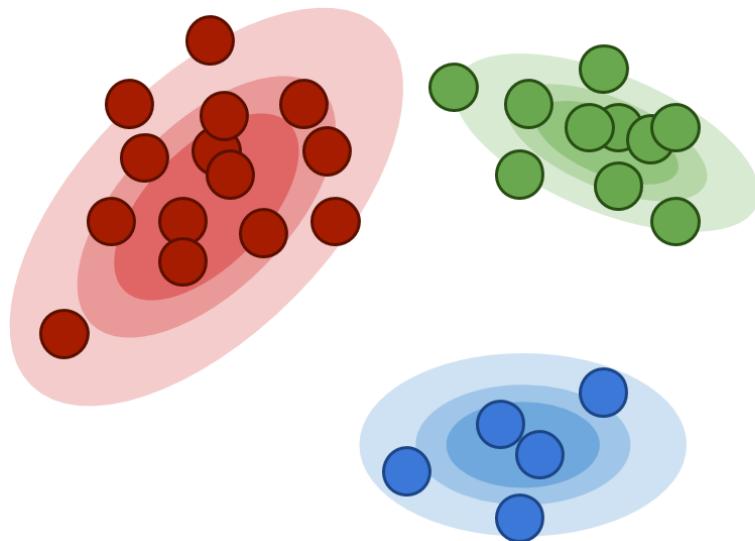


Rapport de stage

Clustering de patients en fonction de leur
type génétique

Luan DECHERY



Université Lumière Lyon 2 - UFR ASSP

24 août 2025

Résumé

Dans la continuité d'un travail consacré à l'exploration de données single-cell RNA-seq provenant de patients atteints de mélanome, ce projet vise à approfondir l'analyse de ces données complexes. Pour cela, plusieurs approches ont été implémentées, notamment le modèle scVI (single-cell variational inference), l'ACP, les méthodes de clustering de type MST et GMM, ainsi que différentes techniques de réduction de dimension. Ces expérimentations ont permis de mettre en évidence certaines caractéristiques structurelles des données et d'ouvrir des pistes pour de futures analyses. En particulier, la complexité et l'hétérogénéité des profils transcriptomiques semblent indiquer que les méthodes de graphe et les embeddings non linéaires offrent une meilleure capacité de représentation et de séparation des sous-populations cellulaires que les approches linéaires classiques. Ce travail fournit ainsi un code modulable et documenté qui pourra être réutilisé et enrichi, notamment en intégrant des stratégies d'optimisation d'hyperparamètres et des approches d'annotation automatique, afin de progresser vers une interprétation biologique plus robuste et généralisable.

Table des matières

1	Introduction	2
1.1	Cadre du stage	4
1.2	Missions du stage	5
1.3	Outils techniques et méthodologie	6
2	Initiation au projet et premières analyses	7
2.1	Revue du code précédent	7
2.2	Préambule à l'analyse des données	9
2.3	Analyse préliminaire des données	10
2.4	Exploration des données	10
2.5	Premiers clusterings	15
3	Méthodologie contextuelle des données single-cell	18
3.1	Présentation théorique	18
3.2	Implémentation	20
3.3	Résultats préliminaires	27
4	Dernières analyses et résultats	30
4.1	Discussion et conclusion	32
5	Annexes	33

Chapitre 1

Introduction

Le cancer constitue l'une des principales causes de mortalité à l'échelle mondiale. Il s'agit d'une maladie complexe, multifactorielle et caractérisée par une forte hétérogénéité, non seulement entre les patients, mais également au sein d'un même tissu tumoral [15]. Cette variabilité, qui peut résulter de différences génétiques, épigénétiques [13] ou environnementales, complique considérablement le diagnostic, le suivi clinique et la mise en place de traitements efficaces. Comprendre cette hétérogénéité constitue donc un enjeu majeur pour améliorer la prise en charge thérapeutique et développer des stratégies de médecine personnalisée.

Avant d'entrer dans les détails, il est important de poser quelques bases théoriques qui guideront par la suite nos analyses et notre démarche de recherche.

Le corps humain est constitué de milliards de cellules, considérées comme l'unité fondamentale du vivant. Depuis le moment de notre conception jusqu'à notre mort, nos cellules se divisent et se renouvellent afin d'accomplir différentes fonctions essentielles au fonctionnement de l'organisme (croissance, réparation, défense immunitaire, etc.)[14].

Chaque cellule contient dans son noyau l'ADN (acide désoxyribonucléique), support de l'information génétique. L'ADN est organisé en gènes, qui sont des segments codant pour des protéines ou régulant leur production [16]. L'ensemble de ces gènes, par leur expression (processus par lequel l'information contenue dans un gène est traduite en protéine), détermine la fonction spécifique de chaque cellule.

Ainsi, bien que toutes les cellules d'un même individu contiennent le même génome, leur profil d'expression génique diffère : c'est ce qui permet à certaines de devenir des cellules musculaires, d'autres des cellules de la peau, des neurones, etc. Ce processus est orchestré par des mécanismes biologiques complexes de régulation qui garantissent le bon fonctionnement de l'organisme.

Cependant, le bon fonctionnement des cellules repose sur un équilibre fin dans la régulation de l'expression des gènes et dans l'intégrité de l'ADN. Lorsque cet équilibre est rompu, par exemple à la suite de mutations (changements dans la séquence de l'ADN), d'altérations épigénétiques (modifications de l'expression des gènes sans changement dans la séquence de l'ADN) ou de dérèglements dans les mécanismes de réparation de l'ADN, les cellules peuvent perdre leur capacité à se contrôler correctement [11].

Dans le cas du cancer, ces altérations s'accumulent progressivement au fil du temps et peuvent être provoquées par des causes internes (erreurs spontanées lors de la réPLICATION de l'ADN, dérèglements hormonaux, inflammation chronique) ou externes (exposition aux rayons ultraviolets, tabagisme, consommation excessive d'alcool, agents chimiques, etc.) [12]

Cette dérégulation permet aux cellules anormales d'échapper aux signaux qui limitent leur croissance, d'éviter la mort cellulaire programmée (apoptose) et de se multiplier de façon incontrôlée. Au fil du temps, elles forment des tumeurs, qui peuvent continuer à croître sans contrôle.

Le cancer est donc fondamentalement une maladie génétique de la cellule, mais il ne s'agit pas toujours d'une maladie héréditaire : dans la majorité des cas, les mutations apparaissent au cours de la vie, sous l'effet de facteurs environnementaux (rayonnements, substances chimiques, virus oncogènes, etc.) ou simplement du vieillissement cellulaire [12].

Aujourd'hui, grâce aux progrès technologiques, nous pouvons approfondir notre compréhension du cancer et de ses mécanismes de développement. L'une des avancées majeures, apparue pour la première fois en 2009 et devenue progressivement plus performante et accessible, est la technique d'analyse unicellulaire (single-cell analysis).

Cette méthode permet d'examiner de manière fine l'expression génétique de chaque gène dans chaque cellule d'un échantillon étudié. Elle offre ainsi la possibilité de réaliser des analyses complexes et précises, telles que : détecter des cellules rares ou atypiques au sein d'une tumeur, qui pourraient être responsables de la résistance à un traitement, ou encore, identifier des sous-groupes de patients en fonction de leur profil d'expression génique, via des approches de clustering.

Depuis sa première apparition en 2009, avec l'étude fondatrice de Tang et al. [7], la technique de single-cell RNA sequencing (scRNA-seq) n'a cessé de gagner en popularité. Cette progression est due à la réduction substantielle des coûts et aux améliorations constantes des technologies associées [3].

À ses débuts, seuls quelques laboratoires spécialisés pouvaient y avoir recours. Aujourd'hui, grâce à l'émergence de plateformes commerciales et à la maturation des outils bio-informatiques, cette technologie est devenue largement accessible, tant dans le domaine de la recherche biomédicale que dans les contextes cliniques [2].

Alors que les approches traditionnelles, comme la bulk RNA-seq, ont permis d'importants progrès dans la compréhension des signatures géniques associées aux cancers, elles restent limitées par le fait qu'elles mesurent une moyenne d'expression sur des millions de cellules. Cette approche globale masque ainsi les différences cruciales entre sous-populations cellulaires, différences qui peuvent être déterminantes dans l'évolution de la maladie ou la réponse aux traitements. [4].

C'est précisément dans ce contexte que le scRNA-seq se révèle particulièrement pertinent pour l'étude du cancer car en offrant une résolution au niveau de la cellule individuelle, il permet de cartographier l'hétérogénéité cellulaire d'une tumeur, d'identifier des cellules rares potentiellement responsables de la résistance aux traitements et de retracer les trajectoires évolutives menant d'un état sain à un état pathologique.

Cette technologie, apparue pour la première fois dans un article fondateur en 2009 et ayant connu une montée en puissance depuis 2015 [1], permet de capturer l'expression génétique de chaque cellule individuellement. Elle offre ainsi une résolution sans précédent, essentielle pour :

- décrypter l'hétérogénéité des tumeurs
- détecter des sous-types cellulaires rares
- et mieux comprendre les trajectoires évolutives de la maladie

Bien que riches en informations, les données issues de la single-cell analysis sont particulièrement complexes à manipuler. Elles sont souvent dispersées (sparse) : la majorité des gènes ne sont pas détectés dans une cellule donnée, soit à cause d'artefacts techniques, soit

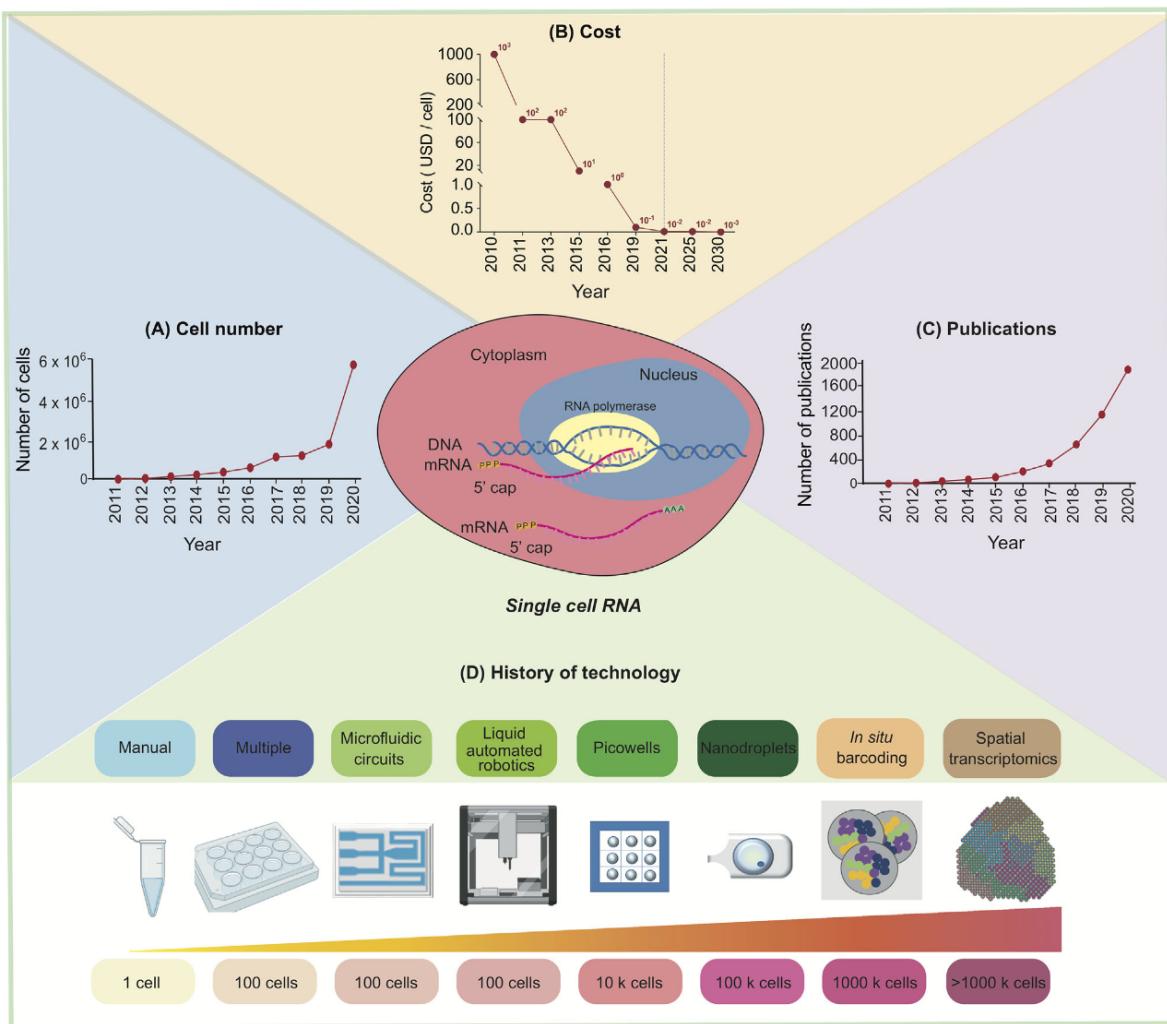


FIGURE 1.1 – Évolution de l’analyse unicellulaire [3]

parce qu’ils ne présentent effectivement aucune expression dans cette cellule (phénomène de dropout).

Ces données sont également bruitées : des difficultés techniques, comme le batch effect, peuvent masquer ou déformer les signaux biologiques réels. Elles présentent aussi une très haute dimensionnalité, avec plusieurs milliers de gènes mesurés sur des milliers, voire des millions, de cellules.

Pour tenter de surmonter ces défis, des pipelines de prétraitement ont été développés et largement décrits dans la littérature. Parmi les outils les plus utilisés, on trouve Seurat (R) et Scanpy (Python), deux bibliothèques largement documentées et spécifiquement conçues pour le traitement et l’analyse de données single-cell.

1.1 Cadre du stage

Dans le cadre du développement de la nouvelle technologie *single cell analysis* le Centre de Recherche en Cancérologie de Lyon (CRCL) a commencé à utiliser cette technologie dans le but d’approfondir ces recherches et connaissances dans les mécanismes du cancer. En partant du principe que si nous connaissons l’expression génétique sur chaque cellule,

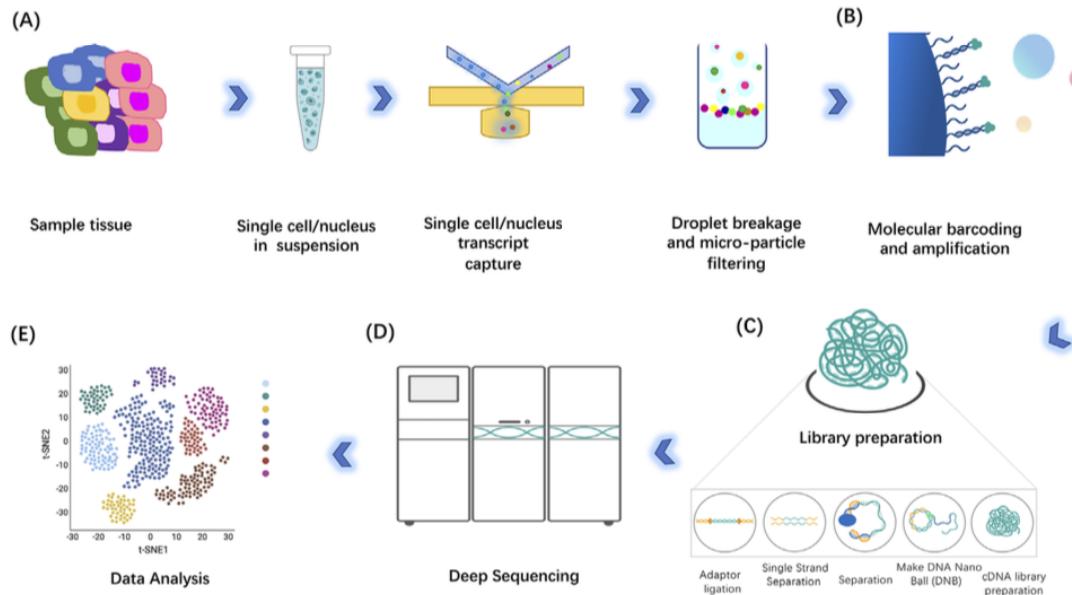


FIGURE 1.2 – Pipeline de l'analyse unicellulaire [3]

nous arrivons à mieux capter les différences génomiques entre les patients, nous pourrons ensuite proposer et adapter les traitements en fonction de ces nouvelles informations et établir plus précisément un diagnostic pour ce patient.

