HAU901I - Bioanalyse, transcriptomique

Rapport du projet : Introduction to single-cell transcriptomics
Dataset 1

Réalisé par :

Fadwa EL KHADDAR

Introduction

Les cellules macrophages sont les cellules possédant la propriété d'ingérer et de détruire de grosses particules (cellules vieilles, des agents infectieux comme les bactéries) par le processus de la phagocytose.
Cependant, les caractéristiques de ces cellules peuvent changer et devenir nuisibles dans certaines circonstances. C'est ce qui se produit dans le cas du cancer, ces macrophages produisent des substances comme les facteurs de croissance favorisant le développement de la tumeur.
Dans cet ensemble de données, les auteurs ont séquencé les TAMs (Tumor-associated macrophages) obtenues à partir d'un modèle murin de cancer du sein afin d'explorer la diversité potentielle de leurs phénotypes. En d'autres termes, à l'aide de leurs transcriptomes unicellulaires, nous voudrions identifier des groupes, vérifier s'ils expriment des programmes transcriptionnels différents bien qu'ils soient tous des TAMs, et de postuler des rôles potentiels basés sur les gènes spécifiques de chaque groupe.

I. Traitement de données

1. Prétraitement

Dans ce projet, nous avons utilisé plusieurs librairies, mais Seurat est la librairie de base pour le contrôle qualité, traitement et l'analyse des données de séquençage du transcriptome. Trois données d'entrée sont stockées dans un dossier nommé « SingleCell », *(voir II la procédure de lancement du code)*, Il s'agit de la matrice d'expression sous format **mtx**, le fichier contenant les *barcodes* sous format **tsv** et le fichier contenant les *features* sous format **tsv**.

```
library(dplyr)
library(Seurat)
library(readxl)

# Importation des fichiers

TAM_data <- Read10X(data.dir='SingleCell/')

# Initialisation de variables
TAM <- CreateSeuratObject(counts = TAM_data, project = "tumor-associated macrophages",
min.cells = 10, min.features = 800)

# Récupération des noms de gènes
gene.names <- rownames(TAM)
length(gene.names)

# Récupération des gènes mitochondriaux
mt.genes <- gene.names[grep("^mt-", gene.names)]
```

Après avoir importé et lu les données avec Seurat, nous avons gardé les gènes exprimés par au moins 10 cellules et les cellules avec au moins 800 gènes distincts qui remplissent cette condition. Ensuite, nous avons récupéré les gènes mitochondriaux afin de les éliminer dans l'étape du contrôle qualité.

2. Contrôle qualité

Cette étape consiste à nettoyer les données. En effet, la présence des gènes mitochondriaux indique une très faible qualité des données, ces gènes mitochondriaux proviennent une fois les cellules sont endommagées, les ARNs cytoplasmiques enfermés au sein des mitochondries vont également être retenus lors du séquençage, ces gènes sont distingués par le préfixe « mt- ».

```
#Elimination des genes mitochondriaux
names(TAM@meta.data)

TAM[["percent.mt"]] <- PercentageFeatureSet(TAM, pattern = "^mt")

# Visualisation des données par VlnPlot

VlnPlot(TAM, features=c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol=3)
```

VnPlot() est une fonction permettant la génération des trois types de plots, **nFeature_RNA** qui indique le nombre de gènes présents dans chaque cellule qui, dans notre cas varie entre 1000 et 7000, le plot **nCount_RNA**, qui montre le nombre de molécule détectée au sein d'une cellule qui est entre 10000 et 30000 mais ne dépasse pas 90000. Et enfin, le plot **Percent.mt**, qui le pourcentage des gènes mitochondriaux présents dans chaque cellule qui varie entre 1% et 45%.

