

R275

ISTRAŽIVANJE PODATAKA 2

SEMINARSKI RAD

Klasterovanje ekspresija gena

Student :
Mihajlo VIĆENTIJEVIĆ

Profesor :
dr Nenad MITIĆ

13. juni 2019



Sadržaj

1	Uvod	2
2	Podaci	2
3	Klasterovanje gena	5
3.1	Analiza i pretprocesiranje podataka	5
3.2	Aktivnija grupa gena	16
3.3	Geni sa manjim stepenom ekspresije	29
4	Klasterovanje ćelija	34
5	Zaključak	38
	Literatura	38

1 Uvod

Napretkom tehnologije DNK mikročipa (eng. *DNA microarray*) koja omogućava da se odredi nivo ekspresije svih gena u genomu čoveka dovelo je do revolucije u analizi gena i proteina. Pre ovog izuma geni su analizirani jedan po jedan a kako svaki čovek ima najmanje 30000¹ gena proces analize je bio vremenski zahtevan.

Analiza podataka koji su dobijeni pomoću nove tehnologije iziskuje kompleksne statističke metode i primenu biomatematičkih algoritama. Takođe, tehnologija DNK mikročipa je omogućila merenje nivoa ekspresije na desetine hiljada gena istovremeno što je jako korisno jer je prilikom analize podataka zapaženo da aktivnosti gena nisu nezavisne jedna od druge. Drugim rečima, potrebno je identifikovati grupe gena kao i njihov uticaj na biološke procese i funkcionisanje ćelija.

U ovom radu fokus je stavljen na samu identifikaciju grupa, bez biološke interpretacije. Grupe identifikujemo poznatim metodama klasterovanja, tj. grupe gena identifikujemo klasterovanjem gena, a grupe ćelija klasterovanjem ćelija. Početni zajednički korak obuhvata upoznavanje sa podacima i njihovo učitavanje. Nadalje je rad podeljen u dve celine. Prvi deo rada obrađuje gene. Podaci se analiziraju i transformišu u kontekstu klasterovanja gena i nad njim izvršavaju različite metode klasterovanja. Izdvojen je jedan zanimljiv podskup gena koji je zasebno razmatran. Drugi deo rada obrađuje ćelije i uvodi dodatne metode klasterovanja. U oba slučaja nastoji se da se primenom različitih metoda identifikuju smislene grupe.

2 Podaci

Kao što je gore rečeno, prvi zajednički korak obuhvata učitavanje podataka: *Human embryonic stem cell and cortical organoids*. Podaci su organizovani u obliku dvodimenzione matrice gde redovi odgovaraju genima dok kolone predstavljaju PBMC ćelije. Periferne mononuklearne krvne ćelije (eng. Peripheral blood mononuclear cells - PBMCs) uključuju ćelije različitih tipova: limfocite (B ćelije, T ćelije, NK ćelije), monocite i dendritske ćelije. Ova matrica se naziva i matrica ekspresije. Formalnije, možemo da je definisemo kao matricu $M = \{t_{ij} | 1 \leq i \leq n, 1 \leq j \leq m\}$ gde redovi $G = \{\vec{g}_1, \dots, \vec{g}_n\}$ predstavljaju gene, dok kolone $C = \{\vec{c}_1, \dots, \vec{c}_m\}$ predstavljaju

¹Po najnovijem istraživanju procenjuje se na najmanje 46831 humanih gena. [5]

	c_1	c_2	c_3	\dots	c_m
g_1	t_{11}	t_{12}	t_{13}	\dots	t_{1m}
g_2	t_{21}	t_{22}	t_{23}	\dots	t_{2m}
g_3	t_{31}	t_{32}	t_{33}	\dots	t_{3m}
\dots	\dots	\dots	\dots	\dots	\dots
g_n	t_{n1}	t_{n2}	t_{n3}	\dots	t_{nm}

Tabela 1: Format podataka

ćelije. Svako polje matrice ekspresije t_{ij} predstavlja broj transkripta i -tog gena u j -toj ćeliji. Opisani format podataka je dat u tabeli 1.