Les données provenant de ce *single cell analysis* constituent des matrices pour chaque patient composées de l'expression génétique de gènes précis pour les cellules étudiées pour un certain patient. Les données sont souvent hétérogènes entre les patients et doivent être sujettes à une étude minutieuse et développée pour pouvoir aboutir à des résultats pertinents pour l'apport d'informations sur le patient.

C'est dans ce contexte que s'inscrit mon stage. À l'aide de techniques de machine learning non supervisés ma tâche principale est d'arriver à dégager de l'information par rapport à ce jeu de données. Mon travail a donc consisté à explorer ces données complexes, à évaluer des méthodes de prétraitement et de réduction de dimension, et à expérimenter différentes approches de clustering dans le but de dégager des profils cellulaires ou patient-spécifiques pertinents pour la recherche en cancérologie.

1.2 Missions du stage

Sous la supervision de mon professeur principal et tuteur de stage, S. Chrétien, cette mission prolonge un travail déjà engagé sur le clustering de cellules à partir de données single-cell. L'objectif principal est d'étudier, optimiser et faire évoluer un code existant afin de produire un clustering pertinent des groupes cellulaires, en intégrant l'ensemble des données disponibles pour plusieurs patients. Ce travail implique à la fois l'exploration des données brutes, l'évaluation et l'amélioration des étapes de prétraitement et de réduction de dimension, ainsi que l'expérimentation de différentes approches de regroupement. L'enjeu est d'aboutir à une organisation des données qui reflète au mieux les profils cellulaires et biologiques, pour faciliter l'interprétation par les chercheurs en cancérologie. Une fois les clusters identifiés, une deuxième partie du travail consiste à représenter la structure interne de chaque groupe à l'aide d'un graphe de type Minimum Spanning Tree

(MST). Ce graphe permet de visualiser les connexions et les relations de proximité entre les cellules à l'intérieur d'un même cluster, facilitant ainsi l'analyse des trajectoires cellulaires et des transitions potentielles entre états.

Les livrables attendus pour ce stage sont doubles :

- d'une part, une analyse complète des résultats obtenus à partir des différents clustering effectués, appuyée par des visualisations pertinentes et des interprétations biologiques ;
- d'autre part, un code propre, documenté et modulaire, suffisamment clair pour être repris ou adapté par des experts en biologie ou par d'autres étudiants souhaitant poursuivre ce travail.

L'objectif final est donc de produire à la fois un résultat scientifique exploitable et un outil technique réutilisable, dans un contexte où les méthodes d'analyse des données single-cell sont encore en pleine structuration.

Les outils utilisés pour ce travail incluent principalement le langage Python, accompagné de plusieurs bibliothèques standards pour la manipulation de données et l'apprentissage automatique (machine learning), telles que NumPy, Pandas, Scikit-learn et Scanpy (spécialisée dans l'analyse de données single-cell).

1.3 Outils techniques et méthodologie

Comme mentionné précédemment, cette étude s'inscrit dans la continuité du travail initié par un autre étudiant sur le même jeu de données. L'objectif final est de réaliser un clustering des patients à partir de leurs profils d'expression génique. Pour cela, la méthode de Gaussian Mixture Models (GMM) a été employée afin de capturer les sous-populations cellulaires présentes.

Enfin, la visualisation des résultats a été réalisée à l'aide du Minimum Spanning Tree (MST). Contrairement à une simple recherche de « plus court chemin », le MST est une technique issue de la théorie des graphes qui permet de relier l'ensemble des points (ici les clusters) en minimisant la somme des distances entre eux, sans créer de cycles. Cette approche est particulièrement utile pour représenter les trajectoires cellulaires ou les relations hiérarchiques entre groupes.

Dans ce contexte, j'ai d'abord consacré une première phase à l'étude du code existant afin de comprendre les avancées déjà réalisées, l'état du projet ainsi que la problématique scientifique sous-jacente.

Dans un second temps, j'ai entrepris ma propre exploration des données pour me les approprier. Cette étape d'analyse exploratoire s'est révélée cruciale en raison de la nature biologique et complexe des données single-cell, ce qui a nécessité de nombreux allers-retours méthodologiques et plusieurs ajustements dans les pratiques adoptées.

Enfin, dans une troisième phase, j'ai procédé à la construction de clusters en testant différentes méthodes non supervisée, dans le but de comparer leurs performances et d'aboutir à un résultat le plus pertinent et exploitable possible pour l'étude.

Chapitre 2

Initiation au projet et premières analyses

2.1 Revue du code précédent

Avant de débuter mes propres analyses, il était essentiel d'examiner l'état du travail déjà réalisé. Cette étape m'a permis de mieux cerner la problématique, de définir plus clairement mes objectifs et d'orienter mes analyses. Elle a également offert l'opportunité d'identifier des parties du code existant pouvant être réutilisées ou adaptées afin de gagner en efficacité dans la suite du travail.

Le code commence par l'importation des librairies nécessaires aux tâches de machine learning, de clustering et de manipulation de données.

	AAACCCAGTCTCAGAT.1_1	AAACCCAGTGTCCGGT.1_1	AAACCCATCGGCTATA.1_1	AAACGAAAGAGCCCAA.1_1	AAACGAATCTTACGTT.1_1
Unnamed: 0					
Bt_Ot_matrix	BT	BT	BT	BT	BT
patient_matrix	14	14	14	14	14
sample_matrix	scrCMA036	scrCMA036	scrCMA036	scrCMA036	scrCMA036
AL627309.1	0	0	0	0	0
AL669831.5	0	0	0	0	0
...
AP000311.1	0	0	0	0	0
AC002480.3	0	0	0	0	0
TRBV7-4	0	0	0	0	0
AC068775.1	0	0	0	0	0
IGHV3OR15-7	0	0	0	0	0

22900 rows x 9075 columns

FIGURE 2.1 – dataset "bt"

Le jeu de données utilisé correspond à une matrice d'expression génique issue de données single-cell RNA-seq. Dans cette matrice, les lignes représentent les gènes et les colonnes correspondent aux cellules individuelles. Les trois premières lignes du tableau contiennent des métadonnées (Bt-Ot-matrix, patient-matrix et sample-matrix), permettant de relier chaque cellule à son patient d'origine ou à son échantillon. Les valeurs contenues dans la matrice représentent le niveau d'expression de chaque gène dans chaque cellule. Dans notre cas, la matrice est de taille 22 900 gènes × 9 075 cellules.

Afin de faciliter la manipulation des données, des fonctions ont été construites pour extraire, à partir du dataframe principal, des sous-jeux de données correspondant à un seul patient. Cette sélection s'appuie sur la ligne « patient-matrix », utilisée comme clé d'identification pour filtrer les cellules associées à chaque patient.

Dans cette partie, plusieurs étapes de traitement sont mises en œuvre pour structurer et visualiser les données. Tout d'abord, l'algorithme NNDescent de la librairie pynndescent est utilisé pour construire un graphe de voisinage à partir des cellules d'un patient. Cet algorithme calcule pour chaque cellule ses voisins les plus proches ($n_neighbors = 10$) et retourne à la fois leurs indices et les distances associées. Ces distances sont ensuite mises en forme et servent de données d'entrée à un modèle de clustering basé sur les mélanges gaussiens (Gaussian Mixture Model, GMM). Le GMM, paramétré ici avec 12 composantes et 50 initialisations ($n_components = 12$, $n_init = 50$), permet de regrouper les cellules en sous-populations probabilistes, chaque cellule étant affectée à un cluster en fonction de sa vraisemblance d'appartenance.

Une fois cette classification obtenue, une réduction de dimensionnalité par Analyse en Composantes Principales (PCA) est réalisée. Trois composantes principales sont extraites, ce qui permet de projeter les données de haute dimension dans un espace 3D. Cette étape facilite à la fois l'interprétation et la visualisation, car elle conserve l'essentiel de la variabilité des données tout en réduisant le bruit.

Ensuite, pour chaque cluster identifié, un graphe est construit en utilisant les distances entre cellules appartenant à ce cluster. À partir de ce graphe, l'algorithme du Minimum Spanning Tree (MST) est appliqué plusieurs fois. Le MST sélectionne les arêtes minimales permettant de relier toutes les cellules du cluster sans cycles redondants, ce qui met en évidence une structure hiérarchique simplifiée. En itérant trois fois le calcul du MST et en supprimant les arêtes déjà utilisées, le code cherche à révéler plusieurs chemins de liaison caractéristiques au sein du cluster. Cette approche permet de dégager une structure plus robuste en éliminant les connexions superflues et en gardant uniquement les liens les plus significatifs.

Enfin, les résultats sont projetés et visualisés : les clusters issus du GMM sont représentés dans l'espace réduit de la PCA, et chaque MST est superposé pour montrer les relations internes entre cellules. Cette visualisation combinée (clusters + MST) fournit une meilleure compréhension des sous-structures présentes dans les données biologiques et permet d'identifier des motifs potentiels dans l'organisation des cellules.

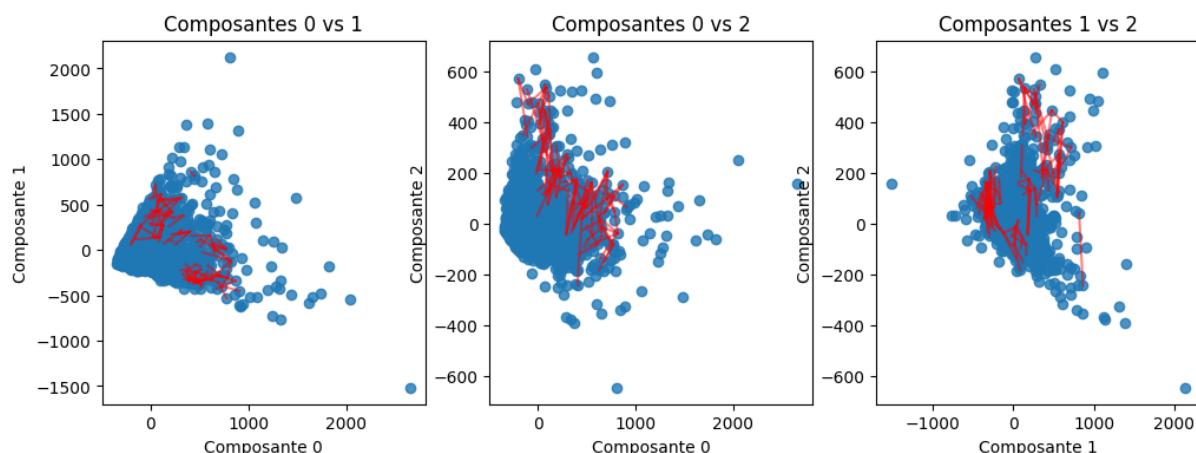


FIGURE 2.2 – Exemple de cluster

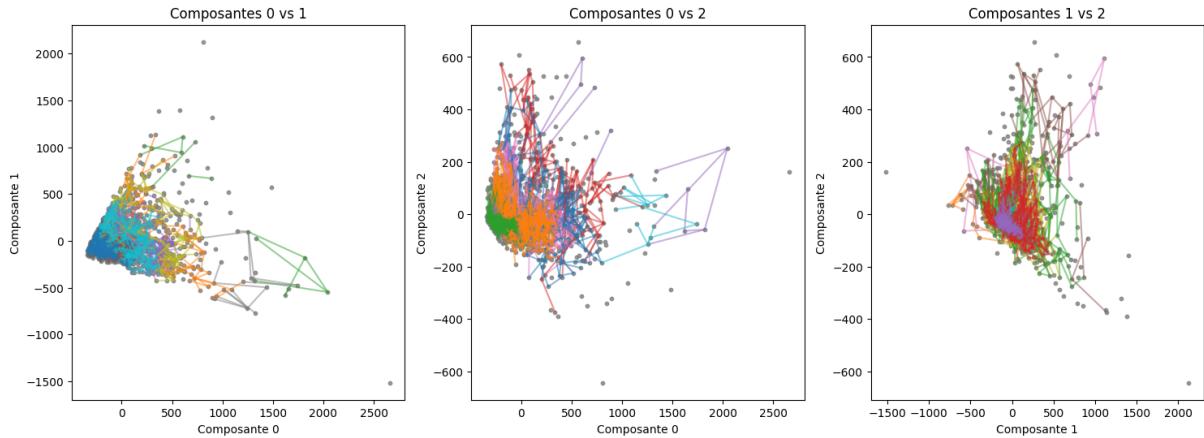


FIGURE 2.3 – Exemple de clusters - MST

Donc ici nous pouvons voir à la fin des clusters aboutis sur les données sur chaque patient, ces clusters représentent bien le lien entre les cellules à l'intérieur des patients.

Dans ce code, nous avons vu la mise en place d'un ensemble d'opérations permettant d'analyser les données d'un patient de manière détaillée. À partir des distances calculées entre voisins proches via nearest neighbors, un clustering a été réalisé avec un modèle de mélanges gaussiens (GMM). Ensuite, une réduction de dimension par ACP a permis de projeter les données dans un espace de plus faible dimension afin de mieux visualiser les groupes identifiés. Enfin, à l'intérieur de chaque cluster, des graphes de distances ont été construits et raffinés par l'application d'arbres couvrants minimaux (MST), permettant de représenter la structure interne des clusters et de les visualiser.

La suite logique de ce travail consisterait désormais à dépasser l'échelle d'un seul patient pour intégrer les données de plusieurs patients. Cela représente une étape essentielle de la problématique, puisque l'objectif final est de dégager des regroupements pertinents entre patients et non seulement à l'intérieur d'un patient isolé. Pour atteindre cet objectif, il sera nécessaire de passer par une phase approfondie d'exploration des données afin de bien comprendre leur organisation et leurs éventuelles spécificités. Cette étape pourra s'accompagner d'un prétraitement ou d'un nettoyage des données, afin de garantir que les clusters obtenus soient significatifs et comparables entre patients. Certaines parties du code déjà existantes, comme la fonction permettant de sélectionner efficacement un patient, pourront être directement réutilisées. Elles serviront également d'inspiration pour développer une méthode plus générale de clustering intégrant plusieurs patients dans une même analyse.

2.2 Préambule à l'analyse des données

Cette étape d'analyse et d'exploration des données a constitué l'un des moments centraux de ce projet. La nature biologique particulière des données, dont je ne maîtrisais pas au départ toutes les spécificités, a représenté un véritable défi et m'a conduit à entreprendre un travail de recherche approfondi. Cela a façonné ma méthodologie et fait évoluer ma manière de comprendre et manipuler les données tout au long du projet.

Dans ce rapport, je présenterai cette démarche de façon linéaire pour en faciliter la lecture. En réalité, mon analyse s'est construite dans un va-et-vient constant entre exploration et modélisation (clustering, projections, etc.). Une première exploration a

permis d'obtenir des résultats initiaux, qui ont ensuite orienté ma compréhension des données et révélé certaines limites. Ces nouvelles connaissances ont à leur tour guidé mes choix méthodologiques et affiné les résultats obtenus.

Cette dynamique entre exploration et modélisation m'a permis non seulement de justifier mes choix analytiques, mais aussi d'inscrire mon travail dans une logique de recherche plus large, en m'appuyant à la fois sur mes expérimentations et sur la littérature scientifique. Les étapes présentées doivent donc être comprises comme une reconstruction ordonnée d'un processus qui, en pratique, s'est déroulé de manière itérative et parallèle.

De ce fait, la structuration de la partie « Résultats » a soulevé certaines interrogations. J'ai néanmoins choisi de présenter les résultats au fur et à mesure de leur obtention. Cette approche reflète plus fidèlement le déroulement du projet et permet de suivre le cheminement réel des analyses, plutôt que de l'enfermer dans une organisation purement conventionnelle (séparer strictement analyses et résultats).

