

# mIF spatial data

Aleksandra Prorok

7 maja 2024

## 1 Opis

Dane przestrzenne mIF są złożonym zbiorem informacji obejmującym lokalizację poszczególnych komórek w tkance oraz ich poziomy ekspresji dla wielu protein lub markerów interesujących. Ta precyzyjna informacja przestrzenna umożliwia dogłębne zbadanie organizacji komórek oraz ich wzajemnych interakcji w obrębie tkanki, co jest kluczowe dla zrozumienia odpowiedzi immunologicznej w raku oraz innych chorobach.

### 1.1 Analiza

Analizę danych przestrzennych rozpoczęliśmy wyczyszczenia danych - pozbycia się wybrakowanych oraz nieużywanych plików. Następnie przeszliśmy do wizualizacji tkanek z podziałem na typ komórek za pomocą biblioteki `matplotlib`. Wykonane zostało więc mapowanie, przenoszące fenotyp do odpowiedniego rodzaju komórek.

Jednakże najważniejszym krokiem było stworzenie grafów sąsiedztwa na podstawie funkcji `radius_neighbors_graph` z pakietu `scikit-learn`. Pozwoliło nam to na stworzenie grafu sąsiedztwa dla komórek typu Bcell, które uznajemy za załączek tworzenia się TLS, czyli specyficznej struktury tkanki nowotworowej oraz dla wszystkich komórek, by móc porównywać, co dzieje się w obrębie danego TLS.

Następnie za pomocą funkcji `connected_components` z pakietu `scipy`, odtworzyliśmy stworzone przez sąsiadów komponenty i wyegzekwowaliśmy te, w których znajduje się co najmniej ... komórek typu Bcell. Pozwala nam to podejrzewać, że może być to potencjalnie TLS.

Dla odpowiednich komponentów, odnaleźliśmy ID komórek znajdujących się w nich i z ich pomocą odnieśliśmy się do grafu sąsiedztwa wszystkich komórek. Pozwoliło nam to na odnalezienie wszystkich komórek znajdujących się w obrębie sąsiedztwa danego komponentu, i sprawdzenie czy mamy do czynienia z TLS.

Finalnie, przeprowadziliśmy klastrowanie z liczbą klastrów 3 na podstawie procentowego rozkładu typów komórek w komponentach. Wydaje mi się, że za pomocą PCA zwizualizowaliśmy przynależność komponentów do klastrów.

Przedstawione zostały także histogramy z procentowym rozkładem typów komórek w danych komponentach.

## **1.2 Implementacja**

Projekt i wizualizacje zostały zamknięte w frameworku Streamlit. Wykorzystane zostały pakiety scipy, scikit-learn oraz matplotlib.

## **1.3 Co można jeszcze poprawić?**

Definitywnie można poprawić klastrowanie, poprzez skorzystanie z algorytmu decydującego o odpowiedniej liczbie klastrow.

Przydatne byłoby także dodanie ciekawych wizualizacji, które pozwoliłyby na dogłębnější analizę i interpretację wyników.