile à observer. On est alors contraint d'acquérir de l'information sur cet objet par des moyens détournés, à travers l'observation de ses manifestations. Par exemple, en écologie, l'objet d'intérêt est le réseau des interactions entre les espèces d'un écosystème. Ce réseau est un objet latent, il n'est pas possible de l'observer directement, on l'infère à partir des mesures de la quantité d'individus des espèces d'un écosystème au cours du temps. Un autre exemple en archéologie, où l'objet d'intérêt est le mode de vie des humains et la structure de leur société à une époque donnée, encore une fois, il n'est pas possible d'observer cet objet d'études directement, et l'on est contraint de le déduire de manière détournée à travers l'étude des documents, objets et fossiles révélés lors de fouilles. De même, la technologie single-cell RNA sequencing permet l'observation détournée de l'ensemble des mécanismes biologiques d'une population de cellules, à travers la quantification des ARN messagers dans leur cytoplasme. Dans toutes ces situations, non seulement l'observation effectuée est incomplète et incertaine, mais en plus, la déduction d'une information sur l'objet d'intérêt est difficile. Le formalisme des statistiques Bayésiennes est approprié à la description de ce type de problème. En effet, il permet de prendre en compte à la fois l'incertitude de l'observation et l'incertitude de la déduction.

4.1.2 Formalisme.

Avant de commencer, je précise que dans la littérature scientifique relative à l'inférence variationnelle, il est d'usage de noter les densités avec leur variable. J'ai gardé ce formalisme, ainsi, j'écris $p(z)$ pour désigner la densité sur z $p(.)$. De même pour une densité conditionnelle, j'écris $p(x|z)$ pour désigner la densité conditionnelle de x sachant z $p(.|z)$. En cas d'ambiguïté, je préciserai clairement les objets dont je parle.

Notons $x = (x_n)_{n=1}^N$ les N réalisations de la quantité observable, et $z = (z_m)_{m=1}^M$ les M variables latentes. Dans un contexte probabiliste, nous souhaitons déduire à partir des observations x non pas la valeur exacte de la variable latente z, mais plutôt la densité dite postérieure $p(z|x)$ de la variable z sachant x. L'estimation de cette densité sert à établir un intervalle de confiance pour z et prédire les observations suivantes de x.

Pour déterminer la densité postérieure $p(z|x)$, on doit d'abord avoir une idée du lien entre les observations x et les variables latentes z. Le choix du modèle génératif consiste à modéliser la loi de z de densité $p(z)$ appelée le *prior*, et la façon dont x dépend de la réalisation de z avec la densité conditionnelle $p(x|z)$. La formule des probabilités conditionnelles nous donne accès à la densité jointe $p(x, z)$,

$$p(x, z) = p(z)p(x|z).$$

On remarque que le choix du *prior* est relativement arbitraire puisqu'il n'y a en général aucun moyen de connaître la véritable distribution des variables latentes. En général, on s'arrange surtout pour que la distribution conditionnelle soit compatible avec les observations x, par exemple en choisissant une loi discrète pour un comptage. La formule de Bayes relie le modèle génératif à la densité postérieure qui nous intéresse.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x,z)}{\int_z p(x|z)p(z)\,dz}.$$

On voudrait utiliser cette formule pour déterminer explicitement la densité postérieure, sauf que le dénominateur $p(x)$, appelé la densité marginale ou évidence, est en général impossible ou trop long à calculer. Nous avons donc besoin d'une méthode pour estimer la densité postérieure sans calculer l'évidence $p(x)$.

4.1.3 État de l'art

Il existe deux grandes familles de méthodes pour contourner le problème de la densité marginale $p(x)$ et estimer la densité postérieure $p(z|x)$ [Blei et al., 2017] .

- **Méthode de Monte Carlo basée sur des chaines de Markov.** Les méthodes appelées *Monte-Carlo Markov Chain* (MCMC) sont les plus anciennes. Elles consistent en la construction d'une chaine de Markov ergodique ayant pour état stationnaire la distribution postérieure $p(z|x)$. On estime donc $p(z|x)$ en réalisant cette chaine jusqu'à la convergence. Ces méthodes présentent l'avantage d'être asymptotiquement exactes, mais le coût computationnel nécessaire à la convergence peut s'avérer élevé.
- **Inférence Variationnelle.** Les méthodes de VI reformulent le problème de l'estimation de la densité postérieure en un problème d'optimisation. On choisit une famille variationnelle, c'est à dire un ensemble de densités $\mathcal{Q}$ sur la variable z. La VI consiste à déterminer la densité $q^* \in \mathcal{Q}$ la plus proche de la densité postérieure $p(z|x)$. q^* est une approximation de la densité postérieure, excepté le cas particulier où $p(z|x) \in \mathcal{Q}$. Pendant mon stage, je me suis uniquement intéressé aux méthodes d'inférence variationnelle.

4.1.4 Borne inférieure de l'évidence.

On appelle log-vraisemblance d'une densité $p(x)$ la fonction :

$$x \longmapsto log\, p(x).$$

La plupart des méthodes d'inférence variationnelle utilisent la mesure de dissimilarité appelée divergence de Kullback-Leibler ($\mathbb{KL}$) pour mesurer la proximité entre une distribution variationnelle $q \in \mathcal{Q}$ et la densité postérieure sur z $p(z|x)$. On parle de distribution variationnelle pour bien faire la distinction entre la densité réelle p et la densité approximée q. On définit la divergence $\mathbb{KL}$ entre deux densités de probabilité $p_1(z)$ et $p_2(z)$ ainsi :

$$\mathbb{KL}(p_1(z)\|p_2(z)) = \mathbb{E}_{p_1}\left[log\,\frac{p_1(z)}{p_2(z)}\right] \text{ si } p_1 \ll p_2.$$

On note que cette divergence n'est pas symétrique, et qu'elle est positive. Cette seconde propriété s'obtient en appliquant l'inégalité de Jensen.

$$\begin{aligned}\mathbb{KL}(p_1(z)\|p_2(z)) &= -\mathbb{E}_{p_1}\left[log\,\frac{p_2(z)}{p_1(z)}\right] \\ &\geq -log\,\mathbb{E}_{p_1}\left[\frac{p_2(z)}{p_1(z)}\right] = -log(\int_z p_2(z)\,dz) = 0.\end{aligned}$$

Soit $q(z) \in \mathcal{Q}$ une densité variationnelle, on a la divergence $\mathbb{KL}$ entre $q(z)$ et $p(z|x)$:

$$\begin{aligned}\mathbb{KL}(q(z)\|p(z|x)) &= \mathbb{E}_{q(z)} log\, q(z) - \mathbb{E}_{q(z)} log\, p(z|x) \\ &= \mathbb{E}_{q(z)} log\, q(z) - \mathbb{E}_{q(z)} log\, p(x,z) + \mathbb{E}_{q(z)} log\, p(x) \\ &= \mathbb{E}_{q(z)} log\, q(z) - \mathbb{E}_{q(z)} log\, p(x,z) + log\, p(x).\end{aligned}$$

On remarque que ce calcul passe par la détermination de la log-vraisemblance de l'évidence $log_p(x)$, qui est, comme l'évidence $p(x)$, en général impossible ou trop longue à calculer.

On définit l'*evidence lower bound* (ELBO) pour une observation x et une densité variationnelle q par

$$\begin{aligned}\mathcal{ELBO}(x,q) &= \mathbb{E}_{q(z)}\ [log\, p(x,z)] - \mathbb{E}_{q(z)}\ [log\, q(z)] &(4.1)\\ &= \mathbb{E}_{q(z)}\ [log\, p(x|z)] - \mathbb{KL}(q(z)\|p(z)). &(4.2)\end{aligned}$$

On a alors

$$\mathbb{KL}(q(z)\|p(z|x)) = log\, p(x) - \mathcal{ELBO}(x,q). \tag{4.3}$$

Comme $\mathbb{KL}$ est positive, l'ELBO minore la log-vraisemblance de l'évidence. Notons qu'on peut aussi obtenir cette inégalité à partir de l'inégalité de Jensen :

$$\begin{aligned}log\, p(x) &= log \int_z p(x,z)\, dz = log \int_z p(x,z) \frac{q(z)}{q(z)}\, dz \\ &= log\, \mathbb{E}_q \frac{p(x,z)}{q(z)} \geq \mathcal{ELBO}(x,q).\end{aligned}$$

Ainsi, minimiser $\mathbb{KL}(q(z)\|p(z|x))$, revient à maximiser l'ELBO car $log\, p(x)$ est constant (et inconnu). Les méthodes d'inférences variationnelle se résument donc à choisir une famille variationnelle $\mathcal{Q}$ et à maximiser l'ELBO sur cette famille.

4.1.5 Lien avec EM

La famille d'algorithmes appelés *Expectation Maximization* (EM) maximisent aussi la log-vraisemblance marginale $log\, p(x|\theta)$ où θ est l'ensemble des paramètres de la distribution dans un problème à variables latentes z [Dempster et al., 1977]. Comme on l'a vu, cette log-vraisemblance est en général inaccessible, et on ne peut pas non-plus maximiser la log-vraisemblance complète $log\, p(x,z|\theta)$ puisqu'on ne connait pas les valeurs des variables latentes. A défaut, les algorithmes EM passent par le calcul de l'espérance de la log-vraisemblance complète $\mathbb{E}_{p(z|x,\theta)}[log\, p(x,z|\theta)|x]$, qui peut être estimée.

L'identité de Fisher permet de faire le lien avec la log-vraisemblance marginale :

$$log\, p(x|\theta) = \mathbb{E}_{p(z|x,\theta^{(t)})}\left[log\, p(x,z|\theta)|x\right].$$

On remarque que dans le cas où $p(z|x,\theta^{(t)}) \in \mathcal{Q}$, l'équation (4.3) mène directement à :

$$log\, p(x|\theta) = \mathcal{ELBO}(\theta).$$

4.1.6 Choix de la famille variationnelle

La famille variationnelle $\mathcal{Q}$ classiquement utilisée est appelée *mean-field variational family* et chaque élément vérifie une hypothèse d'indépendance entre les composantes latentes. Pour $q \in \mathcal{Q}$, on a :

$$q(z) = \prod_{m=1}^{M} q_{z_m}(z_m). \tag{4.4}$$

L'hypothèse d'indépendance limite la capacité de q^* à approcher $p(z|x)$ car elle est incapable de saisir les éventuelles dépendances. Cela donne lieu à une variance de q^* inférieure à la variance de la densité postérieure $p(x|z)$. L'avantage de la famille *mean-field* est de permettre une optimisation variable latente par variable latente et d'appliquer ainsi un algorithme de descente cyclique de coordonnées.

4.1.7 Optimiser l'ELBO sur la famille variationnelle

Soit $q \in \mathcal{Q}$. On va mettre q à jour de façon itérative, en modifiant les facteurs du produit (4.4) un à un. Pour $k \in [|1, M|]$, on optimise l'ELBO par rapport à la densité q_{z_k} en fixant $\{q_{z_j}, \forall j \neq k\}$. On a :

$$\begin{aligned}
\mathcal{ELBO}(x, q) &= \mathbb{E}_{q(z)} \log p(x, z) - \mathbb{E}_{q(z)} \log q(z) \\
&= \mathbb{E}_{q_{z_k}(z_k) \prod_{j \neq k} q_{z_j}(z_j)} \log p(x, z) - \sum_j \mathbb{E}_{q z_j(z_j)} \log q_{z_j}(z_j) \\
&= \mathbb{E}_{q_{z_k}(z_k)} \left[\mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z) \right] - \mathbb{E}_{q_{z_k}(z_k)} \log q_{z_k}(z_k) + \mathcal{C}_{-k} \\
\mathcal{ELBO}(x, q) &= -\mathbb{KL}(q_{z_k}(z_k) \| \tilde{p}(x, z)) + \mathcal{C}_{-k},
\end{aligned} \tag{4.5}$$

où $\mathcal{C}_{-k}$ désigne une constante et

$$\begin{aligned}
q_{z_{-k}}(z_{-k}) &= \prod_{j \neq k} q_{z_j}(z_j), \\
\log \tilde{p}(x, z) &= \mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z).
\end{aligned}$$

L'écriture (4.5) montre que maximiser l'ELBO par rapport à q_{z_k} revient à minimiser $\mathbb{KL}(q_{z_k}(z_k) \| \tilde{p}(x, z))$, donc la densité optimale $q^*_{z_k}$ vérifie

$$\log q^*_{z_k}(z_k) = \mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z) + \mathcal{C}_n,$$

où $\mathcal{C}_n$ est une constante de normalisation. Autrement dit :

$$q^*_{z_k}(z_k) \propto exp\left(\mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z) \right). \tag{4.6}$$

On peut réécrire cette densité $q^*_{z_k}$ en version conditionnelle. Pour cela, on remarque que :

$$\mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z) = \mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(z_k | x, z_{-k}) + \overbrace{\mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(x, z_{-k})}^{\text{ne dépend pas de } z_k}. \tag{4.7}$$

Ainsi,

$$q^*_{z_k}(z_k) \propto exp\left(\mathbb{E}_{q_{z_{-k}}(z_{-k})} \log p(z_k | z_{-k}, x) \right). \tag{4.8}$$

On peut utiliser indifféremment l'une ou l'autre des équations (4.6) et (4.8) pour mettre à jour q_{z_k}. Ces équations permettent de déduire le facteur $q^*_{z_k}$ de la densité variationnelle à partir de $q_{z_{-k}}$ et du modèle génératif. L'algorithme 1 montre comment on implémente cette approche en pratique.

Algorithm 1 Inférence variationnelle

Initialiser aléatoirement $(q_{z_m}^{(0)})_{m=1}^{M}$
$t = 1$
while $t \leq t_{\max}$ ou autre critère d'arrêt **do**
  for $i \in [|1, N|]$ **do**

$$q_{z_m}^{(t)}(z_m) \propto exp\left(\mathbb{E}_{q_{-m}(z_{-m})} \log p(x_i, z)\right)$$

où

$$q_{-m}(z_{-m}) = \prod_{j<m} q_{z_j}^{(t)}(z_j) \prod_{j>m} q_{z_j}^{(t-1)}(z_j)$$

  end for
  $t+ = 1$
end while

4.1.8 Inférence variationnelle stochastique.

Il existe des situations où cette approche "naïve" est trop couteuse. C'est par exemple le cas lorsque le modèle génératif contient des variables locales et des variables globales, et que N est très grand.

Par exemple, si x est une variable aléatoire réelle distribuée selon une mixture de deux lois normales de moyennes et variances (μ_1, σ_1) et (μ_2, σ_2), alors μ_1,σ_1,μ_2 et σ_2 sont des variables latentes globales car elle conditionnent l'ensemble des observations. Par contre, chaque réalisation x_i de x est une réalisation de l'une de ces deux lois. Si on note $c_i \in \{1, 2\}$ la loi dont est issue x_i, alors c_i est une variable latente locale car chacune de ses réalisations ne conditionne qu'une seule observation.

