Problem selekcije marker gena na Zeisel skupu podataka

Jurić, Katarina

Sveučilište Josipa Jurja Strossmayera u Osijeku

Fakultet primijenjene matematike i informatike

Sveučilišni prijediplomski studij Matematika i računarstvo

Problem selekcije marker gena na Zeisel skupu podataka

Završni rad

Mentor:

izv. prof. dr. sc. Domagoj Matijević

Komentor:

dr. sc. Luka Borozan

Student:

Katarina Jurić

Osijek, 2024

Sadržaj

Sažetak

Ovaj rad istražuje problem selekciije marker gena na Zeisel skupu podataka. Obrađuje se usporedba između različitog broja markera s različitim ograničenjima i različitom vremenu izvršenja, tabličnim prikazom i vizualnim prikazom preko grafova. Uspoređuju se također rezultati dobiveni trima različitim metodama. Metodom ograničenja usmjerenih na centar, metodom parnih ograničenja udaljenosti i metodom parnih ograničenja usmjerenih na centar. Također, u radu se definiraju alati, koji su korišteni za rješavanje tog problema.

Ključne riječi

marker geni, metoda ograničenja usmjerenih na centar, metoda parnih ograničenja udaljenosti, metoda parnih ograničenja usmjerenih na centar, TSNE, vrijeme izvršavanja

Marker gene selection problem

Summary

This thesis investigates the marker gene selection problem on Zeisel dataset. It covers a comparison between different number of markers with different constraints and execution times, with tabular representation and visual representation via graphs. The results obtained from three different methods, are also compared. Center based constraints, pairwise distance constraints and pairwise center based constraints. In this thesis, as well, are defined programing tools, that are used in solving marker gene selection problem.

Keywords

marker genes, center based constraints, pairwise distance constraints, pairwise center based constraints, TSNE, execution time

1 | Uvod

Bioinformatika je grana znanosti koja usko povezuje biologiju i računarstvo. Sve veća dostupnost tehnologije sekvenciranja rezultirala je stvaranju velikih skupova bioloških podataka. Ti podatci su motivirali razvoj novih računalnih metoda koje pohranjuju, obrađuju, analiziraju i prikazuju veličinu i posebnost tih podataka.

Jedna od tih metoda je scRNA-seq, ona omogućava detaljan uvid pojedinačnih stanica gena. Jedan od ključnih problema je odrediti marker gene, koji su značajni za razlikovanje staničnih tipova. Selekcija marker gena je značajna za optimizaciju daljnjih analiziranja kompleksnih podataka i smanjenju tih istih. Primjer koji će se kroz rad primjenjivat je Zeisel skup podataka, koji je namijenjen za optimiziranje selekcije markera.

Pristupit će se različitim selekcijama markera. Analizirati će se njihove performanse preko točnosti i vremena izvršenja.

U radu će se opisati linearno programiranje te Gurobi, alat koji se koristi za optimizaciju. Nakon toga će se razmortiti selekcija marker gena, gdje se prvo opisuje implementacija, korištena za rješavanje problema selekcije marker gena, prikazana na sljedećem linku `https://github.com/jurick1910/Zavrsni-prakticki-projekt`. Zatim se opisuje analiziranje dobivenih rezultata tom implementacijom.

1.1 Osnovne definicije

Definicije koje su potrebna za shvaćanje ovoga rada:

Marker geni: specifične sekvence u DNK koje se koriste za prepoznavanje genetskih razlika između pojedinaca ili vrsta te oni djeluju poput otisaka prstiju, prikazuju jedinstvene genetske informacije

scRNA-seq: tehnologija sekvenciranja jednostanične RNA, pristup za otkrivanje heterogenosti i složenosti RNA transkripata pojedinih stanica te za otkrivanje sastava različitih tipova stanica i funkcija unutar tkiva, organa ili organizama.

TSNE(*t-distributed Stochastic Neighbor Embeddin*): tehnika za redukciju dimenzija i vizualni prikaz podataka, nelinearna stoga algoritam omogućuje razdvajanje podataka koji se ne mogu odvojiti „ravnom crtom“.

2 | Linearno programiranje

Linearno programiranje (LP) je matematička metoda za optimizaciju (minimizaciju ili maksimizaciju) linearnih funkcija, uz linearna ograničenja. Softverski alat koji će se koristiti za računanje LP je Gurobi.

Definicija 1 (vidjeti [3, Poglavlje 2.1]). *Neka su* $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{a_i} \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $i \in M = M_1 \cup M_2 \cup M_3$, *gdje su* M_i, $i = 1, 2, 3$ *skupovi indeksa takvi da je* $M_i \cap M_j = \emptyset$, $i \neq j$, *te* $f : \mathbb{R}^n \to \mathbb{R}$ *linearna funkcija cilja zadana formulom* $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x}$. *Promatrajmo sljedeći problem uvjetne optimizacije:*

$$\begin{aligned} & f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \quad \to \quad \min_x \\ & \textit{uz uvjete} \\ & \mathbf{a_i}^T\mathbf{x} \geq b_i, i \in M_1 \\ & \mathbf{a_i}^T\mathbf{x} \leq b_i, i \in M_2 \\ & \mathbf{a_i}^T\mathbf{x} = b_i, i \in M_3. \end{aligned} \tag{2.1}$$

Problem (2.1) zovemo ***problem linearnog programiranja****. Prijetimo da je dopustivo područje određeno s linearnim funkcijama uvjeta.*

2.1 Gurobi

Gurobi se koristi za računanje složenih matematičkih problema optimizacije s visokom efikasnošću i brzinom. Koristi se u znanstveni područjima, kao što su financije, logistika, energetika i druga, gdje se traže optimalna rješenja unutar zadanih ograničenja.