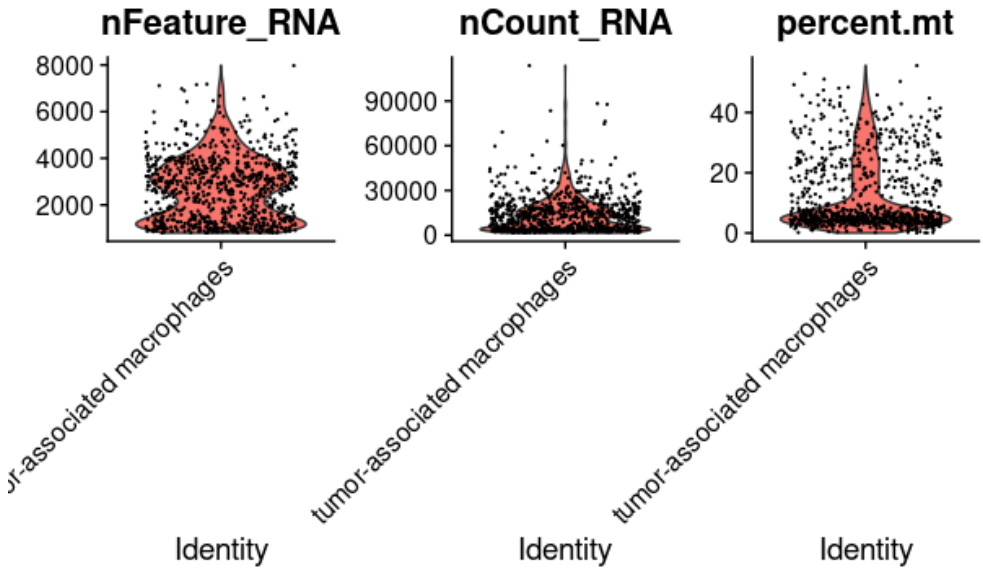

Figure1 : VnPlot de nFeature_RNA, nCount_RNA, percent.mt

L'étape suivante, consiste à filtrer les données et éliminer celles dont le pourcentage de gènes mitochondriaux est supérieure à 12 et dont l'UMIs est supérieure à 65000.

```
TAM<- subset(TAM, subset = percent.mt < 12 & nCount_RNA < 65000)
```

3. Normalisation et identification des gènes variables

La normalisation est une étape importante qui consiste à ajuster les données brutes pour tenir compte des facteurs qui empêchent la comparaison directe des mesures d'expression.

```
# Normalisation des données
TAM <- NormalizeData(TAM)
```

Ensuite, nous avons identifié les 2000 gènes les plus variables dans nos données.

```
# Les gènes variables
Gene_variable <- FindVariableFeatures(TAM, selection.method = "vst", nfeatures=2000)
```

4. Elimination des gènes du cycle cellulaire

Dans cette étape, et à partir de la liste générée dans l'étape précédente, nous avons éliminé les gènes impliqués dans le cycle cellulaire pour la souris. La liste des gènes est récupérée à partir du lien suivant **:** ***http://www.informatics.jax.org/vocab/gene_ontology/GO:0007049***

Le fichier récupéré est sous format Excel. Nous avons effectué une intersection entre les deux listes pour récupérer un nombre de gène de 1662. Ensuite, nous avons récupéré les 25 premiers gènes pour construire un LabelPoints.

```
# GO des gènes de cycles cellulaire
GO <- read_excel("GO_term_summary_20221217_042818.xlsx")
GO <- data.frame(GO)

# Récupération des noms de gènes dans le tableau des gènes GO de la souris
symbol <- GO$Symbol

# Elimination des gènes GO.
TAM<- Gene_variable[!rownames(Gene_variable) %in% symbol,]

#Plot de Labelpoints
top25 <- head(VariableFeatures(TAM), 25)
plot1 <- VariableFeaturePlot(TAM)
LabelPoints(plot=plot1, points=top25, repel=TRUE, ynudge = 0, xnudge = 0)
```

