

Tuto Pré Annotation

Importer le code

Placer le dossier /pre_annot/ à côté du fichier de code.

⚠ Ne pas oublier de définir le répertoire de travail au niveau du code :

```
setwd("répertoire/absolu/de/mon/code")
```

Importer le code de pré-annotation

```
source("../pre_annot/pre_annot.R")
```

Prérequis

Pour pré-annoter il faut :

- L'objet seurat de travail `SeurOBJ`
- Le tableau des gènes différentiellement exprimés `diff.expressed.genes` (obtenu après filtration des gènes marqueurs). Il contient les champs :
 - `p_val` : p-valeur classique.
 - `avg_log2FC` : Coefficient qui mesure la proportion d'expression différentielle.
 - `p_val_adj` : p-valeur plus stricte, utilisée pour la pré-annotation.
 - `cluster`
 - `gene`

Fonctions de pré-annotation

Matrice de comptage

On peut calculer la matrice de comptage des gènes en communs. Cette matrice correspond à la méthode manuelle d'annotation.

```
type.annot.matrix <- get.annot.matrix(SeurOBJ, diff.expressed.genes)
```

Matrice d'expression différentielle

La matrice de pré-annotation assigne à chaque couple (cluster, type cellulaire) un score qui correspond à la moyenne des coefficients `avg_log2FC` des gènes en communs du couple.

Elle capture à quel point les gènes en commun sont différentiellement exprimés. (pertinent ? par la p-valeur est validée de toute façon)

```
type.avg.matrix <- get.avg.matrix(SeurOBJ, diff.expressed.genes)
```

Plus le score est grand, plus les coefficients associés aux gènes en commun sont grands.

⚠ Les résultats sont à croiser avec la matrice de comptage des gènes ci dessus.

Afficher la matrice de pré-annotation

```
display_heatmap(my.matrix)
```

On peut comparer les deux matrices :

```
display_heatmap(type.avg.matrix) + display_heatmap(type.alt.matrix)
```

Pré-assigner automatiquement

Choisit le type cellulaire le plus probable à partir de la moyenne des coefficients avg_log2FC. Utilise une fonction basique pour marquer les choix incertains.

```
clusters.annot <- pre_labels(type.annot.matrix, seuil = 3) # Seuil optionnel
```

Plus le seuil est haut, plus les assignations seront considérées comme incertaines.

On peut afficher directement les types cellulaires choisis :

```
clusters.annot
```

Ajouter les labels à l'objet Seurat et à l'UMAP

```
names(clusters.annot) <- levels(SeurOBJ)
LabeledSeurOBJ <- RenameIdents(SeurOBJ, clusters.annot)
DimPlot(LabeledSeurOBJ, reduction = "umap", label = TRUE, pt.size = 0.25) + NoLegend()
```

Fonctions d'aide à l'annotation finale

Marquer les gènes connus

On peut rajouter une colonne à la table des gènes différentiellement exprimés selon deux critères :

- Est-ce que le gène est déjà dans les données de pré-assignation ?
- Est-ce que le gène est dans la liste des gènes inexploitable ? (fichier ./pre_annot/uselessGenes.csv)

```
diff.expressed.genes <- mark_knowns(diff.expressed.genes)
```

Puis on peut continuer l'assignation manuelle en se concentrant sur les gènes pas encore exploités et en s'aidant de la pré-assignation.

Exemple de code final

```
source("./pre_annot/pre_annot.R")

diff.expressed.genes <- mark_knowns(diff.expressed.genes) # optionnel

type.annot.matrix <- get.annot.matrix(SeurOBJ, diff.expressed.genes)
type.avg.matrix <- get.avg.matrix(SeurOBJ, diff.expressed.genes)
display_heatmap(type.annot.matrix) + display_heatmap(type.avg.matrix)

# Display annotations on UMAP
clusters.annot <- pre_labels(type.annot.matrix, seuil = 6)
clusters.annot

names(clusters.annot) <- levels(SeurOBJ)
LabeledSeurOBJ.alt <- RenameIdents(SeurOBJ, clusters.annot)
DimPlot(LabeledSeurOBJ.alt, reduction = "umap", label = TRUE, pt.size = 0.25) + NoLegend()
```