Primjer 1. *Riješite sljedeći problem uvjetne optimizacije*

$$x_1^2 + x_2^2 \quad \to \quad \min_{x_1, x_2}$$

uz uvjete:

$$\begin{aligned} x_1 + x_2 &\geq 1 \\ x_1 - x_2 &\leq 4 \end{aligned}$$

Rješenje. Iz definicije 1 se može uočiti da su odgovarajući skupovi indeksa $M = \{1,2\}$, $M_1 = \{1\}$, $M_2 = \{2\}$, $M_3 = \emptyset$, a odgovarajući vektori su $\mathbf{x} = \begin{bmatrix} x_1^2 & x_2^2 \end{bmatrix}^T$, $\mathbf{c} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, $\mathbf{a_1} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, $\mathbf{a_2} = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$, $\mathbf{b} = \begin{bmatrix} 1 & 4 \end{bmatrix}^T$. Zatim se optimizira funkcija $x_1^2 + x_2^2$ preko Gurobi-a (u Python programu) na sljedeći način:

```
from gurobipy import *
Model()

opt_mod = Model(name = "Primjer 1")

x1 = opt_mod.addVar(name = 'x1', vtype = GRB.CONTINUOUS, lb = 0)
x2 = opt_mod.addVar(name = 'x2', vtype = GRB.CONTINUOUS, lb = 0)

obj_fn = pow(x1, 2) + pow(x2, 2)
opt_mod.setObjective(obj_fn, GRB.MINIMIZE)

c1 = opt_mod.addConstr(x1 + x2 >= 1, name = 'c1')
c2 = opt_mod.addConstr(x1 - x2 <= 4, name = 'c2')

opt_mod.optimize()
opt_mod.write("Primjer_1.lp")

print('Objective Function Value: %f' % opt_mod.ObjVal)
for v in opt_mod.getVars():
    print('%s: %g' % (v.VarName, v.x))
```

Kreira se model preko funkcije `Model(name = "")`, zatim se dodaju kontinuirane nenegativne varijable x_1 i x_2. Nakon toga se definira funkcija cilja `obj_fn` te se minimizira (`GRB.MINIMIZE`). Metodom `addConstr` dodaju se ograničenja (uvjeti). Optimizira se model te se ispisuje vrijednost funkcije (`opt_mod.ObjVal`) i varijable s optimalnim vrijednostima, sljedećim prikazom:

```
Optimalna vrijednost funkcije: 1.0
x1 = 0.5
x2 = 0.5
```

3 | Selekcija marker gena

Za problem selekcije marker gena koristi se problem linearnog programiranja uz scGeneFit, alat koji analizira genetske podatke iz scRNA-seq eksperimenata, za optimizaciju selekcije marker gena u Zeisel skupu podataka. Zeisel skup podataka je ključan za analiziranje gena u različitim vrstama stanica. Također, bitno je za razumijevanje složenih bioloških procesa i prepoznavanja genetskih informacija specifičnih za različite stanice. Taj skup podataka ima hijerarhijsku strukturu[1], koja doprinosi preciznijem i boljem analiziranju rezultata.

Rezultati u nastavku su dobiveni trima različitim metodama, opisanim u podpoglavlju 3.2, koje su korištene za određivanje najboljih marker gena koji zadovoljavaju postavljene uvjete.

Problem LP je rješavan pomoću Gurobi-a, tako da su prvo definirane varijable, što su odabrani markeri, ciljane funkcije i ograničenja (uvjeti) vezana uz točnosti i vrijeme izvršavanja. Zatim se traže optimalni marker geni, pomoću metode ograničenja usmjerenih na centar, metode parnih ograničenja udaljenosti te metode parnih ograničenja usmjerenih na centar. Nakon optimizacije markera gena analiziraju se dobivene vrijednosti točnosti i vremena izvršavanja, u obliku tablica i grafova.

3.1 Implementacija

```
import numpy as np
import matplotlib.pyplot as plt
import scipy
import time
import sklearn
import sklearn.manifold
import scipy.io
from . import data_files
import gurobipy as gp
from gurobipy import Model, GRB
import random
```

[1]Hijerarhija je struktura u kojoj su elementi organizirani nekim redoslijedom, npr. važnost, generalizacija. Biološki se može gledati, stanični tipovi su organizirani hijerarhijski s glavnim stanicama na vrhu i specifičnim tipovima tih stanica ispod vrha. [5]

U priloženom se vide biblioteke koje su korištene za rješavanje problema selekcije marker gena u scGeneFit-u. Modul „numpy“ iz biblioteke NumPy je korištenq za numeričke operacije, iz biblioteke Matplotlib smo koristili „matplotlib.pyplot“ koji je ispisivao vizualni prikaz, koji će biti prikazan u nastavku. Iz SciPy biblioteke koristimo modul „scipy“ za znanstveni i tehničko računanje, te „time“ i „random“, koji su ugrađen moduli u Pythonu. „Time“ računa vrijeme izvršavanja, dok „random“ bira nasumične brojeve. Iz Scikit-learn biblioteke, „sklearn“, za algoritme strojnog učenja te iz Gurobi-a „gurobipy“. Također iz `data_files` učitavamo Zeisel skup podataka.

```
1 def get_markers_hierarchy(data, labels, num_markers, method='
    centers', sampling_rate=0.1, n_neighbors=3, epsilon=10,
    max_constraints=1000, redundancy=0.01, verbose=True)
```