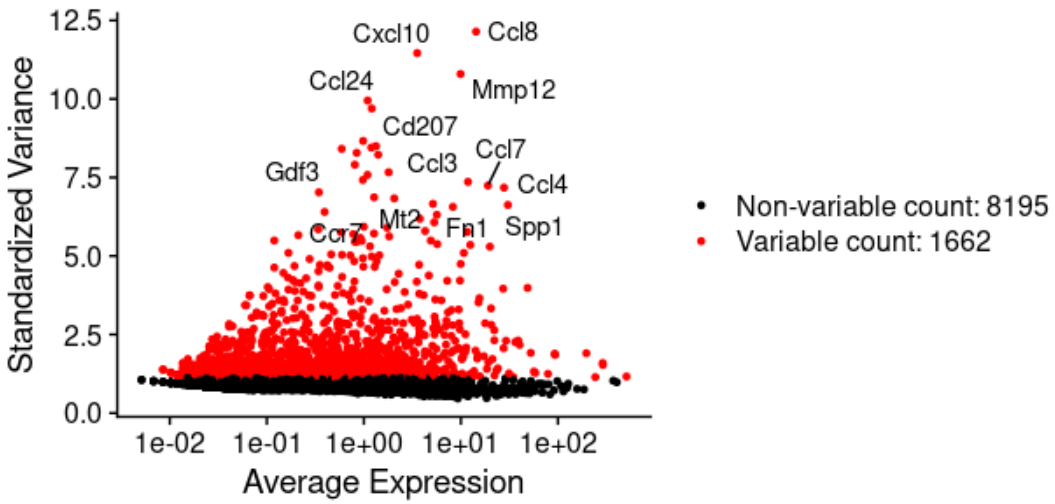

Figure 2 : LabelPoint des gènes variables

5. Réduction des dimensions :

La réduction de la dimensionnalité vise à réduire le nombre de dimensions distinctes dans les données. Ceci est possible car différents gènes sont corrélés s'ils sont affectés par le même processus biologique. Parmi les outils utilisés pour la réduction des dimensions est la PCA (Analyse en composante principale) qui découvre les axes dans l'espace de grande dimension qui capturent la plus grande quantité de variation.

```
# Application du PCA

all.genes <- rownames(TAM)
TAM <- ScaleData(TAM, features=all.genes)
TAM <- RunPCA(TAM, features=VariableFeatures(TAM), ndims.print=1:3,
nfeatures.print=10)
DimPlot(TAM, reduction="pca")
ElbowPlot(TAM)


# t-SNE
TAM <- RunTSNE(TAM, dims=1:35)
DimPlot(TAM, reduction="tsne")

# UMAP
TAM <- RunUMAP(TAM, dims=1:35)
DimPlot(TAM, reduction="umap")
```

6. Clustering :

Dans cette étape Seurat construit grâce à la fonction FindNeighbors() un graphe KNN (K-nearest Neighbor Graph) basé sur les distances euclidiennes dans l'espace PCA et affinent le poids de bords entre deux cellules en fonction de chevauchement partagé dans leur voisinages locaux. Ensuite, j'ai appliqué la fonction FindCluster() qui regroupe les cellules en se basant sur des techniques d'optimisation. Comme indiqué dans l'énoncé, nous avons dû récupérer 8 clusters, c'est la raison pour laquelle j'ai dû réduire la valeur de la résolution pour obtenir 8 clusters.

```
# Clustering

TAM <- FindNeighbors(TAM, dims = 1:35) ##Pour les 35 premiers PCA
TAM <- FindClusters(TAM, resolution=1.3) ## J'ai dû réduire la résolution pour pouvoir
récupérer 8 populations
DimPlot(TAM, reduction="pca", label= T)
DimPlot(TAM, reduction="tsne", label=T)
DimPlot(TAM, reduction="umap", group.by = "seurat_clusters",label=T)
```