Dati podaci se sastoje iz dva dela i zapisani su u *CSV* formatu. Za učitavanje podataka kao i za svaku sledeću obradu koristi se programski jezik *R*. Svaki navedeni kôd prati ispis rezultata koji počinje sa oznakom `#`. Konkretno u ovom slučaju se učitavaju obe datoteke. Prvu datoteku smeštamo u okvir podataka koji skraćeno nazivamo *HESC.CO.1* a drugu u *HESC.CO.2* i ispisujemo njihove dimenzije.

```
HESC.CO.1 = read.csv("data/011_Human_embryonic_stem_cell_and_cortical_organoids_csv.csv",
  header = TRUE, row.names = 1)
HESC.CO.2 = read.csv("data/012_Human_embryonic_stem_cell_and_cortical_organoids_csv.csv",
  header = TRUE, row.names = 1)

dim(HESC.CO.1)

## [1] 31221 2083

dim(HESC.CO.2)

## [1] 31221 1864
```

U ispisu rezultata vidi se da obe datoteke sadrže isti broj redova (gena), tj. 31221. Broj kolona (ćelija) je različit i u prvoj datoteci iznosi 2083 dok je u drugoj 1864. Kako je broj u oba slučaja veliki, radi lakšeg uvida ispisujemo samo prvih 10 redova i kolona.

```
HESC.CO.1[1:10, 1:10]

##          X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
```

```

## hg38_A1BG      0 0 0 0 0 0 0 0 0 0
## hg38_A1BG-AS1 0 0 0 1 0 0 0 0 0 1
## hg38_A1CF     0 0 0 0 0 0 0 0 0 0
## hg38_A2M       0 0 0 0 0 0 0 0 0 0
## hg38_A2M-AS1  0 3 0 0 0 0 0 0 0 0
## hg38_A2ML1    0 0 0 0 0 0 0 0 0 0
## hg38_A2ML1-AS1 0 0 0 0 0 0 0 0 0 0
## hg38_A2ML1-AS2 0 0 0 0 0 0 0 0 0 0
## hg38_A3GALT2  0 0 0 0 0 0 0 0 0 0
## hg38_A4GALT   0 0 0 1 1 0 0 0 0 0

```

HESC.CO.2[1:10, 1:10]

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
## hg38_A1BG	0	0	0	0	0	0	0	0	0	0
## hg38_A1BG-AS1	1	0	0	0	0	0	0	0	0	1
## hg38_A1CF	0	0	0	0	0	0	0	0	0	0
## hg38_A2M	0	0	0	0	0	0	0	0	0	0
## hg38_A2M-AS1	0	1	0	2	0	2	2	0	0	1
## hg38_A2ML1	0	0	0	0	0	0	0	0	0	0
## hg38_A2ML1-AS1	0	0	0	0	0	0	0	0	0	0
## hg38_A2ML1-AS2	0	0	0	0	0	0	0	0	0	0
## hg38_A3GALT2	0	0	0	0	0	0	0	0	0	0
## hg38_A4GALT	0	0	1	0	0	0	1	0	0	0

Primetimo da svi nazivi gena sadrže isti prefiks. Naime prefiks *hg38* označava da su podaci vezani za verziju 38. humanog genoma. Nazivi ćelija su označeni sa X_i gde i predstavlja redni broj ćelije.

Ulagni podaci se interpretiraju na različit način u zavisnosti da li se posmatraju geni ili ćelije. U prvom slučaju, redove posmatramo kao instance dok kolone označavaju vrednosti atributa. Kod posmatranja ćelija, instance predstavljaju kolone dok attribute predstavljaju redovi pa se u tom slučaju vrši transponovanje kako bi se podaci doveli do standardnog oblika. Takođe treba voditi računa da je broj gena isti dok je broj ćelija različit. Zbog svega ovoga neophodno je različito preprocesiranje pa nadalje razdvajamo postupak u zavisnosti da li posmatramo gene ili ćelije.