Kod iznad prikazuje Python funkciju za dohvaćanje markera s hijerarhijom, gdje parametri predstavljaju sljedeće:

- `data`: Nxd numpy niz s koordinatama točaka (N broj točaka, d dimenzija),
- `labels`: lista s oznakama (N oznaka, svaka oznaka je jedinstvena),
- `num_markers`: broj markera, koji se odabiru proizvoljno
- `sampling_rate`: odabir ograničenja iz nasumičnog uzroka s proporcijom `sampling_rate` (zadan 0.1),
- `n_neighbors`: bira ograničenja iz n najbližih susjeda (zadano 3),
- `epsilon`: ograničenje će biti oblika `expr > Delta` (Delta jednaka epsilon puta norma najmanjeg ograničenja, zadano 10), najvažniji parametar koji određuje opseg ograničenja, dok ostali parametri samo određuju veličinu LP-a kako bi se prilagodili ograničenim računalnim resursima,
- `max_constraints`: maksimalan broj ograničenja (zadan 1000)
- `redundancy`: koristi se samo kod metode ograničenja usmjerenih na centar,
- `verbose`: ako je istinito, ispiše informacije o napretku,
- `method`: može biti „centers“, „pairwise“ ili `pairwise_centers` (zadano „centers“), gdje je sljedeće:
  - metoda ograničenja usmjerenih na centar je ,,`centers`‘‘
  - metoda parnih ograničenja udaljenosti je ,,`pairwise`‘‘
  - metoda parnih ograničenja usmjerenih na centar je ,,`pairwise_centers`‘‘

čija su objašnjenja u nastavku.

```
def __lp_markers(constraints, num_markers, epsilon):
    m, d = constraints.shape
    c = np.concatenate((np.zeros(d), np.ones(m)))
    l = np.zeros(d + m)
    u = np.concatenate((np.ones(d), np.array([float('inf') for i in
    range(m)])))
    aux1 = np.concatenate((constraints, -np.identity(m)), axis=1)
    aux2 = np.concatenate((np.ones((1, d)), np.zeros((1, m))), axis
    =1)
    A = np.concatenate((aux1, aux2), axis=0)
    b = np.concatenate((-epsilon * np.ones(m), np.array([
    num_markers])))

    model = gp.Model()
    x = model.addVars(d + m, lb = l, ub = u, name = "x")
    model.addConstrs((gp.quicksum(A[i, j] * x[j] for j in range(d +
    m)) == b[i] for i in range(m + 1)), name = "constraints")
    model.setObjective(gp.quicksum(c[i] * x[i] for i in range(d + m
    )), gp.GRB.MINIMIZE)
    model.optimize()
    return {"x": [x[i].x for i in range(d + m)]}
```

Najvažnija funkcija, koja preko Gurobi optimizatora minimizira problem selekcije marker gena, je ,,__lp_markers‘‘. Ona poprima parametre ,,num_marker‘‘, „epsilon“ i „constraints“ (ograničenja u obliku matrice dimenzije mxd, m broj ograničenja, d broj značajki).

```
def __select_constraints_summarized(data, labels, redundancy=0.01)
```

Ova funkcija odabire ograničenja preko formule $c_a - c_{a+1}$, gdje su c_i centri različitih klasa.

```
def __select_constraints_pairwise(data, labels, samples,
    samples_labels, n_neighbors)
```

Ova funkcija odabire ograničenja preko formule x - y, gdje x i y imaju različite oznake.

```
def __select_constraints_centers(data, labels, samples,
    samples_labels)
```

Ova funkcija odabire ograničenja preko formule:

$$(x - c_{t'})^2 - (x - c_t)^2 > \Delta^2$$

gdje x pripada klasteru s centrom c_t.

Ispis rješenja se dobije pokretanjem sljedeće implementacije:

```
from scGeneFit.functions import *

%matplotlib inline
import numpy as np
np.random.seed(0)
```

Učitavaju se funkcije iz modula „scGeneFit.functions“, zatim se omogućava prikaz grafova.

```
from sklearn.neighbors import NearestCentroid
clf=NearestCentroid()

def performance(X_train, y_train, X_test, y_test, clf):
    clf.fit(X_train, y_train)
    return clf.score(X_test, y_test)
```

Učitava se „NearestCentriod“ i inicira se kao clf. To jednostavan modul koji koristi centroid[2] (središnju točku) svake klase kao referentnu točku za predviđanje. Treniraju se podatci `X_train` i oznake `y_train`, te se vraća točnost modela na testnom skupu.

```
[data, labels, names]=load_example_data("zeisel")
N,d=data.shape
```

Učitavamo Zeisel podatke.

```
num_markers=25
method='centers'
redundancy=0.1

markers= get_markers_hierarchy(data, labels, num_markers, method=
    method, redundancy=redundancy)

accuracy=performance(data, labels[0], data, labels[0], clf)
accuracy_markers=performance(data[:,markers], labels[0], data[:,
    markers], labels[0], clf)

print("Accuracy (whole data,", d, " markers): ", accuracy)
print("Accuracy (selected", num_markers, "markers)",
    accuracy_markers)
```

S metodom ograničenja usmjerenih na centar računamo točnost (accuracy) i vrijeme izvršavanja preko funkcije ,,`get_markers_hierarchy`‘‘.