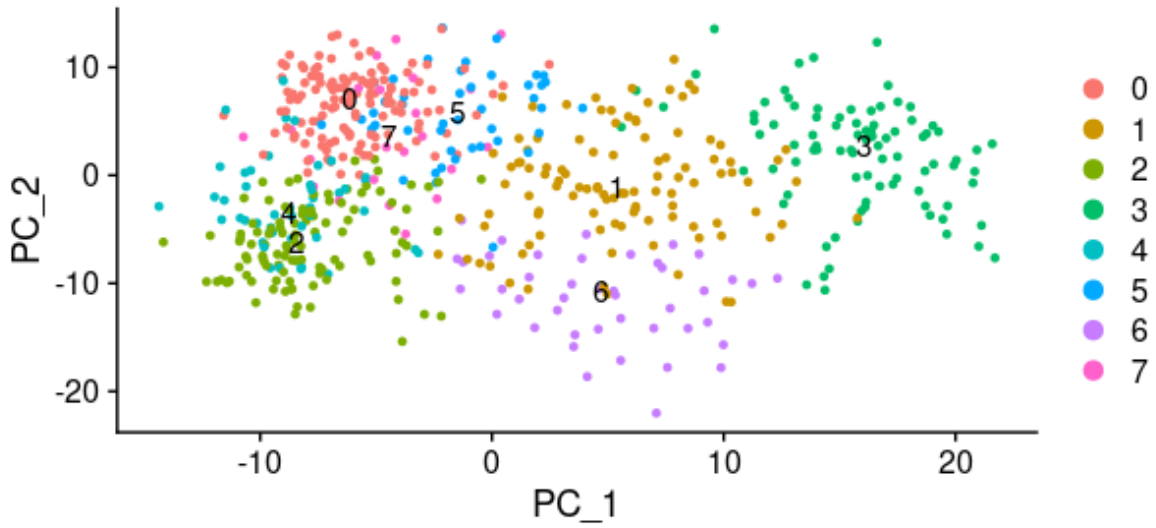

Figure 3 : les clusters générés

7. Récupération des gènes marqueurs et les gènes les mieux exprimés

La librairie Seurat permet d'identifier les gènes marqueurs des clusters à travers une expression différentielle, et cela à l'aide de la fonction FindAllMarkers() qui permet d'automatiser ce processus pour tous les clusters en définissant les marqueurs positifs et négatifs. J'ai ensuite sélectionné que les gènes ayant une valeur de *p-value* ajustée de 10^{-2} .

Afin d'annoter les clusters, j'ai opté pour la méthode qui se base sur la recherche sur web (*term GO enrichment*). Tout d'abord, j'ai récupéré les 5 premiers gènes en fonction de la valeur de log2FC

(Log2 fold change), étant donné qu'on travaille sur un seul type de population « macrophage », j'ai cherché sur internet pour pouvoir identifier le rôle de chaque cluster. Le tableau suivant (Tableau 1) montre les gènes les mieux exprimés dans chaque cluster (je n'ai présenté ici que les deux premiers).
A l'aide de la version web de ***www.genontology.org***, les noms gènes sont récupérés et recherchés dans le moteur de recherche, en précisant certains paramètres, comme l'origine des gènes « ***Mus Musculus*** » et spécifiant la catégorie « ***Biological process*** », les informations correspondantes aux rôles immunitaires de chaque cluster sont illustrées dans le tableau2 (Tableau 2).

```
# Les gènes marqueurs
TAM_markers <- FindAllMarkers(TAM, only.pos=TRUE, min.pct=0.25, logfc.threshold=0.25)
TAM_markers<- TAM_markers[TAM_markers$p_val_adj<0.01,]

#Les genes les mieux exprimés
a <- TAM_markers %>% group_by(cluster)%>% slice_max(n=5, order_by = avg_log2FC))
View(a)
```

Tableau1 : Les gènes les mieux exprimés en fonction du log2 Fold Change pour chaque cluster

Cluster	Avg_log2FC	Gènes
0	1.9874	*Blc2a1b*
0	1.8170	*Dcstamp*
1	1.4479	*Ccl8*
1	1.0234	*Fcrls*
2	1.1723	*Tmem119*
2	1.0157	*Cx3cr1*
3	3.2436	*Ccl7*
3	2.8225	*Ccl6*
4	1.3945	*H2afz*
4	1.0889	*H2afx*
5	1.2036	*H2-Ab1*
5	1.1560	*Il1rn*
6	2.0234	*Ccl8*
6	1.7223	*Maf*
7	4.3666	*Mmp12*
7	2.4877	*Lgals3*

Les rôles sont définis dans le tableau suivant :

Tableau 2 : Les fonctions immunitaires de chaque cluster

Les clusters	Fonctions
0	Ce cluster a moins de fonctions immunitaires. Régulation négative du traitement et de la présentation des antigènes par les macrophages.

1	Régulation négative par l'hôte de la réplication du génome viral Régulation positive de l'induction de la tolérane aux cellules tumorale
2	Régulation négative de la cytotoxicité assurée par les cellules microglies Régulation positive de la migration cellules microglies
3	Régulation positive du chimiotactisme des cellules NK (Natural Killer) Régulation positive du chimiotactisme des lymphocytes
4	Régulation négative du traitement et de la présentation des antigènes des cellules T Régulation négative du traitement et de la présentation des antigènes par les macrophages
5	Régulation positive du traitement et de la présentation des antigènes. Présentation des antigènes peptidiques exogènes via le CMH de classe II
6	Ce cluster a moins de fonctions immunitaires. Régulation négative du traitement et de la présentation des antigènes par les macrophages
7	Régulation négative de l'activation des lymphocytes T par la reconnaissance de l'antigène liée à la molécule CMH sur la cellule présentatrice de l'antigène Activation des cellules microglies impliquées dans la réponse immunitaire

Le tableau ci-dessous montre que les macrophages présents autour de cette tumeur n'effectuent pas le même rôle, ils se caractérisent par une certaine plasticité et une certaine hétérogénéité qui joue en faveur de la tumeur.