3 Klasterovanje gena

3.1 Analiza i preprocesiranje podataka

U prethodnom pogavlju isписан је само део података где се виђа да су вредности целобројне. Потребно је још проверити да ли постоје недостајуће вредности. Нaredним kodom se то проверава.

```
sum(sapply(HESC.CO.1, function(x) sum(is.na(x))))  
## [1] 0  
  
sum(sapply(HESC.CO.2, function(x) sum(is.na(x))))  
## [1] 0
```

У оба slučaja је резултат 0 па закључујемо да подаци не садрže недостајуће вредности. Следеће што је потребно урадити јесте укланjanje података који не доносе додатне информације, тј. чија је вредност кроз ћелије константна. У наšem slučaju укланjamо све one gene који nemaju nikakvu ekspresiju, тј. чија је вредност кроз ћелије jednak nuli. Treba napomenuti да, како је број gena u оба skupa jednak, могуће je razmatrati vrednosti na objedinjenom skupu. Za sada odlučujemo se да posmatramo одвојено. Укланjamо redove који садрже све nule i испisujemo број preostalih gena.

```
G.HESC.CO.1 = HESC.CO.1[apply(HESC.CO.1, 1, function(row) any(row !=  
0)), ]  
G.HESC.CO.2 = HESC.CO.2[apply(HESC.CO.2, 1, function(row) any(row !=  
0)), ]  
dim(G.HESC.CO.1)  
  
## [1] 20571 2083  
  
dim(G.HESC.CO.2)  
  
## [1] 20387 1864
```

Za razliku od прошlih 31221 gena, број gena је у првом skupu smanjen na 20571 dok u drugom na 20387 што је približno za trećinu.

Pre samog klasterovanja potrebno je izračunati matricu sličnosti. Kako se ne bavimo biološkim kontekstom ćelija i gena ne znamo da li neke kolone sadrže značajno veće vrednosti od drugih kolona. Takvo odstupanje bi se zasigurno odrazilo na matricu sličnosti tako što bi se kolonama (ćelijama) sa znatno većim vrednostima dalo više na značaju. To proveravamo izračunavanjem maksimalnog i minimalnog opsega vrednosti kolona (ćelija).

```
col.range.1 = data.frame(min = sapply(G.HESC.CO.1, min), max = sapply(G.HESC.CO.1,
                                max))
col.range.2 = data.frame(min = sapply(G.HESC.CO.2, min), max = sapply(G.HESC.CO.2,
                                max))

apply(col.range.1, 2, max)

##   min   max
##     0 3069

apply(col.range.1, 2, min)

## min max
##    0  59

apply(col.range.2, 2, max)

##   min   max
##     0 2295

apply(col.range.2, 2, min)

## min max
##    0  49
```

U prvom skupu podataka kolona sa najmanjim opsegom uzima vrednosti u rasponu [0, 59] dok kolona sa najvećim opsegom [0, 3069]. Veliki jaz je i kod drugog skupa podataka. Radi dodatnog ispitivanja računamo srednju vrednost, medijanu i standardnu devijaciju kolona. Kod srednje vrednosti i medijane uzimamo samo maksimalnu vrednost jer je poznato da su sve vrednosti veće od nule pa nam minimalna nije od interesa.

```

max(apply(G.HESC.CO.1, 2, median))

## [1] 1

max(apply(G.HESC.CO.2, 2, median))

## [1] 0

max(apply(G.HESC.CO.1, 2, mean))

## [1] 6.077925

max(apply(G.HESC.CO.2, 2, mean))

## [1] 3.823711

min(apply(G.HESC.CO.1, 2, sd))

## [1] 1.894174

max(apply(G.HESC.CO.1, 2, sd))

## [1] 43.80583

min(apply(G.HESC.CO.2, 2, sd))

## [1] 1.776009

max(apply(G.HESC.CO.2, 2, sd))

## [1] 27.95808

```

U tabeli 2 su predstavljeni ukupni rezultati. Stoji napomena da se ove mere odnose na kolone.