Considérons un modèle génératif avec une variable latente globale β et N variables latentes locales $z = z_1^N$ tel que chaque observation ne dépend que de la variable locale correspondante et toutes les variables du modèle dépendent de la variable globale, on a

$$\forall n \in [|1, N|], p(x_n, z_n, \beta) = p(\beta) \prod_{n=1}^{N} p(z_n|\beta) p(x_n|z_n, \beta).$$

Soit $\mathcal{Q}$ une famille variationnelle *mean-field*, l'hypothèse *mean-field* pour une densité variationnelle $q \in \mathcal{Q}$ s'écrit :

$$q(\beta, z_1^n) = q_\beta(\beta) \prod_{i=1}^{N} q_{z_i}(z_i).$$

L'algorithme 2 décrit la VI implémentée pour ce problème.

On constate que chaque nouvelle mise à jour de β nécessite un passage sur l'ensemble des observations. Quand N est grand, le temps nécessaire à l'optimisation de q_β est excessivement grand. On contourne ce problème en adaptant une approche d'optimisation stochastique à ce problème d'inférence variationnelle. Pour cela, on a besoin d'estimer l'espérance de l'équation (4.9) à partir d'un mini-batchs d'observations pour mettre à jour la densité q_β plus régulièrement. Ces approches d'inférence variationnelle stochastique sont efficaces essentiellement dans les contextes où l'estimation de cette espérance n'est pas coûteuse, elles sont donc principalement utilisées lorsqu'on en a une forme close, c'est par exemple le cas lorsque la famille variationnelle est dans la famille exponentielle.

4.1.9 Conclusions et limitations.

Dans le cas où il est possible de déterminer une forme close des espérances d'intérêt, la seule limitation réside dans la mise à jour des densités des variables latentes globales, auquel cas l'approche stochastique est justifiée. Cependant, la plupart du temps, les espérances à calculer n'ont pas de forme close et il est alors

Algorithm 2 Inférence variationnelle pour un modèle à variables latentes locales et globales

Initialiser aléatoirement les densités $q_\beta^{(0)}$ et $(q_{z_i}^{(0)})_{i=1}^N$
$t = 1$
while $t \leq t_{\max}$ ou autre critère d'arrêt **do**
  for $i \in [|1, N|]$ **do**

$$q_{z_i}^{(t)}(z_i) \propto exp\left(\mathbb{E}_{q_{-i}(\beta, z_{j\neq i})} log\, p(x_i, \beta, z_i)\right)$$

où

$$q_{-i}(\beta, z_{j\neq i}) = \prod_{j<i} q_{z_j}^{(t)}(z_j) \prod_{j>i} q_{z_j}^{(t-1)}(z_j) q_\beta^{(t-1)}(\beta)$$

  end for

$$q_\beta^{(t)}(\beta) \propto exp\left(\mathbb{E}_{q_z(z_1^N)} \sum_{i=1}^N log\, p(x_i, \beta, z_i)\right) \tag{4.9}$$

où

$$q_z(z_1^N) = \prod_{i=1}^N q_{z_i}^{(t)}(z_i)$$

  $t+ = 1$
end while

nécessaire de les estimer. Dans ces situations, on peut utiliser des méthodes MCMC mais cela devient rapidement très couteux lorsque les observations sont nombreuses, et même les approches stochastiques ne parviennent pas à ramener le temps de calcul à quelque chose d'acceptable.

4.2 Les auto-encodeurs variationnels

4.2.1 Introduction.

Les VAE apportent une solution aux limitations de la VI et de la VI stochastique dans la situation où l'on souhaite prédire uniquement des variables latentes locales, c'est à dire les variables latentes réalisées une fois pour chaque observation. Dans le contexte du scRNAseq, on utilise les VAE comme des outils de réduction de dimension probabiliste : x est une observation en grande dimension, et z est la représentation en dimension réduite de x dont on cherche à inférer la distribution. z est donc une variable latente locale car chaque observation x possède une représentation latente caractéristique z. Les VAE furent introduits simultanément en 2014 par deux équipes de recherche[Kingma and Welling, 2013] [Rezende et al., 2014] et définissent une famille variationnelle *mean-field* et un modèle génératif à partir de réseaux de neurones.

4.2.2 Cadre.

Le cadre d'application des VAE décrit dans la publication de Kingma [Kingma and Welling, 2013] est celui d'un problème d'inférence variationnelle où l'on veut inférer la distribution des variables latentes locales, les valeurs des éventuelles variables latentes globales sont estimées par *maximum a posteriori* ou par *maximum de vraisemblance*. On garde la notation $(x_n)_{n=1}^N$ pour les observations et $(z_n)_{n=1}^N$ pour les variables latentes locales. Dans le cadre de la réduction de dimension probabiliste, une observation $x \in \mathbb{R}^G$ a pour représentation latente $z \in \mathbb{R}^D$ avec $D < G$. Le lien supposé entre une observation x et sa représentation z est définit par le choix des lois du modèle génératif, le *prior* $p(z)$ et la densité conditionnelle $p(x|z)$. Notre objectif est d'estimer la distribution $p(z|x)$ de la représentation z à partir de l'observation x.

4.2.3 Famille variationnelle définie par un réseau de neurones.

La famille variationnelle du VAE est dans la même famille de distributions que le *prior* et elle est paramétrique. Chaque densité de cette famille variationnelle est caractérisée par un paramètre λ, c'est donc ce paramètre qu'on voudrait déterminer à partir de l'observation x. On note donc

$$\mathcal{Q}_\Lambda = \{q_{\lambda(x)}(z) = q(z|\lambda(x)) \ \ |\lambda(.) \in \Lambda \ : x \in \mathbb{R}^G \mapsto \lambda(x) \in \mathbb{R}^D\}.$$

Trouver $q^* \in \mathcal{Q}_\Lambda$ équivaut à trouver la fonction déterministe $\lambda^*(.) \in \Lambda$ tel que pour toute observation x, le paramètre variationnel $\lambda^*(x)$ caractérise la densité $q(z|\lambda^*(x))$ la plus pertinente pour la représentation z correspondante. Cette famille variationnelle paramétrique vérifie l'hypothèse *mean-field*, pour une observation x, on a :

$$\forall \lambda(.) \in \Lambda, q(z|\lambda(x)) = \prod_{d=1}^{D} q_d(z_d|\lambda_d(x)).$$

Λ est défini comme l'ensemble des fonctions d'un réseau de neurones à l'architecture fixée. Pour simplifier, on considère ici que l'architecture qui définit Λ est un réseau à une couche cachée de taille 3 mais on peut généraliser ce qui suit à d'autres architectures de réseau de neurones. Dans la suite, on identifie Λ à son architecture. Soient

- $W_G = (w_{G,g,1}, w_{G,g,2}, w_{G,g,3})_{g=1}^{G} \in \mathbb{R}^{G\times 3}$ la matrice des poids des observations.
- $b_G = b_{G,1}, b_{G,2}, b_{G,3} \in \mathbb{R}^3$ le vecteur de biais des observations.
- $W_\lambda = (w_{\lambda,1,d}, w_{\lambda,2,d}, w_{\lambda,3,d})_{d=1}^{D} \in \mathbb{R}^{3\times D}$ la matrice des poids des neurones de la couche cachée.
- $b_\lambda = (b_{\lambda,d})_{d=1}^{D} \in \mathbb{R}^D$ le vecteur des biais de la couche cachée.
- $\theta_\Lambda = (W_G, b_G, W_\lambda, b_\lambda) \in \Theta_\Lambda = \{\mathbb{R}^{G\times 3} \times \mathbb{R}^3 \times \mathbb{R}^{3\times D} \times \mathbb{R}^D\}$ l'espace des paramètres de Λ.
- $a : \mathbb{R} \to \mathbb{R}$ une fonction bornée, continue et non décroissante appelée fonction d'activation.

L'ensemble des valeurs possibles du paramètre θ_Λ caractérise l'ensemble des fonctions de Λ. Pour tout $\lambda(.) \in \mathcal{Q}_\Lambda$, il existe $\theta_\Lambda \in \Theta_\Lambda$ tel que :

$$\lambda(x) = \lambda_{\theta_\Lambda}(x) = \left(a \left(\sum_{c=1}^{3} w_{\lambda,c,d}\, a(\sum_{g=1}^{G} w_{G,g,c}\, x_g + b_{G,c}) + b_{\lambda,d} \right) \right)_{d=1}^{D}.$$

On peut réécrire l'équation du paramètre variationnel $\lambda(x)$ de l'observation x de façon à mieux comprendre où intervient la couche cachée :

$$\forall c \in [|1,3|], f_{G,c}(x) = a(\sum_{g=1}^{G} w_{G,g,c}\, x_g + b_{G,c}),$$

$$\lambda(x) = (\lambda_d(x))_{d=1}^{D} = \left(a(\sum_{c=1}^{3} w_{\lambda,c,d}\, f_{G,c}(x) + b_{\lambda,d}) \right)_{d=1}^{D}.$$

Dans le contexte du VAE, la fonction $\lambda_{\theta_\Lambda}(.)$ est appelé l'encodeur, car elle relie l'observation x à la distribution de sa représentation z. L'encodeur $\lambda_{\theta_\Lambda}(.)$ caractérise la densité $q(z|\lambda_{\theta_\Lambda}(x))$, on identifie la densité à son paramètre variationnel. La figure 4.1 est une représentation de cette famille variationnelle définie par un réseau de neurones. Pour simplifier, on écrira :

$$\lambda_{\theta_\Lambda}(.) = \lambda(.).$$

Théorème d'approximation universelle. On choisit un réseau de neurones pour définir la famille variationnelle car le théorème d'approximation universelle [Hornik, 1991] dit que pour toute fonction f continue définie sur un compact de $\mathbb{R}^D$, il existe une suite $(\theta_\Lambda^{(n)})_{n\geq 1} \in \Theta_\Lambda^{\mathbb{N}}$ telle que $(\lambda_{\theta_\Lambda^{(n)}}(.))_{n\geq 1}$ converge simplement vers f. On a donc une famille variationnelle plastique, ce qui augmente nos chances d'obtenir une bonne approximation de $p(z|x)$.

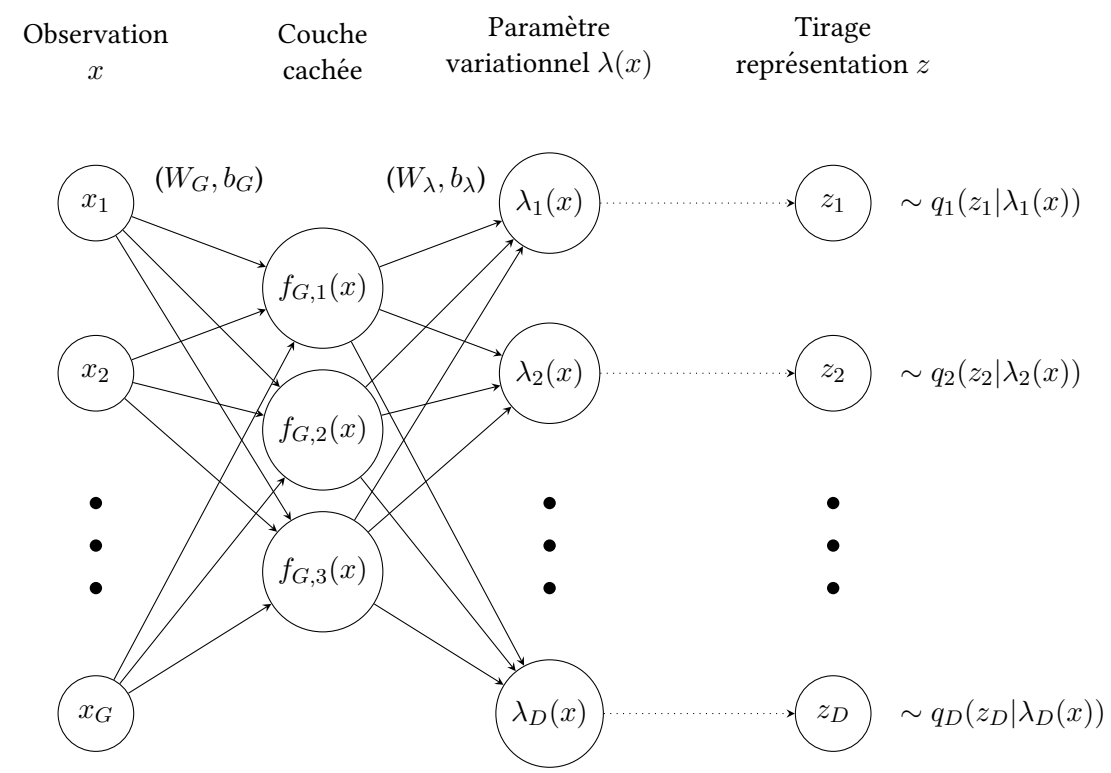

FIGURE 4.1 – Réseau de neurones associé à la famille variationnelle

4.2.4 Un réseau de neurones pour le modèle génératif.

On voit $z \in \mathbb{R}^D$ comme la représentation en dimension réduite de l'observation $x \in \mathbb{R}^G$. Conditionnellement à z, chaque composante x_g de x est tirée selon la densité conditionnelle $p_g(x_g|z)$. Les paramètres spécifiques de cette distribution sont caractérisés par z, mais la contribution de z est indirecte, car z n'est pas lui-même un paramètre. Cette contribution indirecte est moins contraignante qu'une contribution directe et permet de ne pas être trop dépendant du choix arbitraitre du *prior* (cf. 4.1.2). Pour expliciter cette contribution indirecte, on définit $\varphi(.) \in \Phi : \mathbb{R}^D \to \mathbb{R}^G$ et on réécrit la densité conditionnelle ainsi :

$$p(x|z) = p(x|\varphi(z)) = (p_g(x_g|\varphi_g(z))_{g=1}^G = \left(p_{\varphi_g(z)}(x_g)\right)_{g=1}^G . \tag{4.10}$$

Dans le contexte du VAE, on identifie l'espace fonctionnel Φ à l'architecture d'un réseau de neurones symétrique à Λ, soit :

- $W_D = (w_{D,d,1}, w_{D,d,2}, w_{D,d,3})_{d=1}^D \in \mathbb{R}^{D\times 3}$ la matrice des poids des variables latentes.
- $b_D = b_1, b_2, b_3 \in \mathbb{R}^3$ le vecteur de biais des variables latentes.
- $W_\varphi = (w_{\varphi,1,g}, w_{\varphi,2,g}, w_{\varphi,3,g})_{g=1}^G \in \mathbb{R}^{3\times G}$ la matrice des poids de la couche cachée.
- $b_\varphi = (b_g)_{g=1}^G \in \mathbb{R}^G$ le vecteur des biais de la couche cachée.
- $\theta_\Phi = (W_D, b_D, W_\varphi, b_\varphi) \in \Theta_\Phi = \{\mathbb{R}^{D\times 3} \times \mathbb{R}^3 \times \mathbb{R}^{3\times G} \times \mathbb{R}^G\}$ l'espace des paramètres de Φ.
- $a : \mathbb{R} \to \mathbb{R}$ une fonction d'activation.