2.3 Analyse préliminaire des données

Dans le but de réaliser un clustering basé sur l'expression génétique de patients atteints de cancer de la peau, j'ai structuré mon travail en plusieurs étapes successives. Une première phase exploratoire a été menée afin d'effectuer une analyse descriptive des données et de me familiariser avec leur structure. Cette étape m'a permis de formuler des premières hypothèses qui ont orienté la mise en place d'un premier clustering.

Cependant, ces résultats initiaux présentaient certaines limites : ils ne prenaient pas encore en compte l'ensemble des patients, et l'analyse portait sur les gènes à l'échelle des patients plutôt que sur leurs groupes cellulaires dans l'ensemble. La suite du travail est née de ce constat, mais également d'un approfondissement sur les spécificités de la *single-cell analysis*, qui nécessite un traitement particulier des données.

Ces recherches ont permis d'aboutir à une méthodologie plus adaptée, avec des outils spécifiquement conçus pour ce type de données, aboutissant à des résultats plus solides et interprétables, que je vais présenter dans la suite de ce rapport.

2.4 Exploration des données

L'analyse a débuté par une phase d'exploration du jeu de données BT, issu de patients atteints de cancer de la peau (/mélanome). Les jeux de données obtenus par RNA-seq single-cell mesurent l'expression de gènes dans chaque cellule mesurée, ce qui signifie que chaque ligne représente un gène, avec pour colonnes chaque cellule mesurée ; chaque élément de cette matrice est l'expression d'un gène sur une cellule. L'objectif de cette phase était d'évaluer la qualité des données, d'en comprendre la structure pour passer ensuite à l'étape de traitement pour pouvoir en faire un clustering.

	AAACCCAGTCTCAGAT.1_1	AAACCCAGTGCCGGT.1_1	AAACCCATCGGCTATA.1_1	AAACGAAAGAGCCAA.1_1	AAACGAATCTTACGTT.1_1
Unnamed: 0					
Bt_Ot_matrix	BT	BT	BT	BT	BT
patient_matrix	14	14	14	14	14
sample_matrix	scrCMA036	scrCMA036	scrCMA036	scrCMA036	scrCMA036
AL627309.1	0	0	0	0	0
AL669831.5	0	0	0	0	0
...
AP000311.1	0	0	0	0	0
AC002480.3	0	0	0	0	0
TRBV7-4	0	0	0	0	0
AC068775.1	0	0	0	0	0
IGHV3OR15-7	0	0	0	0	0

22900 rows x 9075 columns

FIGURE 2.4 – dataset "bt"

L'une des premières analyses a consisté à vérifier si tous les patients disposaient du même nombre de cellules mesurées. Il s'est avéré que le jeu de données est fortement déséquilibré à ce niveau. Cette disparité a rapidement posé problème dans ma première tentative d'analyse : à ce stade, mon objectif était de comparer les données entre patients, d'appliquer une méthode de réduction de dimension comme l'ACP, puis d'effectuer un clustering. Cependant, une telle comparaison se révélait peu pertinente, puisque chaque patient semblait disposer de volumes de données très différents, rendant difficile toute mise en relation directe.

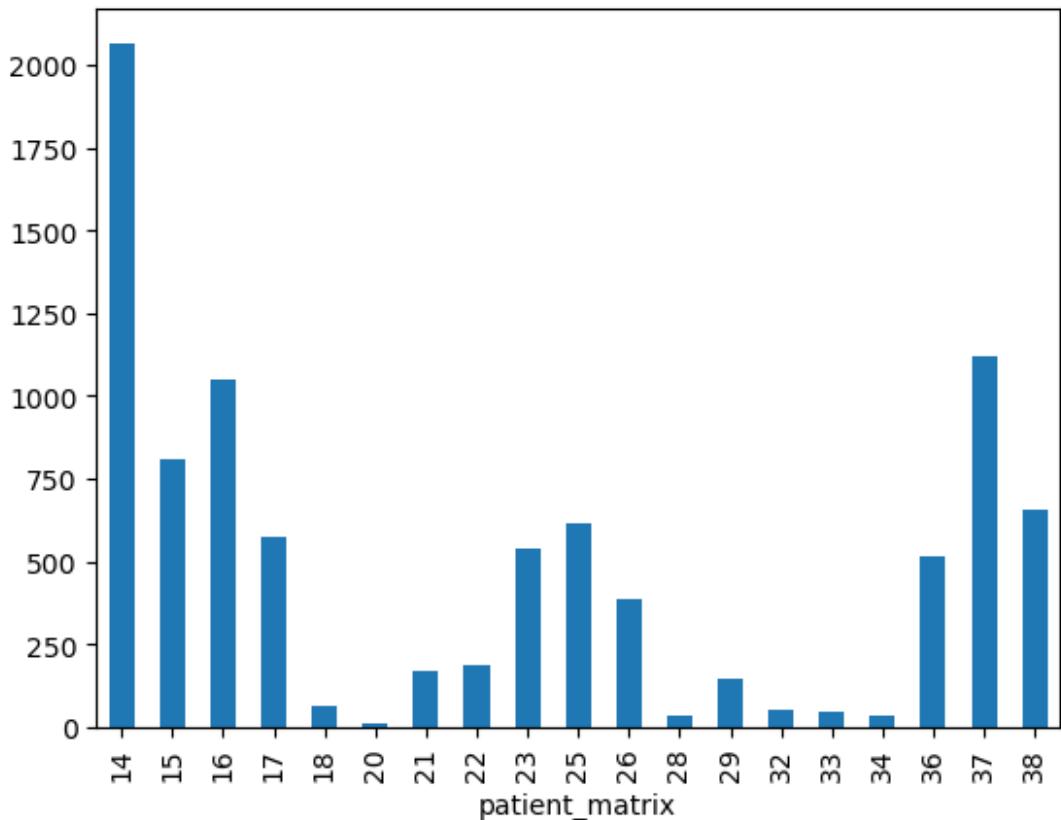


FIGURE 2.5 – Nombre de cellules mesurées par patient

On observe notamment une forte variabilité dans le nombre de cellules mesurées par patient : par exemple, le patient n°14 compte 2 067 cellules, tandis que le patient n°20 n'en possède que 10.

Dans l'objectif d'explorer d'éventuelles similarités entre patients, et dans l'espoir d'identifier des ensembles de gènes communs potentiellement exploitables, j'ai mis en place une fonction permettant de détecter les cellules ayant le même nom à travers différents patients. Pour rendre cette comparaison possible, j'ai pris soin de retirer l'identifiant spécifique à chaque cellule, afin de comparer uniquement les parties du nom véritablement informatives. Le résultat de cette analyse a révélé seulement six doublons de cellules, répartis entre différents patients. Ce nombre très faible indique que les cellules sont, dans leur grande majorité, spécifiques à chaque patient et que les échantillons sont donc faiblement interconnectés à ce niveau.

```
from collections import defaultdict #defaultdict permet de créer un dictionnaire avec des listes comme valeurs par défaut
vu = defaultdict(list)

for idx, col in enumerate(bt.columns): #enumerate pour obtenir l'index et le nom de la colonne
    vu[col].append(idx)

duplicates = {col: indices for col, indices in vu.items() if len(indices) > 1}

print(duplicates)
print(len(duplicates))

[{'CGTAAACAAAGCGAAC': [826, 2387], 'GACCCAGTCCTCTAA': [1011, 6578], 'GCATGTAAGATCACGG': [7083, 8807], 'TACCTTATCAACACCA': [7175, 8179], 'ATCGA': 6}
```

FIGURE 2.6 – Code pour trouver les doublons

À partir de ce constat, j'ai formulé une première hypothèse méthodologique concernant le clustering : la faible correspondance inter-patient des cellules rendait difficile, voire non pertinent, un clustering global basé directement sur les cellules (du moins avec des techniques traditionnelles de machine learning). En effet, les cellules ne pouvant pas être directement comparées entre patients (ni sur la base de leurs identifiants, ni même de leur nombre), une autre stratégie s'imposait.

J'ai alors envisagé une approche inverse : au lieu de projeter les cellules sur un espace défini par les gènes, j'ai choisi de projeter les gènes sur un espace vectoriel structuré par les cellules, ce qui permet d'étudier les relations entre l'expression des mêmes gènes à travers tout les patients.

Pour continuer cette exploration, j'ai également tracé la distribution du nombre de gènes exprimés par cellule pour chaque patient. Cela permet de comparer la complexité transcriptionnelle entre patients et d'identifier d'éventuelles différences globales dans la qualité ou la nature des données. Par exemple, un patient présentant un grand nombre de cellules exprimant peu de gènes pourrait indiquer soit une particularité biologique, soit un effet technique (qualité des échantillons, profondeur de séquençage). À l'inverse, des distributions plus homogènes suggèrent une cohérence entre les patients.

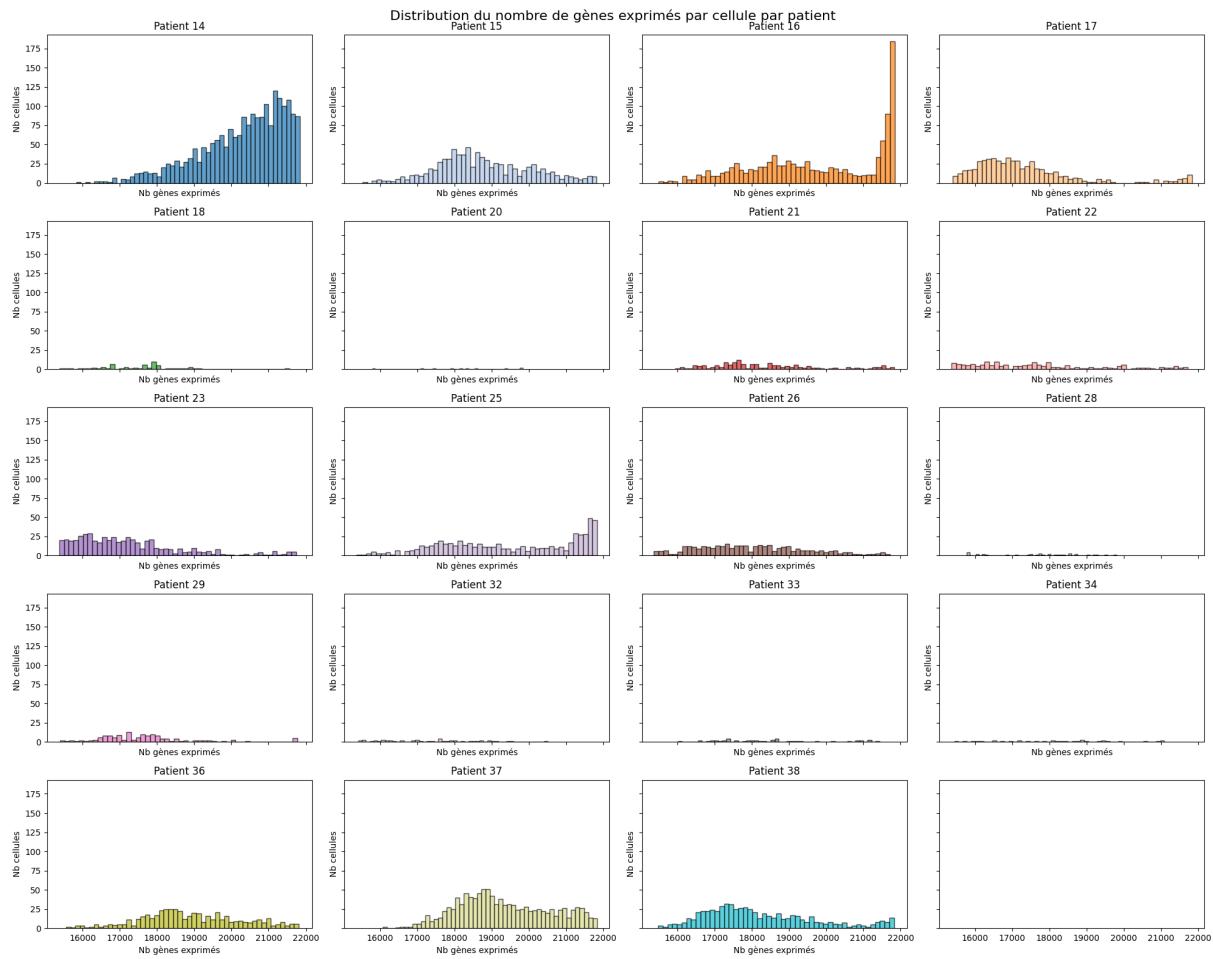


FIGURE 2.7 – Distribution de nombre de gènes exprimés par cellule

La distribution du nombre de gènes exprimés par cellule varie fortement entre patients. Certains présentent un grand nombre de gènes exprimés de façon homogène (patients 36–38), tandis que d'autres montrent une forte variabilité ou des valeurs globalement faibles (ex. patients 18, 21, 28, 32). Cette hétérogénéité reflète à la fois des différences biologiques potentielles et des effets techniques, qu'il sera nécessaire de corriger lors de l'analyse.

Dans la continuité de l'exploration des données, l'utilisation de la fonction *describe()* de la bibliothèque Pandas a permis de dégager plusieurs statistiques descriptives sur la distribution des valeurs d'expression génique. Il apparaît très clairement que la matrice d'expression est majoritairement composée de zéros. En effet, pour une large majorité de gènes, l'expression est absente ou très faible dans la plupart des cellules. Ce phénomène, bien connu dans les données issues de la single-cell RNA-seq, résulte du fait que chaque cellule n'exprime qu'un sous-ensemble restreint de gènes à un instant donné.

```

bt.iloc[3:].T.describe().T
✓ 24.1s
```

	count	unique	top	freq
Unnamed: 0				
AL627309.1	9075	2	0	9039
AL669831.5	9075	6	0	7723
LINC00115	9075	7	0	8643
FAM41C	9075	4	0	8903
SAMD11	9075	6	0	9009
...
AP000311.1	9075	1	0	9075
AC002480.3	9075	1	0	9075
TRBV7-4	9075	1	0	9075
AC068775.1	9075	1	0	9075
IGHV3OR15-7	9075	1	0	9075

22897 rows × 4 columns

```

print((bt.iloc[3:].T.describe()['top'] == 0).sum())
✓ 23.9s
```

22659

FIGURE 2.8 – Describe - gènes

En approfondissant cette analyse, j'ai souhaité vérifier s'il existait des gènes totalement inactifs, c'est-à-dire dont l'expression était nulle dans l'ensemble des cellules du jeu de données. À l'aide du code ci-dessous, j'ai pu identifier ces gènes sans expression, qui ne contiennent donc aucune information exploitable pour l'analyse.

```

print(bt.iloc[3:].apply(lambda gene: (gene == 0).all(), axis=1).value_counts())
```

False	21223
True	1674
Name:	count, dtype: int64

FIGURE 2.9 – Gènes vides

Cependant, avant de traiter les données, j'ai voulu observer comment ces données brutes se comportaient face à l'algorithme de clustering, en particulier au regard des critères BIC et AIC. L'analyse du point d'inflexion de ces critères (souvent appelée méthode du coude) devait m'aider à estimer le nombre optimal de clusters à retenir. (annexe 5.1)

On observe néanmoins que, pour plusieurs patients, les courbes ne mettent pas en évidence un nombre de clusters clairement défini. Cette ambiguïté est probablement liée à la faiblesse du volume de données disponible pour ces patients spécifiques.

Ces éléments posés, j'ai engagé un premier traitement et nettoyage des données, en commençant par l'exclusion des gènes vides. N'apportant aucun signal exploitable, ces

gènes peuvent être considérés comme non informatifs dans le cadre du clustering et ont donc été retirés de la suite de l'analyse. Leur suppression permet non seulement de réduire la dimension de la matrice, mais aussi d'améliorer l'efficacité des traitements ultérieurs, en concentrant l'étude sur les gènes effectivement exprimés.