Gore izvršene analize su veoma grube i ne dopuštaju tačno trvđenje ili zaključivanje ali na osnovu njih se može naslutiti da na pojavu velikih opsega vrednosti po kolonama, utiče mali broj gena koji imaju veliku vrednost ekspresije. Zbog toga dalje analiziramo gene. Kako je dimenzionalnost velika, ovakve gene je teško vizualizovati bez prethodne redukcije dimenzija. Postavlja se pitanje da li postoje geni koji su važniji od drugih. Preciznije,

	G.HESC.CO.1	G.HESC.CO.2
Minimalni opseg	[0, 59]	[0, 49]
Maksimalni opseg	[0, 3069]	[0, 2295]
Maksimalna medijana	1	0
Maksimalna srednja vrednost	6.077925	3.823711
Minimalna SD	1.894174	1.776009
Maksimalna SD	43.80583	27.95808

Tabela 2: Gruba statisticka analiza kolona

da li je moguće razdvojiti gene na one sa većim informacionim doprinosom i na one koji su irrelevantni za dalju analizu i klasterovanje. Autori, koji upućuju na metode koje se tom prilikom koriste, tvrde da odstranjivanje irrelevantnih gena u velikoj meri utiče na redukciju dimenzionalnosti i na samo klasterovanje[7]. Kako nije poznato da li je nad datim podacim izvršena neka vrsta filtriranja kao i do koje mere, pokušaćemo da nekim naivnim i jednostavnim statističkim alatom izvučemo informacije o vrednostima samih gena. U tu svrhu, za svaki gen se računa medijana i srednja vrednost transkripta.

```
gene_mean_1 = apply(G.HESC.CO.1, 1, mean)
```

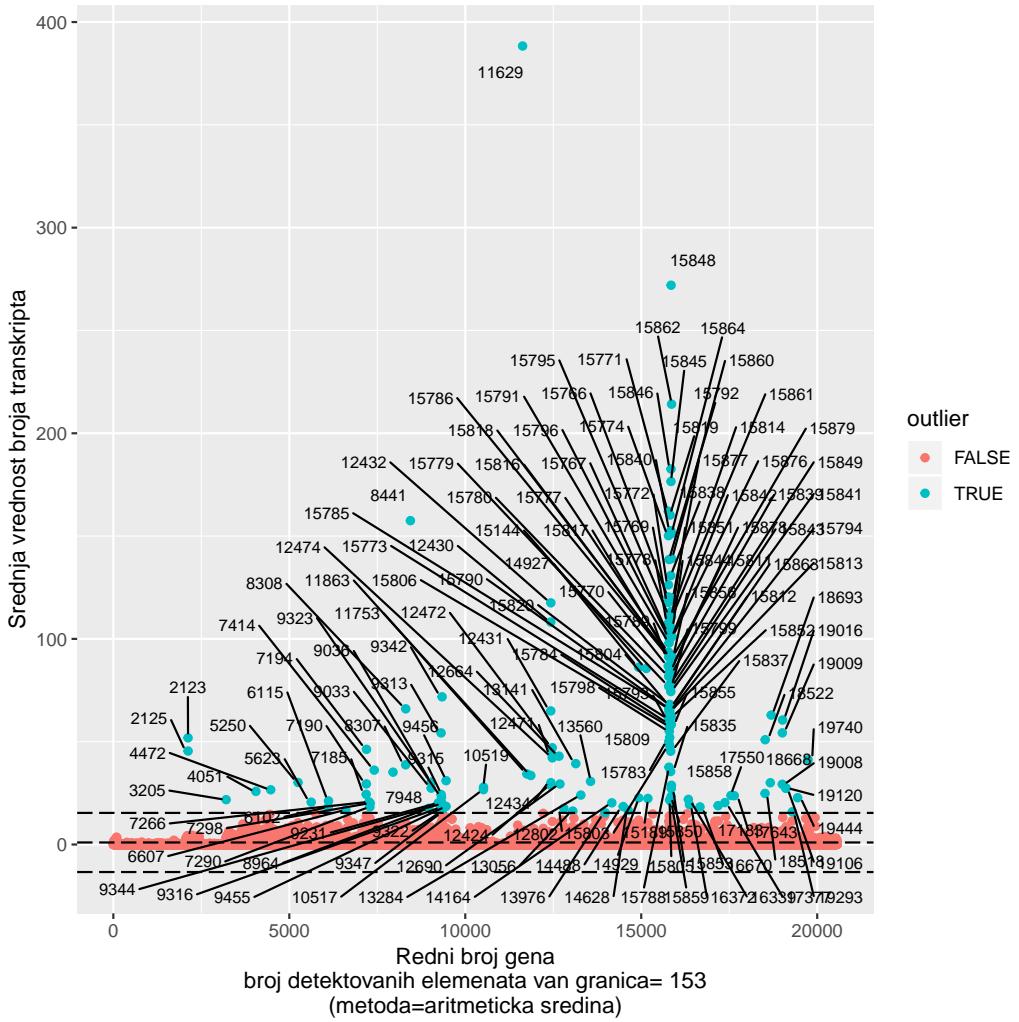
```
gene_mean_2 = apply(G.HESC.CO.2, 1, mean)
```