Comme pour l'encodeur, on a le décodeur $\varphi_{\theta_\Phi}(.)$ du VAE définit par :

$$\forall c \in [|1,3|], f_{D,c}(x) = a(\sum_{d=1}^{D} w_{D,d,c}\, z_d + b_{D,c}),$$

$$\varphi(x) = \varphi_{\theta_\Phi}(z) = (\varphi(x))_{g=1}^G = \left(a(\sum_{c=1}^{3} w_{\varphi,c,g}\, f_{D,c}(z) + b_{\varphi,g}) \right)_{g=1}^G .$$

Le décodeur relie la représentation z à la distribution conditionnelle de l'observation $p(x|\varphi_{\theta_\Phi}(z))$. La figure 4.2 montre l'architecture du décodeur et la figure 4.3 résume l'architecture globale du VAE.

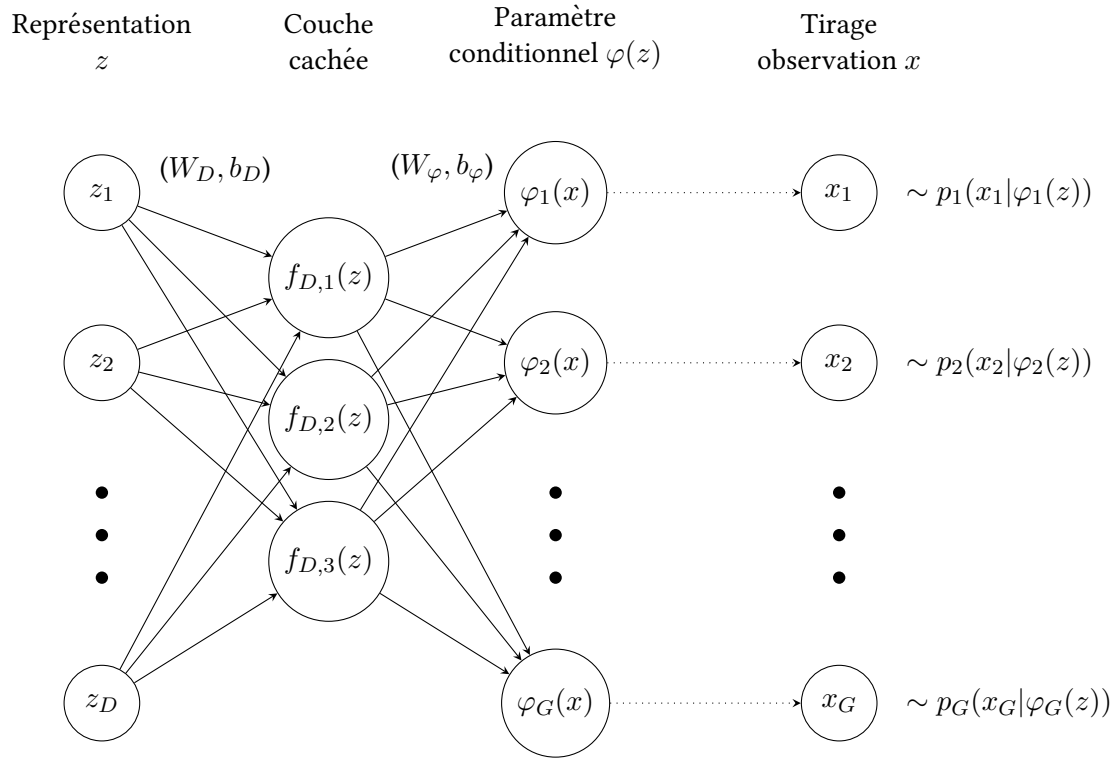

FIGURE 4.2 – Réseau de neurones associé à la densité conditionnelle

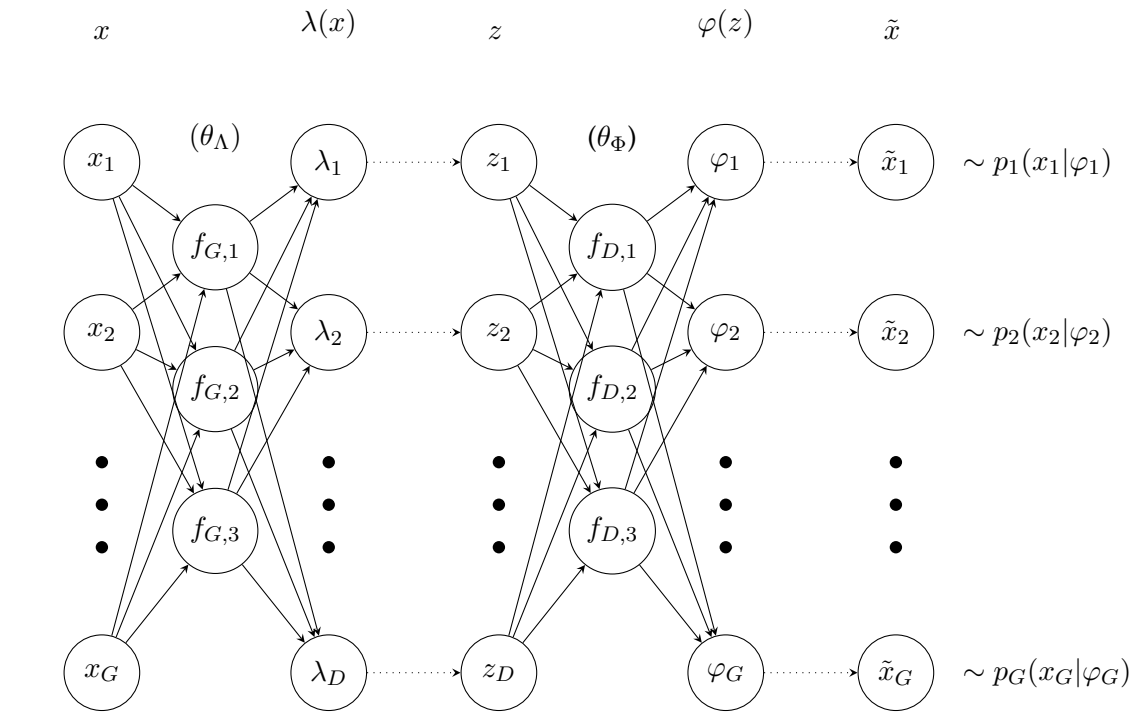

FIGURE 4.3 – Réseau de neurones du VAE complet

4.2.5 Maximiser l'ELBO.

On peut réécrire l'ELBO définit en (4.2) comme une fonction de x, de l'encodeur $\lambda(.)$ et du décodeur $\varphi(.)$. Pour cela, on remarque que toute densité variationnelle $q(z) \in \mathcal{Q}_\Lambda$ est caractérisée par l'encodeur $\lambda(.)$ et l'observation correspondante x, donc $q(z) = q(z|\lambda(x))$. On a :

$$\mathcal{ELBO}(x, \lambda(.), \varphi(.)) = \mathbb{E}_{q(z|\lambda(x))} log\ p(x|\varphi(z)) - \mathbb{E}_{q(z|\lambda(x))} log\ \frac{q(z|\lambda(x))}{p(z)}. \tag{4.11}$$

Lors de l'apprentissage, on met à jour conjointement les densités $(q(z|\lambda(x)), p(x|\varphi(z))$ identifiées par $(\lambda(.), \varphi(.))$ selon une approche cyclique similaire à celle décrite précédemment. On souhaite donc déterminer comment ajuster les paramètres $(\theta_\Lambda, \theta_\Phi)$ pour maximiser l'ELBO. Pour cela, on doit expliciter l'ELBO par rapport au couple $(\lambda(.), \varphi(.))$, puis dériver cette quantité par rapport à chacun des paramètres. La dérivation d'une fonction définie par un réseau de neurones nécessite l'implémentation d'un algorithme de *back-propagation*, qui consiste à réécrire la fonction comme une composée de fonctions et à appliquer une dérivation en chaine. Par exemple, si l'on souhaite dériver $\varphi(.)$, on commence par écrire

$$\begin{aligned}
\varphi_{\theta_\Phi}(z) = f_\varphi(z) &= (a(h_{\varphi,g}(z))_{g=1}^G. \\
\forall g \in [|1, G|], h_{\varphi,g}(z) &= \sum_{c=1}^{3} w_{\varphi,c,g}\, f_{D,c}(z) + b_{\varphi,g}. \\
f_D(z) &= (f_{D,c}(z))_{c=1}^3 = (a(h_{D,c}(z))_{c=1}^3. \\
\forall c \in [|1, 3|], h_{D,c}(z) &= \sum_{d=1}^{D} w_{D,d,c}\, z_d + b_{D,c}.
\end{aligned}$$

Puis la *back-propagation* nous donne chacune des dérivées d'intérêt :

$$\begin{aligned}
\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial w_{\varphi,c}} &= \left(\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial h_{\varphi,g}(z))} \frac{\partial h_{\varphi,g}(z))}{\partial w_{\varphi,c,g}} \right)_{c\in[|1,3|],g\in[|1,G|]} \\
&= \left(a'(h_{\varphi,g}(z)) f_{D,c}(z) \right)_{c\in[|1,3|],g\in[|1,G|]}. \\
\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial b_{\varphi}} &= \left(\frac{\partial a(h_{\varphi,g}(z))}{\partial b_{\varphi,g}} \right)_{g\in[|1,G|]} \\
&= \left(a'(h_{\varphi,g}(z)) \right)_{g\in[|1,G|]}. \\
\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial w_{D}} &= \left(\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial h_{\varphi,g}(z))} \frac{\partial h_{\varphi,g}(z))}{\partial f_{D,c}(z)} \frac{\partial f_{D,c}(z)}{\partial h_{D,c}(z)} \frac{\partial h_{D,c}(z)}{\partial w_{D,d,c}} \right)_{d\in[|1,D|],c\in[|1,3|],g\in[|1,G|]} \\
&= \left(a'(h_{\varphi,g}(z)) w_{\varphi,c,g} a'(h_{D,c}(z)) z_d \right)_{d\in[|1,D|],c\in[|1,3|],g\in[|1,G|]}. \\
\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial b_{D}} &= \left(\frac{\partial \varphi_{\theta_\Phi}(z)}{\partial h_{\varphi,g}(z))} \frac{\partial h_{\varphi,g}(z))}{\partial f_{D,c}(z)} \frac{\partial f_{D,c}(z)}{\partial h_{D,c}(z)} \frac{\partial h_{D,c}(z)}{\partial b_{D,d,c}} \right)_{d\in[|1,D|],c\in[|1,3|],g\in[|1,G|]} \\
&= \left(a'(h_{\varphi,g}(z)) w_{\varphi,c,g} a'(h_{D,c}(z)) \right)_{d\in[|1,D|],c\in[|1,3|],g\in[|1,G|]}.
\end{aligned}$$

De même avec $\lambda(.)$.

Terme de reconstruction

L'ELBO (4.11) est la somme de deux termes. On définit la perte de reconstruction de l'ELBO par

$$\mathcal{R}(x, \lambda(.), \varphi(.)) = \mathbb{E}_{q(z|\lambda(x))} log\ p(x|\varphi(z)).$$

Ce terme est interprété comme une perte car il s'agit de la log-vraisemblance d'une observation x si elle avait été générée à partir de la variable z tirée selon la densité variationnelle $q(z|\lambda(x))$. Il est d'autant plus grand que la densité variationnelle est proche de la densité postérieure. En d'autres termes, la perte de reconstruction quantifie la pertinence de la densité inférée sur z conditionnellement à x. La méthode naturelle pour estimer la valeur de ce terme étant donnée une observation x est de constituer un échantillon $(\tilde{z}_j)_{j=1}^J$ tiré aléatoirement selon $q(z|\lambda(x))$ et d'appliquer une méthode de Monte-Carlo, en effet, on a :

$$\hat{\mathcal{R}}(x, \lambda(.), \varphi(.)) = \sum_{j=1}^{J} log\ p(x|\varphi(\tilde{z}_j)).$$

On peut ensuite dériver cette quantité pour mettre à jour les paramètres du VAE $(\theta_\Lambda, \theta_\Phi)$ dans l'optique de la maximiser. Le problème est qu'avec cette méthode, la réalisation de $\tilde{z}$ cache le lien avec la densité variationnelle et donc avec $\lambda(.)$. On ne peut donc pas mettre à jour θ_Λ. L'astuce de reparamétrisation permet de faire apparaître la fonction $\lambda(.)$ dans cette équation pour mettre à jour tous les paramètres. Pour cela, on choisit une fonction g différentiable et on tire aléatoirement un échantillon $(\tilde{\varepsilon}_j)_{j=1}^J$ selon une loi $P(\varepsilon)$ indépendante de x et de $\lambda(.)$, tels que :

$$\tilde{z}|x = g(\lambda(x), \tilde{\varepsilon})|x \sim q(z|\lambda(x)).$$

On a alors une approximation de la perte de reconstruction $\mathcal{R}(x, \lambda(.), \varphi(.))$ par

$$\hat{\mathcal{R}}(x, \lambda(.), \varphi(.)) = \sum_{j=1}^{J} log\ p(x|\varphi(g(\lambda(x), \tilde{\varepsilon}_j))).$$

De plus, on peut dériver cet estimateur pour mettre à jour $(\theta_\Lambda, \theta_\Phi)$ en utilisant la *back-propagation*, puisqu'on connait la forme de la densité conditionnelle définie par le modèle génératif, et g est différentiable. Cette reparamétrisation est presque toujours possible, le cas le plus simple est celui où $q(z|\lambda(x))$ est dans une famille de distributions de type *moyenne-variance*, dans ce cas, on a $g(\lambda(x), \varepsilon) = \text{moyenne} + \text{variance} \times \varepsilon$, mais on peut aussi calculer ou approximer la fonction de répartition inverse et choisir $P(\varepsilon) = \mathcal{U}(0, 1)$.

