Paris-Saclay university
Master 2 Data science

Data-camp project
Single-cell RNA-seq classification

Realized by : Data Saiyentist

Supervised by :
Dr. Nicolas Jouvin

University year : 2022 / 2023

Contents

List of Figures

1 Abstract

Single-cell RNA sequencing (scRNA-seq) has enabled the measurement of gene expression at the scale of individual cells, providing insights into the specialization of cells with different biological functions. In this data challenge, we aim to classify cell types using scRNA-seq data from the scMARK benchmark dataset [1]. Our focus is on a subset of the data consisting of four cell types: `Cancer cells`, `NK cells`, `Tcells CD4+`, and `Tcells CD8+`.
We present a supervised classification approach using machine learning algorithms to accurately classify the cell types based on gene expression profiles. Our approach leverages feature selection, data normalization, and outlier removal techniques to improve the accuracy of the classification. The results demonstrate the effectiveness of our approach in classifying cell types using scRNA-seq data. All the programming and computing will be done in python.

Key words: *single-cell RNA sequencing, scRNA-seq, gene expression, cell type classification, machine learning, quality control, feature selection, data normalization, dimension reduction.*

2 Introduction

La classification des types de cellules est d'un intérêt primordial pour de nombreuses applications biologiques et médicales, car elle peut fournir des informations sur le diagnostic des maladies. Nous proposons une approche d'apprentissage automatique supervisée pour étudier les types de cellules et leurs fonctions biologiques en utilisant des données de séquençage d'ARN à cellule unique (**scRNA-seq**). En fait, les données scRNA-seq mesurent le nombre de molécules d'ARN à l'intérieur de chaque cellule d'un échantillon donné. Cette information fournit un instantané du transcriptome (ie. les gènes qui étaient en cours de transcription) au moment où les cellules ont été récoltées. Par rapport à ce que nous avons mentionné en début d'introduction, il est vrai qu'on devrait plutôt étudier des protéines (qui sont le produit final de l'expression d'un gène). Pour autant, la détection de son ARN messager (ARNm) indique que le gène est activé et a donc le potentiel d'être ensuite traduit et exprimé.
Un exemple concret d'utilisation peut être d'identifier des sous-populations de cellules dont la transcription est anormale, telles que les cellules malignes.

Ensuite, il faut savoir que ce projet fait l'objet d'un data challenge. En effet, les performances de nos modèles sont évaluées avec l'outil Ramp studio mis en place par l'université de Paris-Saclay. Chaque groupe peut soumettre différentes modélisations puis obtenir des scores de performance, dont l'un d'eux est uniquement accessible à notre superviseur de projet. L'objectif d'un tel projet est d'évaluer nos compétences à :

- travailler et s'organiser en groupe
- comprendre des techniques pré-existantes et à proposer de nouveaux concepts
- résoudre des problèmes complexes et hiérarchiser les tâches
- s'adapter à un nouvel environnement de travail (dans notre cas, au workflow de Github)

Enfin, notre projet est accessible sur le Github suivant
https://github.com/DataSaiyentist/DataCamp_scRNAseq. Il respecte les contraintes du challenge, dont les limitations de librairies Python et le formatage des fichiers.

3 Exploration des données

Nous disposons d'un jeu de données scRNA-seq extraits du benchmark scMARK [1] de Mendonca et al. Il contient plus précisément 1500 échantillons, séparés en 1000 échantillons dans notre jeu de données d'entraînement et 500 dans notre jeu de données test. Concernant les colonnes, nous avons 13551 colonnes. D'ailleurs, il faut savoir que notre jeu de données est en fait limité à la classification de quatre types de cellules :

- `Cancer_cells`
- `NK_cells`
- `T_cells_CD4+`
- `T_cells_CD8+`

NB : Nous appellerons parfois abusivement, "gènes" les colonnes de notre dataset.

Concernant la répartition de ces labels dans notre jeu de données, nous avons environ la même proportion de `T_cells_CD4+` et de `T_cells_CD8+` (34% chacun), puis 24% de `Cancer_cells` et 8% de `NK_cells` (notre jeu de données est disproportionné et il faudra s'y adapter).

3.1 Visualisation des lignes et des colonnes

Visualisons quelques lignes pour comprendre ce qui nous attend :

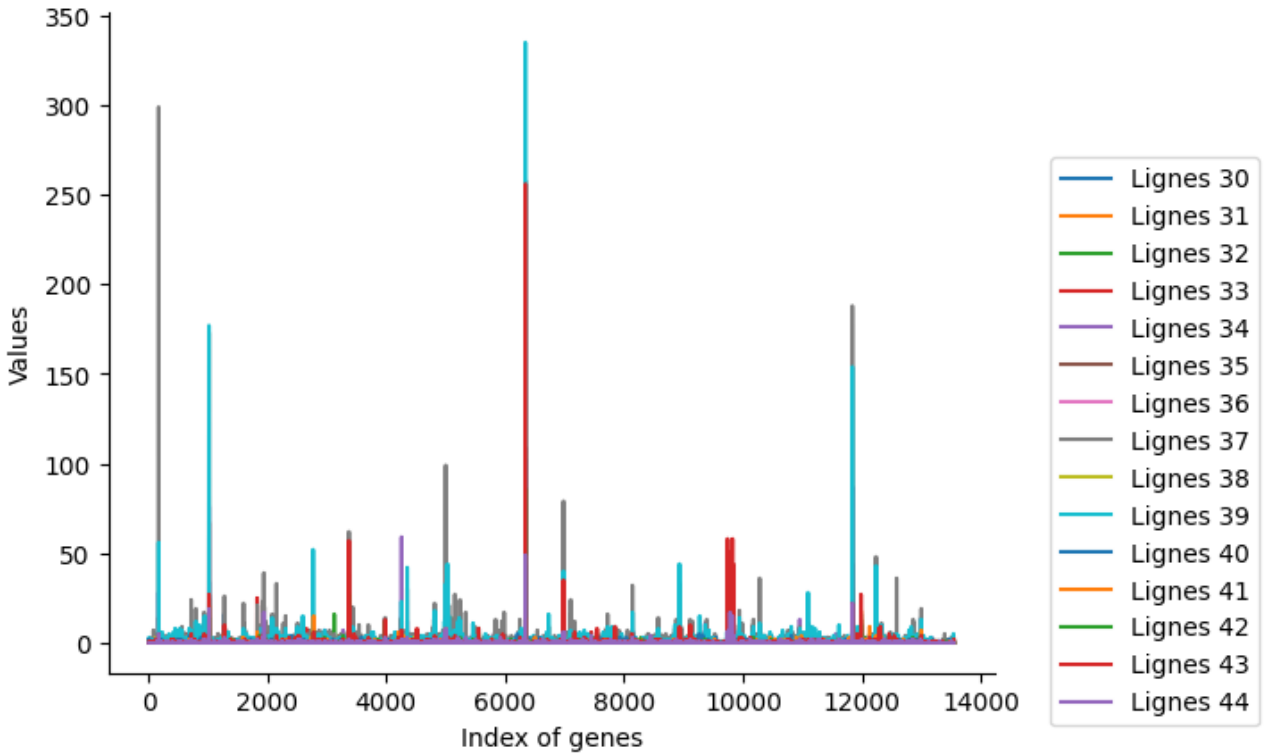

Figure 1: Visualisation de certaines lignes (1)