[2]U geometriji, centar mase dvodimenzionalnog lika ili trodimenzionalnog tijela. Stoga centroid dvodimenzionalnog lika predstavlja točku na kojoj bi se moglo uravnotežiti kada bi npr. bila izrezana iz lima. Centroid kružnice ili kugle je njezin centar. Općenitije, centroid predstavlja točku određenu srednjom vrijednošću koordinata svih točaka u nekom skupu. [7]

```
a=plot_marker_selection(data, markers, names[0])
```

Ispisuje vizualni prikaz grafova preko TSNE-a.

```
num_markers=25
method='pairwise'
sampling_rate=0.05
n_neighbors=3
epsilon=10
max_constraints=500
use_centers=False

markers= get_markers_hierarchy(data, labels, num_markers, method=
    method, sampling_rate=sampling_rate, n_neighbors=n_neighbors,
    epsilon=epsilon)

accuracy=performance(data, labels[0], data, labels[0], clf)
accuracy_markers=performance(data[:,markers], labels[0], data[:,
    markers], labels[0], clf)

print("Accuracy (whole data,", d, " markers): ", accuracy)
print("Accuracy (selected", num_markers, "markers)",
    accuracy_markers)
```

S metodom parnih ograničenja udaljenosti računamo točnost (accuracy) i vrijeme izvršavanja preko funkcije ,,`get_markers_hierarchy`''. Metodu parnih ograničenja usmjerenih na centar učitavamo preko istog koda kao i za metodom parnih ograničenja udaljenosti uz promjenu „pairwise" u ,,`pairwise_centers`'' te ,,`use_centers=False`'' u ,,`use_centers=True`''.

Za sve tri metode koristimo TSNE za prikaz grafova.

3.2 Rezultati

Korištenjem Zeisel skupa podataka s tri različite metode su dobiveni sljedeći rezultati.

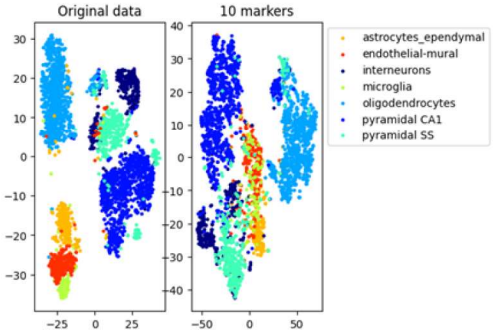
3.2.1 Metoda ograničenja usmjerenih na centar

Metoda koristi razlike između centara klasa za određivanje ograničenja. Od svake klase, srednji centar se koristi kao točka za generiranje ograničenja koja su namijenjena da razlike između klasa budu veće od određenog praga. Za ovu metodu maksimalni broj ograničenja zadan je na 89. Sljedeća tablica 3.1 prikazuje točnosti i vrijeme izvršavanja u sekundama za nekoliko brojeva markera:

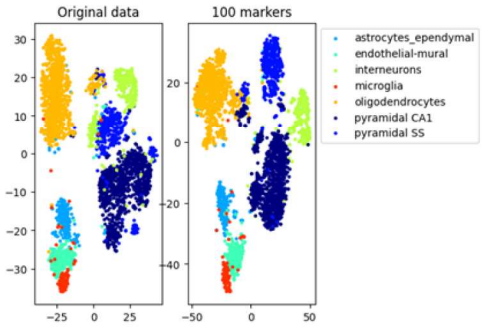
Broj markera	Broj ograničenja	Točnost	Vrijeme izvršavanja (sekunde)
10	89	0.8099833610648919	12.588299036026001
15	89	0.8662229617304492	12.973503828048706
20	89	0.8685524126455907	12.699063301086426
25	89	0.8805324459234609	12.87053370475769
50	89	0.9108153078202995	12.696515560150146
100	89	0.9391014975041597	13.041755676269531

Tablica 3.1: Rezultati metode ograničenja usmjerenih na centar

Za ovu metodu iz tablice 3.1 se može sljedeće zaključiti:
Za 10 markera, točnost je 0.8099833610648919 dok je za 100 markera jednaka 0.9391014975041597. Točnost od 100 markera je veća od točnosti od 10 markera te iz toga možemo zaključiti, što je manji broj markera to je točnost manja, dok je preciznija za veći broj markera. Iz tablice se također može zaključiti, što je veći broj markera to je vrijeme izvršavanja duže.



Slika 3.1: 10 markera



Slika 3.2: 100 markera

Slika 3.1 prikazuje rezultate za 10 markera i 100 markera pomoću TSNE-a. Lijevi grafovi prikazuju originalne podatke dok desni podatke za određeni broj markera. Graf s više markera pomaže očuvanju strukture podataka, povećavajući točnost. Stoga desni graf na slici 3.2 je precizniji, no vrijeme izvršavanja je duže.