Terme de dissimilarité

Le terme de divergence $\mathbb{KL}$ de l'ELBO ne dépend que de x et de $\lambda(.)$:

$$\mathbb{KL}(q(z|\lambda(x))\|p(z)) = \mathbb{E}_{q(z|\lambda(x))}\left[log\ \frac{q(z|\lambda(x))}{p(z)}\right].$$

Pour maximiser l'ELBO, on veut minimiser ce terme. Cela contraint la distribution variationnelle à être proche du *prior*, et l'espace image de $\lambda(.)$ à ne pas recouvrir tout l'espace. Cette contrainte incite à associer des distributions variationnelles proches aux observations similaires entre elles. Dans le cas où on suppose que la densité variationnelle et le *prior* sont dans la même famille, il y a des choix de *prior* pour lesquels ce terme est explicite, c'est le cas par exemple pour un *prior* Gaussien. Il suffit alors de dériver cette quantité explicite par rapport aux paramètres de l'encodeur θ_Λ. Dans les cas où cette dissimilarité n'est pas explicite, on utilise une méthode de Monte-Carlo pour l'estimer, et on met à jour les paramètres à l'aide de l'astuce de reparamétrisation et la *back-propagation*. On pourrait croire qu'on se retrouve dans la même situation que la VI stochastique, sauf qu'on a déplacé le problème d'estimation : ici, on estime une divergence $\mathbb{KL}$ entre deux densités de même famille, ce qui est en général plus simple que d'estimer la log-vraisemblance de la densité jointe ou conditionnelle.

4.3 scVI

4.3.1 Introduction.

L'outil d'analyse de données scRNAseq *single cell variational inference* (scVI) a été développé par le doctorant Romain Lopez en 2018 dans le laboratoire de Nir Yosef à Berkeley et sous la supervision de Michael Jordan [Lopez et al., 2018]. Cet outil basé sur les VAE propose un modèle bayésien hiérarchique pour modéliser

plusieurs source de variances. scVI permet de nombreuses analyses statistiques telles que la normalisation des données, la réduction de dimension, la correction de l'effet batch, l'analyse de l'expression différentielle, l'imputation de données manquantes, l'inférence de la dynamique d'expression cellulaire, la clusterisation et la visualisation des données. La parallelisation possible des calculs permet à l'outil d'étudier plusieurs millions de cellules en un temps raisonnable sur GPU. L'outil scVI se démarque des autres approches par sa modélisation spécifique des biais propres à la technologie scRNAseq. De plus, c'est un projet entièrement Open Source basé sur le package Torch de Python, cela rend envisageable de l'adapter à des contextes très particuliers, comme c'est le cas dans l'étude du Myélome Multiple.

4.3.2 Choix du modèle génératif.

Variables observées et variables latentes.

On note $x_n \in \mathbb{N}^G$ le vecteur de comptage des gènes transcrits observés dans la cellule n parmi les G gènes de référence. Si l'échantillon est issu de B échantillons quantifiés lors de différentes expérimentations, $s_n \in \{0,1\}^B$, l'effet batch, est un vecteur de zéros avec un un à l'indice de l'expérimentation dont est issue la cellule n. Le couple (x, s) constitue la seule information que nous ayons sur chaque cellule. On suppose qu'il existe deux variables latentes locales pour la cellule n, son facteur de taille $\ell_n \in \mathbb{R}$ et sa représentation en dimension réduite $z_n \in \mathbb{R}^D$. ℓ est aussi appelé taille de la librairie et est supposé modéliser toute la variance du comptage liée à la manipulation expérimentale scRNAseq. Dans le cadre de la VI, on suppose que l'observation x est conditionnée par les variables (s, z, ℓ).

Choix du *prior*.

Pour que l'assignation des sources de variances à l'issue de l'apprentissage ait du sens, il convient de choisir des *prior* adéquats pour les variables z et ℓ. La variable s est observée, on peut donc la considérer comme une quantité déterministe. La quantité d'ARN messagers capturés en scRNAseq peut varier de façon logarithmique entre deux cellules similaires, on fait l'hypothèse forte que les écarts logarithmiques entre le comptage des cellules est principalement dû à des effets techniques, et cette source de variance est assignée à la variable ℓ, modélisée par un *prior* log-normal de paramètres (ℓ_μ, ℓ_σ) déterminés par des estimateurs classiques sur l'ensemble de la matrice de comptage. On a

$$\ell \sim \log \mathcal{N}(\ell_\mu, \ell_\sigma^2). \tag{4.12}$$

La seconde variable latente z est vue comme une représentation de x en dimension réduite construite sur la base de la variabilité biologique une fois que la variabilité technologique a été prise en compte. Comme on l'a vu dans les sections précédentes, le choix de son *prior* a peu d'importance et on choisit donc un *prior* simple, une distribution Gaussienne multivariée centrée et réduite.

$$z \sim \mathcal{N}(O, I_D). \tag{4.13}$$

On suppose l'indépendance entre ℓ et z, le *prior* de notre problème est la densité $p(z, \ell) = p(z)p(\ell)$.

Choix de la densité conditionnelle.

La variable aléatoire de comptage x de scVI est modélisée selon le consensus scientifique actuel concernant les données scRNAseq : x est un comptage sur-dispersé avec un excès de zéros, ce qui justifie l'utilisation d'une loi *zero inflated negative binomiale* (ZINB) pour la densité conditionnelle. C'est à dire une loi *negative binomiale* (NB) ou distribution de Pólya à laquelle on a ajouté une composante *zero inflated* (ZI). La loi NB est une loi de comptage sur-dispersée. La densité d'une variable y distribuée selon une loi NB de paramètres r et p est définie par :

$$p(y|r,p) = \sum_{k=0}^{+\infty} \delta_k(y) \frac{\Gamma(r+k)}{\Gamma(r)\Gamma(k+1)} p^r (1-p)^k.$$

Si on ajoute h une composante ZI, on obtient la densité de l'observation x distribuée selon une loi ZINB de paramètres (r, p, h) :

$$p(x|r,p,h) = \ h\delta_0(x) + (1-h)\sum_{k=0}^{+\infty}\delta_k(x))\frac{\Gamma(r+k)}{\Gamma(r)\Gamma(k+1)}p^r(1-p)^k.$$

On remarque que la variable y n'a pas de sens interprétable, il sert seulement à construire x, de même pour les paramètres (r,p), leur signification n'est pas évidente. Seul le paramètre h est inteprétable. Si x est le comptage d'un gène dans une cellule, alors h est un paramètre local correspondant à la probabilité de présence de ce gène dans la cellule. Afin de donner plus de sens aux paramètres de la densité conditionnelle, on considère la loi NB comme un mélange Gamma-Poisson. En effet, si y est une variable aléatoire distribuée selon une loi de Poisson de paramètre $(\ell \times w)$, et si w est une variable aléatoire distribuée selon une loi Γ de paramètres (ρ,θ), alors on a :

$$y|w,\ell \sim \mathcal{P}(\ell \times w).$$

mais aussi :

$$y|\rho,\theta,\ell \sim \mathcal{NB}(\rho, \frac{\theta}{\ell+\theta}).$$

En ajoutant h la composante ZI, on obtient la loi de x :

$$x|\rho,\theta,\ell,h \sim \mathcal{ZINB}(\rho, \frac{\theta}{\ell+\theta}, h).$$

De cette façon, x dépend des paramètres (ρ,θ,ℓ,h). On peut alors donner du sens à chacun des paramètres. On considère que x est le comptage d'un gène dans une cellule. Alors ρ est une variable locale correspondant à l'expression moyenne du gène dans la cellule, et θ est un paramètre global de dispersion spécifique au gène. On retrouve le paramètre de taille de la librairie de la cellule ℓ et celui de la probabilité de présence du gène dans la cellule h. On retrouve la variable latente ℓ dans la densité conditionnelle, sa contribution est directe car il s'agit de l'un des paramètres. Par contre avec ce choix de densité conditionnelle, la contribution des variables latentes (s,z) est indirecte, elles n'apparaissent pas directement dans les paramètres. On utilise le formalisme des VAE pour relier (z,s) aux variables locales (ρ,h) de la densité conditionnelle, comme dans l'équation (4.10), et la variable globale θ est approximée par *maximum a posteriori*. On définit deux architectures de réseaux de neurones Φ_ρ et Φ_h de paramètres $(\theta_\rho,\theta_h) \in \Theta_\rho \times \Theta_h$. Et on redéfinit les paramètres ρ et h comme des fonctions de la représentation z et du batch s, $\rho(.)$ et $h(.)$ sont les deux encodeurs du VAE de scVI. On a donc :

$$\Phi_\rho = \{\rho(.,.) = \rho_{\theta_\rho}(.,.) \,|\, \forall(z,s), \rho(z,s) \in \mathbb{R}^G, \sum_{g=1}^{G}\rho_g(z,s) = 1\},$$
$$\Phi_h = \{h(.,.) = h_{\theta_h}(.,.) \,|\, \forall(z,s), h(z,s) \in [0,1]^G\}.$$

En pratique, B couples de réseaux sont implémentés et entrainés en parallèle, chaque observation x participe à la mise à jour des paramètres du couple correspondant à son batch. En pratique, une partie des paramètres de chaque couple est commune.

Résumé du modèle génératif.

En résumé, le modèle génératif est défini ainsi :
- $\ell_\sim \log\mathcal{N}(\ell_\mu, \ell_\sigma^2)$: taille de la librairie.
- $z_\sim\mathcal{N}(O, I_D)$: représentation en dimension réduite.
- $s_\in\{0,1\}^B$: effet batch.
- $\rho(z,s) \in \mathbb{R}^G$: expression génique normalisée par la taille de la librairie.
- $\theta \in \mathbb{R}^G$: dispersion génique approximée par *maximum a posteriori*.
- $h(z,s) \in \mathbb{R}^G$: effet dropout.
- $x|\rho(z,s),\theta,\ell,h(z,s) \sim \mathcal{ZINB}(\rho(z,s), \frac{\theta}{\ell+\theta}, h(z,s))$: observation.

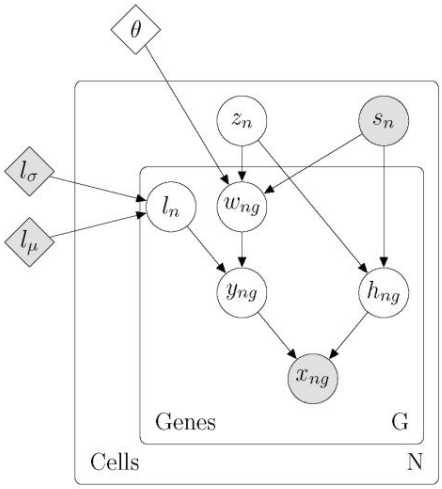

FIGURE 4.4 – Graphe acyclique dirigé de la modélisation hierarchique de scVI. Les noeuds grisés représentent les variables observés et les noeuds blancs représentent les variables latentes. Les carrés grisés représentent les constantes fixées a priori et le carré blanc représente une variable latente globale fixée *Maximum a posteriori* [Lopez et al., 2018]
.

On peut voir le graphe acyclique dirigé de scVI sur la figure 4.4, et la densité jointe s'écrit :

$$p(x, \ell, z) = p(\ell, z)p(x|\rho(z, s), \theta, \ell, h(z, s)).$$

4.3.3 Choix de la famille variationnelle.

Pour estimer la densité postérieure $p(\ell, z|x, s)$, on fait l'hypothèse que les variables latentes sont indépendante de l'effet batch, donc la vraie densité postérieure qu'on estime est $p(\ell, z|x)$. Soit $\mathcal{Q}_\Lambda$ une famille variationnelle paramétrique définie par une architecture de réseau de neurones pour chaque variable latente parmi (ℓ, z) : $\Lambda = (\Lambda_\ell, \Lambda_z)$, de paramètres $(\theta_\ell, \theta_z) \in \Theta_\ell \times \Theta_z$. La famille variationnelle vérifie l'hypothèse *mean-field*, pour toute dansité variationnelle $q(\ell, z|x)$ dans $\mathcal{Q}_\Lambda$, on a :

$$q(\ell, z|x) = q(\ell|x)q(z|x) = q(\ell|x) \prod_{d=1}^{D} q_d(z_d|x).$$

De plus, $\mathcal{Q}_\Lambda$ est choisie dans la même famille que le *prior*. On note $\lambda_\ell = (\lambda_{\ell,\mu}, \lambda_{\ell,\sigma}) \in \mathbb{R}^2$ la moyenne et la variance de la densité log-normale $q(\ell|x)$, et $\lambda_z = (\lambda_{z,\mu}, \lambda_{z,\Sigma}) \in \mathbb{R}^D \times \mathbb{R}^{D\times D}$ le vecteur moyen et la matrice de variance-covariance de la densité gaussienne multivariée $q(z|x)$. L'hypothèse *mean-field* impose que $\lambda_{z,\Sigma}$ est diagonale, on note $(\lambda_{z,\mu,d})_{d=1}^D$ les composantes du vecteur moyen et $(\lambda_{z,\sigma,d})_{d=1}^D$ les termes diagonaux. L'ensemble des densités variationnelles de $\mathcal{Q}_\Lambda$ est caractérisé par l'ensemble des fonctions $(\lambda_\ell(.), \lambda_z(.)) \in (\Lambda_\ell, \Lambda_z)$, on note :

$$\mathcal{Q}_\Lambda = \{q(\ell|\lambda_\ell(x))q(z|\lambda_z(x))|(\lambda_\ell(.), \lambda_z(.)) \in (\Lambda_\ell, \Lambda_z)\}.$$

4.3.4 Maximiser l'ELBO.

Le problème de VI dans scVI est maintenant entièrement défini selon le cadre d'approche des VAE. Pour les VAE, on a écrit l'ELBO en fonction de l'observation x, l'encodeur $\lambda(.)$ de paramètre $\theta_\Lambda \in \Theta_\Lambda$ et le décodeur $\varphi(.)$ de paramètre $\theta_\Phi \in \Theta_\Phi$. Dans scVI, nous avons deux encodeurs $\lambda_\ell(.)$ et $\lambda_z(.)$ de paramètres $(\theta_\ell, \theta_z) \in$

$\Theta_\ell \times \Theta_z$. Et nous avons deux décodeurs $\rho(.,.)$ et $h(.,.)$, de paramètres $(\theta_\rho, \theta_h) \in \Theta_\rho \times \Theta_h$. L'ELBO de scVI s'écrit donc

$$\begin{aligned}\mathcal{ELBO}(x, \rho(.,.), h(.,.), \lambda_\ell(.), \lambda_z(.)) =& \mathbb{E}_{q(\ell,z|\lambda_\ell(x),\lambda_z(x))} log\ p(x|\rho(z,s), h(z,s), \ell) \\ &- \mathbb{KL}(q(\ell, z|\lambda_\ell(x), \lambda_z(x))\|p(\ell, z))\end{aligned}. \tag{4.14}$$