Dans un second temps, j'ai choisi d'écartier les patients pour lesquels le nombre de cellules mesurées était inférieur à 100. En effet, dans le cadre d'une analyse multivariée comme l'ACP, un volume d'observations trop faible par rapport au nombre de variables peut engendrer deux types de difficultés : d'une part, l'ACP peut échouer ou générer des résultats numériquement instables ; d'autre part, même si elle aboutit, l'interprétation des résultats reste très limitée, car un échantillon trop réduit ne reflète pas de manière fiable une structure sous-jacente.

Ce choix est par ailleurs cohérent avec les observations préalables faites à l'aide des critères AIC et BIC (annexe 5.1), où certains patients présentaient une absence de structure de clustering claire. Leur faible couverture en cellules explique probablement cette instabilité, renforçant ainsi la pertinence de leur exclusion dans les étapes suivantes.

2.5 Premiers clusterings

Afin de faciliter l'expérimentation et de pouvoir généraliser les étapes du clustering, j'ai choisi de construire une fonction appelée GMM_clustering. Cette fonction prend en entrée les données à analyser, ici le jeu de données bt, ainsi que le nombre de clusters souhaité, à utiliser avec le modèle *Gaussian Mixture Model* (GMM).

Dans un premier temps, les données sont centrées et réduites à l'aide de la classe StandardScaler de la bibliothèque scikit-learn, afin d'uniformiser l'échelle des variables et d'éviter que certains gènes ne dominent l'analyse en raison de leur amplitude.

Ensuite, une Analyse en Composantes Principales (ACP) est appliquée pour réduire les données à deux dimensions, en conservant les deux premiers vecteurs propres, et en notant au passage la variance expliquée par ces composantes.

Les données projetées dans ce nouvel espace sont ensuite transmises à l'algorithme GMM, qui procède à la création des clusters selon une modélisation probabiliste. Enfin, les résultats sont visualisés sur un graphique 2D, avec les cellules réparties selon leurs appartances aux clusters. Sur ce graphique, j'ai également mis en évidence les cinq points les plus éloignés du centre des groupes, correspondant aux gènes avec des profils d'expression les plus marqués.

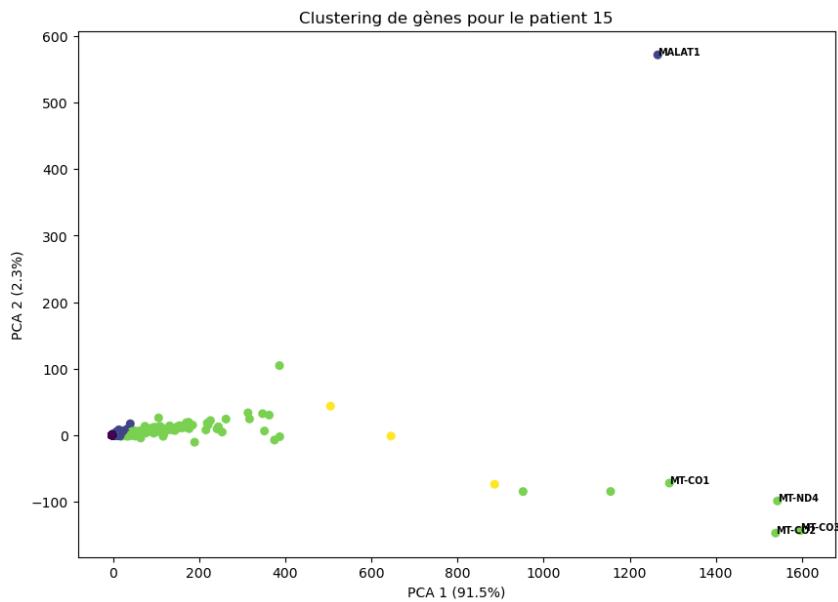


FIGURE 2.10 – Clustering gènes GMM - patient 15

Voici un exemple de clustering appliqué au patient n°15. On observe tout d'abord que la variance expliquée par les deux premières composantes principales de l'ACP s'élève à 93,8 %, ce qui indique que ces deux axes capturent la quasi-totalité de la variance présente dans les données projetées. Cela justifie l'utilisation de cet espace réduit pour visualiser les résultats du clustering.

Parmi les gènes les plus représentatifs (ceux situés aux extrémités de l'espace des composantes), on retrouve le gène MALAT1 ainsi que quatre autres gènes mitochondriaux commençant par MT-. Ces gènes présentent une forte expression dans les cellules de ce patient, et cette tendance se retrouve également chez la majorité des autres patients du jeu de données. Après mes recherches, j'ai découvert que le gène MALAT1 est considéré comme un potentiel biomarqueur du cancer, ce qui corrobore les observations réalisées dans notre cas.[9]

Par curiosité, j'ai ensuite appliqué la même procédure de clustering, mais cette fois en projetant les cellules sur un espace défini par les gènes, ce qui correspondait à notre objectif initial : analyser la similarité entre cellules à partir de leurs profils d'expression génique.

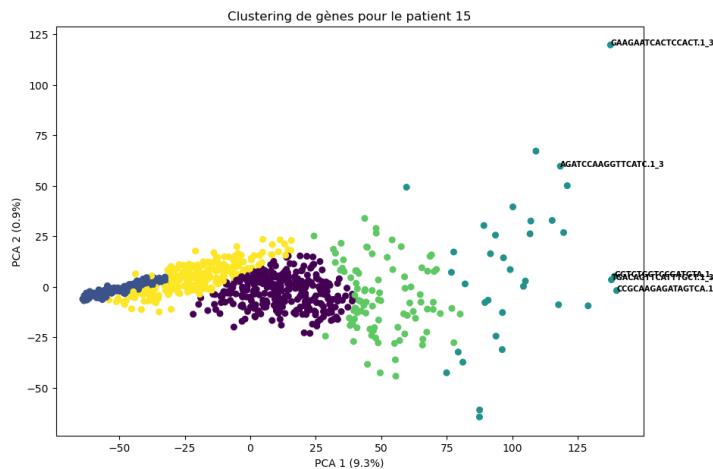


FIGURE 2.11 – Clustering cellules GMM - patient 15

Le résultat obtenu s'est toutefois révélé peu concluant. En effet, la variance expliquée par les deux premières composantes principales dépasse légèrement les 10%, ce qui indique que ces deux axes ne capturent qu'une infime partie de la structure réelle des données. Cela suggère que les cellules, considérées dans leur ensemble, présentent une variabilité très diffuse qui ne peut pas être résumée efficacement dans un espace à faible dimension via une ACP classique.

Il est également important de noter que, lors de ces premiers essais de clustering, je ne me suis pas encore préoccupé de l'optimisation des représentations visuelles. Dans les analyses ultérieures, les données seront traitées de manière plus appropriée, notamment par l'utilisation d'échelles logarithmiques pour corriger l'effet de valeurs trop extrêmes, par l'exploration de techniques d'embedding alternatives comme le spectral embedding, et par l'examen de composantes supplémentaires ou d'autres espaces latents plus adaptés à la nature non linéaire des données single-cell.

Les visualisations associées à cette étape préliminaire sont présentées en annexe 2 5.2. On y observe notamment une surexpression marquée des gènes mitochondriaux, phénomène souvent lié à des cellules en stress ou de faible qualité.

Enfin, même si certaines cellules semblent présenter une expression génique globalement plus élevée, l'interprétation des résultats reste limitée. Comme évoqué précédemment, la forte hétérogénéité entre patients, tant en nombre qu'en type de cellules, rend toute comparaison directe peu fiable. Ces constats renforcent l'idée qu'un clustering pertinent nécessite des approches plus adaptées, capables de mieux tenir compte des spécificités et de la complexité des données single-cell.

Chapitre 3

Méthodologie contextuelle des données single-cell

Face aux limites clairement identifiées lors des premières tentatives de clustering, il est rapidement devenu évident qu'une approche plus poussée, spécifiquement adaptée à la nature des données single-cell, était nécessaire. Pour pouvoir analyser efficacement ces données à forte dimension, sparse, et biologiquement complexes, il fallait non seulement mieux comprendre leurs spécificités, mais aussi adopter des outils conçus pour y répondre.

C'est dans ce contexte que j'ai découvert la bibliothèque Scanpy [17](Single Cell Analysis in Python). Cette librairie open source, largement utilisée dans la communauté scientifique, a été développée pour accompagner la montée en puissance des technologies de single-cell RNA sequencing. Elle propose une suite complète de fonctions dédiées à toutes les étapes de l'analyse : prétraitement, normalisation, réduction de dimension, construction de graphes de voisinage, clustering (avec Leiden ou Louvain), et visualisation (UMAP, t-SNE, etc.).

Grâce à son architecture optimisée autour du format AnnData, Scanpy facilite la gestion de gros volumes de données cellulaires tout en intégrant des méthodes robustes et reproductibles, validées par la recherche. Cette transition vers Scanpy marque une seconde phase dans mon analyse, orientée vers une pipeline plus rigoureuse, modulaire et adaptée au contexte biologique de mon jeu de données.

Avant d'entrer dans la description des étapes techniques, il est nécessaire de présenter le cadre théorique ainsi que les raisons qui motivent l'utilisation du package Scanpy. Celui-ci constitue en effet une référence incontournable pour l'analyse de données single-cell, comme j'ai pu le confirmer à travers mes recherches.

3.1 Présentation théorique

Tout d'abord, l'outil Scanpy repose fortement sur le format de gestion de données AnnData (abréviation de Annotated Data), avec lequel il interagit de manière optimale. À l'instar de Pandas, AnnData permet de manipuler des données tabulaires, mais il s'avère bien plus adapté à l'analyse single-cell en raison de sa structure spécifique.

En effet, AnnData offre une organisation en couches (layers) des matrices de données et la possibilité d'ajouter différentes annotations, qu'il s'agisse d'informations techniques (qualité des mesures, normalisation) ou biologiques (types cellulaires, conditions expérimentales). Ces métadonnées sont directement intégrées à l'objet AnnData et facilitent

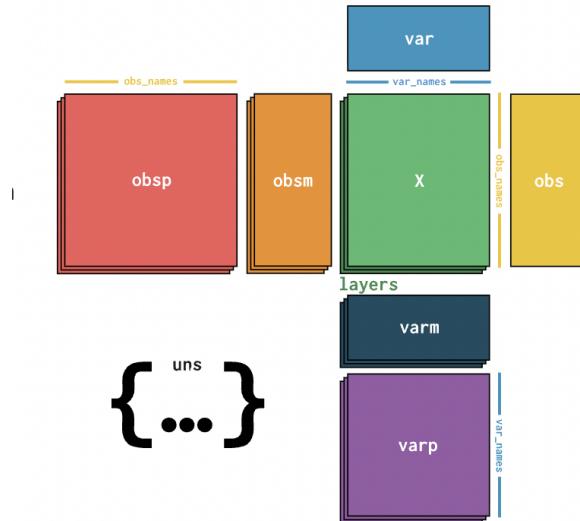


FIGURE 3.1 – Schéma - AnnData /citeanndata

grandement les analyses, en particulier lorsque l'on manipule des jeux de données volumineux et hétérogènes.

Les différentes couches de ce format de données permettent notamment d'accéder facilement à :

- la matrice brute de comptage des gènes,
- les données normalisées ou transformées,
- les graphes de voisinage utilisés pour le clustering ou l'embedding

Cette structuration simplifie considérablement l'implémentation de modèles de machine learning, car elle met à disposition toutes les étapes du pipeline d'analyse dans un même objet cohérent et bien organisé.

Dans le cas de notre dataset BT, ces fonctionnalités sont particulièrement utiles, car elles permettent de gérer simultanément plusieurs patients et de les projeter dans un même espace d'embedding. Cela favorise non seulement la comparaison entre patients, mais aussi l'exploration de structures cellulaires communes ou spécifiques, étape essentielle pour garantir la robustesse et l'interprétabilité des analyses downstream (clustering, visualisation, identification de marqueurs, etc.).

Avant ce stage, je ne connaissais pas cet outil et je suis ravi d'avoir pu le découvrir, tant son approche est adaptée et efficace pour les données single-cell.

La librairie Scanpy est une bibliothèque open source en Python, très similaire à Seurat (développé en R), et qui a été créée dans le but de fournir un cadre flexible et efficace pour analyser des données issues de la single-cell RNA-seq. Conçue pour être scalable (capable de gérer des millions de cellules), elle permet de traiter des jeux de données massifs de manière rapide et accessible.

Scanpy s'intègre parfaitement avec le format AnnData, ce qui en fait un outil central dans les pipelines d'analyse single-cell. Il propose un large éventail de fonctionnalités spécifiques, allant du prétraitement à l'analyse exploratoire avancée, parmi lesquelles :

- Normalisation et filtrage des données (qualité des cellules et des gènes, gestion des doublets).
- Réduction de dimension (PCA, UMAP, t-SNE, Diffusion Maps).
- Construction de graphes de voisinage et clustering (algorithmes de type Louvain

- et Leiden, très utilisés pour détecter des sous-populations cellulaires).
- Détection de marqueurs différentiels (identification des gènes caractéristiques d'un cluster ou d'un type cellulaire).
- Intégration multi-échantillons / multi-patients afin de corriger les effets de batch et d'harmoniser les données.
- Visualisation interactive (cartes de clusters, heatmaps, dot plots, violon plots, etc.).

Grâce à ces capacités, Scanpy constitue une solution complète permettant non seulement d'obtenir des résultats robustes et reproductibles, mais aussi de s'intégrer à des workflows plus complexes, incluant par exemple l'annotation automatique des cellules, l'étude des trajectoires différentielles (pseudotime analysis) ou encore l'intégration avec des données spatiales.

Le dernier outil technique que je vais présenter ici est le modèle de machine learning appelé scVI (single-cell Variational Inference). Ce modèle a été spécialement conçu pour répondre aux défis propres aux données single-cell RNA-seq, qui présentent des caractéristiques particulières telles que leur forte dimensionnalité, leur sparsité (beaucoup de valeurs nulles) et la présence de bruit technique lié aux processus de mesure [5].

L'intérêt majeur de scVI est qu'il regroupe dans un cadre probabiliste unifié la plupart des étapes de traitement traditionnellement appliquées aux données single-cell (normalisation, correction de batch, réduction de dimension, intégration inter-patients, etc.). Concrètement, scVI repose sur un modèle hiérarchique bayésien implémenté via des réseaux de neurones variationnels (VAE – Variational Autoencoders). L'idée est de projeter les données brutes de comptage de gènes dans un espace latent de faible dimension où les relations biologiquement pertinentes entre cellules sont mieux capturées.

Une hypothèse clé de scVI est que les comptages de gènes suivent une distribution binomiale négative à inflation de zéros (Zero-Inflated Negative Binomial, ZINB). Cette distribution est bien adaptée car elle prend en compte à la fois la variabilité biologique et les nombreux zéros techniques (issus de la non-détection d'ARN dans une cellule donnée).

En pratique, scVI permet :

- de produire des embeddings de faible dimension robustes pour la visualisation et le clustering,
- de corriger automatiquement les effets de batch entre échantillons ou patients,
- d'améliorer la robustesse des analyses downstream (classification, trajectoires cellulaires, intégration multi-échantillons).

Ce modèle a été introduit par Lopez et al. (2018)[5] et est aujourd'hui au cœur de la librairie scvi-tools, qui propose différentes extensions.

3.2 Implémentation

Dans cette démarche, j'ai choisi de m'appuyer sur une pipeline d'analyse assez générale pour le traitement des données de single-cell RNA-seq. Cette approche, largement utilisée dans la littérature et mise en œuvre à l'aide des outils Scanpy, suit des étapes bien établies : filtration, normalisation, réduction de dimension, construction du graphe de voisinage, puis clustering et visualisation (**pipeline**).

Pour me familiariser avec cette méthodologie et en comprendre les subtilités, j'ai suivi plusieurs tutoriels de la chaîne "Sanbomics", qui constitue une ressource particulièrement riche pour développer ses propres analyses en single-cell.

J'ai ainsi utilisé cette pipeline standard comme base de travail, tout en l'adaptant aux spécificités de mon jeu de données BT. La méthode présentée est donc à la fois générale

et reproductible, mais aussi ajustée aux particularités biologiques et techniques du projet, ce qui en renforce la pertinence.