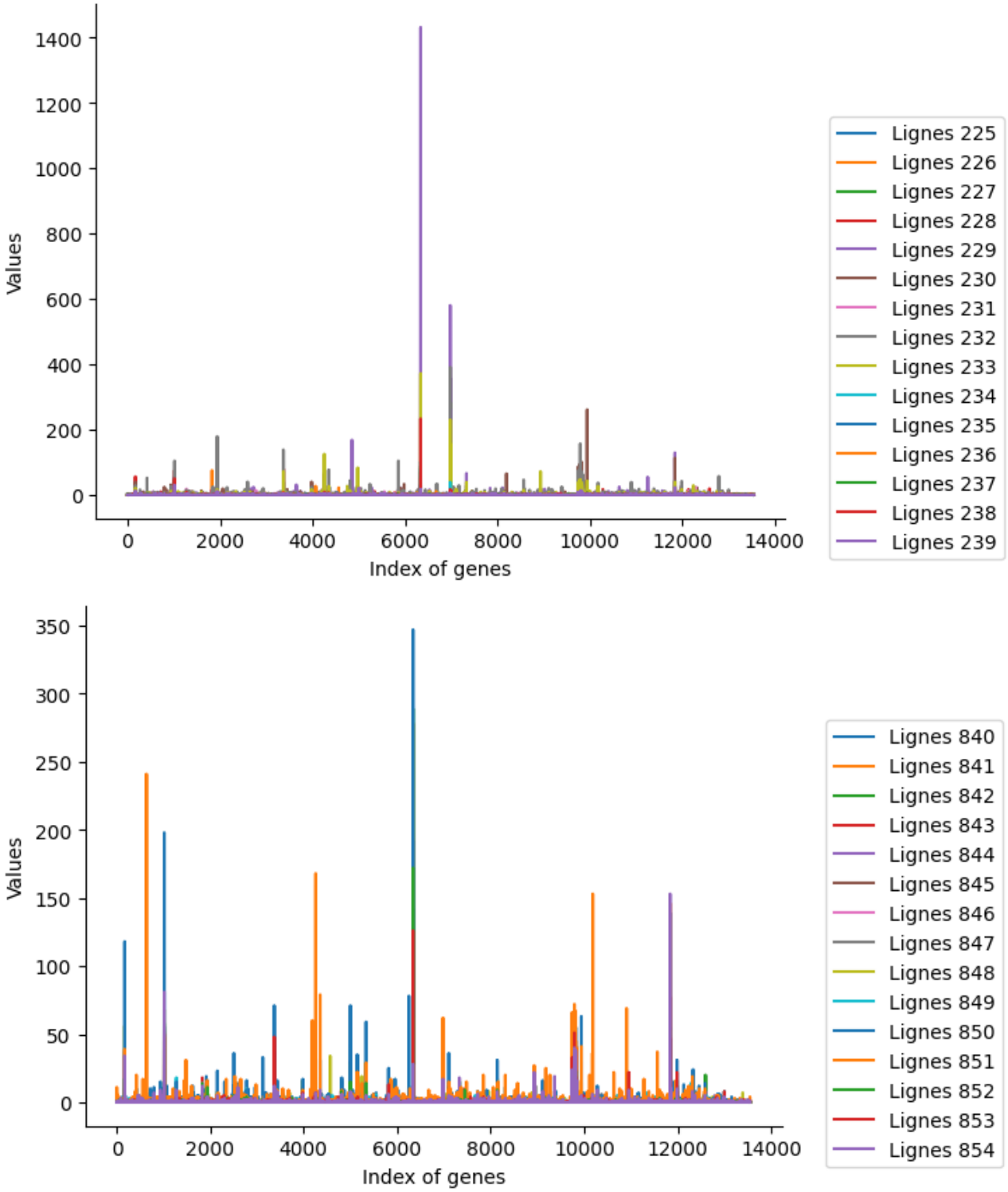

Figure 2: Visualisation de certaines lignes (2)

Nous remarquons plusieurs choses. Il est d'abord difficile d'extraire des tendances à partir des gènes. En plus, certains des gènes s'expriment peu ou quasiment tout le temps (empêchant ainsi de potentiellement bien distinguer des types de cellules). Enfin, les valeurs sont anormalement réparties, ie. nous avons beaucoup de 0 et quelques fois de très grosses valeurs.

Essayons à présent de confirmer nos dires sur les gènes vis-à-vis des types de cellules :

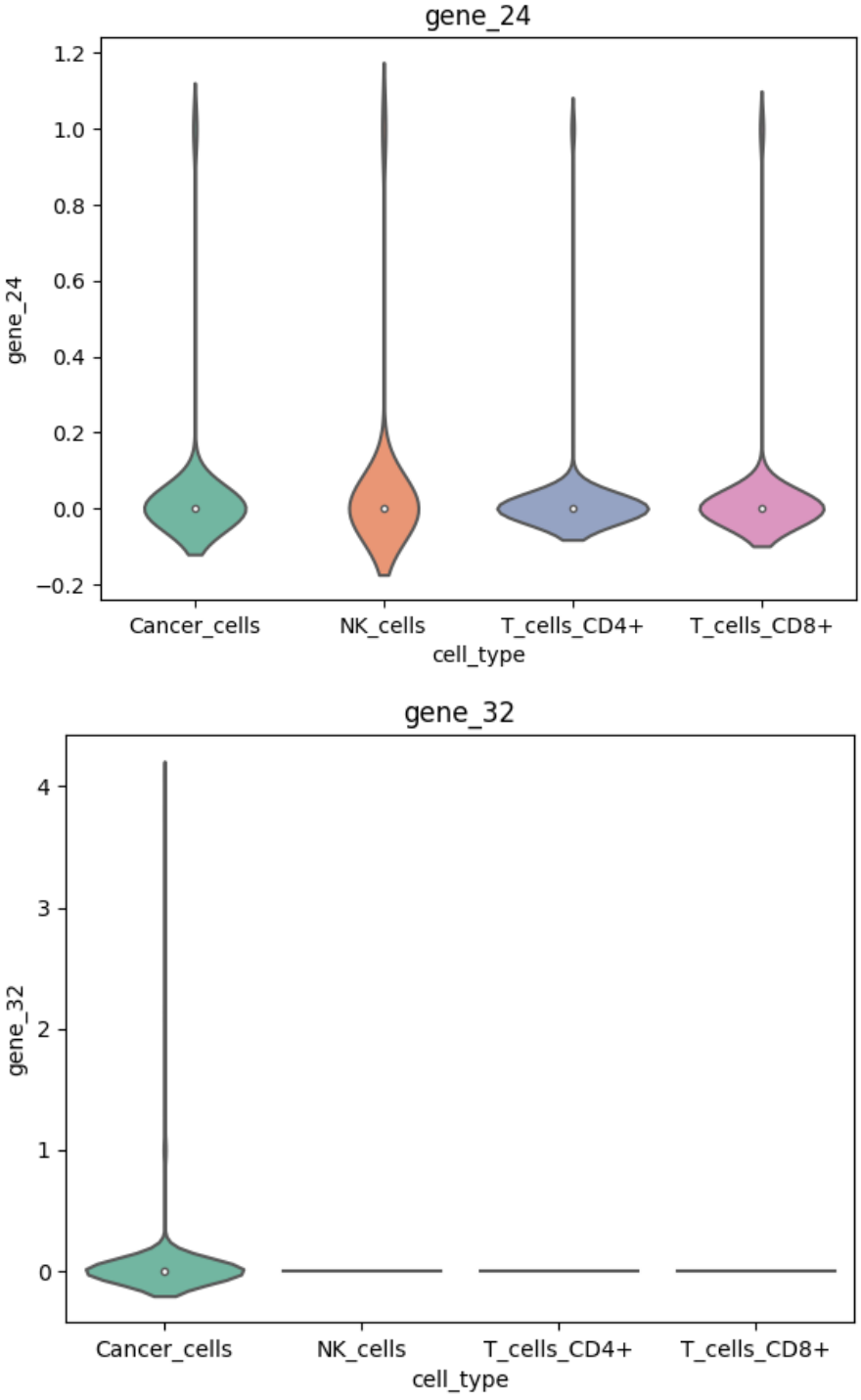

Figure 3: Visualisation de certaines colonnes

Nos propos semblent être confirmés, à savoir que certains gènes s'expriment uniformément parfois pour les 4 types de cellules. Il faudra donc les retirer lors du traitement des données.

3.2 Visualisation des données avec réduction de dimension

Lorsque nous travaillons avec des données, il peut être difficile de les visualiser et de les interpréter en tant qu'humain. Cela est particulièrement vrai lorsque nous travaillons avec des données de haute dimensionnalité, qui peuvent comporter des milliers, voire des millions, de variables. Dans ce cas, il peut être nécessaire d'utiliser des techniques de réduction de dimensionnalité, telles que l'Analyse en Composantes Principales (ACP), pour pouvoir visualiser les données plus facilement :