Prvo su izračunate srednje vrednost. Sada kada svaki gen ima jednu vrednost, pozivamo funkciju *outlier* koja detektuje elemente van granica. Metoda koja se tom prilikom koristi zasnovana je na standardnoj devijaciji pri čemu se za prag uzima razdaljina dve standardne devijacije od aritmetičke sredine. Isti postupak vršimo za oba skupa podataka.

```
outlier(gene_mean_1, addthres = TRUE)
```

```
outlier(gene_mean_2, addthres = TRUE)
```

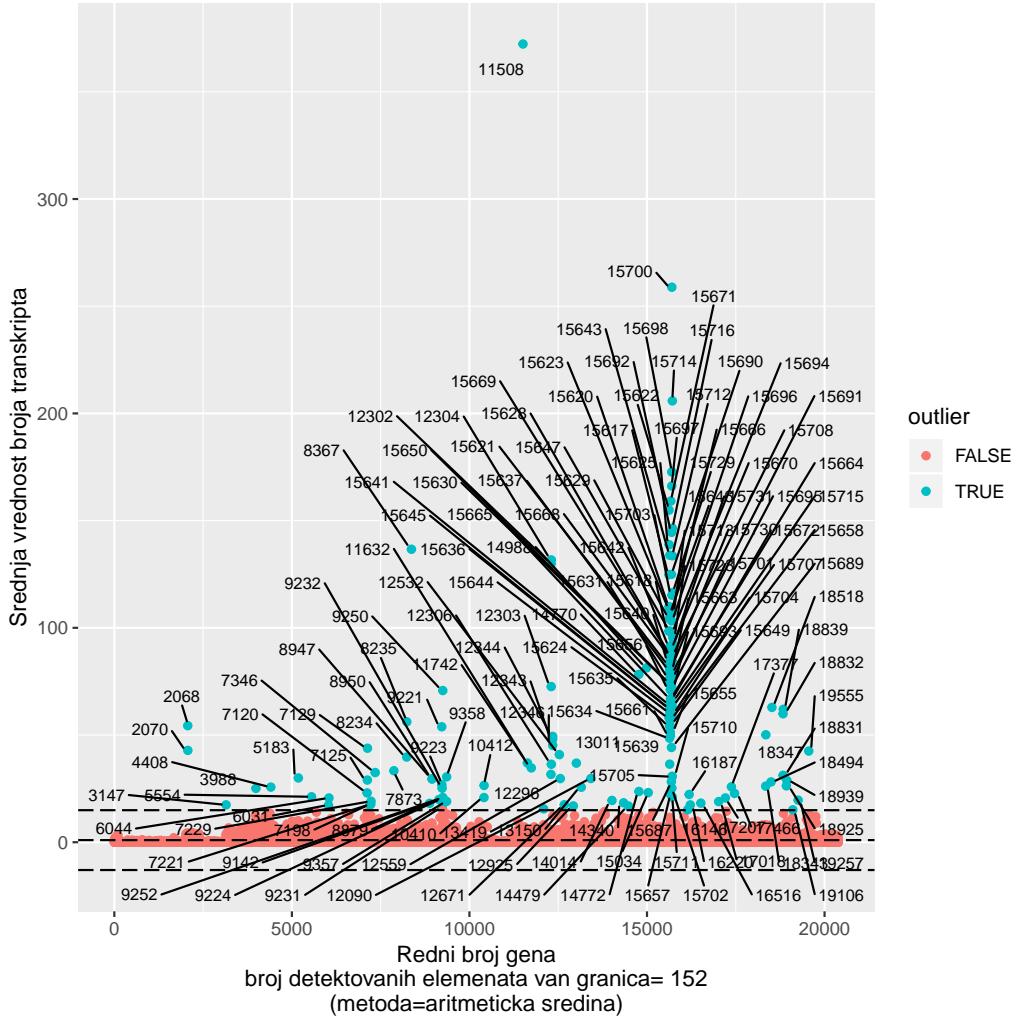
Funkcija *outlier* generiše grafik gde *x*-osa označava redni broj gena dok *y*-osa označava srednju vrednost transkripta. Geni su predstavljeni tačkicom plave boje ukoliko su detektovani kao elementi van granica i crvenom ako