Et on veut le maximiser par rapport aux paramètres $(\theta_\rho, \theta_h, \theta_\ell, \theta_z)$ des encodeurs et décodeurs $(\rho(.,.), h(.,.), \lambda_\ell(.), \lambda_z(.$
Le choix des *prior* log-normal et gaussien multivarié de ℓ et z implique que le terme de divergence $\mathbb{KL}$ est explicite. De manière générale, si $p_1(z) \sim \mathcal{N}(\mu_1, \Sigma_1)$ et $p_2(z) \sim \mathcal{N}(\mu_2, \Sigma_2)$ où $\Sigma_1 = diag(\sigma_{1,i}^2)_{i=1}^D$ et $\Sigma_2 = diag(\sigma_{2,i}^2)_{i=1}^D$, on a :

$$\begin{aligned}
&\mathbb{KL}(p_1(z)\|p_2(z)) = \mathbb{E}_{p_1}\left(log\frac{p_1}{p_2}\right) \\
&= \frac{1}{2}log\frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}\mathbb{E}_{p_1}(z-\mu_1)^t\Sigma_1^{-1}(z-\mu_1) + \frac{1}{2}\mathbb{E}_{p_1}(z-\mu_2)^t\Sigma_2^{-1}(z-\mu_2) \\
&= \frac{1}{2}\left[log\frac{|\Sigma_2|}{|\Sigma_1|} - \sum_{i=1}^D \frac{1}{\sigma_{1,i}^2}\mathbb{E}_{p_1}(z_i-\mu_{1,i})^2 + \sum_{i=1}^D \frac{1}{\sigma_{2,i}^2}\mathbb{E}_{p_1}(z_i-\mu_{2,i})^2\right] \\
&= \frac{1}{2}\left[log\frac{|\Sigma_2|}{|\Sigma_1|} - \sum_{i=1}^D \frac{\sigma_{1,i}^2}{\sigma_{1,i}^2} + \sum_{i=1}^D \frac{1}{\sigma_{2,i}^2}\mathbb{E}_{p_1}\left((z_i-\mu_{1,i})^2 + 2(z-\mu_{1,i})(\mu_{1,i}-\mu_{2,i}) + (\mu_{1,i}-\mu_{2,i})^2\right)\right] \\
&= \frac{1}{2}\left[log\frac{|\Sigma_2|}{|\Sigma_1|} - D + \sum_{i=1}^D \frac{1}{\sigma_{2,i}^2}\left(\sigma_{1,i}^2 + 0 + (\mu_{1,i}-\mu_{2,i})^2\right)\right] \\
&= \frac{1}{2}\left[log\frac{|\Sigma_2|}{|\Sigma_1|} - D + \sum_{i=1}^D \frac{\sigma_{1,i}^2 - (\mu_{1,i}-\mu_{2,i})^2}{\sigma_{2,i}^2}\right].
\end{aligned}$$

De même, si $p_1(\ell) \sim log\mathcal{N}(\mu_1, \sigma_1)$ et $p_2(\ell) \sim log\mathcal{N}(\mu_2, \sigma_2)$, on a

$$\begin{aligned}
\mathbb{KL}(p_1(\ell)\|p_2(\ell)) &= \mathbb{E}_{p_1}\left(log\frac{p_1}{p_2}\right) \\
&= \mathbb{E}_{p_1} log\frac{\sqrt{2\pi\ell^2\sigma_2^2}}{\sqrt{2\pi\ell^2\sigma_1^2}} - \frac{1}{2\sigma_1^2}\mathbb{E}_{p_1}\left[(log\ell - \mu 1)^2\right] + \frac{1}{2\sigma_2^2}\mathbb{E}_{p_1}\left[(log\ell - \mu_2)^2\right] \\
&= log\frac{\sigma_2}{\sigma_1} - \frac{1}{2} + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2}.
\end{aligned}$$

Donc

$$\begin{aligned}
&\mathbb{KL}(q(\ell, z|\lambda_z(x), \lambda_\ell(x))\|p(\ell, z)) = \mathbb{KL}(q(\ell|\lambda_\ell(x))\|p(\ell)) + \mathbb{KL}(q(z|\lambda_z(x))\|p(z)) \\
&= log\frac{\ell_\mu}{\lambda_{\ell,\sigma}} - \frac{1}{2} + \frac{\lambda_{\ell,\sigma}^2 + (\lambda_{\ell,\mu} - \ell_\mu)^2}{2\ell_\sigma} + \frac{1}{2}\left[log\frac{1}{|\lambda_{z,\Sigma}|} - D + \sum_{d=1}^D (\lambda_{z,\sigma,d}^2 - \lambda_{z,\mu,d}^2)\right].
\end{aligned} \tag{4.15}$$

On peut dériver cette partie de l'ELBO directement grâce à une *back-propagation* pour mettre à jour θ_ℓ et θ_z.

La perte de reconstruction $\mathcal{R}$ de scVI s'écrit de façon explicite :

$$\begin{aligned}
\mathcal{R}(x, \rho(.,.), h(.,.), \lambda_\ell(.), \lambda_z(.)) &= \sum_{g=1}^G \mathbb{E}_{q(\ell,z|\lambda_\ell(x),\lambda_z(x))} log\ p(x_g|\rho(z,s), h(z,s), \ell) \\
&= \sum_{g=1}^G \int_l \int_z q(z|\lambda_z) q(\ell|\lambda_\ell) log\mathcal{ZINB}(x_g|\rho, \frac{\theta}{\ell+\theta}, h), \ell) dz\ d\ell
\end{aligned} \tag{4.16}$$

$$= \sum_{g=1}^{G} \left(\int_l \int_z \frac{1}{\sqrt{(2\pi)^{D+1}|\lambda_{z,\Sigma}|\ell\lambda_{\ell,\sigma}}} exp - \frac{1}{2}(z - \lambda_{z,\mu})\lambda_{z,\Sigma}^{-1}(z - \lambda_{z,\mu}) - \frac{1}{2\lambda_{\ell,\sigma}^2}(\ell - \lambda_{\ell,\mu})^2 \right.$$
$$\left. log\left[h\delta_0(x^g) + (1-h)\sum_{k=0}^{+\infty} \delta_k(x^g) \frac{\Gamma(k+\rho)}{\Gamma(k+1)\Gamma(\rho)} \left(\frac{\ell}{\ell+\theta}\right)^k \left(\frac{\theta}{\ell+\theta}\right)^\rho \right] dz d\ell \right).$$

Plutôt que de calculer cette somme d'intégrales, on profite du fait que notre famille variationnelle $\mathcal{Q}_\Lambda$ est une famille de densités de type *moyenne-variance* pour appliquer l'astuce de reparamétrisation et tirer un échantillon de variables aléatoires selon la densité variationnelle puis appliquer une méthode de Monte Carlo, et enfin, calculer la dérivée de la perte de reconstruction afin de mettre à jour $(\theta_\rho, \theta_h, \theta_\ell, \theta_z)$. Toutes les composantes de l'algorithme sont définies pour appliquer la VI implémentée sur VAE de scVI et effectuer une réduction de dimension des observations de comptage x.

Chapitre 5

Analyses des données biologiques

5.1 Objectif général

Certains mécanismes de la biologie cellulaire sont la conséquence d'un motif d'expression génique particulier. L'analyse des données scRNAseq est d'identifier des motifs d'expression particuliers pour les relier à des mécanismes cellulaire. La réduction de dimension permet de faciliter leur analyse, mais il est alors plus difficile de détecter l'information relative aux motifs d'expression. En particulier, la représentation d'une cellule obtenue à partir de la réduction de dimension non-linéaire et non supervisée de l'outil scVI ne peut être interprétée biologiquement. L'objectif de ce chapitre est de chercher comment il serait possible de donner plus de sens à cette représentation latente. L'idéal étant d'attribuer à chaque composante de la représentation un motif d'expression génique ou un mécanisme biologique particulier. On aimerait aussi quantifier les liens entre ces mécanismes, mais l'hypothèse *mean-field* rend cette analyse impossible sur une telle représentation latente, car chaque composante est indépendant des autres.

En pratique, il est difficile de distinguer le signal du bruit dans les données scRNAseq, et l'interprétation des données nécessite une connaissance profonde de la biologie sous-jacente. Actuellement, on ne connait pas parfaitement l'ensemble des gènes impliqués dans un mécanisme donnée, et la contribution des gènes identifiés n'est pas toujours comprise. De plus, des gènes participent à plusieurs mécanismes, ce qui rend les mécanismes dépendants les uns des autres. Il y a aussi une forte variabilité temporelle, certains mécanismes sont a l'oeuvre en continu, et d'autres se produisent uniquement dans quelques cellules à un instant donné. Enfin, il y a une variabilité quantitative entre les gènes exprimés pour un même mécanisme. Deux gènes participants à un même mécanisme peuvent être exprimés dans des proportions très variables.

Pour étudier cette problématique, je me suis particulièrement intéressé au cycle cellulaire. Ce mécanisme est relativemet bien connu du point de vue de l'ARN messager, et il est particulièrement intéressant dans le myélome multiple. En effet, ce cancer est caractérisé par la prolifération de cellules normalement incapables de se diviser. Il est possible qu'une partie des cellules tumorales des échantillons séquencés soit en cours de division, et que la variabilité d'expression qui en résulte nous empêche de percevoir la variabilité d'expression liée à une hypothétique organisation tumorale en sous-clones. Il s'agit là de mon hypothèse de travail. Je me suis demandé si il était possible d'utiliser l'outil de réduction de dimension probabiliste scVI présenté au chapitre précédent pour identifier les effets de la division cellulaire et comprendre comment s'organisent les cellules indépendamment de cet effet.

5.2 Préparation des données

La matrice de comptage sur laquelle j'ai effectué mes analyses est celle qu'on obtient après avoir appliqué les méthodes décrites à la partie 3.1. Dans la suite, on considère que tous les codes barres spécifiques d'une encapsulation correspondent à une unique cellule, et chaque observation est donc une cellule. On considère aussi que chaque UMI correspond à l'expression d'un gène dans le cytoplasme d'une cellule. Les variables sont des gènes et leur valeur dans la matrice de comptage correspond au nombre de copies distinctes présente dans la cellule. A partir de la matrice de comptage, un premier travail de réduction des données à la main a pour but

d'identifier les gènes et les cellules qui ne serviront pas aux analyses ultérieures. Ce travail d'analyse est long et minutieux, chaque décision d'élimination d'un gène ou d'une cellule doit prendre en compte l'expertise des biologistes. Parmi les métriques à partir desquelles on identifie les variables et les observations à éliminer, on trouve la proportion de zéros, l'expression moyenne et l'écart type, ainsi que des métriques construites à partir de ces dernières comme le coefficient de variation, égal à l'écart type sur la moyenne.

5.3 Appliquer l'outil scVI pour étudier le cycle cellulaire

On identifie trois approches pour contraindre un VAE à modifier l'encodage de son espace latent.
— Faire varier les hyperparamètres du modèle.
— Modifier l'information donnée en input (normalisation, sélection de cellules ou de gènes)
— Modifier l'algorithme, choisir une autre modélisation hierarchique bayesienne, une autre famille variationnelle ou une autre fonction objectif à maximiser.

Dans cette section, je présente comment j'ai utilisé scVI avec l'objectif de corriger le cycle cellulaire en ligne de mire. Dans le premier paragraphe, je présente ma prise en main de l'outil et l'influence des hyperparamètres. Ensuite, je montre qu'il est possible d'identifier les types cellulaires à partir de l'outil scVI. Dans une troisième partie j'explique ma tentative de faire passer le cycle cellulaire pour un biais expérimental pour tenter de le corriger. Le paragraphe suivant est consacré à une approche plus astucieuse basée sur la sélection de gènes. Enfin, la dernière partie présente des pistes de modification de l'algorithme avec une famille variationnelle basée sur des distributions issues des statistiques directionnelles.

5.3.1 Prise en main et hyperparamètres

La librairie scVI est codée en python et est principalement basée sur la librairie de machine-learning Torch. Pour une utilisation optimale, je l'ai utilisée en complément de la librairie SCANPY afin d'avoir accès à un large panel d'outils statistiques usuels pour les données single-cell RNA-seq. Cela permet entre autres d'effectuer un pré-traitement des données. Tous les calculs effectués dans cette partie sont effectués sur des données pré-traitées comme présenté dans la section 'Prétraitements' du chapitre 3.

Pour créer et entrainer le VAE décrit dans le chapitre 3, il faut créer des instances des classes VAE et Trainer. L'instanciation du VAE nécessite de spécifier les paramètres du modèle et les caractéristiques des données :
— n_input : dimension de l'espace de départ, nombre de gènes considérés.
— n_batch : nombre de batchs dont sont issues les données.
— n_latent : dimension de l'espace latent, i.e. dimension de la variable z (4.13).
— n_layers : nombre de couches par réseau (paramètre unique pour tous les réseaux du VAE).
— n_hidden : nombre de neurones par couche (les couches sont densément connectées par défaut).

La classe Trainer effectue les calculs nécessaires à l'ajustement des poids des réseaux de neurones du VAE instancié à partir des données présentées en input, on lui spécifie donc les paramètres d'apprentissage suivants :
— gene_dataset : le jeu de données d'apprentissage.
— train_size : la taille de l'échantillon train.
— n_epochs : le nombre de passages sur les données à l'apprentissage.

Le choix du triplet (n_hidden,n_layers,n_latent) détermine la capacité du modèle à construire une représentation latente pertinente. J'ai entrepris d'éclairer ce choix en testant différentes configurations. La figure 5.1 représente l'évolution de l'ELBO (4.14) au cours des époch, chaque courbe correspond à un choix de paramètres. On voit que la capacité d'apprentissage du VAE est relativement constante.