Afin de rendre la démarche claire et pédagogique, je vais détailler dans un premier temps les différentes étapes de la pipeline appliquées à un patient spécifique. Le processus sera ensuite généralisé à l'ensemble des patients afin d'obtenir une intégration globale et cohérente des données.

Avant de commencer le traitement proprement dit des données, il est essentiel de passer par une étape de contrôle de qualité, incontournable dans toute analyse en single-cell RNA-seq. La première opération consiste à transposer la matrice d'expression pour qu'elle soit compatible avec les modèles de Scanpy : dans notre cas, chaque cellule devient une observation (ligne), et chaque gène une variable (colonne), ce qui correspond à la structure attendue.

Une fois cette réorganisation effectuée, nous procédons à un filtrage initial des gènes. L'objectif ici est d'éliminer ceux dont l'expression est trop rare pour être informatives : nous supprimons les gènes exprimés dans moins de dix cellules. Ensuite, parmi les gènes restants, nous sélectionnons ceux présentant la plus grande variance d'expression à travers l'ensemble des cellules (ici nous prenons les 2000 gènes avec la plus haute variance). Ce critère est basé sur l'hypothèse que les gènes les plus variables sont les plus susceptibles de contenir une information discriminante pour le clustering.

```
sc.pp.filter_genes(adata, min_cells = 10)

sc.pp.highly_variable_genes(adata, n_top_genes = 2000, subset = True, flavor = 'seurat_v3')

adata

AnnData object with n_obs × n_vars = 811 × 2000
var: 'n_cells', 'highly_variable', 'highly_variable_rank', 'means', 'variances', 'variances_norm'
uns: 'hvg'
```

FIGURE 3.2 – Premier filtrage

Ce processus de sélection permet donc de réduire la dimensionnalité tout en conservant l'information pertinente, ce qui facilite les étapes ultérieures (réduction de dimension, construction des graphes, clustering) et améliore la qualité des résultats.

Un problème courant dans les données issues de la single-cell RNA-seq est la présence de cellules doublons. Cela signifie que, lors du processus de capture et de séquençage, deux cellules distinctes ou une même cellule peuvent être encapsulées ensemble, créant un profil d'expression artificiel qui brouille les analyses ultérieures. Ces doublons techniques peuvent fausser la représentation des populations cellulaires et biaiser le clustering.

Pour remédier à ce problème, nous faisons appel au modèle scVI (single-cell Variational Inference), un modèle probabiliste bayésien profond basé sur des autoencodeurs variationnels. scVI est conçu spécifiquement pour modéliser les données single-cell RNA-seq en prenant en compte des phénomènes tels que la surdimensionnalité, le bruit technique, ou encore les batch effects.

Nous utilisons ici plus précisément sa fonction Solo, qui est un classificateur supervisé.

Solo permet de prédire, pour chaque cellule, une valeur d'être un doublon en s'appuyant sur des simulations de doublons artificiels et sur la structure latente des données. Elle classe ainsi les cellules en deux catégories : "doublet" ou "singlet".

L'identification des doublons constitue une étape fortement recommandée afin de fiabiliser l'analyse. En effet, la présence de doublons, c'est-à-dire des cellules artificiellement considérées comme uniques alors qu'elles proviennent en réalité de la capture simultanée de deux cellules, peut biaiser les résultats des étapes ultérieures telles que la normalisation, la réduction de dimension ou encore le clustering. Exclure ces cellules suspectes permet donc d'améliorer la qualité et la robustesse des analyses.

	doublet	singlet	prediction
AAACCCAGTGCACCGA.1	0.277908	0.722092	singlet
AAACGAAAGCAACAT.1	0.167621	0.832379	singlet
AAACGAAGTCTAACGT.1	0.005573	0.994427	singlet
AAACGCTTCTCATAGG.1	0.282275	0.717725	singlet
AAAGAACCAAATGGCG.1	0.062035	0.937965	singlet
...
TTTGACTTCTCAGGCG.1	0.218148	0.781852	singlet
TTTGGAGAGCAATAGT.1	0.135964	0.864036	singlet
TTTGGAGTCTCGTTA.1	0.005838	0.994162	singlet
TTTGGTTTCTGTCCGT.1	0.578969	0.421031	doublet
TTTGGTTCTCGGTAAGG.1	0.280071	0.719929	singlet

811 rows × 3 columns

FIGURE 3.3 – Jeu de données - prédiction des doublons

Dans notre pipeline, la prédiction des doublons a produit deux colonnes supplémentaires dans la table de métadonnées, l'une indiquant la probabilité qu'une cellule soit un doublon et l'autre précisant sa classification binaire en tant que singlet ou doublon. Ces informations nous ont permis de filtrer plus facilement les cellules et de conserver uniquement celles de haute qualité pour la suite du traitement.

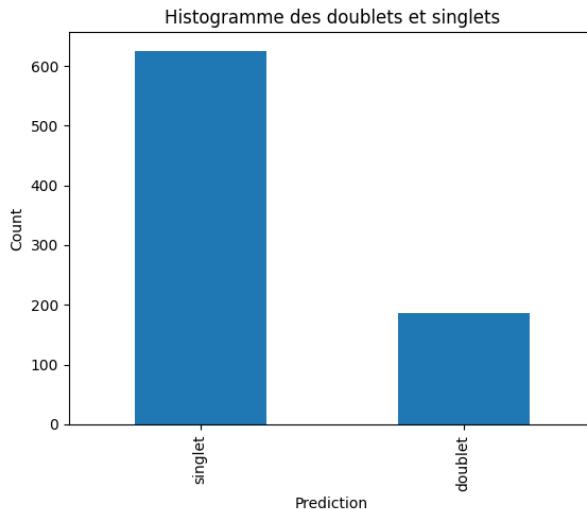


FIGURE 3.4 – Histogramme - prédiction des doublons

```

prediction
singlet    625
doublet    186
Name: count, dtype: int64

```

FIGURE 3.5 – Valeurs - prédiction des doublons

Lors de l’application du modèle de détection des doublons, nous avons constaté qu’environ un tiers des cellules étaient prédictes comme étant des doublons, ce qui constitue une proportion relativement élevée et qui se retrouve également chez les autres patients. Or, retirer directement toutes ces cellules reviendrait à écarter une part trop importante de nos données, ce qui risquerait d’appauprîrir considérablement l’analyse. Pour affiner ce filtrage, nous avons choisi d’examiner plus en détail les scores de prédiction fournis par le modèle, au-delà de la simple classification binaire.

En effet, pour chaque cellule, le modèle attribue à la fois une probabilité d’être un doublet et une probabilité d’être un singlet. En considérant la différence entre ces deux valeurs (probabilité doublet – probabilité singlet), nous obtenons un indicateur plus nuancé : lorsque cette différence est proche de zéro, cela signifie que le modèle est incertain, et la cellule a donc peu de chances d’être réellement un doublet. À l’inverse, plus cette différence est grande et positive, plus la probabilité que la cellule soit effectivement un doublon augmente. Ce choix méthodologique nous permet de mieux équilibrer la rigueur du filtrage avec la conservation d’un maximum d’informations biologiquement pertinentes.

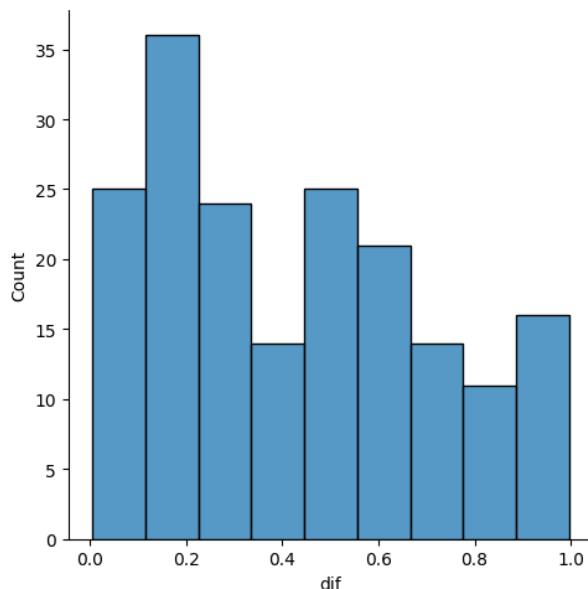


FIGURE 3.6 – Histogramme - différence prédiction

```

df['dif'] = df.doublet - df.singlet
df

```

	doublet	singlet	prediction	dif
AAACCCAGTGCACCGA.1	0.277908	0.722092	singlet	-0.444184
AAACGAAAGCAACAT.1	0.167621	0.832379	singlet	-0.664757
AAACGAAGTCTAACGT.1	0.005573	0.994427	singlet	-0.988854
AAACGCTCTCATAGG.1	0.282275	0.717725	singlet	-0.435450
AAAGAACCAAATGGCG.1	0.062035	0.937965	singlet	-0.875929
...
TTTGACTTCTCAGGCG.1	0.218148	0.781852	singlet	-0.563703
TTTGGAGAGCAATAGT.1	0.135964	0.864036	singlet	-0.728071
TTTGGAGTCCTCGTTA.1	0.005838	0.994162	singlet	-0.988324
TTTGGTTTCTGTCCGT.1	0.578969	0.421031	doublet	0.157939
TTTGGTTCTGGTAAGG.1	0.280071	0.719929	singlet	-0.439858

811 rows x 4 columns

FIGURE 3.7 – Valeurs - différence prédition

Pour le contrôle de qualité, j'ai considéré comme doublons les cellules dont la différence entre les valeurs de prédition dépassait 0,4. Après avoir adapté le code en conséquence, ces cellules ont été exclues des analyses suivantes afin de garantir la fiabilité des résultats.

Une autre étape essentielle du prétraitement des données single-cell, que je n'avais pas anticipée avant de faire des recherches approfondies, consiste à identifier et surveiller les gènes mitochondriaux, généralement notés par le préfixe "MT" dans les noms de gènes.

En effet, une expression élevée de gènes mitochondriaux dans une cellule peut être le signe d'un stress cellulaire important, souvent causé par des dommages ou une apoptose (mort cellulaire programmée), ou encore par un stress induit pendant le processus de séquençage (par exemple, mauvaise conservation ou manipulation des échantillons). Ainsi, un pourcentage élevé de l'expression totale provenant des gènes mitochondriaux est souvent utilisé comme critère d'exclusion pour éliminer les cellules de mauvaise qualité [10].

En contexte oncologique, plusieurs études montrent que les cellules tumorales présentent souvent des altérations du métabolisme mitochondrial, ce qui peut favoriser leur progression ou réduire leur sensibilité à la mort cellulaire. Par exemple, certains travaux ont mis en évidence un lien entre ces changements et un moins bon pronostic dans certains cancers comme le carcinome de l'œsophage [8].

De même, les gènes ribosomaux, bien qu'ils soient naturellement très exprimés, peuvent dominer le signal transcriptomique et masquer des variations plus subtiles entre les cellules. Il est donc fréquent, selon les objectifs de l'étude, de les traiter séparément, voire de les exclure temporairement, afin de mieux faire ressortir les différences d'expression liées à des processus biologiques spécifiques.

Ces deux types de gènes, mitochondriaux et ribosomaux, sont donc des indicateurs critiques pour le contrôle qualité, et leur gestion adéquate conditionne la fiabilité des analyses comme le clustering ou l'identification de sous-populations cellulaires.

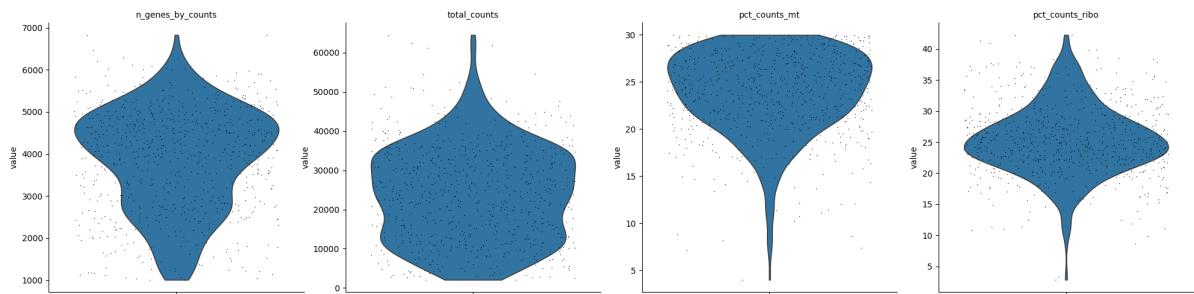


FIGURE 3.8 – Violin plot - patient 15

Voici un exemple de distribution obtenue après le prétraitement des données.

- "n genes by counts" représente le nombre de gènes exprimés dans chaque cellule, c'est-à-dire le nombre de gènes dont l'expression est détectable (avec un comptage supérieur à zéro).
- "total counts" indique le nombre total de transcrits détectés pour chaque cellule, soit la somme des valeurs d'expression de tous les gènes exprimés dans cette cellule.
- "pct counts mt" et "pct counts ribo" correspondent respectivement au pourcentage de l'expression totale attribuée aux gènes mitochondriaux (prefixés par "MT-") et aux gènes ribosomaux.

Ces indicateurs sont essentiels pour évaluer la qualité des cellules individuelles. Par exemple, une cellule présentant un faible nombre de gènes exprimés et un fort pourcentage de gènes mitochondriaux peut être considérée comme stressée ou endommagée, et donc potentiellement à exclure de l'analyse. Les autres plots pour les autres patients sont présents à l'annexe 5.3.

Les gènes ribosomaux n'ayant pas de préfixe au nom il est nécessaire de charger une liste depuis internet pour récupérer leur nom. Une fois ces gènes identifiés nous pouvons produire des statistiques comme le pourcentage de ces gènes présents sur une cellule ou encore un groupe de cellules pour un patient spécifique par exemple, ces indicateurs nous donnent une idée générale de la qualité et la santé et ce groupe choisi.

```
sc.pp.calculate_qc_metrics(adata, qc_vars=['mt', 'ribo'], percent_top=None, log1p=False, inplace=True)
```

`adata.obs`

	doublet	n_genes_by_counts	total_counts	total_counts_mt	pct_counts_mt	total_counts_ribo	pct_counts_ribo
AAACCCAGTCAACGA	False	3327	18744.0	5000.0	26.675203	5081.0	27.107340
AAACGAAAGCAACAAT	False	4809	33530.0	9592.0	28.607216	7226.0	21.550850
AAACGAAGTCTAACGT	False	1663	4976.0	1267.0	25.462219	1042.0	20.940514
AAACGCTTCTCATAGG	False	5124	35903.0	7222.0	20.115311	9333.0	25.995043
AAAGAACCAAATGGCG	False	2932	14366.0	3752.0	26.117220	4066.0	28.302938
...
TTTGACTTCTCAGGCG	False	4543	24005.0	5090.0	21.203917	4123.0	17.175589
TTTGGAGAGCAATAGT	False	4474	29879.0	7569.0	25.332174	9086.0	30.409319
TTTGGAGTCTCGTTA	False	2313	11667.0	2939.0	25.190708	3537.0	30.316275
TTTGGTTCTGTCCGT	False	3356	16264.0	2959.0	18.193556	5236.0	32.193802
TTTGGTTGTCGGTAAGG	False	3946	24719.0	7067.0	28.589344	6531.0	26.420971

719 rows x 7 columns

FIGURE 3.9 – Statistiques données mt ribo

Il est généralement recommandé d'éliminer les cellules qui présentent une expression

élevée en gènes mitochondriaux et ribosomaux, car cela peut indiquer un stress cellulaire, une apoptose ou une mauvaise qualité de séquençage. Cependant, la première fois que j'ai appliqué ces filtres de manière stricte, je me suis retrouvé avec à peine plus de 150 cellules valides sur l'ensemble du dataset, ce qui représentait une perte considérable d'information.

En examinant plus attentivement les données, j'ai constaté que la majorité des patients présentaient plus ou moins des taux élevés d'expression en gènes mitochondriaux. Cela m'a poussé à approfondir mes recherches, notamment en me focalisant sur le contexte du mélanome, qui est ici le type de cancer étudié.