3.2.2 Metoda parnih ograničenja udaljenosti

Metoda se služi razlikama između svih uzoraka s različitim oznakama. Generira ograničenja za dovoljnu udaljenost između različitih oznaka. Sljedeća tablica 3.2 prikazuje točnosti i vrijeme izvršavanja u sekundama za nekoliko brojeva markera s ograničenjima iz segmenta [1000, 6000]:

Broj markera	Broj ograničenja	Točnost	Vrijeme izvršavanja (sekunde)
10	1000	0.7038269550748752	147.2401442527771
10	2000	0.7038269550748752	342.573762178421
10	3000	0.7038269550748752	474.43251514434814
10	4000	0.7038269550748752	731.0720653533936
10	5000	0.7038269550748752	1014.2989234924316
10	6000	0.7038269550748752	1349.761330127716
50	1000	0.919134775374376	154.3847315311432
50	2000	0.908153078202995	348.3114495277405
50	3000	0.9071547420965058	487.3961489200592
50	4000	0.907820299500832	746.0194842815399
50	5000	0.9098169717138103	1037.54208445549
50	6000	0.9088186356073211	1384.5690088272095
100	1000	0.946089850249584	149.40975689888
100	2000	0.9477537437603993	277.9124641418457
100	5000	0.9454242928452579	1042.5717034339905
100	6000	0.9504159733777038	1409.2790305614471
500	1000	0.9484193011647255	147.33414578437805
500	2000	0.9490848585690516	284.39072823524475
500	5000	0.9504159733777038	1055.198569059372
500	6000	0.9517470881863561	1387.8972718715668

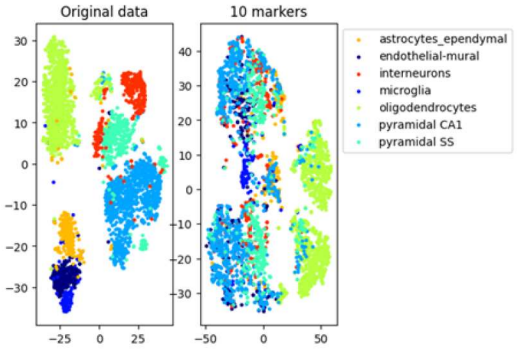
Tablica 3.2: Rezultati metode parnih ograničenja udaljenosti

Za ovu metodu iz tablice 3.2 se može sljedeće zaključiti:

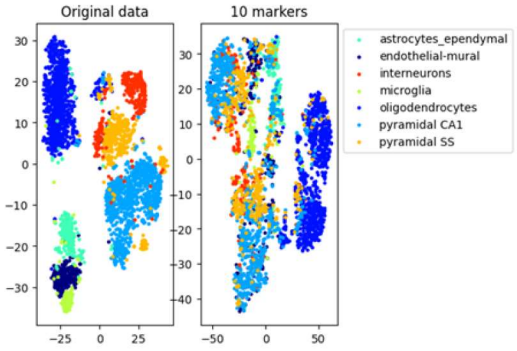
Za 10 markera s ograničenjem jednakim 1000, točnost je 0.7038269550748752 te se za 10 markera s ograničenjem od 6000 nije promijenila, no vrijeme izvršavanja je se znatno promijenilo, s 147.24s za ograničenje od 1000 na 1349.76 za 6000.

Za 500 markera s ograničenjem od 1000, točnost je 0.9484193011647255 s vremenom izvršavanja 147.33s, a za ograničenje od 6000, točnost je 0.9517470881863561 s vremenom izvršavanja 1387.89s. Iz toga se vidi da s porastom ograničenja, raste i vrijeme izvršavanja i točnost je preciznija.

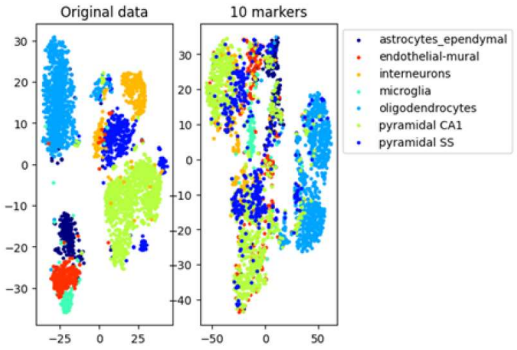
Može se zaključiti da s većim brojem markera i ograničenja, točnost postaje veća i vrijeme izvršavanje je duže, no ima izuzetaka, kao što je vidljivo u tablici 3.2 za 50 markera točnost s ograničenjem od 2000 je bolja nego s ograničenjem od 4000. Također točnost može biti ista za različit broj markera s različitim ograničenjima, npr. 100 markera s ograničenjem od 5000 i 500 markera s ograničenjem od 6000 čija je točnost 0.9504159733777038, ali vrijeme izvršavanje im je drugačije.