Slika 1: Prvi skup podataka: Prosek srednjih vrednosti

to nisu. Na grafiku 1 se jasno vidi pik tj. da jedna grupa gena odstupa od drugih. Isti je slučaj i sa drugim skupom podataka 2. Ponavlja se isti postupak ali se za meru uzima medijana.

```
gene_medians_1 = apply(G.HESC.CO.1, 1, median)
gene_medians_2 = apply(G.HESC.CO.2, 1, median)
```

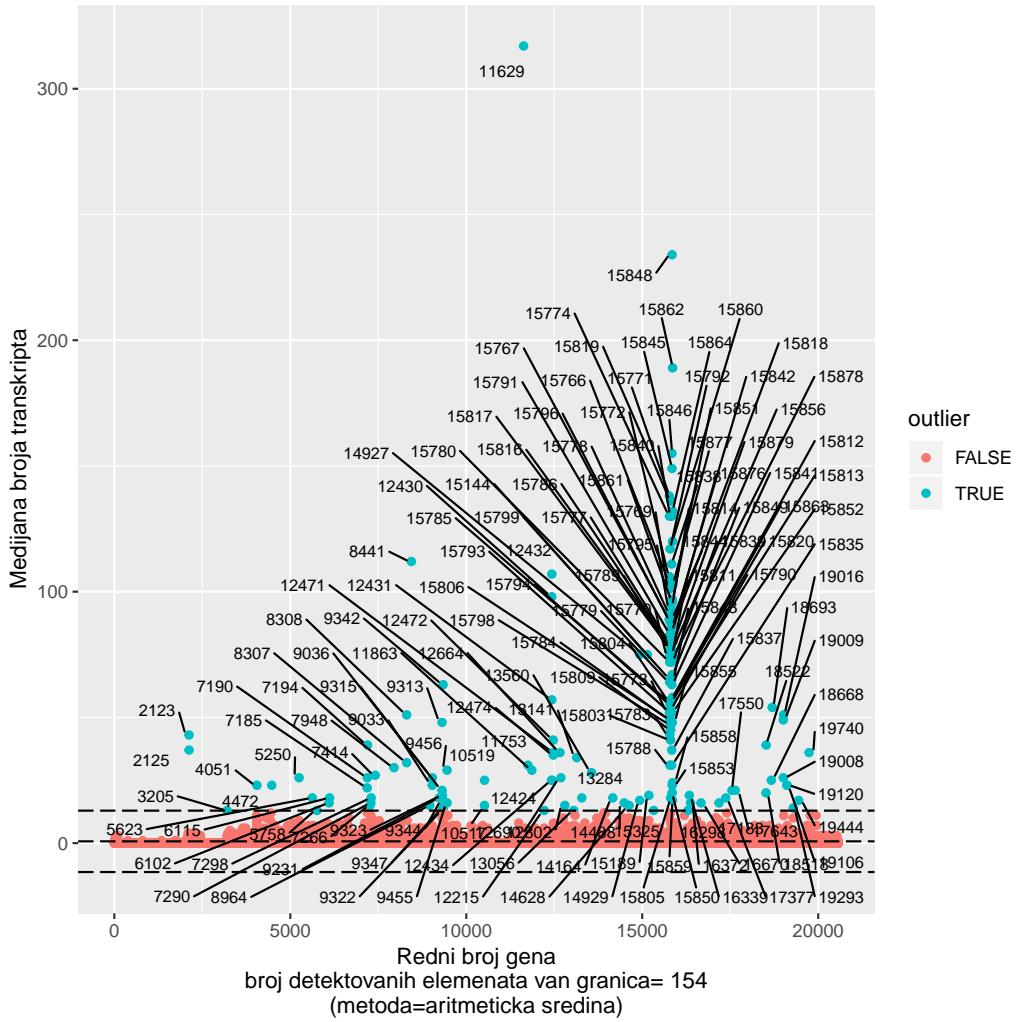


Slika 2: Drugi skup podataka: Prosek srednjih vrednosti

```
outlier(gene_medians_1, addthres = TRUE)
```

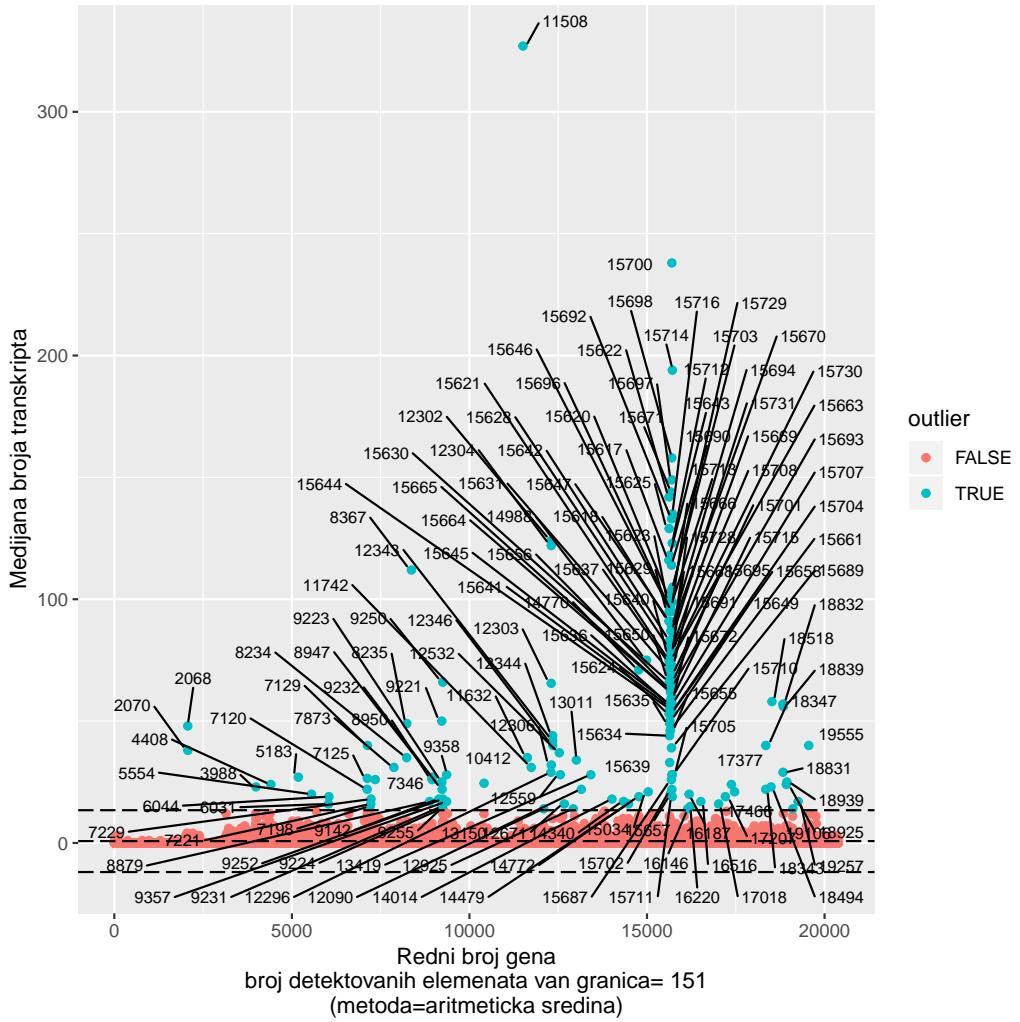
```
outlier(gene_medians_2, addthres = TRUE)
```