La figure 5.2 permet de visualiser l'effet spécifique du paramètre n_hidden relativement aux autres, les lignes et les colonnes correspondent respectivement à n_latent et à n_layers. On constate qu'augmenter la valeur de chacun des hyper-paramètres implique une augmentation de l'effet de sur-apprentissage sans améliorer la capacité d'apprentissage globale du modèle. En particulier, on préfèrera la valeur de 128 pour n_hidden afin de limiter le sur-apprentissage. Cela n'est pas surprenant, augmenter ces hyper-paramètres

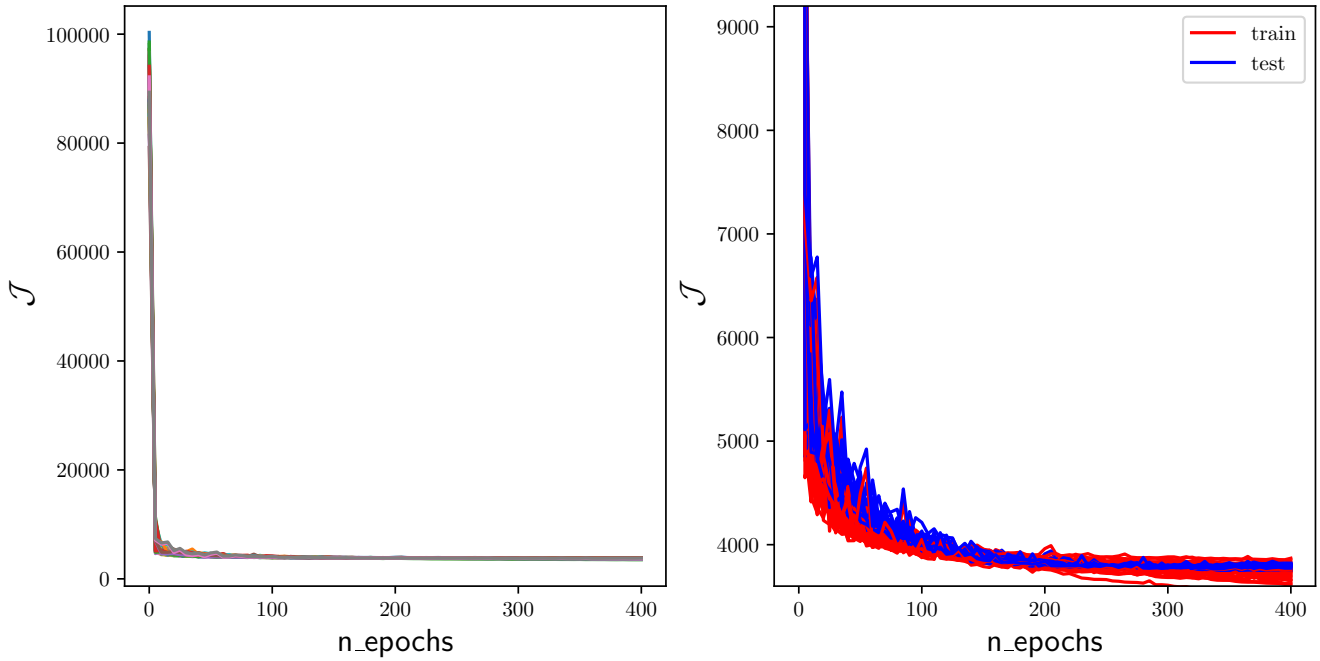

FIGURE 5.1 – Courbes d'erreurs et réglages des hyperparamètres : Tous les paramétrages testés. J'ai testé un large choix de paramètrages, en faisant varier le nombre de couches et le nombre de neurones par couches ainsi que la dimension de la représentation latente. On constate que pour tous ces paramétrages, les performances sont grossièrement similaires. Patient : 0136, Echantillon : Contrôle.

revient à donner plus de degrés de liberté aux fonctions approximées par les réseaux de neurones. Comme l'effet des différents choix n'est pas flagrant, on déduit que le modèle le plus contraint testé est déjà suffisant pour saisir la part généralisable de l'information contenue dans nos échantillons séquencés.

Enfin, sur la figure 5.3 chaque position correspond à un choix de paramètres, les points affichés sont ceux que j'ai testés. La couleur du point correspond à la valeur optimale de l'ELBO obtenue au cours de l'apprentissage sur le sous-échantillon test de l'échantillon d'apprentissage, deux échantillons contrôle et deux échantillons traités sont représentés. Les scores les plus intéressants sont représentés par des points violets. On observe que la configuration la plus contrainte, en bas à gauche, correspond à l'apprentissage le moins efficace, et en règle générale, la contrainte liée à la dimension de l'espace latent est celle qui a le plus d'effet, il faut donc privilégier un nombre de dimensions latentes supérieur à 2. Le constat de la figure 5.2 est corroboré par ce cube, l'apprentissage fonctionne mieux avec la valeur basse pour n_hidden.

En conclusion, cette étude préliminaire permet de définir de bonnes pratiques concernant le choix des paramètres, même si les différences ne sont pas grandes d'un jeu de paramètres à l'autre. Ainsi, on choisira pour la suite $\texttt{n\_latent} = 10, \texttt{n\_layers} = 1$ et $\texttt{n\_hidden} = 128$. Un travail d'optimisation plus approfondi étudierait spécifiquement les valeurs du terme de divergence (4.15) de l'ELBO et de la perte de reconstruction (4.16), et testerait une gamme plus large de paramètres. Par contrainte technique et par souci de temps, je n'ai pas encore pu effectuer ce travail.

5.3.2 Sélection des plasmocytes

Le VAE de scVI entraîné sur un jeu de données donne accès à la représentation latente de chaque cellule. Cette représentation peut prendre la forme d'une distribution de probabilité où d'une position dans l'espace latent. Dans le cas d'une famille variationnelle Gaussienne dans lequel nous sommes, la position dans l'espace latent peut être considérée comme déterministe si l'on ne s'intéresse qu'à la moyenne dans chaque dimension. C'est à partir de cette information que sont effectués les traitements qui suivent.

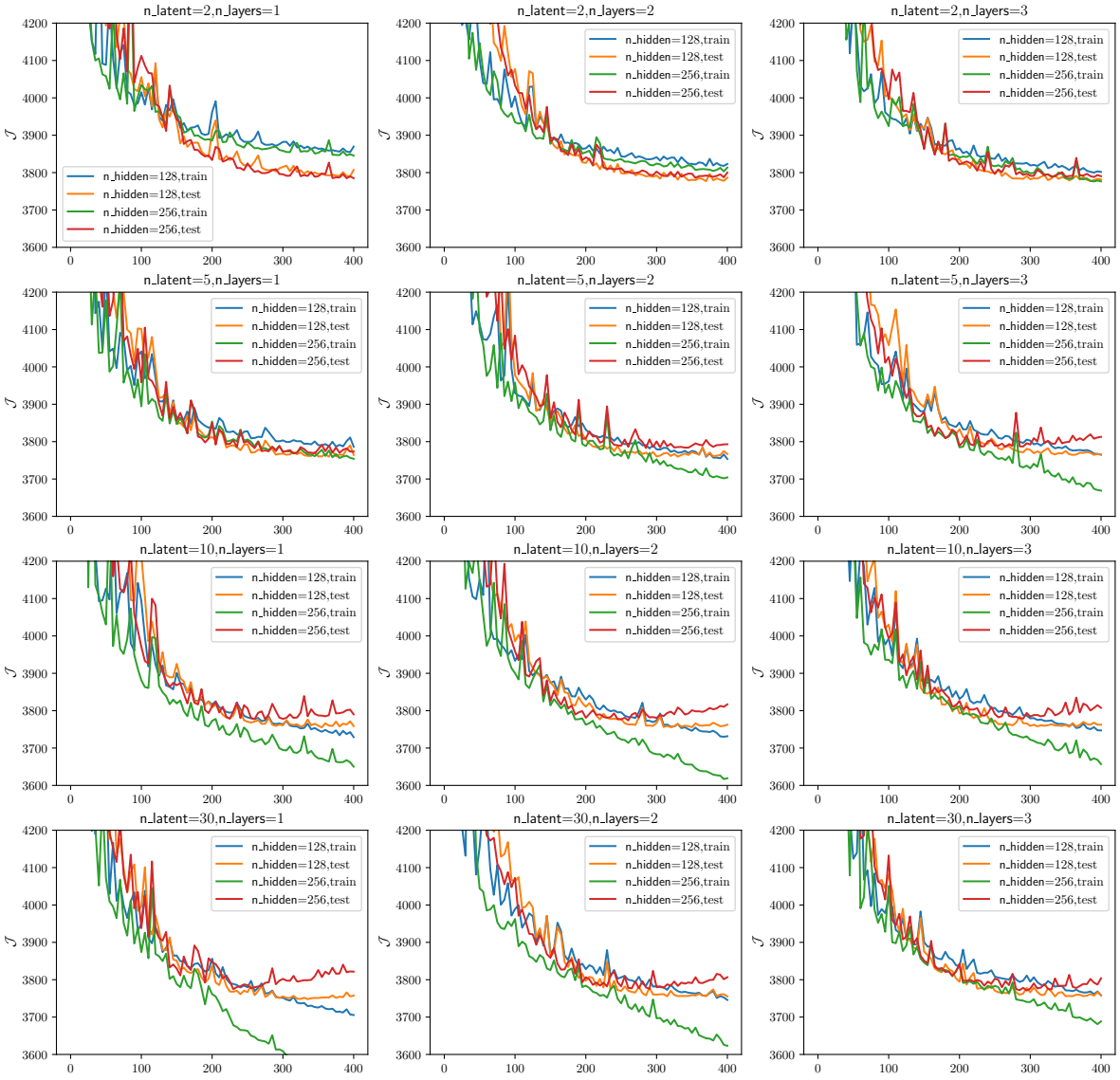

FIGURE 5.2 – Courbes d'erreurs et réglages des hyperparamètres. Chaque case correspond à un réglage différent des paramètres n_latent(de haut en bas) et n_layers(de gauche à droite). Les courbes d'une case correspondent à l'évolution de la perte lors de l'apprentissage pour les différentes valeurs de n_hidden. Patient : 0136, Echantillon : Contrôle.

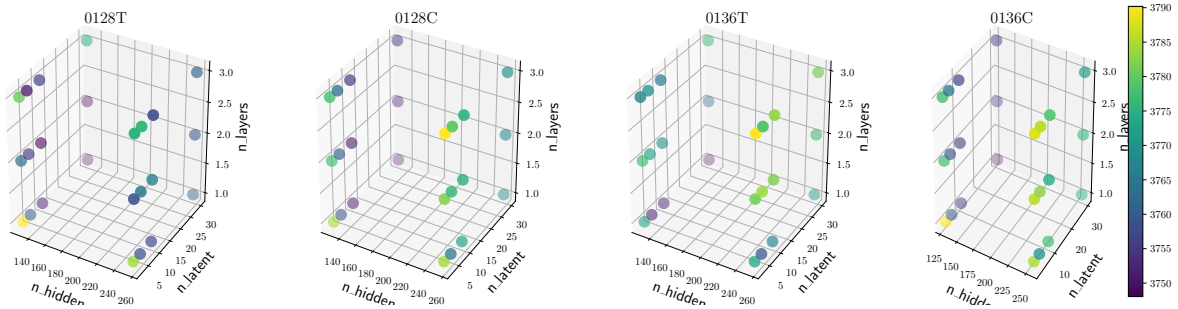

FIGURE 5.3 – Courbes d'erreurs et réglages des hyperparamètres : Chaque cube correspond à un échantillon de cellules, chaque direction du cube correspond à un paramètre, et chaque point est coloré en fonction du meilleur score obtenu sur l'échantillon test au cours de l'apprentissage.

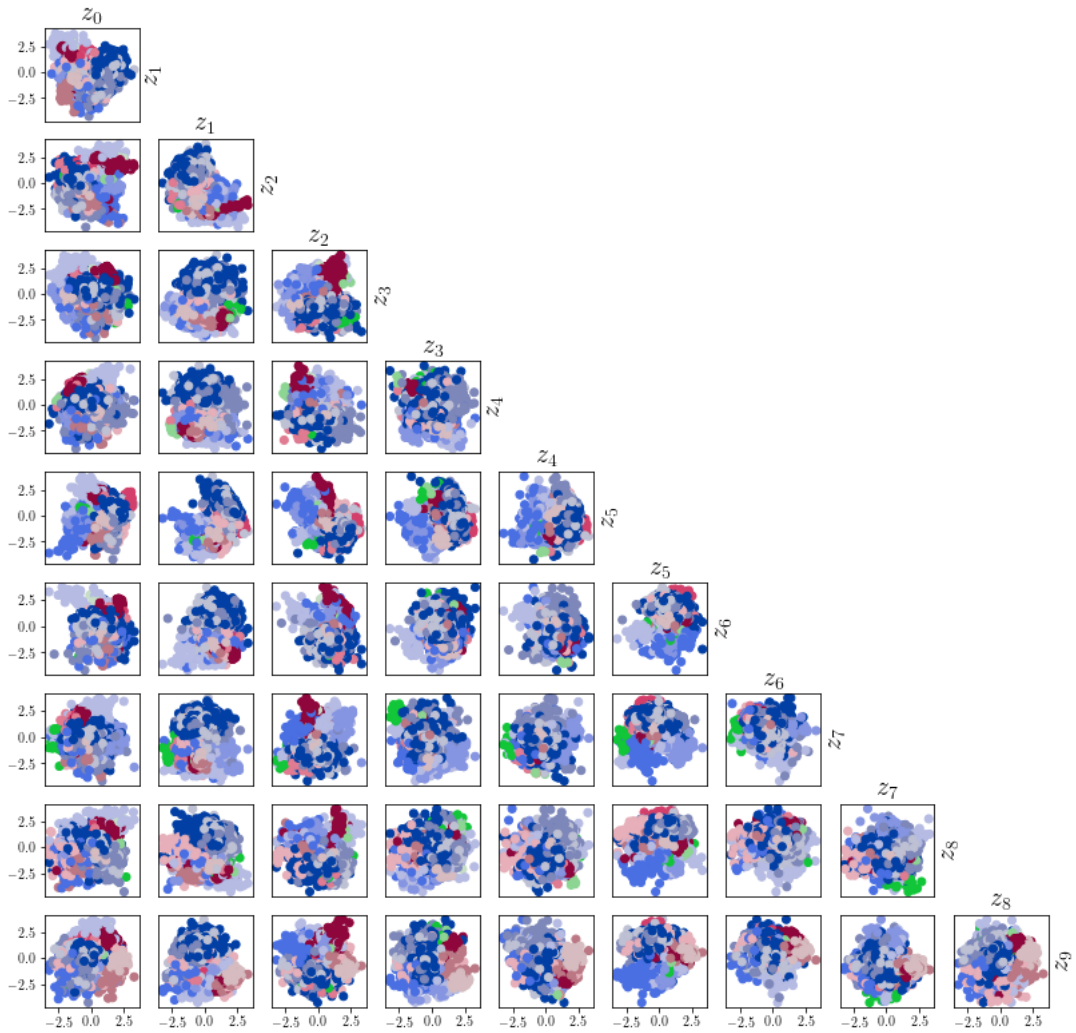

FIGURE 5.4 – Cellules représentées dans l'espace latent à 10 dimensions par la moyenne de leur distribution postérieure approximée après apprentissage. Chaque case correspond à la projection du même échantillon de cellules sur deux directions différentes. On voit bien l'effet du choix du *prior* : les représentations dans chaque direction semblent distribuées selon une gaussienne. Patient :0128, Échantillon :Contrôle.

La figure 5.4 affiche la projection d'un jeu de données après apprentissage sur les plans orthogonaux constitués par les couples de directions issues de l'encodeur. On constate que la distribution est grossièrement centrée autour de zéro dans chaque direction. Cela s'explique par la contrainte imposée à la distribution postérieure d'être proche de la distribution *a priori*, due au terme de divergence de l'ELBO (4.15). Cette représentation 5.4 n'est pas très lisible, on préfère donc les représentations issues des algorithmes de réduction de dimension non-linéaire comme tSNE [Maaten and Hinton, 2008] ou UMAP [McInnes et al., 2018]. Les représentations qui suivent sont générées à partir de la réduction de dimension UMAP, car cette méthode intègre le calcul d'un graphe des plus proches voisins, et ce graphe de proximité est aussi utilisé pour le clustering. Sur la figure 5.5, on peut voir l'effet du nombre de voisins considérés sur la réduction de dimension. On ajuste ce paramètre afin que les clusters qu'on distingue correspondent globalement aux types cellulaires en présence.