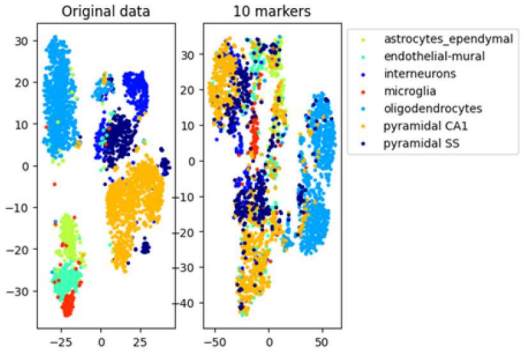
Slika 3.3: 10 m. s 1000 ograničenja



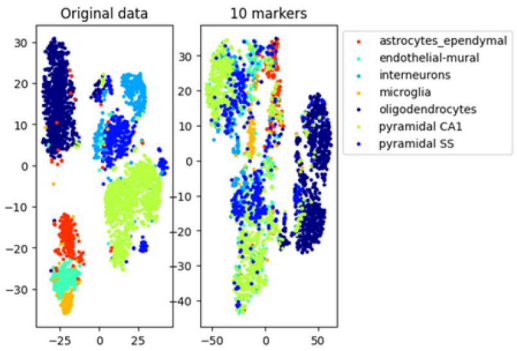
Slika 3.4: 10 m. s 2000 ograničenja



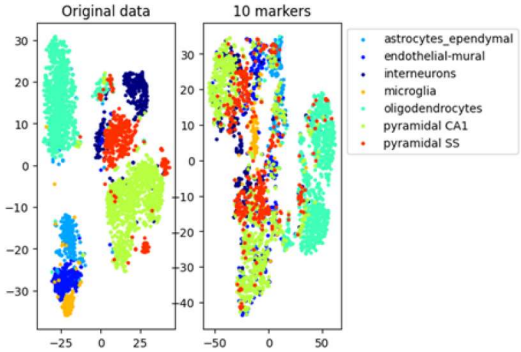
Slika 3.5: 10 m. s 3000 ograničenja



Slika 3.6: 10 m. s 4000 ograničenja

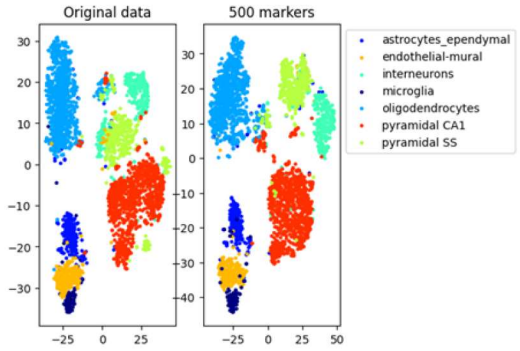


Slika 3.7: 10 m. s 5000 ograničenja

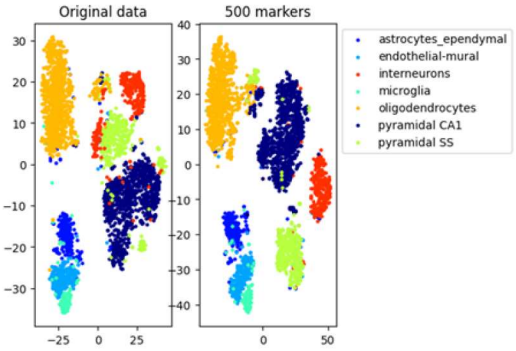


Slika 3.8: 10 m. s 6000 ograničenja

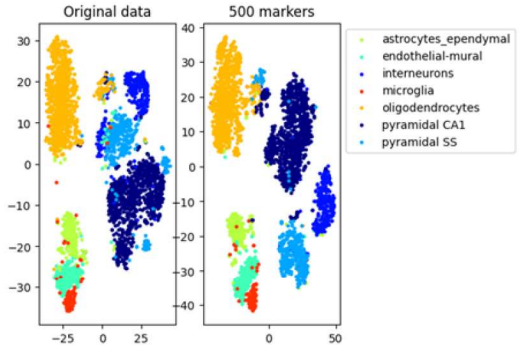
Slike (3.3 - 3.8) prikazuju TSNE grafove s 10 markera s različitim ograničenjima. Graf s ograničenjem od 1000 je znatno drugačiji od grafova s višim ograničenjem, dok su grafovi s višim ograničenjima identični. U njima su stanice jasnije prikazane.



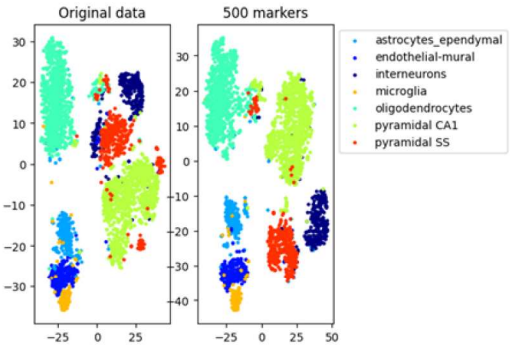
Slika 3.9: 500 m. s 1000 ograničenja



Slika 3.10: 500 m. s 2000 ograničenja



Slika 3.11: 500 m. s 5000 ograničenja



Slika 3.12: 500 m. s 6000 ograničenja

Slike (3.9 - 3.12) prikazuju TSNE grafove s 500 markera s različitim ograničenjima. Graf s ograničenjem od 1000 je znatno drugačiji od grafova s višim ograničenjem. Razlika između ostala tri grafa je minimalna, razlikuju se u stanicama microglia, endothelial-mural i `astrocytes_ependymal`. Slika 3.12 prikazuje precizniji prikaz stanica, sličan kao za originalne podatke.