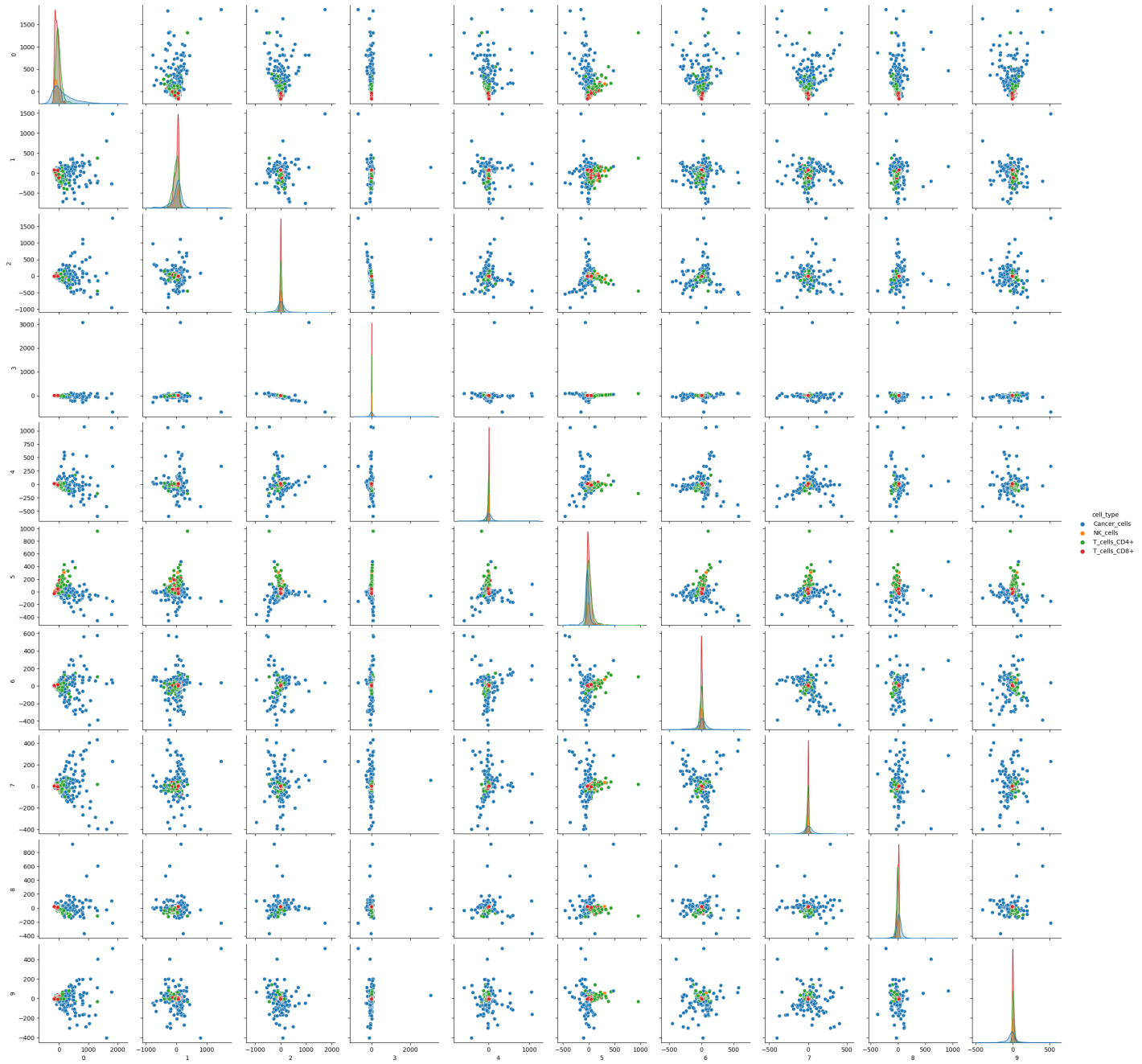

Figure 4: Pairplot des composantes de l'ACP

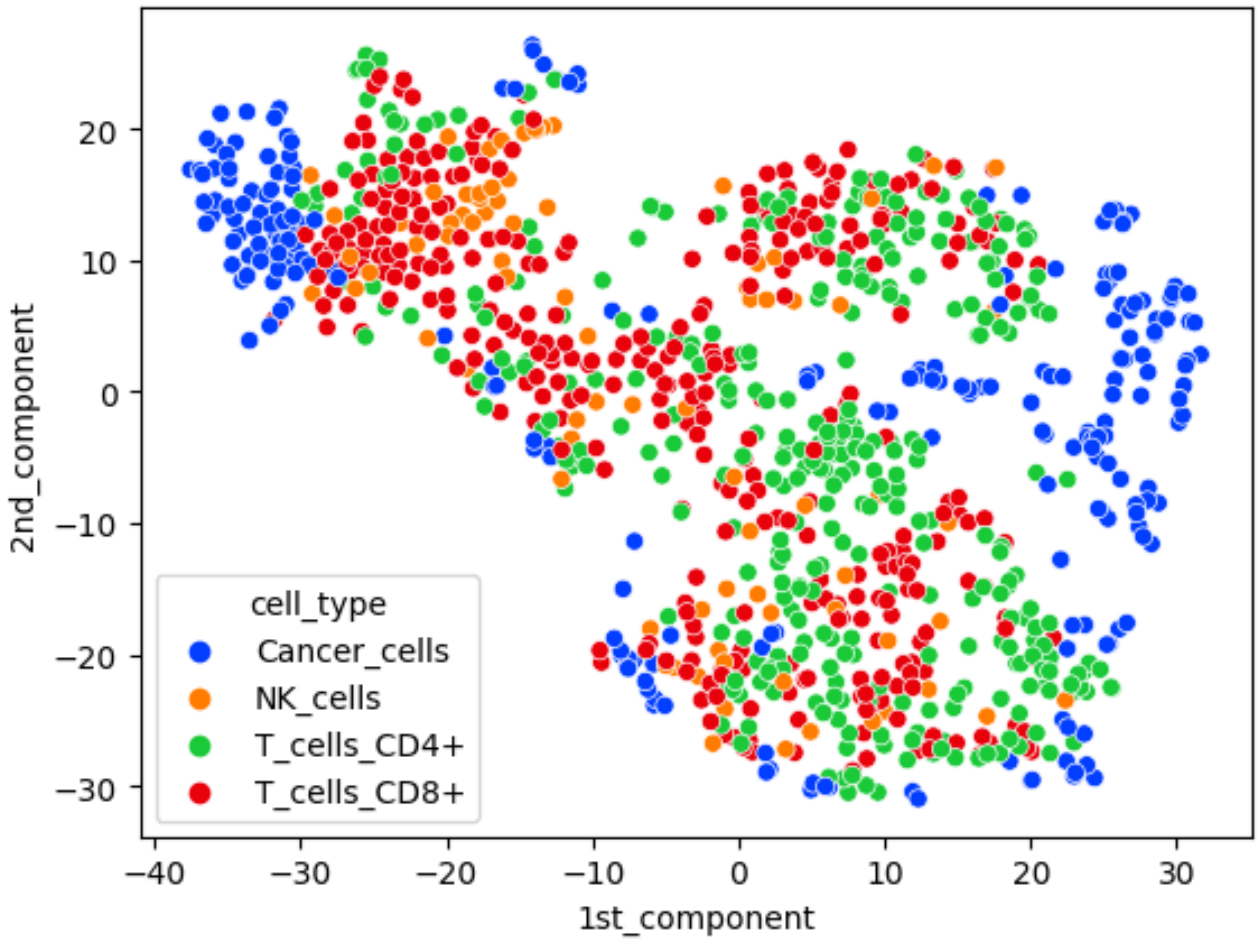

Figure 5: Visualisation de train avec un t-SNE

Nous remarquons que même avec des techniques de réduction de dimensionnalité telles que l'ACP ou le t-SNE, il nous est difficile de distinguer les différents types de cellules. Si nous regardons de plus près nos réductions de dimension, alors nous remarquons que certains types sont imbriqués les uns sur les autres, entre autres les `T_cells_CD4+` et les `T_cells_CD8+`.

Par conséquent, cela nous indique que la tâche de classification sera difficile. Dans de tels cas, nous devrons alors nous adapter à nos données et ainsi considérer :

- des stratégies de traitement de données pour mieux distinguer les types de cellules
- des méthodes de classification puissantes

4 Prétraitement des données

4.1 Qualité contrôle

En raison de la nature sensible des données de scRNA-seq, il est essentiel de procéder à un contrôle de qualité rigoureux pour identifier et éliminer les lignes/ cellules qui pourraient biaiser les résultats de notre classifieur. En effet, les bibliothèques de mauvaise qualité dans les données de scRNA-seq peuvent provenir de diverses sources telles que des dommages aux cellules lors de la dissociation ou des échecs dans la préparation des bibliothèques.

Lors de cette étape de qualité contrôle, nous avons testé différents critères sur les cellules :

- le nombre total de gènes qui se sont exprimés
- le nombre total de gènes différents qui se sont exprimés
- le call rate, ie. le pourcentage de gènes exprimés
- le novelty score, ie. le logarithme du call rate

Si le critère ne dépasse pas un certain seuil, alors on supprimait la cellule. Cependant, nous avons remarqué que les résultats n'augmentaient pas significativement avec cette approche. Nous avons donc conservé l'ensemble des cellules lors de l'entraînement des modèles.

NB : La recherche d'une meilleure normalisation n'a pas été traité dans ce projet. Nous avons simplement considéré une normalisation par rapport à chaque ligne, puis une standardisation.

4.2 Choix des techniques de sélection de variables

La sélection de variables est une étape importante pour réduire la dimensionnalité des données scRNA-seq. Elle consiste à identifier les variables les plus informatives pour la classification des cellules, tout en éliminant les variables redondantes ou non pertinentes.
Dans notre projet, plusieurs techniques de sélection de variables ont été utilisées. D'une part, nous avons implémenté des techniques de réduction de dimension, les principales étant :

- L'**ACP** qui est une méthode de transformation linéaire qui permet de projeter les données sur un nombre réduit d'axes principaux, appelés composantes principales. Les composantes principales sont choisies de manière à maximiser la variance des données projetées tout en minimisant leur corrélation. Ainsi, l'ACP permet de réduire la dimensionnalité des données tout en préservant l'information.
- **t-SNE** (t-Distributed Stochastic Neighbor Embedding), une méthode non-linéaire de réduction de la dimensionnalité qui est souvent utilisée pour la visualisation des données scRNA-seq. t-SNE est basé sur la similarité entre les points dans un espace de haute dimension et dans un espace de basse dimension. Il cherche à préserver la similarité locale entre les points tout en diminuant la similarité globale.