En parallèle de la visualisation, notre objectif principal est d'identifier les cellules tumorales de l'échantillon. Pour cela, on utilise un algorithme de clustering. L'algorithme de Louvain [Blondel et al., 2008] infère un clustering à partir du même graphe des plus proches voisins que celui utilisé pour UMAP. Le paramètre resolution de l'algorithme de Louvain influence le nombre de clusters, il faut donc le calibrer pour pouvoir distinguer les types cellulaires. La figure 5.6 illustre l'effet de ce paramètre sur le nombre de clusters obtenus. La masse de cellules à gauche correspond aux plasmocytes tumorales, on constate que les autres types cellulaires sont rapidement identifiés, puis que l'algorithme de Louvain identifie de nouveaux clusters essentiellement dans

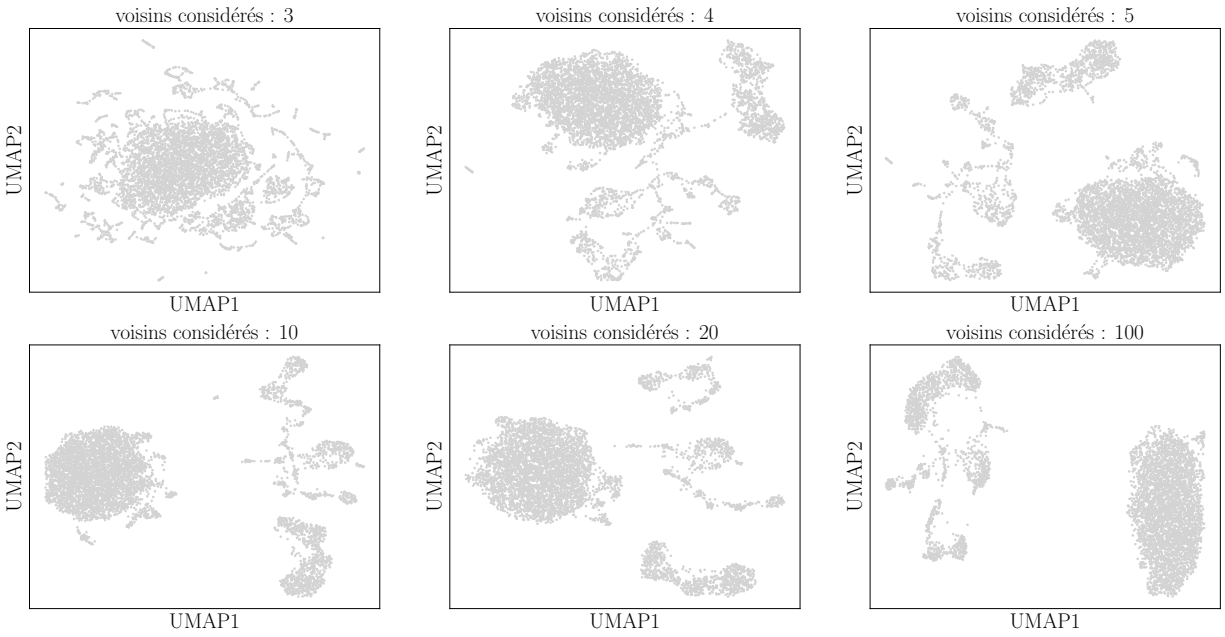

FIGURE 5.5 – UMAP appliqué à l'espace latent, effet du nombre de voisins sur la réduction de dimension. On voit clairement une grosse masse de cellules se démarquer des autres, il s'agit des cellules tumorales. Patient :0128, Échantillon :Contrôle.

la masse tumorale. Cela laisse penser que cette masse tumorale est bien plus hétérogène que les autres types de cellules en présence.

Enfin, on identifie le type cellulaire correspondant à chaque cluster à partir des gènes exprimés par les cellules. Il nous suffit d'afficher la valeur du comptage des gènes spécifiques aux types cellulaires que les biologistes s'attendent à trouver dans l'échantillon. Sur la figure 5.7, les gènes sont colorés en fonction de la quantité d'ARN messagers présents dans la cellule correspondants aux gènes codants pour l'immunoglobuline, la protéine dont la production est la fonction principale des plasmocytes. On identifie alors aisément la grosse masse de cellules excentrée comme le cluster des plasmocytes tumoraux.

5.3.3 Cycle cellulaire en effet batch

Objectif. L'objectif du travail présenté dans cette partie est le suivant : Dans quelle mesure la correction de l'effet du cycle cellulaire permet de faire apparaitre une information biologique pertinente et auparavant inaccessible dans la réduction de dimension effectuée avec le VAE de scVI. Autrement dit, comment le cycle cellulaire influence le comptage de gènes mesuré et comment corriger ce 'biais biologique' ? Puisque cette tentative est exploratoire, les résultats présentés n'ont pas été approfondis, et il y a peut être des éléments intéressants à côté desquels nous sommes passés dans ce travail. L'intérêt à long terme de cette approche était avant tout de prendre en main scVI et d'évaluer la facilité à utiliser cet outil pour l'adapter à des problématiques précises.

Corriger ou harmoniser ? On parle de correction en scRNA-seq lorsqu'on veut corriger un biais expérimental sur des cellules supposées identiques. Lorsqu'on souhaite corriger les effets induits par le cycle cellulaire d'une cellule, on doit donc supposer que la cellule en division est par ailleurs identique aux cellules qui ne sont pas en cours de division. Cette hypothèse est critiquable dans le contexte d'une pathologie complexe comme le Myélome Multiple, dont la particularité est la division de cellules censées être matures et stabilisées. Dans cette situation, il est préférable de parler d'harmonisation [Xu et al., 2019] entre les cellules. L'harmonisation a pour objectif de corriger les effets technologique entre des cellules supposées être différentes. L'harmonisation est, comme la correction, est une altération de l'information mesurée, et il est très difficile de discerner l'information biologique pertinente du bruit induit par les effets technologiques.

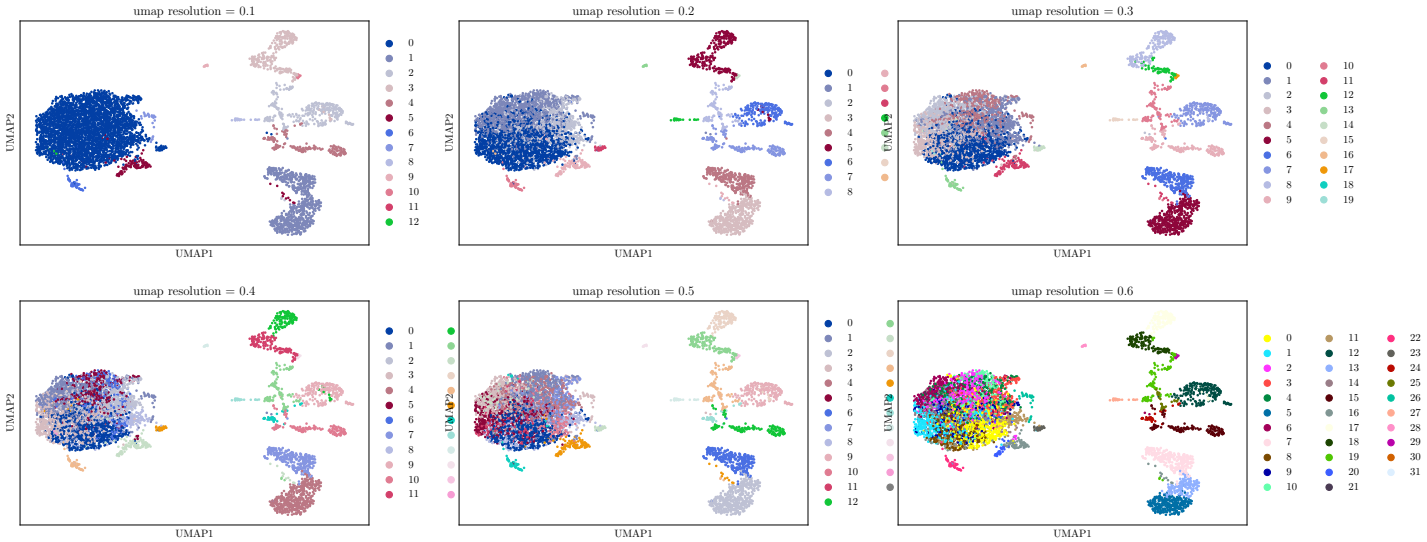

FIGURE 5.6 – UMAP appliqué à l'espace latent, effet du paramètre resolution sur le clustering. Patient :0128, Échantillon :Contrôle.

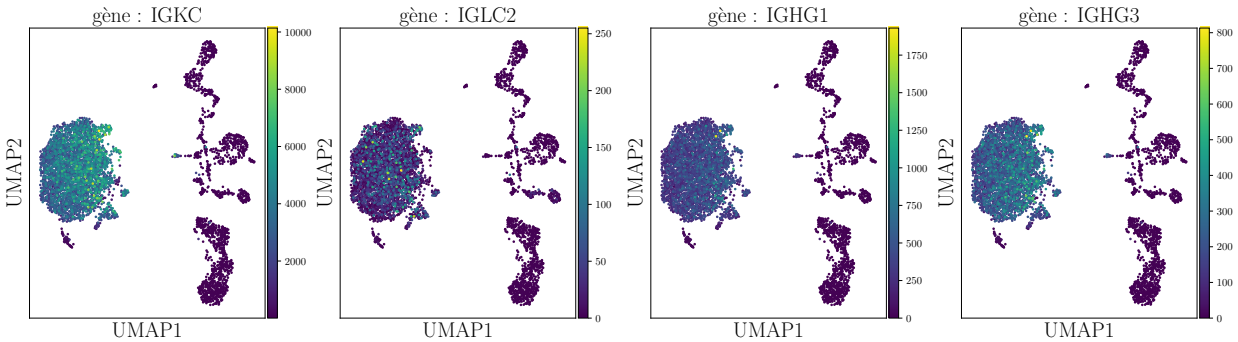

FIGURE 5.7 – UMAP appliqué à l'espace latent, présence des gènes spécifiques des plasmocytes. On voit que la masse de cellules se colore, ce qui confirme le fait qu'il s'agit de la masse des cellules tumorales. Patient :0128, Échantillon :Contrôle.

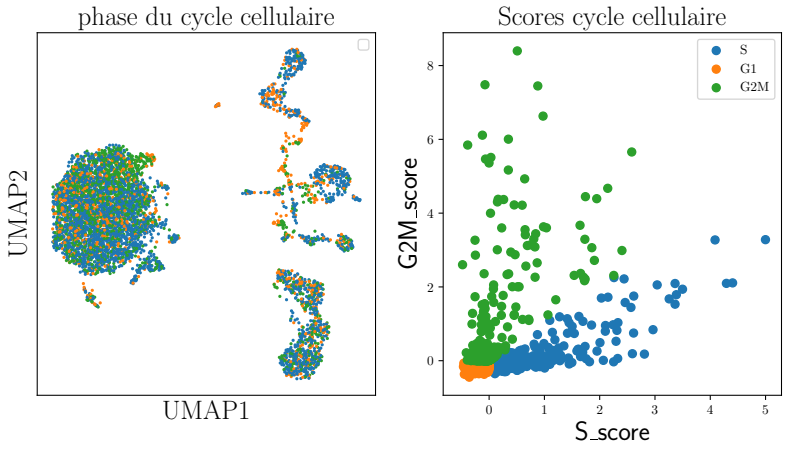

FIGURE 5.8 – Attribution de la phase du cycle cellulaire. Patient :0128, Échantillon :Contrôle.

Intérêt de l'effet batch dans scVI L'outil scVI est conçu de manière à attribuer les différences observées entre les cellules à un effet technologique ou biologique. Les différences d'échelle logarithmique sont attribuées aux biais techniques qui peuvent se produire entre le moment de la séparation des cellules et la capture des ARN messagers jusqu'au séquençage en passant par l'amplification par RT-PCR. Cette source de biais est représentée par la variable ℓ 4.12. Les cellules issues de différentes expériences identifiées par leur information batch s sont traitées à partir de réseaux de neurones distincts dans le décodeur. Cela implique que la même information biologique latente est traité différemment en fonction de l'information batch de la cellule. A travers le terme de perte de reconstruction (4.16), cela influence la représentation latente. Cette particularité de traitement de l'information appelée 'batch' ouvre la voie à la correction d'effets autres que cet effet batch. En effet, il suffirait d'avoir une information catégorielle sur l'état de la cellule par rapport au cycle cellulaire pour pouvoir tenter de le corriger en le faisant passer pour un effet batch lors de l'apprentissage de scVI. Il se trouve que certains chercheurs ont déjà entrepris de déduire l'état de division d'une cellule à partir de son transcriptome.

Déterminer la phase du cycle cellulaire On distingue généralement trois phase pour les cellules en cycle, $S, G1$ et $G2M$. Classiquement, on détermine la phase d'une cellule de façon expérimentale, ici, l'information transcriptomique est la seule dont on dispose, il faut donc inférer la phase. Pour cela, j'ai utilisé la fonction d'inférence du cycle cellulaire de la librairie Scanpy, inspirée d'une étude sur le mélanome métastatique en single-cell [Tirosh et al., 2016]. Elle est basée sur la liste des 93 gènes connus pour être liés au cycle cellulaire. A partir de cette liste de gènes et des données normalisées, les scores S_score pour la phase S et G2M_score pour la phase G2M sont attribués à chaque cellule. On déduit l'information catégorielle de la phase à partir de ces scores. Nous ne rentrerons pas plus dans le détail du calcul des score et de l'attribution de la phase du cycle. La figure 5.8 montre comment est distribuée la phase du cycle sur les cellules de notre échantillon. Pour que le travail présenté ici ait un sens, nous devons faire les hypothèses suivantes :

- Les 93 gènes présentés ci-dessus sont exactement les gènes dont l'expression détermine le cycle cellulaire chez toutes les cellules humaines.
- Les plasmocytes tumoraux se divisent selon le même mécanisme que les cellules humaines saines.
- La méthode d'inférence de la phase du cycle utilisée fonctionne parfaitement.

Résultats. La figure 5.9 représente la projection UMAP des espaces latents construits en considérant ou non le cycle cellulaire en tant qu'effet batch. Visuellement, on constate que l'organisation latente reste similaire, mais il semble que localement, la répartition des cellules en fonction de la phase du cycle inférée est différente. Cependant, on ne peut conclure visuellement sur ce changement et on ne peut pas non plus s'assurer que cette différence n'est pas simplement imputable aux variations qui occurrent lors de l'apprentissage.