Pour faciliter la distinction des sous-populations des macrophages, j'ai choisi d'annoter les clusters en exécutant les commandes suivantes.

```
new.cluster.ids <- c("Macrophage_0", "Macrophage_1", "Macrophage_2", "Macrophage_3",
"Macrophage_4", "Macrophage_5",
                     "Macrophage_6", "Macrophage_7")
names(new.cluster.ids) <- levels(TAM)
TAM <- RenameIdents(TAM, new.cluster.ids)
DimPlot(TAM, reduction = "pca", label = TRUE, pt.size = 0.5) + NoLegend()
```

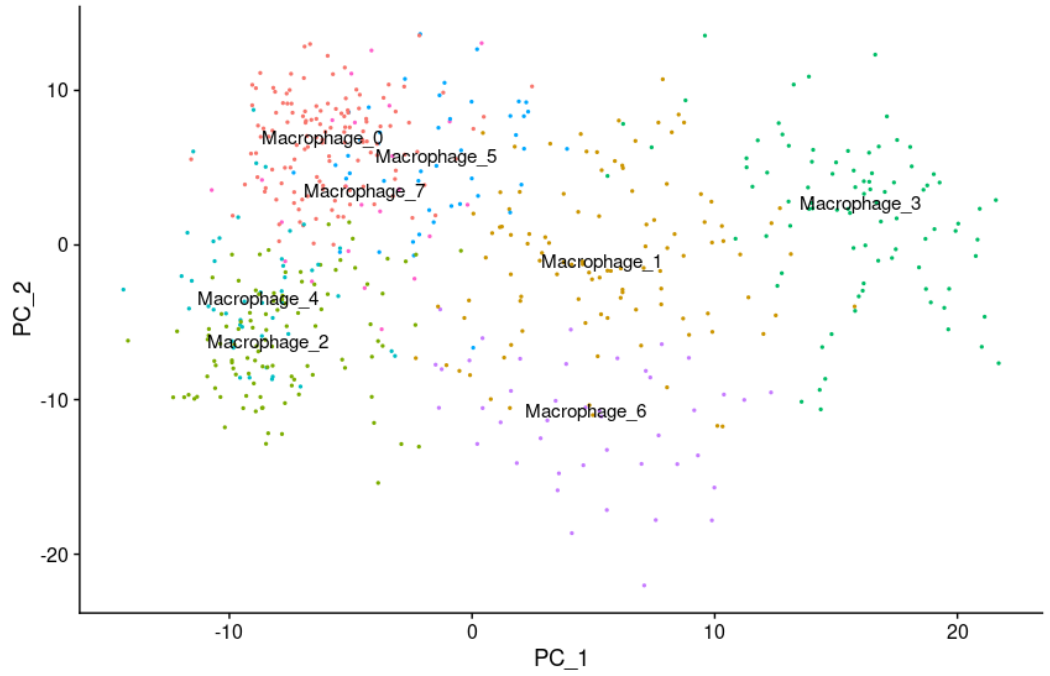

Figure 4 : Illustration de PCA des différents clusters

8. Diffusion Map et trajectoire des pseudos temps

Diffusion map est une méthode spatiale de la réduction des dimensions non linéaire, elle se base sur une métrique de distance nommée (distance de diffusion) qui est potentiellement pertinente pour illustrer la façon dont les cellules en différenciation peuvent suivre une dynamique type diffusion, en passant d'un état pluripotent à des états plus différenciés comme le cas des TAMs. Pour pouvoir construire la carte de diffusion, j'ai utilisé la librairie Destiny qui est exclusivement dédiée aux données Single Cell, et permettant d'organiser les cellules le long de leur trajectoire de développement.

Les processus biologiques qui se manifestent par une succession de changement de l'état cellulaire comme dans le cas des cellules TAM, peuvent être illustrés en identifiant une « trajectoire », c'est-à-dire un chemin à travers l'espace d'expression qui traverse les différents états cellulaires associés à un processus continu. La trajectoire est associée à une notion de pseudo temps (*pseudo time*) qui est définie comme étant le positionnement des cellules le long de la trajectoire qui correspond à la progression du processus biologique sous-jacent. Ces deux éléments sont construits à l'aide de la librairie Slingshot.

```
library(destiny)
library(slingshot)

# La Diffusion map  pour une projection de données

Emb <- Embeddings(TAM, reduction = "pca")
dif_map <- DiffusionMap(Emb)

names_clusters <- Idents(TAM)
sling <- slingshot(dif_map@eigenvectors[,1:15], clusterLabels = names_clusters, reducedDim
= "diffusion", start.clus = "Fibroblast")
sds <- as.SlingshotDataSet(sling)

plot(dif_map@eigenvectors[,1],  dif_map@eigenvectors[,2], col = names_clusters)

#Pseudotemps
lines(sds, type = "c", lwd=3 )
legend(x="topright",col=names_clusters,pch=16,legend=levels(names_clusters))
```