NB : À titre illustratif, pour notre culture générale, nous nous sommes amusés à parfois tester d'autres méthodes de réductions disponibles dans la librairie `scikit-learn` comme `KernelPCA`, `IncrementalPCA`, `FastICA`, `TruncatedSVD`, etc.

D'autre part, nous avons testé des méthodes plus avancées pour la sélection comme :

- **SelectKBest**, une méthode de sélection de variables univariée qui évalue chaque variable indépendamment en fonction de sa corrélation avec la variable cible (labels). Elle utilise une mesure de test statistique pour évaluer l'importance de chaque variable et sélectionne les K variables les plus importantes. Cette méthode est simple et rapide, et peut être utilisée avec différents tests statistiques. Dans notre cas, considérer une telle méthode permet notamment d'éviter de réduire le nombre de gènes si besoin.
- la **Réduction de la Variance** utilisée couramment pour la réduction de dimension. Cette méthode consiste à supprimer les variables ayant une variance inférieure à un seuil donné. Elle est particulièrement utile pour éliminer les variables bruyantes ou peu informatives (comme celles indiquées à la section **3.1**). Ayant appliqué cette méthode, nous avons été surpris par les résultats donnés qui sont généralement les meilleurs.

Au cours de notre étude, nous avons exploré différentes techniques de sélection de variables pour réduire la dimensionnalité des données scRNA-seq. Pour cela, nous avons comparé les performances des différentes combinaisons de techniques possibles en termes de **précision pondérée** du modèle. Nous avons alors constaté que la meilleure combinaison de techniques pour améliorer les performances du modèle était d'utiliser la méthode de réduction de la variance avec SelectKBest. Cette combinaison a ainsi permis de réduire la dimensionnalité des données **tout en préservant les variables les plus informatives** pour la classification des cellules.

5 Modélisation

Après traitement, nos données sont prêtes pour la classification. Pour ce faire, nous avons utilisé plusieurs algorithmes de classification en faisant également une hyperparamétrisation grâce à la fonction **GridSearch** de `scikit-learn`. Parmi ces algorithmes mis en œuvre :

- Le **SVM** (Support Vector Machines) qui permet de trouver une frontière de décision qui sépare les différentes classes de données. L'objectif est de trouver la frontière de décision qui maximise la marge, c'est-à-dire la distance entre la frontière de décision et les points les plus proches de chaque classe.
- Le **KNN** (K-Nearest Neighbors) qui consiste à trouver les k échantillons d'entraînement les plus proches de l'échantillon à prédire, puis à lui attribuer la classe la plus présente parmi les k échantillons comme prédiction.
- Le **Random Forest** qui crée une multitude d'arbres de décision, chacune basée sur un sous-ensemble de caractéristiques sélectionnées aléatoirement à partir des données d'entrée de l'ordre de ($\sqrt{p}$ où p est le nombre de caractéristiques dans la dataset).
- Le **Gradient Boosting** qui combine plusieurs modèles d'arbre de décision faibles pour former un modèle prédictif plus précis. En itérant un nombre fixe de fois, chaque boost tente de corriger les erreurs du modèle précédent en ajoutant une nouvelle estimation modifiée grâce à une descente de gradient.
- Le **MLP** (perceptron multicouche) qui s'inspire de la structure du cerveau humain. Il utilise une approche de "boîte noire" pour modéliser des relations non-linéaires complexes entre les variables et les variables cibles, en utilisant plusieurs couches de neurones (ou unité de calcul) interconnectées.

6 Résultats

Dans cette dernière section, nous allons vous présenter les performances de classification pour les différents modèles testés, et par la même occasion analyser/ comparer leurs performances.

6.1 Évaluation des modèles

Après entraînement avec le jeu d'entraînement, nous évaluons chaque modèle avec le jeu de données test grâce à différents outils statistiques afin d'analyser les performances :

- La **précision pondérée** prennant en compte la disproportion des labels dans nos données.
- La **matrice de confusion**, donnant une représentation tabulaire des résultats. Elle permet en fait de comparer les prédictions du modèle avec les labels réels des données.
- La **courbe ROC** montrant la relation entre le taux de vrais positifs et le taux de faux positifs à différents seuils de classification. Mais ayant plusieurs classes, nous avons construit une courbe par classe (en traitant à chaque fois une classe comme la classe positive et le reste comme la classe négative).

NB : Pour visualiser intuitivement les prédictions de chaque modèle, nous visualiserons celles-ci avec un t-SNE. Plus bas, pour pouvoir comparer, vous retrouverez une visualisation avec les vrais labels du jeu de données test.