Afin de mieux comprendre en quoi l'effet batch influence la construction de l'espace latent, on s'intéresse maintenant à un couple d'échantillons Contrôle/Traité. Les données d'apprentissage sont les plasmocytes

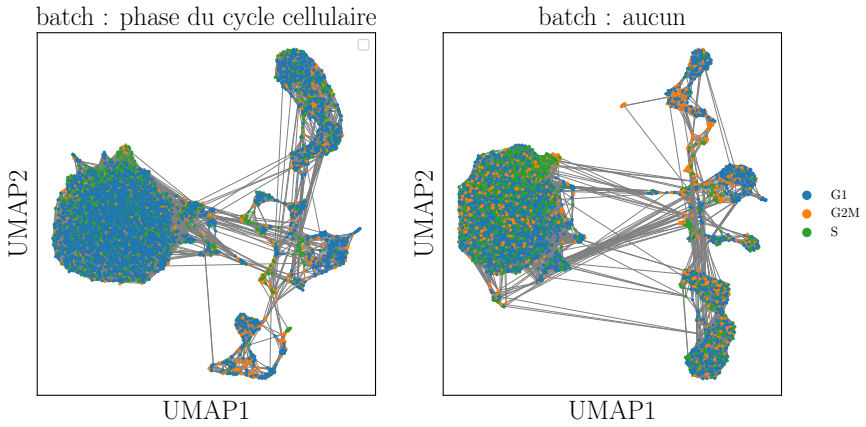

FIGURE 5.9 – Effet de l'harmonisation du cycle cellulaire. Le graphe semble globalement être le même lorsque le cycle cellulaire est pris en compte ou non. Patient :0128, Échantillon :Contrôle.

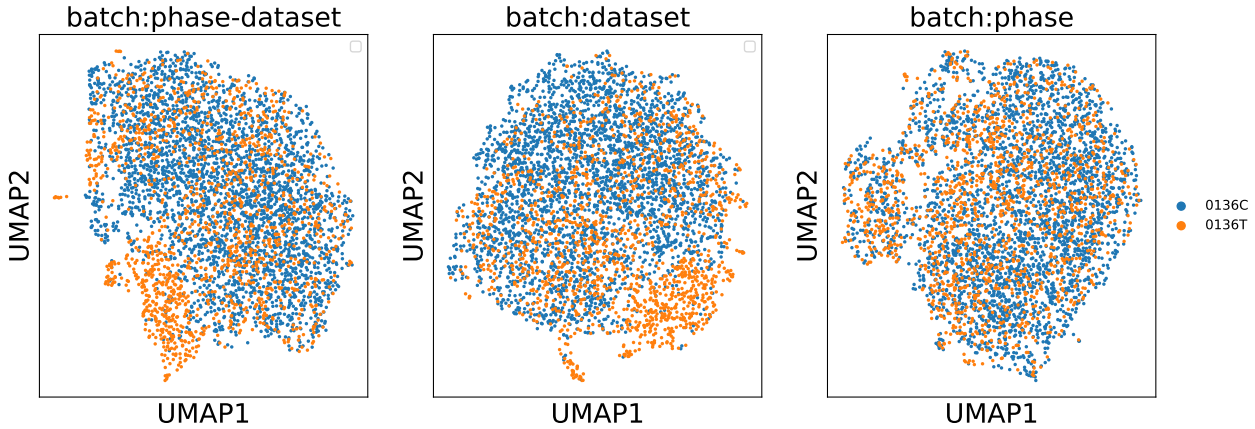

FIGURE 5.10 – UMAP sur les cellules tumorales uniquement, effet de l'information batch. les cellules contrôle et traitées semblent se distinguer sauf dans une région spécifique où l'on voit plus de cellules oranges. Patient :0136, Échantillons :Contrôle/Traité

uniquement, sur lesquelles on effectue trois réductions de dimension, chacune avec une information batch différente annotée dataset, phase et phase-dataset. On rappelle que B correspond au nombre de batchs considérés.

- dataset, $B = 2$: seul le véritable effet batch entre l'échantillon Contrôle et l'échantillon Traité est explicité en information batch pour harmonisation.
- phase, $B = 3$: seule la phase du cycle cellulaire est explicitée en information batch.
- phase-dataset, $B = 6$: la phase et le véritable effet batch sont explicités. Chaque cellule est assignée à un unique batch en fonction de son échantillon d'origine et sa phase du cycle cellulaire estimée.

Sur les projections UMAP de ces trois tentatives en figure 5.10, on observe que lorsque le véritable effet batch est pris en compte, une zone de l'espace projeté contient une forte densité de cellules traitées et très peu de cellules contrôle. Cette masse n'est pas présente lorsque seule la phase du cycle cellulaire a été prise en compte en tant qu'effet batch. D'autres analyses sont nécessaires pour comprendre ce phénomène, mais la prise en compte du cycle cellulaire en effet batch semble avoir un effet intéressant.

5.4 Résultats

Mes analyses ne semblent pas apporter plus d'informations que les analyses déjà effectuées par les membres de l'équipe qui utilisent des outils de réduction de dimension plus classiques comme la PCA. Cependant, il est difficile de conclure car nous n'avons pas d'outils permettant de dire si une réduction de dimension est

meilleure ou moins bonne qu'une autre dans ce contexte biologique où l'on cherche à comprendre notre objet d'étude, et donc on ne sait pas exactement ce que l'on cherche. D'autre part, la possibilité d'établir la phase du cycle cellulaire à partir du transcriptome ne fait pas consensus et il se peut que la méthode utilisée ne fonctionne pas. Enfin, l'utilisation de scVI requiert de grosses capacités de calculs, or, j'ai rencontré beaucoup de problèmes dans l'utilisation des clusters de calcul que j'avais à disposition pour effectuer mon travail, ce qui a grandement limité mes tentatives. Pour toutes ces raisons, l'approche statistique de scVI reste envisagée comme un outil potentiellement utile pour la suite.

Chapitre 6

Conclusion

Bilan personnel Ces six mois ont été l'occasion de découvrir ce qu'étaient la recherche en statistique et la recherche en biologie. J'ai distribué mon temps en trois thématiques complémentaires : la prise en main des ressources de calculs et des outils d'analyse, la compréhension des problématiques biologique et de la complexité liée à cette discipline, et enfin, l'appropriation des concepts mathématiques sous-jacents, en particulier la VI et les VAE. Cette expérience a confirmé mon désir de poursuivre en thèse dans ce domaine. J'ai aussi eu l'occasion de réaliser l'importance de l'aspect humain dans un travail de recherche et l'importance d'avoir des personnes animées par le même désir de comprendre pour avancer.

Bilan technique Même si ce travail n'a pas mené à des résultats positifs, il a été très utile pour poser les fondations d'une collaboration solide entre les biologistes et les statisticiens. Cette collaboration n'est pas toujours facile car la vision rigide et minutieuse des mathématiques est souvent démunie face aux approches globales et approximatives de la biologie. Les méthodes de travail diffèrent mais les deux discipline sont complémentaires lorsque les personnes impliquées arrivent à se comprendre. Pour la suite, nous allons chercher à déterminer les différences de densité locales significatives entre les échantillons contrôle et traité afin d'en déduire l'effet du traitement sur les cellules.

Bibliographie

[Benaniba et al., 2019] Benaniba, L., Tessoulin, B., Trudel, S., Pellat-Deceunynck, C., Amiot, M., Minvielle, S., Gourraud, P. A., de Visme, S., Maisonneuve, H., Lok, A., Le Gouill, S., Moreau, P., and Touzeau, C. (2019). The MYRACLE protocol study : a multicentric observational prospective cohort study of patients with multiple myeloma. *BMC Cancer*, 19(1) :855.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference : A Review for Statisticians. *Journal of the American Statistical Association*, 112(518) :859–877. arXiv : 1601.00670.

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008.

[Butler and Satija, 2017] Butler, A. and Satija, R. (2017). Integrated analysis of single cell transcriptomic data across conditions, technologies, and species. preprint, Genomics.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the *EM* Algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 39(1) :1–22.

[DePasquale et al., 2018] DePasquale, E. A., Schnell, D. J., Valiente-Alandí, i., Blaxall, B. C., Grimes, H. L., Singh, H., and Salomonis, N. (2018). DoubletDecon : Cell-State Aware Removal of Single-Cell RNA-Seq Doublets. preprint, Bioinformatics.

[Ding et al., 2018] Ding, J., Condon, A., and Shah, S. P. (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, 9(1) :2002.

[Ding et al., 2019] Ding, J., Lin, C., and Bar-Joseph, Z. (2019). Cell lineage inference from SNP and scRNA-Seq data. page 9.

[Hafemeister and Satija, 2019] Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. preprint, Genomics.

[Hornik, 1991] Hornik, K. (1991). Approximation Capabilities of Muitilayer Feedforward Networks. page 7.

[Jindal et al., 2018] Jindal, A., Gupta, P., Jayadeva, and Sengupta, D. (2018). Discovery of rare cells from voluminous single cell expression data. *Nature Communications*, 9(1) :4719.

[Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv :1312.6114 [cs, stat]*. arXiv : 1312.6114.

[Kumar et al., 2017] Kumar, S. K., Rajkumar, V., Kyle, R. A., van Duin, M., Sonneveld, P., Mateos, M.-V., Gay, F., and Anderson, K. C. (2017). Multiple myeloma. *Nature Reviews Disease Primers*, 3(1) :17046.

[La Manno et al., 2018] La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastriti, M. E., Lönnerberg, P., Furlan, A., Fan, J., Borm, L. E., Liu, Z., van Bruggen, D., Guo, J., He, X., Barker, R., Sundström, E., Castelo-Branco, G., Cramer, P., Adameyko, I., Linnarsson, S., and Kharchenko, P. V. (2018). RNA velocity of single cells. *Nature*, 560(7719) :494–498.

[Levine et al., 2015] Levine, J., Simonds, E., Bendall, S., Davis, K., Amir, E.-a., Tadmor, M., Litvin, O., Fienberg, H., Jager, A., Zunder, E., Finck, R., Gedman, A., Radtke, I., Downing, J., Pe'er, D., and Nolan, G. (2015). Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell*, 162(1) :184–197.

[Lopez et al., 2018] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12) :1053–1058.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov) :2579–2605.

[Macosko et al., 2015] Macosko, E., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A., Kamitaki, N., Martersteck, E., Trombetta, J., Weitz, D., Sanes, J., Shalek, A., Regev, A., and McCarroll, S. (2015). Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, 161(5) :1202–1214.

[McGinnis et al., 2019] McGinnis, C. S., Murrow, L. M., and Gartner, Z. J. (2019). DoubletFinder : Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors. *Cell Systems*, 8(4) :329–337.e4.

[McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). UMAP : Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv :1802.03426 [cs, stat]*. arXiv : 1802.03426.

[Pierson and Yau, 2015] Pierson, E. and Yau, C. (2015). ZIFA : Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1) :241.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv :1401.4082 [cs, stat]*. arXiv : 1401.4082.

[Schneider et al., 2016] Schneider, V. A., Graves-Lindsay, T., Howe, K., Bouk, N., Chen, H.-C., Kitts, P. A., Murphy, T. D., Pruitt, K. D., Thibaud-Nissen, F., Albracht, D., Fulton, R. S., Kremitzki, M., Magrini, V., Markovic, C., McGrath, S., Steinberg, K. M., Auger, K., Chow, W., Collins, J., Harden, G., Hubbard, T., Pelan, S., Simpson, J. T., Threadgold, G., Torrance, J., Wood, J., Clarke, L., Koren, S., Boitano, M., Li, H., Chin, C.-S., Phillippy, A. M., Durbin, R., Wilson, R. K., Flicek, P., and Church, D. M. (2016). Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. preprint, Genomics.

[Schwann, 1839] Schwann, T. (1839). *Mikroskopische Untersuchungen über die Uebereinstimmung in der Struktur und dem Wachsthum der Thiere und Pflanzen*. Sander, Berlin, 1. auflage edition.

[Shleiden, 1838] Shleiden, M. J. (1838). *Archiv für Anatomie, Physiologie und Wissenschaftliche Medicin.*, volume 1838. Berlin.

[Svensson, 2019] Svensson, V. (2019). Droplet scRNA-seq is not zero-inflated. preprint, Bioinformatics.

[Tirosh et al., 2016] Tirosh, I., Izar, B., Prakadan, S. M., Wadsworth, M. H., Treacy, D., Trombetta, J. J., Rotem, A., Rodman, C., Lian, C., Murphy, G., Fallahi-Sichani, M., Dutton-Regester, K., Lin, J.-R., Cohen, O., Shah, P., Lu, D., Genshaft, A. S., Hughes, T. K., Ziegler, C. G. K., Kazer, S. W., Gaillard, A., Kolb, K. E., Villani, A.-C., Johannessen, C. M., Andreev, A. Y., Van Allen, E. M., Bertagnolli, M., Sorger, P. K., Sullivan, R. J., Flaherty, K. T., Frederick, D. T., Jané-Valbuena, J., Yoon, C. H., Rozenblatt-Rosen, O., Shalek, A. K., Regev, A., and Garraway, L. A. (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science (New York, N.Y.)*, 352(6282) :189–196.

[Vallejos et al., 2015] Vallejos, C. A., Marioni, J. C., and Richardson, S. (2015). BASiCS : Bayesian Analysis of Single-Cell Sequencing Data. *PLOS Computational Biology*, 11(6) :e1004333.

[van der Maaten and Hinton, 2012] van der Maaten, L. and Hinton, G. (2012). Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1) :33–55.

[Wang and Gu, 2017] Wang, D. and Gu, J. (2017). VASC : dimension reduction and visualization of single cell RNA sequencing data by deep variational autoencoder. preprint, Bioinformatics.

[Welch et al., 2018] Welch, J., Kozareva, V., Ferreira, A., Vanderburg, C., Martin, C., and Macosko, E. (2018). Integrative inference of brain cell similarities and differences from single-cell genomics. preprint, Neuroscience.

[Wolock et al., 2018] Wolock, S. L., Lopez, R., and Klein, A. M. (2018). Scrublet : computational identification of cell doublets in single-cell transcriptomic data. preprint, Bioinformatics.

[Xu et al., 2019] Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M. I., and Yosef, N. (2019). Harmonization and Annotation of Single-cell Transcriptomics data with Deep Generative Models. *bioRxiv*.

[Xu and Su, 2015] Xu, C. and Su, Z. (2015). Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12) :1974–1980.

[Young and Behjati, 2018] Young, M. D. and Behjati, S. (2018). SoupX removes ambient RNA contamination from droplet based single cell RNA sequencing data. preprint, Bioinformatics.