J'ai alors découvert que plusieurs études scientifiques établissent une corrélation entre une surexpression des gènes mitochondriaux et ribosomaux et la présence de cellules tumorales dans divers types de cancers, y compris le mélanome. Dans ce contexte, ces cellules pourraient donc représenter des cellules tumorales actives, et leur élimination risquerait de masquer une partie essentielle du signal biologique [6].

C'est pourquoi j'ai décidé de conserver ces cellules dans l'analyse, en assumant qu'elles pourraient contenir des informations biologiques pertinentes liées à l'état tumoral.

Cette même procédure de prétraitement a été appliquée à l'ensemble des patients, puis les résultats ont été consolidés dans une unique matrice, aboutissant à une table finale de 8 652 cellules (lignes) \times 19 207 gènes (colonnes). Cette intégration permet d'obtenir une vision globale tout en garantissant l'homogénéité du jeu de données. Avec cette nouvelle matrice, nous disposons ainsi d'un cadre plus robuste et plus adapté pour produire des analyses fiables et comparables entre patients.

	Sample	doublet	n_genes	n_genes_by_counts	total_counts	total_counts_mt	pct_counts_mt	total_counts_ribo	pct_counts_ribo
TTACAGGAGATTACCC	patient_22	False	2487	2482	8612.0	2585.0	30.016254	88.0	1.021830
TCCATCGCAAATTGGA	patient_25	False	1172	1170	4675.0	1403.0	30.010695	791.0	16.919786
TATCAGGCATGACTAC	patient_25	False	4872	4872	37927.0	11378.0	29.999737	7388.0	19.479527
ACTGATGCAAAGCAAT	patient_16	False	1221	1221	3377.0	1013.0	29.997040	568.0	16.819662
ATTTCTGGTTCACGC	patient_16	False	1051	1051	2807.0	842.0	29.996437	510.0	18.168863
...
CTGATCCGGTGGAGGT	patient_38	False	5529	5529	49064.0	8.0	0.016305	21884.0	44.602966
CTCGAGGAGTACACCT	patient_38	False	1167	1167	6257.0	0.0	0.000000	3147.0	50.295670
CTAATGGAGAGGGCTT	patient_38	False	1479	1478	7073.0	0.0	0.000000	3595.0	50.827087
ACCAGTAGTTACAGGT	patient_38	False	2482	2481	11868.0	0.0	0.000000	5144.0	43.343445
TGCCCTAGTACAGTTC	patient_38	False	1305	1304	3463.0	0.0	0.000000	1311.0	37.857349

8652 rows \times 9 columns

FIGURE 3.10 – Adata

Cependant, il est fortement recommandé que le nombre de variables soit inférieur au nombre d'observations. En effet, avoir plus de variables que d'observations peut poser plusieurs problèmes. Cela complique l'apprentissage des modèles statistiques ou de machine learning, car ceux-ci risquent de surajuster les données (overfitting). Cela augmente fortement le coût computationnel et rend aussi plus difficile l'estimation fiable des paramètres, notamment dans les modèles probabilistes comme scVI.

Pour réduire la dimensionnalité tout en conservant l'information la plus discriminante, nous avons donc sélectionné les 4 500 gènes présentant la plus grande variance sur l'ensemble du jeu de données. Ce filtrage permet de ne garder que les gènes les plus informatifs, c'est-à-dire ceux qui varient le plus entre les cellules, et donc les plus susceptibles de contribuer au bon apprentissage du modèle scVI appliqué ensuite à l'ensemble du dataset.

```

[23] sc.pp.highly_variable_genes(adata, n_top_genes=4500, subset = True, layer = 'counts',
|   |   |   |   flavor = "seurat_v3", batch_key="Sample")
...
[24] /Users/luandechery/Documents/faculdade/python/stage/.conda/lib/python3.11/site-packages/anndata/_core/anndata.py:1756: |
  utils.warn_names_duplicates("obs")

[25] scvi.model.SCVI.setup_anndata(adata, layer = "counts",
|   |   |   |   categorical_covariate_keys=["Sample"],
|   |   |   |   continuous_covariate_keys=['pct_counts_mt', 'total_counts', 'pct_counts_ribo'])

```

FIGURE 3.11 – Configuration du modèle scVI

Ici, pour la configuration du modèle scVI, nous spécifions la variable "sample" comme categorical covariate key. Cela permet au modèle de reconnaître que chaque cellule est associée à un patient particulier. En intégrant cette information, scVI peut mieux contrôler les effets liés au patient (batch effects) et ainsi distinguer plus efficacement la variabilité biologique réelle de la variabilité technique. Cette étape est essentielle, car elle améliore la qualité des embeddings produits par le modèle, ce qui conduit ensuite à un clustering plus pertinent et plus fiable des cellules.

En d'autres termes, sans cette correction, le modèle risque de regrouper les cellules en fonction de leur origine (le patient ou la condition expérimentale) plutôt que de leurs véritables similarités biologiques. En prenant en compte la covariable sample, scVI apprend à "neutraliser" les différences dues à l'échantillonnage et met en avant les signaux transcriptionnels partagés, ce qui rend l'analyse plus robuste et généralisable.

3.3 Résultats préliminaires

Le modèle scVI (single-cell Variational Inference) est un cadre probabiliste conçu pour l'analyse des données single-cell RNA-seq. Basé sur des autoencodeurs variationnels (VAE) et l'inférence bayésienne, il encode chaque transcriptome cellulaire dans un espace latent, puis reconstruit l'expression génique à partir de ce vecteur.

Son principal atout est la correction intégrée des batch effects, qui représentent des variations techniques non biologiques. Plutôt que d'utiliser des modèles séparés, scVI intègre cette correction dans un cadre unique en conditionnant le codage sur des covariables (par ex. lot expérimental, patient, pourcentage mitochondrial). Cette approche permet de distinguer efficacement les effets techniques du signal biologique pertinent.

Après entraînement, scVI produit un espace latent (souvent configuré à 10 dimensions par défaut) qui capture l'essentiel de la variation biologique réelle tout en neutralisant les artefacts techniques. Cet espace est obtenu après entraînement d'un réseau de neurones qui estime les paramètres probabilistes des distributions de chaque gène par cellule.

Grâce à cette approche non linéaire, scVI s'adapte mieux à la structure complexe et bruitée des données single-cell, tout en produisant des représentations utilisables pour des analyses telles que le clustering.

Avec à cet espace latent appris, nous avons pu effectuer un clustering des cellules. Pour cela, nous avons utilisé l'algorithme de Leiden, une méthode particulièrement populaire dans le domaine de la génomique single-cell. Cet algorithme améliore le très connu algorithme de Louvain en garantissant une meilleure cohérence interne des clusters (chaque

groupe est fortement connexe) et en offrant de meilleures performances sur des graphes de grande taille. Le clustering Leiden repose sur la construction d'un graphe de k plus proches voisins (kNN) dans l'espace latent, sur lequel est appliqué un processus d'optimisation de la modularité ou de la CPM (Constant Potts Model), permettant d'identifier des communautés cellulaires pertinentes d'un point de vue structurel et biologique.

Une fois les clusters définis, nous avons projeté les données en deux dimensions à l'aide de UMAP (Uniform Manifold Approximation and Projection). UMAP, comme t-SNE, est une technique de réduction de dimension non linéaire, particulièrement bien adaptée aux données sparse et bruitées comme celles issues de la single-cell RNA-seq. Ces méthodes ne reposent pas sur des hypothèses de linéarité (contrairement à PCA) et permettent de préserver la structure locale des données, ce qui facilite la visualisation des groupes cellulaires dans un espace bidimensionnel.

Il est important de noter que ni UMAP ni t-SNE ne résolvent un problème convexe, ce qui signifie qu'ils peuvent produire des résultats légèrement différents à chaque exécution (en fonction de l'initialisation aléatoire). Néanmoins, leur capacité à préserver la structure topologique locale rend ces techniques incontournables pour l'interprétation visuelle des données single-cell.

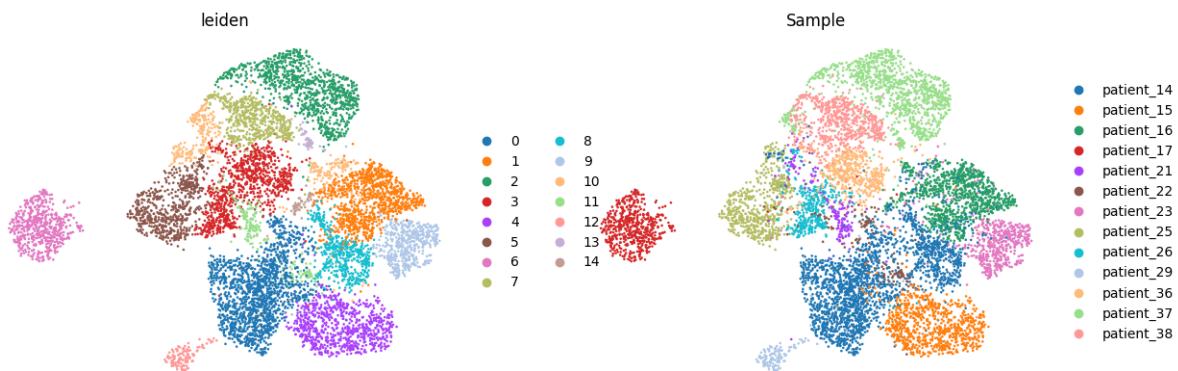


FIGURE 3.12 – Clustering Leiden/UMAP

Nous disposons désormais d'une représentation plus fidèle de nos données, d'un point de vue biologique. À droite, le clustering obtenu à l'aide de l'algorithme Leiden (avec un paramètre de résolution fixé à 0,5) permet de faire émerger des groupes cellulaires cohérents. À gauche, la projection UMAP colore chaque cellule en fonction de son patient d'origine, ce qui permet d'observer la distribution inter-patient dans l'espace latent.

Cette représentation met en évidence la granularité permise par les données single-cell : nous pouvons à la fois observer les grandes tendances globales, mais aussi nous concentrer sur un patient spécifique (par exemple le patient 17) afin de détecter des sous-populations cellulaires particulières ou des signatures tumorales potentielles.

Une fois le clustering réalisé, une étape classique consiste à annoter les groupes cellulaires. Cela implique d'identifier des gènes "marqueurs" exprimés de manière significative dans un cluster donné, et caractéristiques d'un type cellulaire spécifique. Ces gènes nous permettront, dans la suite, de relier les clusters observés à des types cellulaires connus, et d'interpréter biologiquement la composition du tissu analysé.

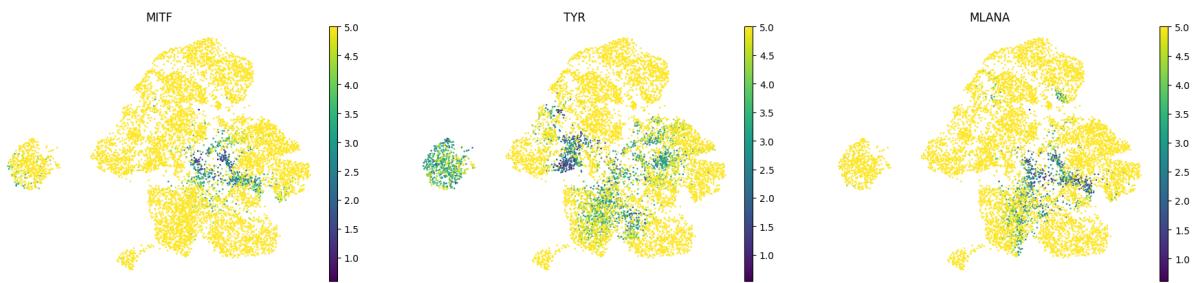


FIGURE 3.13 – Gènes marqueurs - Mélanome

Pour l'étape de labellisation des cellules, il est courant de s'appuyer sur des répertoires de gènes marqueurs déjà établis et disponibles en ligne. Ces bases de connaissances recensent, pour chaque type cellulaire, des ensembles de gènes dont l'expression est caractéristique. Elles constituent donc une ressource précieuse pour interpréter les clusters obtenus après le traitement et le clustering des données.

Dans ce travail, j'ai utilisé le site [PangaoDB](#), une base de données de référence qui propose des marqueurs cellulaires validés à partir de nombreuses expériences de transcriptomique unicellulaire. Ce site permet de rechercher des gènes associés à différents types cellulaires, facilitant ainsi la comparaison entre mes résultats et les connaissances biologiques déjà établies.

Avec les fonctions de Scanpy, il est également possible d'identifier les gènes les plus exprimés dans chaque cluster, ce qui permet d'inférer théoriquement à quel type ou groupe cellulaire il correspond. [5.4](#)

J'ai entamé l'analyse des gènes marqueurs dans le but d'annoter les groupes cellulaires issus du clustering. En m'appuyant sur trois gènes bien connus pour être associés au mélanome, j'ai pu constater que ceux-ci présentent une expression élevée dans la majorité des cellules, chez presque tous les patients. Un fait notable est que l'un de ces gènes est faiblement exprimé chez le patient 17, ce qui pourrait expliquer son éloignement observé dans la projection UMAP par rapport aux autres patients. Ce type d'analyse montre qu'avec l'identification de quelques gènes spécifiques, un expert pourrait, sans information préalable sur le jeu de données, identifier qu'il s'agit probablement de cas de mélanome.

Cependant, l'annotation fine des clusters reste une étape délicate qui demande une expertise en biologie moléculaire et en immunologie, notamment pour distinguer les différents types de cellules immunitaires ou tumorales à partir de profils d'expression génique. Même si les outils computationnels permettent de détecter les gènes différentiellement exprimés, leur interprétation biologique nécessite des connaissances approfondies que je ne possède pas encore entièrement.

Chapitre 4

Dernières analyses et résultats

Une fois ces étapes réalisées, il reste encore plusieurs approfondissements à mener, notamment concernant le clustering, les embeddings et l'application de l'algorithme MST.

En particulier, je prévois d'appliquer l'algorithme GMM (Gaussian Mixture Model) sur ce nouveau jeu de données, désormais prétraité et donc mieux adapté à des analyses robustes. Concernant les embeddings, les approches les plus couramment utilisées dans ce type de problématique sont UMAP et t-SNE, qui permettent de projeter les données dans un espace de plus faible dimension. Toutefois, ces méthodes étant non convexes, elles ne garantissent pas nécessairement une solution optimale. Même si les relations entre les cellules ne sont pas strictement linéaires, il reste pertinent d'explorer d'autres représentations, par exemple en étudiant davantage les composantes principales issues de la PCA ou encore en appliquant un spectral embedding, afin de comparer la cohérence des résultats obtenus.

Pour réaliser ces analyses, j'ai travaillé directement à partir des données prétraitées, sans utiliser l'embedding généré par le modèle scVI. Ce choix s'explique par ma volonté d'explorer d'autres méthodes de réduction de dimensionnalité, telles que l'Analyse en Composantes Principales (ACP) ou le Spectral Embedding, afin de comparer leurs performances et d'évaluer leur impact sur la qualité du clustering.

J'ai ensuite effectué plusieurs essais en variant la combinaison des étapes d'embedding et de clustering. Par exemple, j'ai comparé les résultats obtenus en appliquant un clustering directement sur les données brutes à ceux obtenus après une réduction dimensionnelle préalable par ACP. Les analyses montrent que, dans certains cas, appliquer le clustering directement sur les données brutes peut s'avérer plus robuste, car cela conserve davantage d'information, et donc conduire à de meilleurs résultats que sur les données projetées. Cependant, la réduction de dimensionnalité présente aussi l'avantage de réduire le bruit et de faciliter la visualisation, ce qui illustre l'importance de trouver un équilibre entre robustesse et lisibilité des résultats.