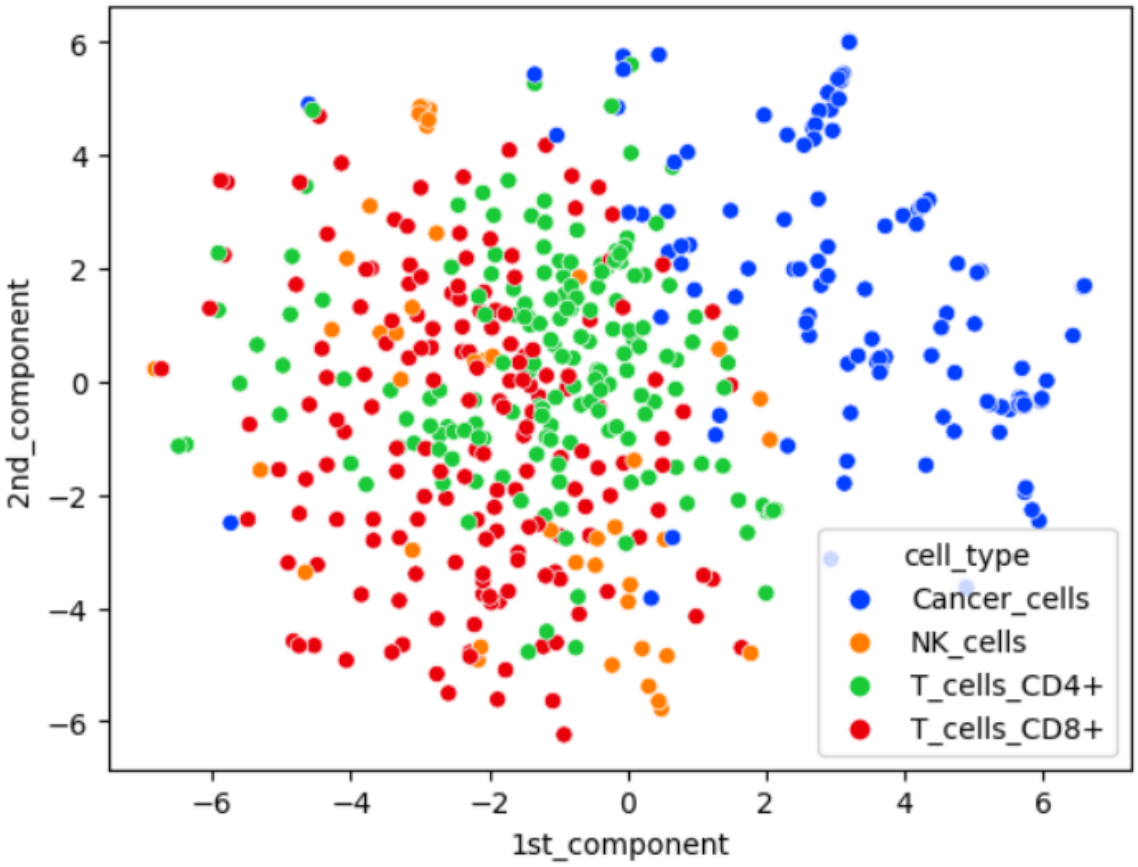

Figure 6: Visualisation de test avec un t-SNE

6.1.1 SVM

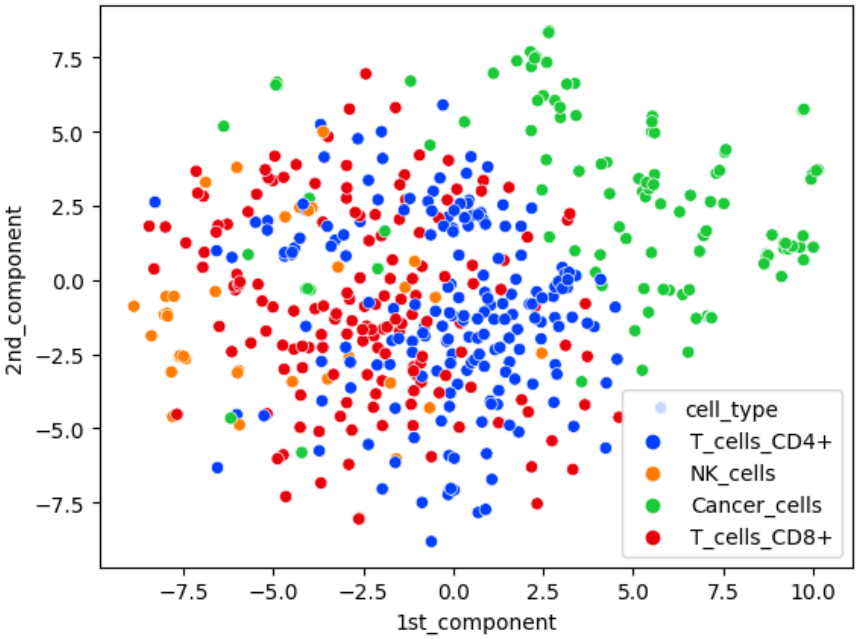

Figure 7: t-SNE avec les prédictions du SVM

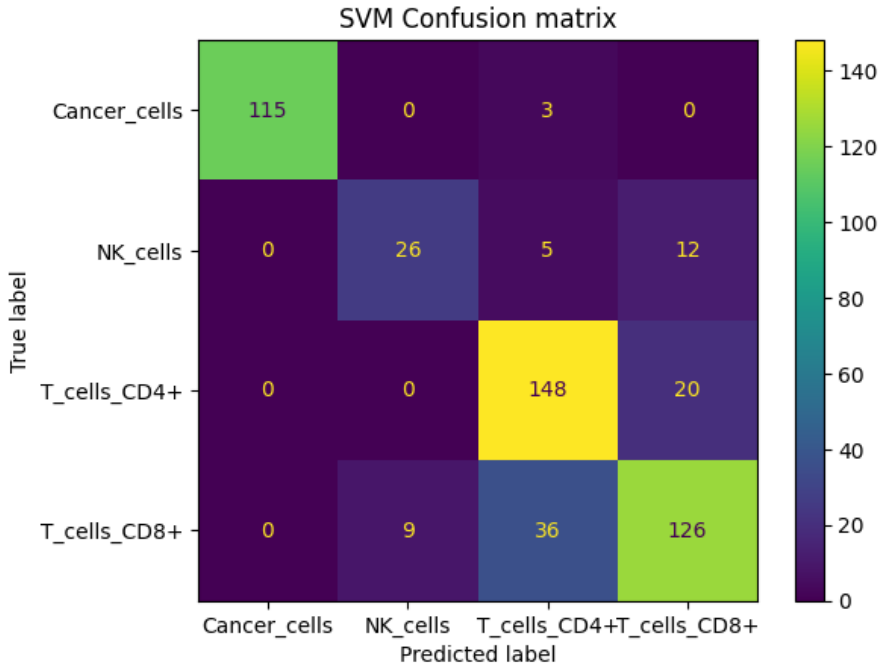

Figure 8: Matrice de confusion du SVM

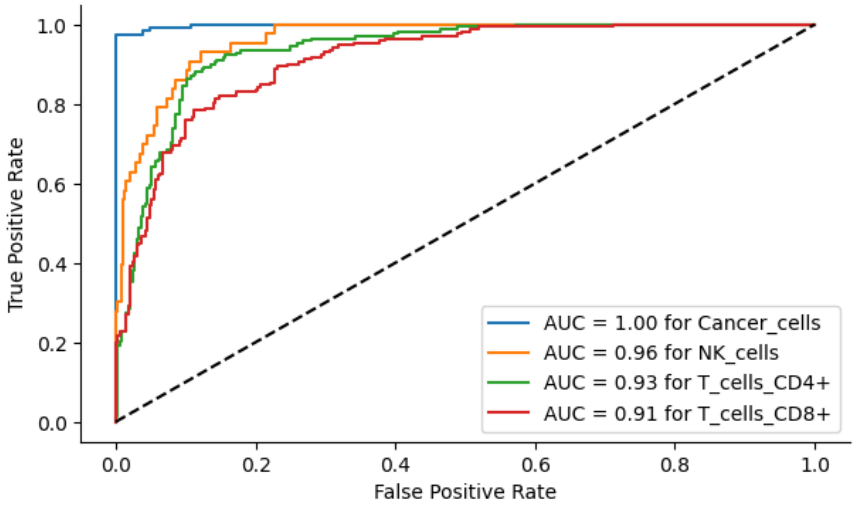

Figure 9: Courbe ROC du SVM

Notre SVM a pu prédire de manière assez forte la plupart des types de cellules. En effet, nous avons obtenu une précision pondérée de 0.80. Maintenant, si nous regardons en détail la matrice de confusion associée, alors notre modèle semble avoir quelques difficultés à distinguer `NK_cells`, `T_cells_CD4+` et `T_cells_CD8+` (ceci est compréhensible lorsqu'on se réfère à la Figure 6). Notre remarque concernant la difficulté de classifier est d'ailleurs renforcée par la courbe ROC. En effet, plus la courbe ROC se rapproche du coin supérieur gauche, plus les performances du modèle sont élevées. Or, on remarque que ce n'est pas vraiment le cas pour les 3 classes mentionnées précédemment.

