POLITECHNIKA OPOLSKA
Wydział Informatyki

Szymon Barczyk

Sprawozdanie z przedmiotu Praca przejściowa

Implementacja algorytmu grupowania SIMLR w języku Python

Prowadzący zajęcia
prof. dr hab. inż. Bronisław Tomczuk

Praca wykonana pod kierunkiem
Dr hab. inż Michała Tomaszewskiego

Opole, 2025 r.

Spis treści

Rozdział 1

Wstęp

W ostatnich latach dynamiczny rozwój technologii biologii molekularnej, w szczególności sekwencjonowania RNA pojedynczych komórek (single-cell RNA sequencing, scRNA-seq), umożliwił odkrycie nieznanej wcześniej heterogeniczności i zróżnicowania funkcjonalnego w obrębie populacji komórkowych. Dzięki tej metodzie możliwe stało się niezależne (de novo) identyfikowanie typów komórek i ich subpopulacji, co odgrywa kluczową rolę m.in. w badaniach nad nowotworami, różnicowaniem komórek czy odpowiedzią immunologiczną.

Pomimo postępu technologicznego, analiza danych pochodzących z eksperymentów scRNA-seq napotyka szereg wyzwań obliczeniowych. Tradycyjne metody analizy, opracowane z myślą o klasycznym (bulk) RNA-seq, nie radzą sobie wystarczająco dobrze z typowymi dla scRNA-seq problemami, takimi jak obecność komórek odstających, szum wynikający z amplifikacji transkryptów czy tzw. dropouty – przypadkowe braki w odczytach ekspresji genów. Co więcej, nowoczesne platformy, takie jak DropSeq czy GemCode, umożliwiają jednoczesne badanie tysięcy komórek, generując przy tym bardzo duże i skrajnie rzadkie zbiory danych, w których nawet 95

W celu skutecznej analizy tak złożonych danych, niezbędne jest zastosowanie specjalistycznych metod, które potrafią uczyć się odpowiednich miar podobieństwa między komórkami bez potrzeby ich uprzedniego zdefiniowania. Jednym z takich nowoczesnych podejść jest algorytm SIMLR (Single-cell Interpretation via Multi-kernel LeaRning), który oferuje ramy do nienadzorowanego grupowania, redukcji wymiarowości i wizualizacji danych jednocelowych poprzez adaptacyjne uczenie się miary podobieństwa między komórkami.

Celem niniejszej pracy jest implementacja algorytmu SIMLR w języku Python, jego weryfikacja na rzeczywistych i syntetycznych zbiorach danych oraz porównanie skuteczności z innymi popularnymi metodami klasteryzacji. Praca koncentruje się na analizie jakości i wydajności algorytmu SIMLR, a także na ocenie jego przydatności w kontekście analizy danych scRNA-seq.

Rozdział 2

Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest implementacja algorytmu grupowania SIMLR (Single-cell Interpretation via Multi-kernel LeaRning) w języku Python oraz ocena jego efektywności na wybranych zbiorach danych. Praca ma na celu nie tylko stworzenie poprawnie działającej implementacji, ale również porównanie jej skuteczności i wydajności z innymi popularnymi metodami klasteryzacji, takimi jak k-means, DBSCAN czy UMAP. Efektem końcowym pracy będzie analiza jakości uzyskanych klastrów oraz praktyczna ocena możliwości zastosowania SIMLR w rzeczywistych problemach analizy danych.

Rozdział 3

Przegląd istniejących rozwiązań

Klasteryzacja to jedna z podstawowych technik eksploracyjnej analizy danych, która umożliwia identyfikację naturalnych grup lub struktur w danych bez uprzedniego oznaczenia klas. W kontekście analizy danych wysokowymiarowych, takich jak dane z eksperymentów single-cell RNA-seq (scRNA-seq), wybór odpowiedniej metody klasteryzacji odgrywa kluczową rolę w uzyskaniu wiarygodnych i biologicznie interpretowalnych wyników.

3.1 K-means

Opis:
Klasyczny algorytm centroidowy do grupowania danych na podstawie minimalizacji odległości między punktami a środkami klastrów.

Zalety:
Szybki i prosty w implementacji.

Wady:
Wymaga wcześniejszego określenia liczby klastrów, wrażliwy na wartości odstające i nierównomiernie rozłożone klastry.

3.2 DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

Opis:
Metoda klasteryzacji oparta na gęstości punktów. Wykrywa skupiska dowolnych kształtów i potrafi identyfikować szum.

Zalety:
Nie wymaga określenia liczby klastrów z góry, dobrze radzi sobie z danymi zawierającymi szum.

Wady:
Wydajność może spadać w wysokich wymiarach, wrażliwa na dobór parametrów (epsilon i minimalna liczba punktów).

3.3 UMAP

(Uniform Manifold Approximation and Projection)

Opis:
Technika redukcji wymiarowości, często używana do wizualizacji i klasteryzacji danych jednocelowych.

Zalety:
Zachowuje strukturę lokalną i globalną danych, szybka, dobrze działa z dużymi zbiorami danych.