Après plusieurs essais, j'ai constaté qu'il était particulièrement intéressant d'examiner les clusterings à travers différents axes des embeddings. En effet, en raison de la complexité des données, certaines structures restent invisibles lorsqu'on se limite à une seule paire de composantes principales, mais deviennent plus claires lorsqu'on change de perspective. Pour répondre à ce besoin, j'ai développé une fonction permettant de générer automatiquement des visualisations pour l'ensemble des combinaisons de composantes. Cette approche facilite une exploration plus riche des données en multipliant les « angles de vue », et permet ainsi de mettre en évidence des relations ou regroupements cellulaires qui auraient pu rester cachés.

Pour analyser la structure des données et explorer les relations entre cellules, nous avons construit des graphes de type Minimum Spanning Tree (MST). Après avoir appliqué un Spectral Clustering basé sur une affinité de type nearest neighbors afin de définir les groupes cellulaires, nous avons projeté les données dans un espace réduit par PCA afin de faciliter la visualisation. À partir de la matrice de distances euclidiennes entre cellules, nous avons ensuite extrait le MST en utilisant les fonctions de scipy. Ce graphe met en évidence les connexions minimales reliant toutes les cellules, et permet d'observer la continuité et les transitions potentielles entre clusters. La combinaison de ces approches offre une représentation plus intuitive des relations entre sous-populations cellulaires et complète l'analyse obtenue par clustering classique.

L'utilisation de cette méthode d'exploration s'est révélée particulièrement efficace : elle a permis de mieux appréhender la structure sous-jacente des données et d'identifier des regroupements cellulaires de manière plus fiable. Dans ce contexte, il apparaît que le spectral embedding 5.6 semble offrir une représentation plus adaptée que la simple ACP, car il fournit des visualisations où les clusters sont plus nettement discernables 5.5.

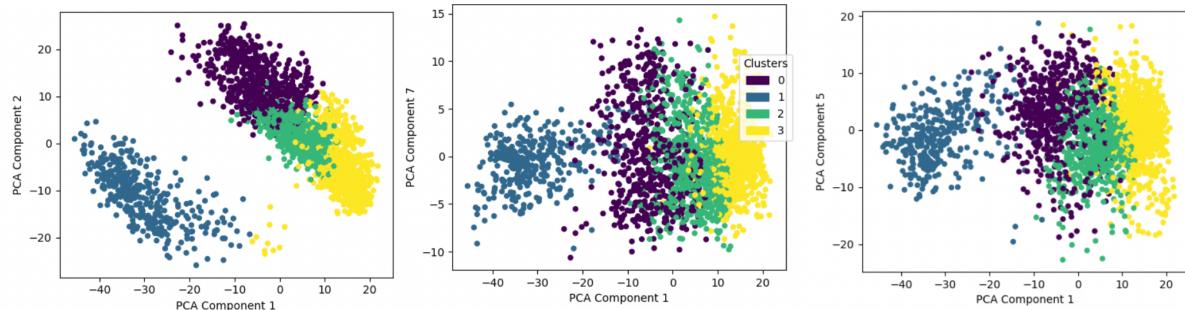


FIGURE 4.1 – PCA + GMM - patient14

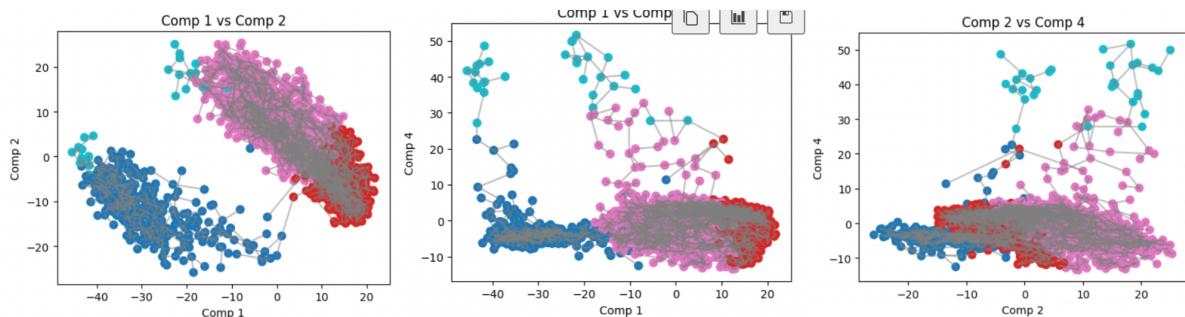


FIGURE 4.2 – PCA + GMM + MST - patient14

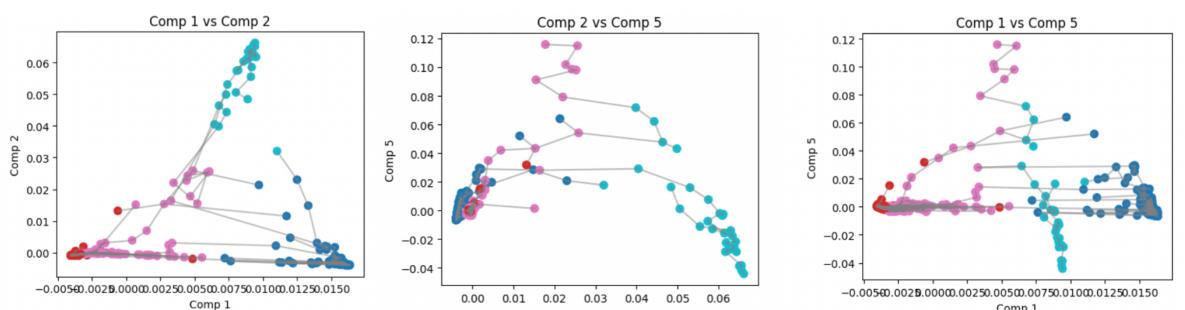


FIGURE 4.3 – Spectral Embedding + Clustering + MST - patient14

En conclusion, cette phase d'analyse montre que le choix de l'embedding joue un rôle crucial dans l'interprétation des résultats de clustering, et que l'exploration multidimensionnelle des composantes constitue un outil indispensable pour une meilleure compréhension des données. Ces constats ouvrent la voie à une discussion plus large sur l'interprétation biologique des clusters identifiés, sur les limites des méthodes employées, ainsi que sur les perspectives d'amélioration pour affiner la qualité des analyses.

4.1 Discussion et conclusion

Telles ont été les principales analyses réalisées dans ce travail. Avant de conclure, il me paraît important de revenir sur certains points critiques et sur les perspectives qui en découlent.

Tout d'abord, l'absence de contact direct avec un spécialiste du domaine de la biostatistique a représenté une difficulté au démarrage. Cela m'a demandé un temps d'adaptation important pour comprendre les spécificités des données et des outils. Néanmoins, j'ai pu surmonter cette contrainte en approfondissant l'exploration des méthodes disponibles et en documentant rigoureusement chaque étape du code. Cette documentation et la modularité du pipeline constituent, à mon sens, une contribution utile, car elles faciliteront une reprise et une extension des analyses par la suite.

Par ailleurs, le manque de temps a limité la possibilité d'explorer davantage de patients ou de comparer plus systématiquement différents algorithmes de clustering. De plus, l'application du modèle scVI s'est révélée particulièrement exigeante. En effet, ce modèle repose sur un grand nombre d'hyperparamètres dont l'optimisation est à la fois complexe et chronophage. Or, ces paramètres influencent directement la qualité de l'embedding et, par conséquent, la pertinence des résultats finaux (clustering, détection de sous-populations, etc.). Cela souligne l'importance, dans des travaux futurs, d'intégrer des approches plus systématiques d'optimisation (méthodes bayésiennes, grid search, random search), afin de garantir des résultats plus robustes et reproductibles.

Une autre piste d'amélioration concerne la possibilité de recourir à des méthodes d'annotation automatique des cellules. Celles-ci permettraient de comparer les clusters identifiés avec des types cellulaires déjà connus. Leur mise en œuvre nécessiterait toutefois l'accès à des jeux de données de référence labellisés, ce qui n'a pas pu être envisagé dans le cadre de ce travail par contrainte de temps.

Enfin, j'ai constaté que l'apprentissage des outils de la librairie Scanpy m'a demandé un temps considérable. Avec une meilleure maîtrise initiale de ces méthodes, j'aurais sans doute pu approfondir des pistes plus avancées, telles que l'utilisation de graphes pour affiner le clustering. Malgré ces limites, je considère avoir mené à bien les objectifs principaux dans le temps imparti et posé des bases solides pour des explorations ultérieures.

En conclusion, ce travail a permis de mieux comprendre les enjeux liés à l'exploration de données single-cell et de mettre en lumière l'importance des choix méthodologiques dans l'interprétation des résultats. Bien que perfectible, il constitue un point de départ pertinent pour des analyses plus poussées, intégrant à la fois une optimisation des modèles, une comparaison sur un plus grand nombre d'échantillons et une validation biologique des clusters identifiés.

Chapitre 5

Annexes

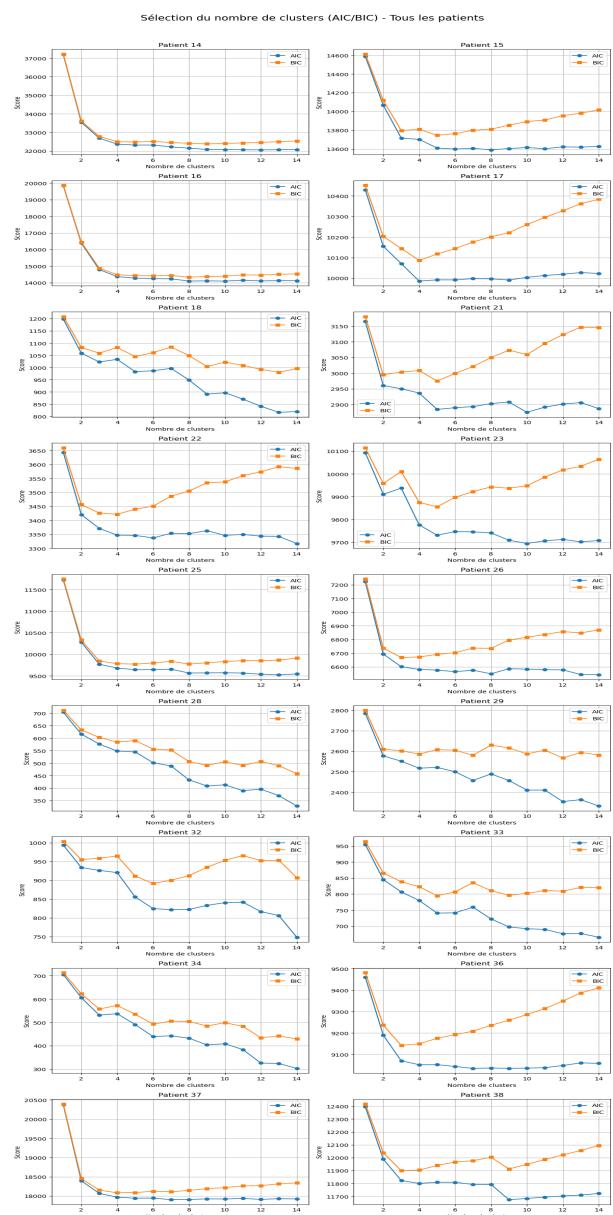
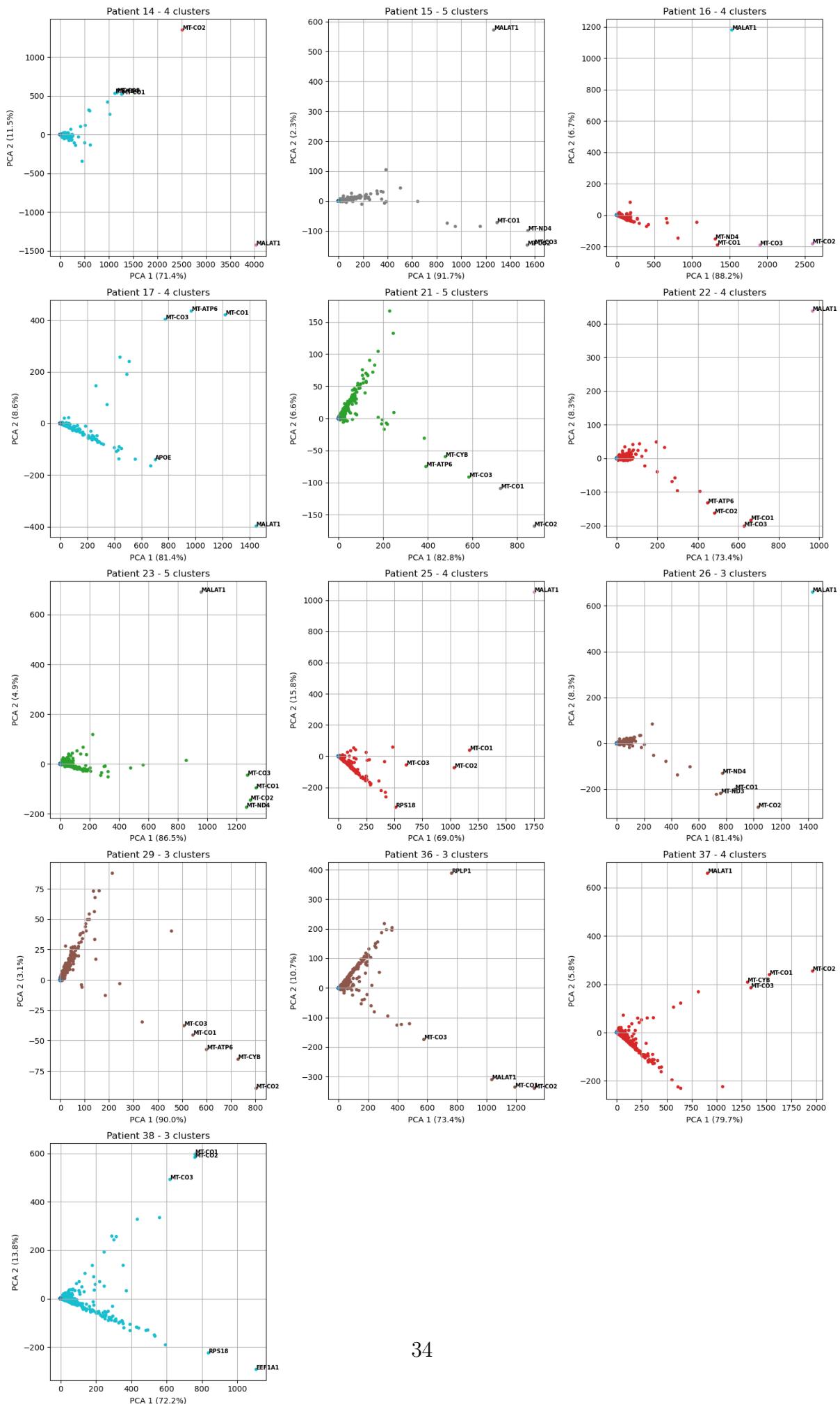