6.1.2 KNN

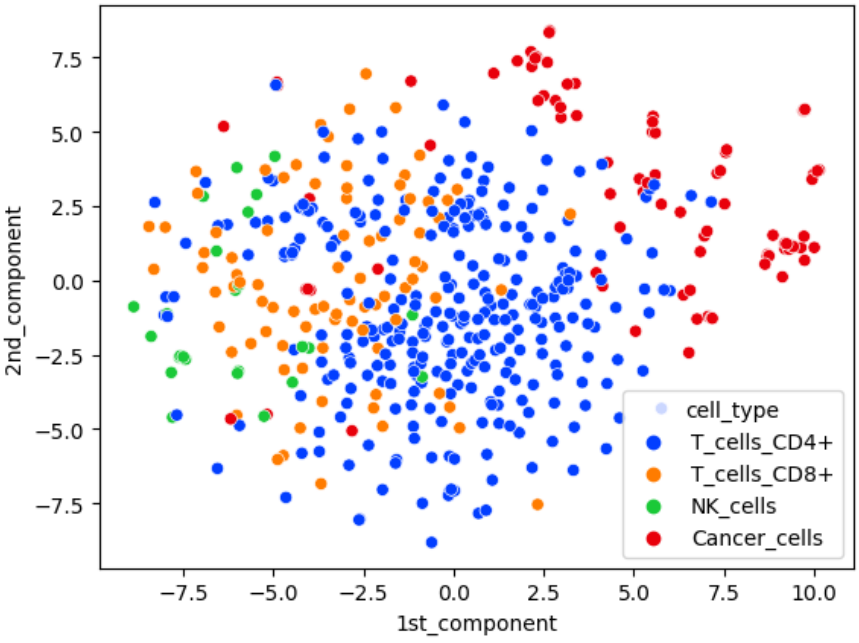

Figure 10: t-SNE avec les prédictions du KNN

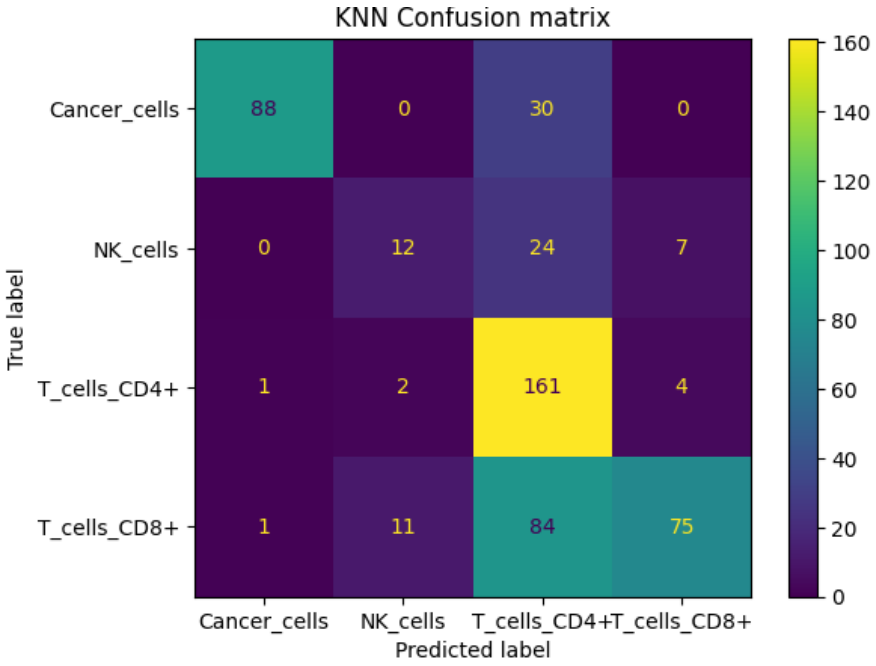

Figure 11: Matrice de confusion du KNN

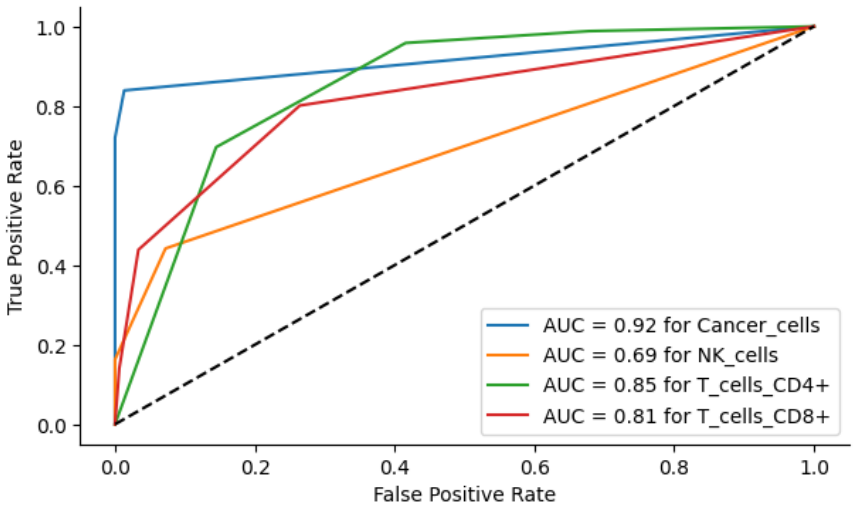

Figure 12: Courbe ROC du KNN

Notre KNN a obtenu les performances les moins satisfaisantes parmi les différents modèles testés. Cela peut être dû à l'incompatibilité de l'architecture de KNN avec les caractéristiques des données scRNA-seq. En effet, les données scRNA-seq sont hautement dimensionnelles et présentent une structure complexe de corrélation entre les variables, ce qui peut rendre difficile l'utilisation de KNN pour la classification, d'où une précision pondérée de 0.61 seulement.
Ces résultats soulignent l'importance de choisir un modèle plus approprié pour les données scRNA-seq. D'autres modèles plus adaptés, tels que les réseaux de neurones ou les méthodes de classification ensemblistes, peuvent être utilisés pour obtenir de meilleures performances de classification sur les données scRNA-seq, et c'est ce qu'on va voir dans les prochains modèles.

6.1.3 Random Forest

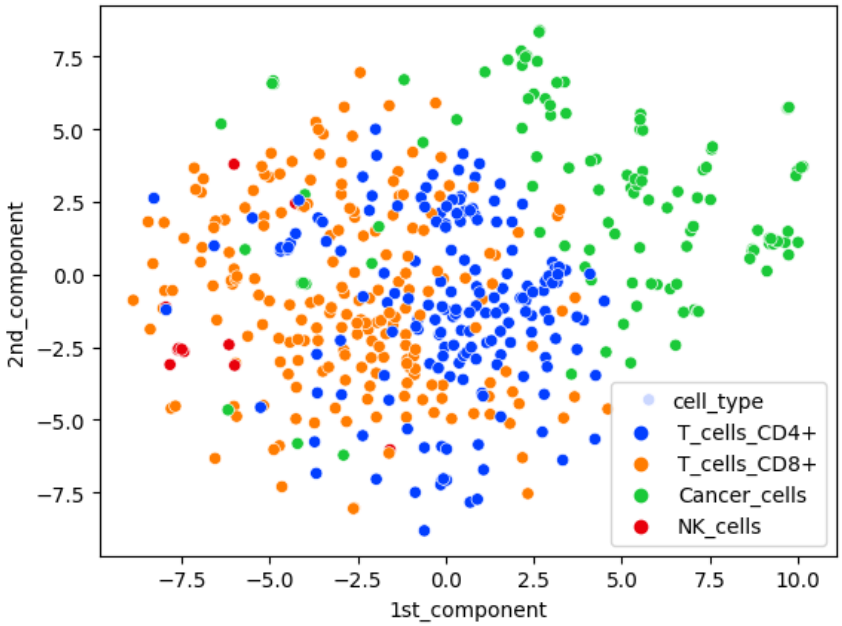

Figure 13: t-SNE avec les prédictions du Random Forest

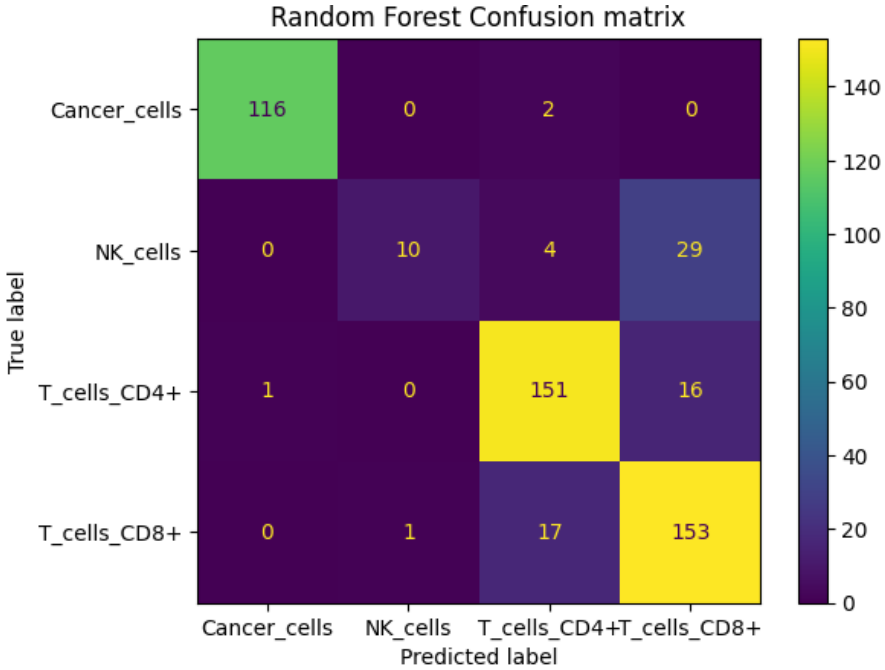

Figure 14: Matrice de confusion du Random Forest

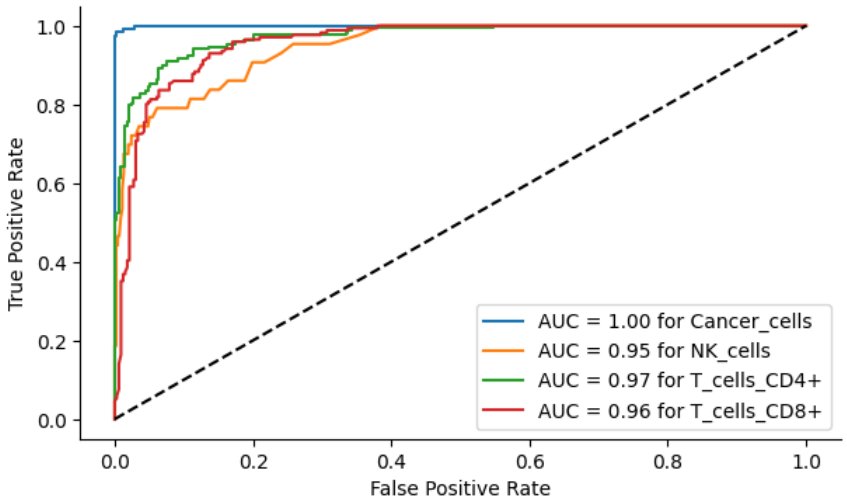

Figure 15: Courbe ROC du Random Forest

Les remarques sont assez semblables aux résultats du SVM, mis à part que notre Random Forest est moins bon que celui-ci puisqu'il obtient une précision pondérée de 0.75. Cette contre-performance est probablement due à ses erreurs fréquentes à mal prédire les `NK_cells` (cf. la Figure 14 où seules 10 cellules sont bien prédites sur les 43 cellules).