Wady:
Głównie metoda wizualizacyjna — do klasteryzacji wymaga dodatkowych kroków (np. zastosowania k-means na danych zredukowanych przez UMAP).

3.4 t-SNE

(t-distributed Stochastic Neighbor Embedding)

Opis:
Technika redukcji wymiarów, popularna do wizualizacji danych wysokowymiarowych.

Zalety:
Bardzo dobrze oddaje lokalne zależności między punktami.

Wady:
Wysoki koszt obliczeniowy, nie zachowuje globalnej struktury, trudny do użycia w celu bezpośredniej klasteryzacji.

3.5 SC3

(Single-Cell Consensus Clustering)

Opis:
Narzędzie zaprojektowane specjalnie do klasteryzacji danych scRNA-seq, łączące różne podejścia i miary podobieństwa w jeden wynik klasteryzacji.

Zalety:
Dobre wyniki w kontekście danych jednocelowych, specjalnie dostosowane do problemów dropoutów i szumu.

Wady:
Działa głównie w środowisku R, wolniejsze dla dużych zbiorów danych.

Rozdział 4

Użyte technologie

4.1 Python

Python to wysokopoziomowy, interpretowany język programowania, który zdobył ogromną popularność w środowisku naukowym, akademickim i przemysłowym dzięki swojej prostocie składni, czytelności kodu oraz rozbudowanemu ekosystemowi bibliotek. W kontekście analizy danych i uczenia maszynowego, Python oferuje szeroki wachlarz narzędzi umożliwiających efektywne przetwarzanie, analizę, wizualizację danych oraz implementację algorytmów matematycznych i statystycznych.

W niniejszej pracy język Python został wykorzystany do implementacji algorytmu SIMLR, przetwarzania danych wejściowych, a także porównania działania różnych metod klasteryzacji.

4.2 NumPy

NumPy to podstawowa biblioteka Pythona służąca do obliczeń naukowych. Udostępnia wydajne struktury danych (przede wszystkim tablice wielowymiarowe – ndarray) oraz szeroki zbiór funkcji do wykonywania operacji matematycznych, algebry liniowej, statystyki i analizy macierzy.

W kontekście implementacji SIMLR, NumPy umożliwia:

- efektywne operacje na macierzach podobieństwa i danych wejściowych,
- szybkie obliczenia numeryczne niezbędne w procesie optymalizacji i normalizacji danych,
- operacje wektorowe znacząco przyspieszające działanie algorytmu.

4.3 SciPy

SciPy to rozszerzenie biblioteki NumPy, oferujące dodatkowe funkcje z zakresu obliczeń naukowych i inżynierskich. Biblioteka zawiera m.in. narzędzia do optymalizacji, interpolacji, analizy statystycznej oraz obliczeń z zakresu algebry liniowej.

W pracy SciPy wykorzystywana jest głównie w:

- rozwiązywaniu problemów optymalizacyjnych występujących podczas uczenia się metryki podobieństwa,
- obliczeniach wartości własnych i dekompozycji macierzy,
- obsłudze operacji wymagających precyzyjnych narzędzi matematycznych poza standardowym zakresem NumPy.

4.4 scikit-learn

scikit-learn to jedna z najpopularniejszych bibliotek do uczenia maszynowego w Pythonie. Udostępnia szeroki zestaw narzędzi do klasteryzacji, klasyfikacji, regresji, walidacji krzyżowej oraz analizy zbiorów danych.

W niniejszej pracy scikit-learn pełni kluczową rolę w:

- porównywaniu SIMLR z innymi metodami klasteryzacji,
- ocenie jakości klasteryzacji przy użyciu wbudowanych metryk,
- przetwarzaniu wstępnym danych,
- budowie i testowaniu eksperymentów porównawczych.

Bibliografia

[1] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, & Serafim Batzoglou *Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning* (June 9, 2016) , Nature Methods.

[2] Python Software Foundation *Python 3.12.10 documentation* (8 April 2025) https://docs.python.org/release/3.12.10/

[3] scikitlearn developers *scikitlearn 1.6.1 documentation* (2025) https://scikit-learn.org/stable/user_guide.html

[4] SciPy community *SciPy documentation* (May 08, 2025) https://docs.scipy.org/doc/scipy/

[5] NumPy Developers *NumPy 2.2 documentation* (2025) https://numpy.org/doc/stable/

[6] R Core Team *An Introduction to R* (version 4.5.0, 2025-04-11) https://cran.r-project.org/doc/manuals/r-release/R-intro.html

[7] Luciano Ramalho *Zaawansowany Python. Jasne, zwięzłe i efektywne programowanie* (2020-11-19) Promise

[8] Jake VanderPlas *Python Data Science. Niezbędne narzędzia do pracy z danymi. Wydanie II* (2023-12-06) Helion

[9] Aurélien Géron*Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow. Wydanie III* (2023-07-04) Helion

[10] Hadley Wickham, Mine Çetinkaya-Rundel, Garrett Grolemund*Język R w data science. Importowanie, porządkowanie, przekształcanie, wizualizowanie i modelowanie danych. Wydanie II* (2024-04-02) Helion