FIGURE 5.1 – Sélection du nombre de clusters



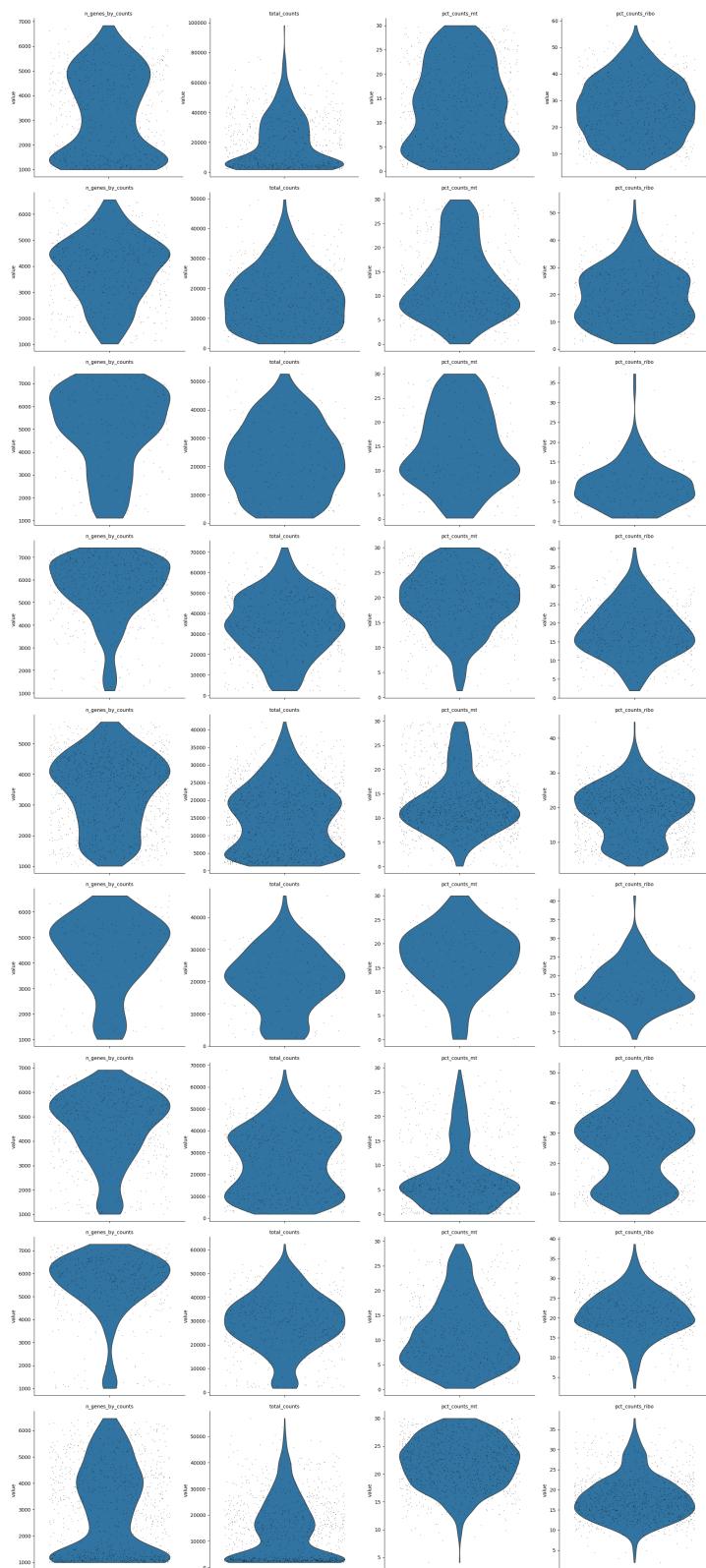


FIGURE 5.3 – Violin plots

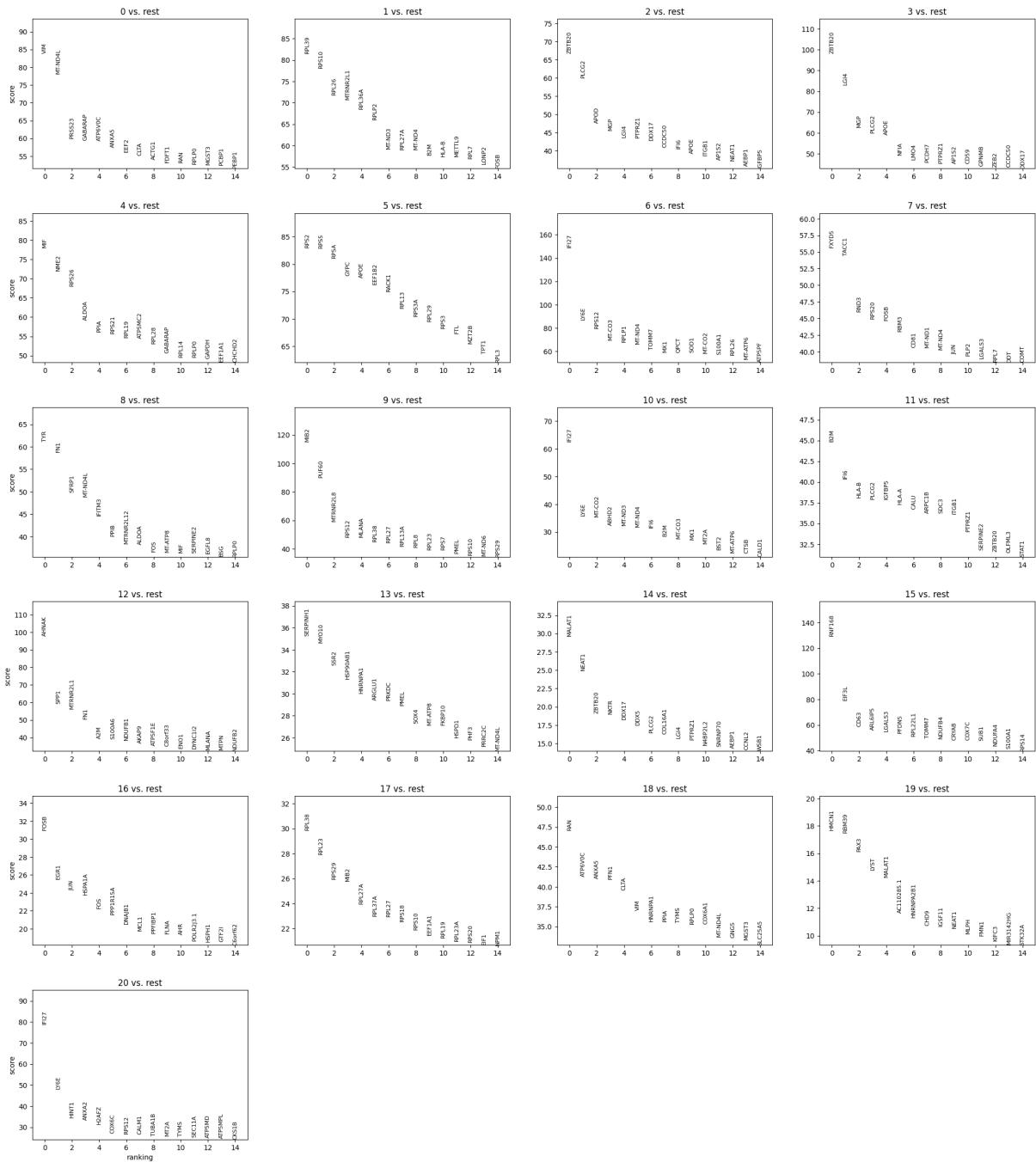


FIGURE 5.4 – Gênes exprimés par cluster

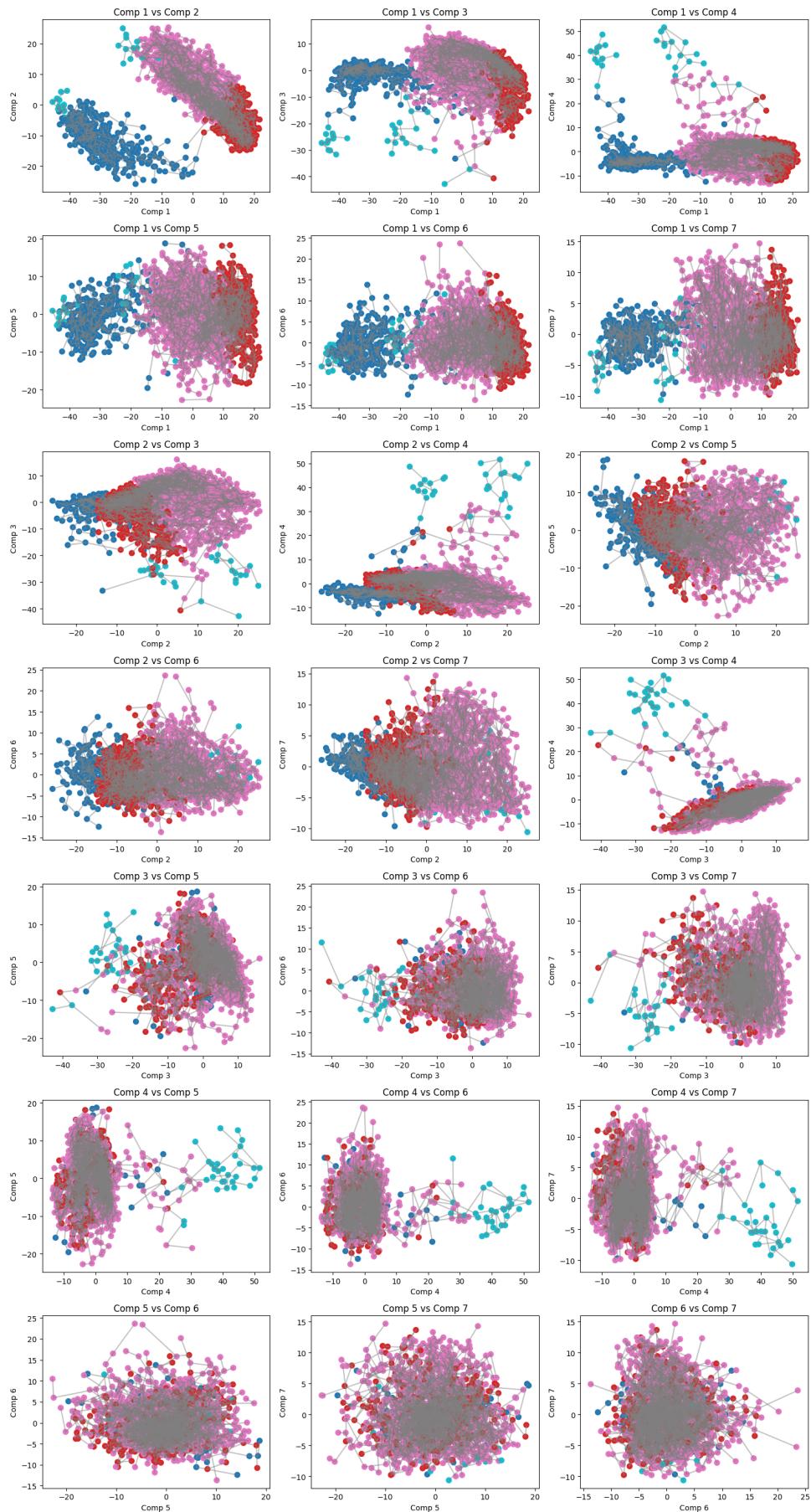


FIGURE 5.5 – pca + gmm + mst

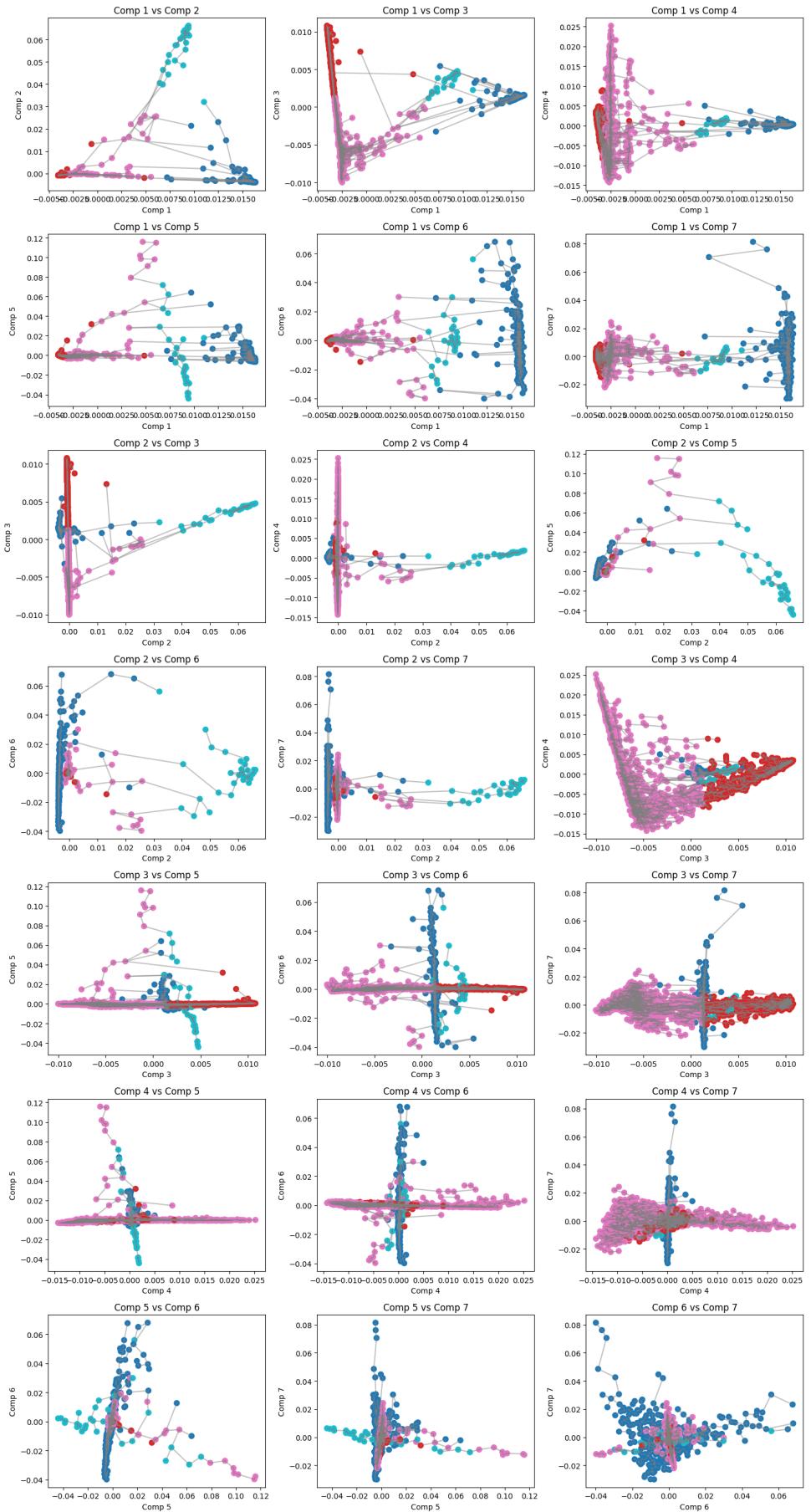


FIGURE 5.6 – spectral + embedding + mst

Bibliographie

- [1] Rahma Adjadj. Single-cell technologies mark the dawn of a new era. [article](#), 2024.
- [2] Ashraful Haque et al. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. [article](#), 2017.
- [3] Dragomirka Jovic et al. Single-cell rna sequencing technologies and applications : A brief overview. [article](#), 2022.
- [4] Nighat Noureen et al. Signature-scoring methods developed for bulk samples are not adequate for cancer single-cell rna sequencing data. [article](#), 2022.
- [5] Romain Lopez et al. Deep generative modeling for single-cell transcriptomics. [article](#), 2018.
- [6] Shunchao Bao et al. Potential of mitochondrial ribosomal genes as cancer biomarkers demonstrated by bioinformatics results. [article](#), 2022.
- [7] Tang Fuchou et al. mrna-seq whole-transcriptome analysis of a single cell. [article](#), 2009.
- [8] Zewei Zhang et al. Mitochondrial energy metabolism correlates with an immunosuppressive tumor microenvironment and poor prognosis in esophageal squamous cell carcinoma. [article](#), 2023.
- [9] Zhi-Xing Li et al. Malat1 : a potential biomarker in cancer technologies mark the dawn of a new era. [article](#), 2018.
- [10] Daniel Osorio et James J Cai. Systematic determination of the mitochondrial proportion in human and mice tissues for single-cell rna-sequencing data quality control. [article](#), 2021.
- [11] Douglas Hanahan et Robert A. Weinberg. The hallmarks of cancer. *Cell Press*, 100 :57–70, 2000.
- [12] InfoCancer. Cancérisation et mutations génétiques. [article](#), 2025.
- [13] Inserm. Épigénétique un génome, plein de possibilité! [article](#), 2015.
- [14] Inserm. Le génome, comment ça marche ? [video](#), 2017.
- [15] Bertrand Jordan. Hétérogénéité des tumeurs : l'apport du séquençage sur cellules individuelles. [lien](#), 2014.
- [16] Génétique médicale. Les notions-clés de la génétique médicale. [article](#).
- [17] Open source. Single-cell analysis in python. [site](#).