Razlika u vizualnom prikazu 10 markera i 500: U prikazima s 500 markera prikaz je jasniji te sličniji originalni podatcima, dok za 10 markera se vidi velika razlika u prikazu stanica. Iz toga se može uspostaviti da s većim brojem markera i ograničenja, vizualni prikaz stanica će bit detaljniji i pregledniji, dok vrijeme izvršavanja traje duže.

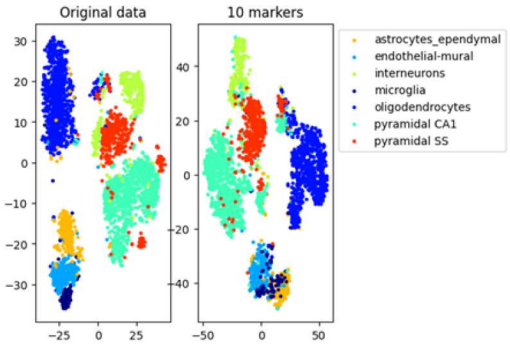
3.2.3 Metoda parnih ograničenja usmjerenih na centar

Metoda koristi razlike između centara klasa i svih uzoraka. Preko razlike svake točke i njezinog centra s razlikom točke i drugih centara. Ova metoda je kombinacija prijašnje dvije metode. Sljedeća tablica 3.3 prikazuje točnosti i vrijeme izvršavanja u sekundama za nekoliko brojeva markera s nekoliko ograničenja, gdje je maksimalno ograničenje 2011:

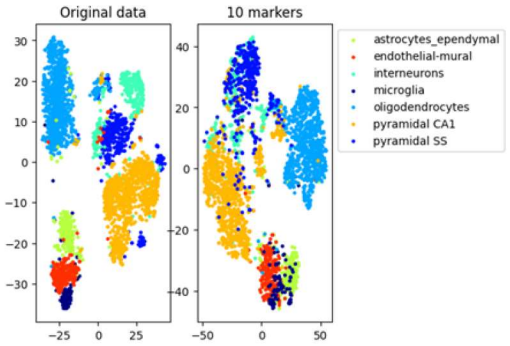
Broj markera	Broj ograničenja	Točnost	Vrijeme izvršavanja (sekunde)
10	500	0.854242928452579	68.15221643447876
50	500	0.9038269550748752	66.86037349700928
100	500	0.930116472545757	67.34651470184326
500	500	0.8123128119800332	69.0920193195343
10	1000	0.8608985024958402	155.26166152954102
50	1000	0.9118136439267887	148.54600644111633
100	1000	0.9227953410981697	154.44333910942078
500	1000	0.8482529118136439	157.47608065605164
10	2000	0.8123128119800332	354.52719378471375
50	2000	0.8935108153078203	377.3004870414734
100	2000	0.9271214642262895	380.136198759079
500	2000	0.8063227953410982	382.2451431751251
10	2011	0.8123128119800332	358.63791370391846
50	2011	0.924459234608985	368.06763339042664
100	2011	0.8978369384359401	382.4927535057068
500	2011	0.8133111480865225	372.44002628326416

Tablica 3.3: Rezultati metoda parnih ograničenja usmjerenih na centar

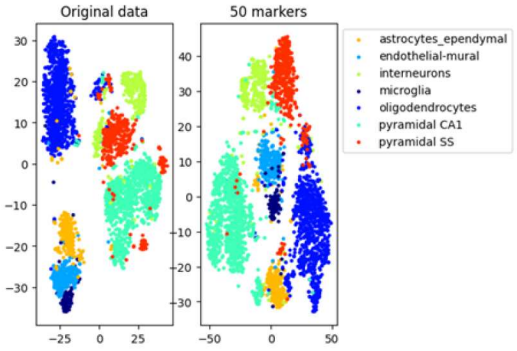
Iz tablice 3.3 se može vidjeti da za ograničenje od 500 za 10, 50 i 500 markera, točnost je manje precizna nego za 100 markera, s tim da je točnost za 500 markera manja od točnosti za 10. To također vrijedi za ograničenje od 1000 i 2000. Za ograničenje od 2011 (maksimalno) 500 markera ima bolju točnost nego točnost za 10. Vrijeme izvršavanja je veće s povećanjem broja markera.