6.1.4 GradientBoosting

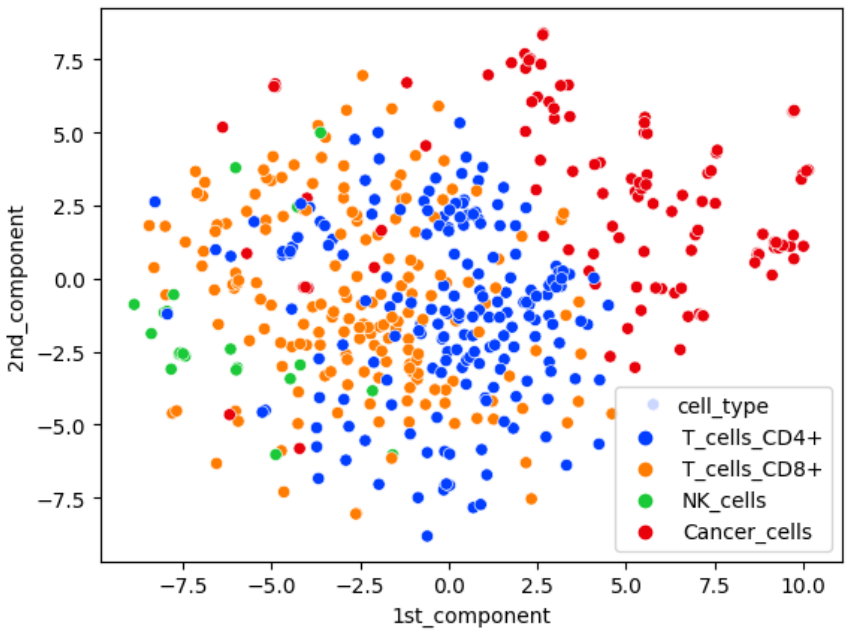

Figure 16: t-SNE avec les prédictions du Gradient Boosting

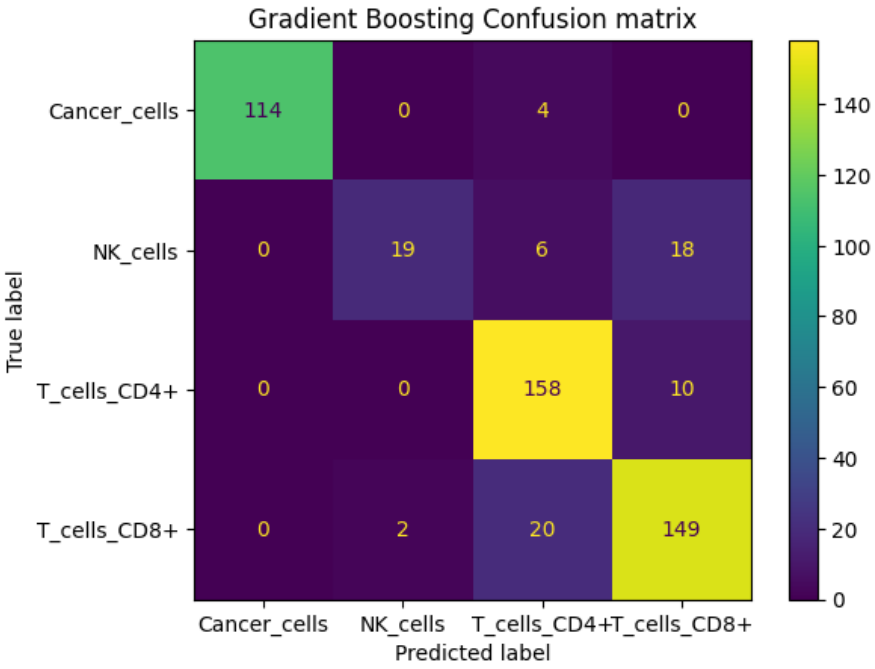

Figure 17: Matrice de confusion du Gradient Boosting

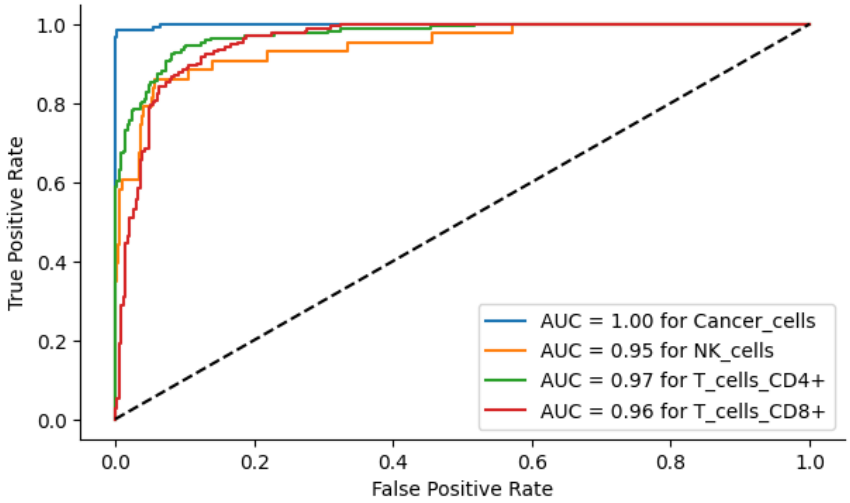

Figure 18: Courbe ROC du Gradient Boosting

Notre Gradient Boosting a obtenu une précision pondérée de 0.8, ce qui est tout aussi remarquable que le SVM. Mais nous remarquons que la courbe ROC du Gradient Boosting est meilleur que celui du SVM (avec de meilleurs AUCs pour toutes les types de cellules).
Et concernant la matrice de confusion, on remarque qu'il arrive à mieux distinguer les `T_cells_CD4+` et `T_cells_CD8+` (mais au détriment des`NK_cells`).

Une extension de notre travail pourrait être de combiner les résultats du SVM et du Gradient Boosting pour pallier les défauts de prédiction de chacun de ces deux modèles.