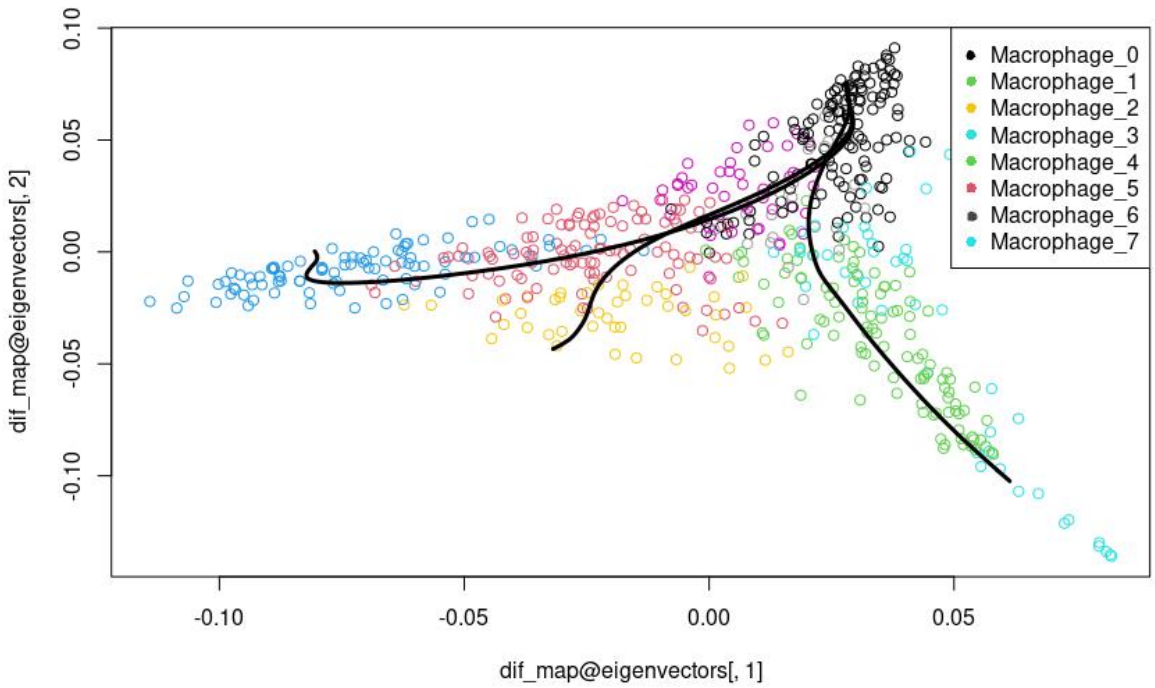

Figure 5 : trajectoire des cellules macrophages

L'analyse pseudo-temporelle a montré les potentiels de différenciation élevés du cluster 0 et 6 le cluster 1, 2 et 7 était à la fin des courbes principales, tandis que les clusters 3, 4 et 5 étaient au milieu. Cela suggère que les clusters 0 et 6 se différencient en cluster 3,4, 5 puis en 1,2,7.

II. Lancement du script :

Entrée :

Le script R généré prend en entrée les fichiers (matrice, barcodes, features) présents dans le dossier SingleCell et le fichier GO_term_summary_20221217_042818.xlsx.

Exécution :

Pour le lancer il faut changer « Directory » sur R (session – Set working directory – Choose Directory) puis choisir le dossier contenant le répertoire SingleCell et le fichier GO_term_summary_20221217_042818.xlsx
Sélectionner toutes les commandes du script puis cliquer sur Run.

Sortie :

Chaque étape du script génère un type de sortie, soit une table, un message ou un plot qui sera affiché dans la section droite.

Référence :

https://bioconductor.org/packages/release/bioc/vignettes/destiny/inst/doc/Diffusion-Maps.html

https://bioconductor.org/packages/devel/bioc/vignettes/slingshot/inst/doc/vignette.html

https://www.singlecellcourse.org/single-cell-rna-seq-analysis-using-seurat.html

https://pubmed.ncbi.nlm.nih.gov/34303328/