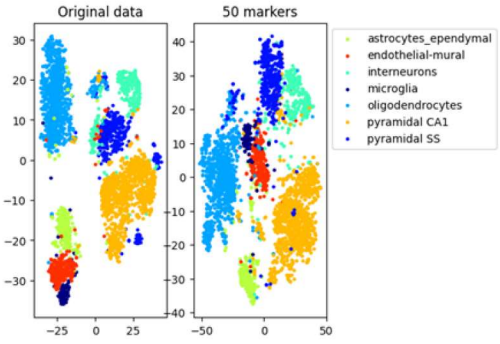
Slika 3.13: 10 m. s 1000 ograničenja



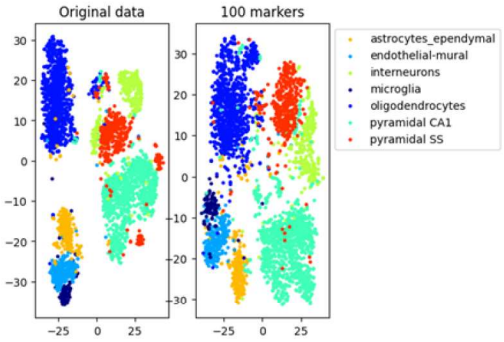
Slika 3.14: 10 m. s 2011 ograničenja



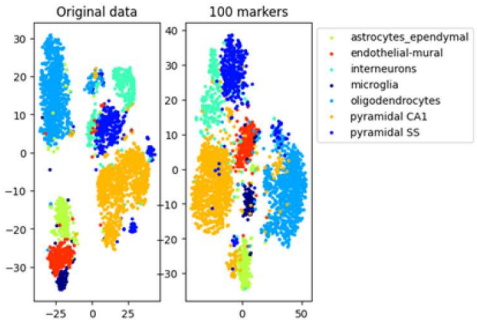
Slika 3.15: 50 m. s 1000 ograničenja



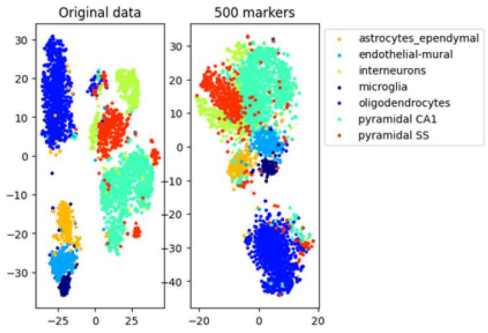
Slika 3.16: 50 m. s 2011 ograničenja



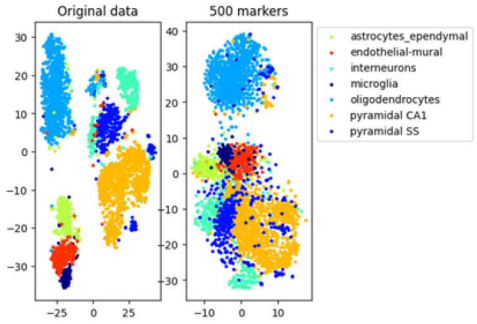
Slika 3.17: 100 m. s 1000 ograničenja



Slika 3.18: 100 m. s 2011 ograničenja



Slika 3.19: 500 m. s 1000 ograničenja



Slika 3.20: 500 m. s 2011 ograničenja

Na grafovima za 10 markera (3.13 i 3.14) vidi se razlika u prikazu stanica, s većim ograničenjem slika je preciznija i detaljnija, neke stanice se više ističu od drugih. Za 50 markera slike 3.15 i 3.16 su sličnije, no još se vidi razlika prikaza stanica. Za 100 markera, preciznije i detaljnije vizualno prikazana, je slika s maksimalnim ograničenjem 3.18, dok za 500, 3.19 i 3.20, prikaz izgleda potpuno drugačije, stanice su raspodijeljene na suprotne strane.

Razlika za ograničenje od 2011: grafovi su najsličniji za 50 i 100 markera, dok je za 500 markera prikaz najviše različit od ostala tri broja markera. Za ograničenje od 1000 se također vizualni prikaz može opisati kao i za maksimalno ograničenje.

Zaključak

Iz dobivenih rezultata u radu možemo uočiti da je metoda parnih ograničenja udaljenosti, najpreciznija metoda od tri obrađene. Ona prikazuje najbolju preciznost točnosti također i prikaz grafova preko TSNE-a, sličniji originalnim podatcima za razliku od drugih. Veći broj markera i ograničenosti su pokazali detaljnije i preciznije rezultate, s povećanim vremenom izvršavanje. Proširenjem na druge skupove podataka, mogli bih usporediti dobivene rezultate te ustanoviti njihovu sličnost.

Literatura

1. A. A. Awan, *Introduction to t-SNE*, DataCamp, Inc., 2023.

2. B. Dumitrascu, S. Villar, D. G. Mixon & B. E. Engelhardt, *Optimal marker gene selection for cell type discrimination in single cell analyses*, Nature Communications, 2021.

3. Ivana Kuzmanović, Kristian Sabo, *Linearno programiranje*, Odjel za matematiku, Sveučilište Josipa Jurja Strossmayera u Osijeku, 2016.

4. Amit Zeisel, Ana B Munoz-Manchado, Simone Codeluppi, Peter Lonnerberg, Gioele La Manno, Anna Jureus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al., *Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq*, Science, 347(6226):1138–1142, 2015.

5. *hijerarhija*, Hrvatska enciklopedija, mrežno izdanje, Leksikografski zavod Miroslav Krleža, 2013. – 2024. Pristupljeno 3.9.2024. https://www.enciklopedija.hr/clanak/hijerarhija.

6. Gurobi Optimization, LLC. *Gurobi Optimizer*, 2024. Preuzeto s https://www.gurobi.com/solutions/gurobi-optimizer/

7. Britannica, The Editors of Encyclopaedia. *centroid. Encyclopedia Britannica.*, 09.07.2024. Pristupljeno 16.09.2024. https://www.britannica.com/science/centroid

Životopis

Rođena sam 2002. u Vinkovcima. Svoje obrazovanje sam započela 2009. godine u Osnovnoj školi „Ivan Kozarac“ u Županji. Zatim sam 2017. nastavila obrazovanje u matematičkoj gimnaziji Županja do 2021. godine te iste godine upisujem sveučilišni preddiplomski studij Matematika i računarstvo na Fakultetu primijenjene matematike i informatike (tada Odjel za matematiku) Sveučilišta Josipa Jurja Strossmayera u Osijeku.