6.1.5 MLP

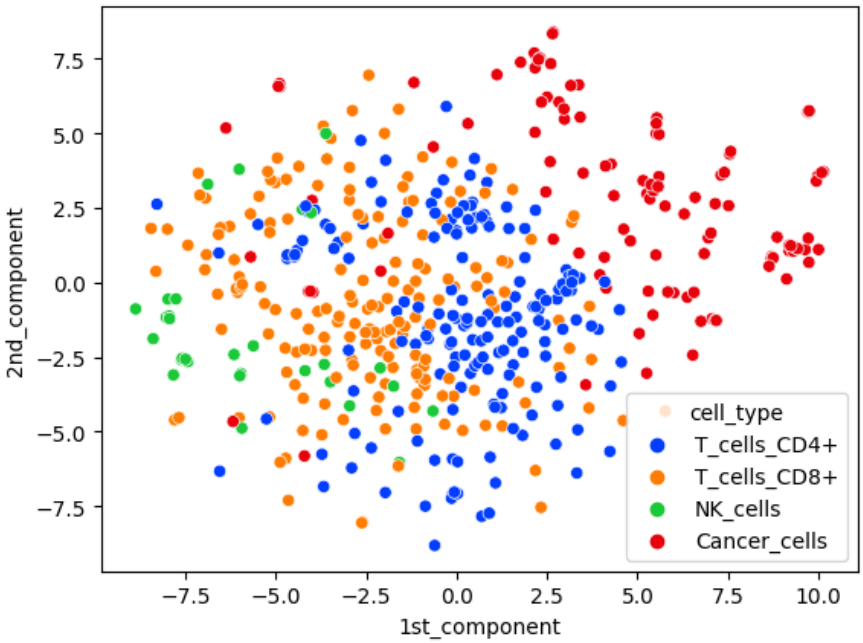

Figure 19: t-SNE avec les prédictions du MLP

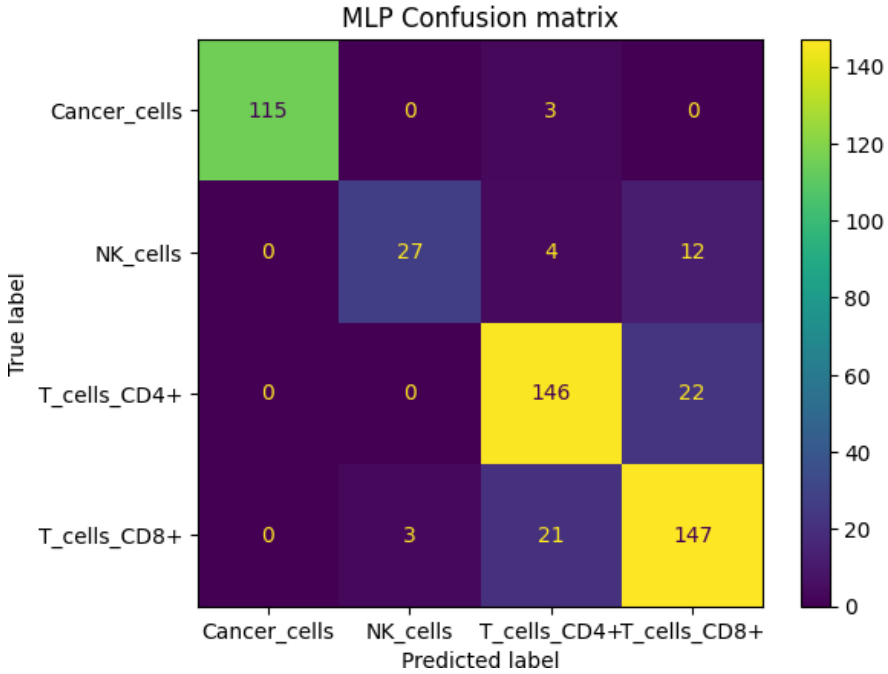

Figure 20: Matrice de confusion du MLP

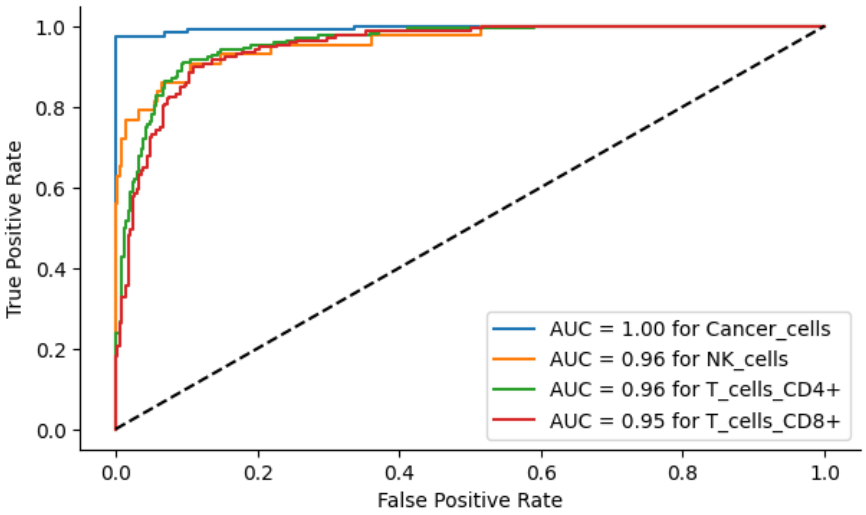

Figure 21: Courbe ROC du MLP

Notre MLP a obtenu de loin les meilleures performances en termes de précision pondérée (de 0.83). L'architecture du MLP est caractérisée par des couches cachées qui permettent d'extraire des caractéristiques pertinentes à partir des variables données en entrée.
Pour le challenge, nous avons en particulier construit un MLP composé de deux couches cachées, chacune composée de 1000 neurones. Cette architecture de MLP est puissante pour plusieurs raisons. D'une part, l'utilisation de plusieurs couches cachées permet d'extraire des caractéristiques de plus en plus abstraites à partir des données d'entrée, ce qui peut améliorer la représentation des données et la précision de la classification. D'autre part, le grand nombre de neurones dans chaque couche permet d'augmenter la capacité de représentation du réseau et d'optimiser la séparation des classes.

Conséquemment, considérer des modèles de Deep Learning plus complexe pourrait mieux répondre au problème de classification de données scRNA-seq. Et notre MLP, très basique en somme, est un début d'esquisse de son potentiel.

6.2 Comparaison des modèles

Les résultats précédents nous donnent un premier aperçu de nos modèles en cas d'usage. Mais nous devrions quand même jeter un coup d'œil aux propriétés de chacun de nos modèles grâce à un K-Fold Cross validation **stratifié** (parce qu'il est nécessaire de garantir la répartition équitable des classes dans chaque fold pour éviter de déséquilibrer davantage les données).

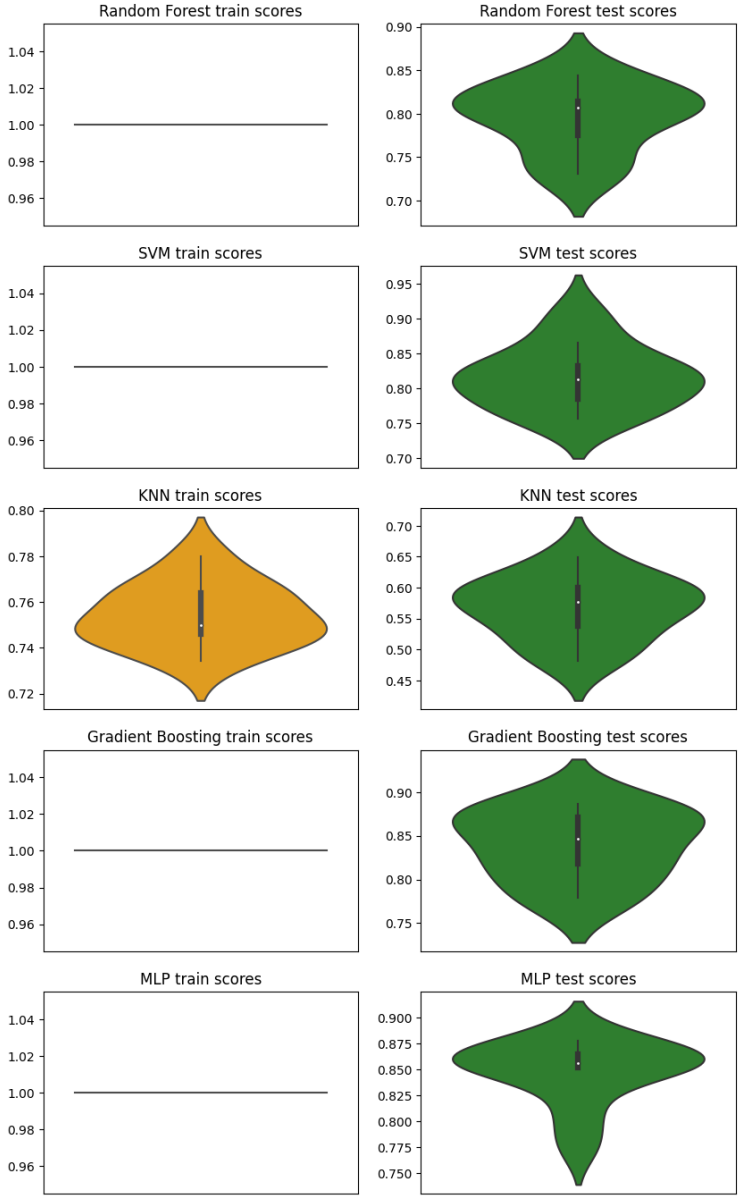

Figure 22: Validation croisée avec 10 fold stratifiés

Nous notons que la plupart de nos modèles surapprennent (les répartitions des précisions pondérées sont différentes). Et si on regarde la stabilité de nos modèles sur le jeu de validation, alors il semble que la plupart le sont, mis à part peut-être le MLP et le Random Forest (qui parfois sous-performent). Mais le SVM sur-performent parfois sur la validation.

Ainsi, si l'on souhaite généraliser la prédiction de nos modèles, le plus à même de répondre à cette question est sans nul doute le SVM.

7 Conclusion

La classification supervisée de types de cellules à partir de données scRNA-seq est un défi important pour la recherche en biologie et en médecine. Dans cette étude, nous avons testé des approches d'apprentissage automatique afin de répondre à ce défi en utilisant les données du benchmark scMARK.

Nos résultats, sur le sous-ensemble de données contenant uniquement quatre types de cellules différents, ont montré des premières pistes d'approches de classification supervisée pour prédire les types de cellules à partir des données scRNA-seq. En particulier, la combinaison de la méthode de réduction de variance avec SelectKBest a permis d'obtenir de bonnes précisions de classification, et pourrait être appliquée à des ensembles de données plus larges pour étudier les mécanismes biologiques des cellules et leurs fonctions spécifiques.

En conclusion, notre étude a démontré l'efficacité de l'approche d'apprentissage automatique supervisée pour la classification des types de cellules à partir de données scRNA-seq.

Et nous concernant, nous avons beaucoup apprécié cette expérience qui nous a notamment permis de mettre en pratique quelques subtilités en tant qu'informaticien, comme le formatage des fichiers, l'utilisation d'environnements conda et l'implémentation de classes ou de pipelines sur Python. Nous avons notamment appris à quel point il est essentiel de pré-traiter les données pour répondre efficacement à une problématique, surtout lorsque les données sont volumineuses.

References

[1] Swechha, D. Mendonca, O. Focsa, JJ. Díaz-Mejía, S. Cooper, "scMARK an MNIST like benchmark to evaluate and optimize models for unifying scRNA data", 2021, bioRxiv, doi: https://doi.org/10.1101/2021.12.08.471773

[2] https://www.technologynetworks.com/genomics/articles/understanding-single-cell-sequencing-how-it-works-and-its-applications-357578

[3] https://bioconductor.org/books/3.15/OSCA.basic/quality-control.html

[4] R. Hong, Y. Koga, S. Bandyadka, A. Leshchyk, Y. Wang et al. "Comprehensive generation, visualization, and reporting of quality control metrics for single-cell RNA sequencing data", 2022, Nature communication, doi: https://doi.org/10.1038/s41467-022-29212-9

[5] M. Su, T. Pan, QZ. Chen et al. "Data analysis guidelines for single-cell RNA-seq in biomedical studies and clinical applications", 2022, Military Medical Research, doi: https://doi.org/10.1186/s40779-022-00434-8

[6] L. Yu, Y. Cao, JHY. Yang et al. "Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data", 2022, Genome Biolology, doi: https://doi.org/10.1186/s13059-022-02622-0