11/05/2022

Spécification du projet GIMOVapp

Graphical Interface for Multi-Omics Visualisation

Objectifs :

Le logiciel a pour but de fournir une interface graphique ergonomique et fluide pour la visualisation de données single cell RNA seq puis multi-omique dans un second temps (priorités Cite-seq, ATAC-seq et spatial). L'interface sera sous la forme d'un site web accessible uniquement par le réseau du CRCL et sera hébergée sur un serveur interne.

L'interface sera organisée en plusieurs onglets thématiques comportant des graphiques interactifs qui feront ressortir au mieux les informations du jeu de donnés. Le but étant de permettre à l'utilisateur de projeter des informations de contrôle qualité ou biologique sur son jeu de données On retrouvera aussi la possibilité de faire des analyses directement depuis l'interface pour donner plus d'autonomie à l'utilisateur.

L'interface sous forme de site web devra permettre un accès sans installation de la part de l'utilisateur ainsi qu'un accès limité aux données qui ne pourront pas être récupérées, ou modifiées. Les analyses ne seront que des modifications temporaires du jeu de données, qui ne sera pas enregistré.

Technologies :

Le logiciel sera développé en Python grâce au module Dash (équivalent de Rshiny pour Python). Le script lancera l'interface grâce à un serveur Flask, configuré par Dash. Elle utilisera des données sous la forme h5ad format AnnData pour le single-cell ou MuData (extension de AnnData) pour le multi-omique. Les analyses proposées seront faites par le module Scanpy (ou autre package python), utilisé pour l'analyse de données single-cell. Les différents graphiques seront construits avec le module Plotly, compatible avec Dash (car même créateur).

Le logiciel sera hébergé sur un serveur interne au CRCL puis déployé dans une machine virtuelle. Les utilisateurs pourront donc accéder à leurs données uniquement depuis le réseau du CRCL et via un code unique qui leur sera envoyé.

Graphiques :

- Umap : Affichage des différents embeddings calculés, ainsi qu'une coloration par les observations ou les gènes. Il sera possible de modifier la taille des points, le pourcentage de cellules affichées, la palette, ainsi que les seuils de coloration. Un switch permettra de passer d'un affichage 2D vers 3D si l'embeddings le permet. La Umap pourra afficher l'image pour le spatial, et les trajectoires de la vélocité. On pourra aussi sélectionner au lasso différents groupes pour faire une analyse différentielle.
- Histplot : Affichage d'observations et variables numériques pour vérifier les QCs appliqués. Si une variable factorielle est sélectionnée pour le boxplot, alors le graphique affichera la variable numérique en fonction de celle factorielle.

- Box plot : Affiche la distribution des données numériques en fonction d'une donnée factorielle.
- Tableau : Affiches les gènes différentiellement exprimés dans un tableau trié par la pvalue. Le tableau comporte au moins le nom, le fold change, la pvalue et la pvalue ajustée. On pourra sauvegarder le tableau en csv soit partiellement soit complètement (fichier de plusieurs dizaines de milliers de lignes)
- Heatmap : Affiche l'expression des gènes différentiellement exprimé pour chaque cellule et chaque cluster sélectionné.
- Dotplot : Affiche l'expression des gènes différentiellement exprimé dans chaque cluster.
- Density plot : Permet d'afficher la distribution de données continue en fonction d'autre données continue. Comme par exemple, le nombre d'UMI en fonction du pseudotime, On pourra aussi créer une ligne par observation factorielle.
- Stacked barplot : Permet d'afficher la répartition des données entre deux observations factorielles.
- Fluvial plot (sankey plot) : Permet d'afficher la répartition des données entre deux observations factorielles.

Analyses :

- Pathways : Permet la sélection de plusieurs gènes pour la création d'un pathways. Scanpy permettra pour chaque cellule un calcul d'affinité avec le pathway. On pourra ensuite l'afficher sur les autres plots.
- Clusterings : Permet de créer un clustering avec plusieurs résolutions différentes, puis de récupérer les gènes différentiellement exprimés par ces mêmes clusters. On pourra comparer les différents clustering directement grâce à un sankey plot.