Medijana ne menja značajno grafik 34. I dalje se može uočiti grupa gena koja ima veću vrednost kroz celije. Šta više, može se reći da postoji grupa gena (na grafiku obeleženi rednim brojevima između 15000 i 16000)



Slika 3: Prvi skup podataka: Prosecne vrednosti medijana

koja aktivnije učestvuje u célijima tj. njihov nivo ekspresije je znatno veći u poređenju sa ostalim genima. Ovo tvrđenje se uzima sa velikom rezervom iz više razloga. Prvo, računanjem srednje vrednosti i medijane je dimenzionalnost svedena na jednu dimenziju tj. na jedan atribut koji kao takav gubi većinu informacija o genu. Drugo, ne znamo da li metoda detekcije elemenata van granica odgovara prirodi ovakvih podataka. I pored toga, smatra se da ovaj postupak identificuje zanimljivu grupu gena. Kasnije će se ispostaviti



Slika 4: Drugi skup podataka: Prosečne vrednosti medijana

da se ta grupa gena "lepo" klasteruje.

Daljom analizom grafika uočavaju se tačke (po jedna na oba grafika) koje značajno odstupaju od ostalih. To su geni sa rednim brojem 11629 i 11508. Njihov naziv i vrednost ispisujemo sledećim kodom:

```
gene_medians_1[11629]
```

```
## hg38_MALAT1
```

```
gene_medians_2[11508]
```

```
## hg38_MALAT1
## 327
```

U pitanju je isti gen, MALAT1². Ovaj gen ćemo kvalifikovati kao element van granica i odstraniti ga iz skupa podataka.

```
G.HESC.CO.1 = G.HESC.CO.1[-11629,]
G.HESC.CO.2 = G.HESC.CO.2[-11508,]
```

Tumačenjem gore dobijenih grafika mogu se izvući neka opažanja. Prvo, svi grafici su jako slični. I u slučaju kada je za meru uzeta srednja vrednost i kada je to bila medijana. Jasno je uočljiv pik koji čine tačke sa jako izraženim vrednostima. Ovakvo ponašanje je isto u oba skupa podataka tj. ponašanje se nije promenilo u zavisnosti da li gledamo podatke iz jednog ili drugo skupa podataka. Prisetimo se da je broj gena pre eliminisanja nula redova bio jednak u oba skupa. Broj ćelija se razlikovao. To dovodi do zaključka da je ponašanje gena slično, nezavisno od toga da li se radi o skupu ćelija iz prve datoteke ili iz druge datoteke. Kako se ne bi ponavljali isti postupci nadalje, spojićemo ova dva skupa. Postojala je mogućnost da se prvo izvrši spajanje a nakon toga eliminacija nula redova. U ovom našem slučaju sada, broj gena više nije jednak pa spajanje vršimo na osnovu zajedničkih gena, tj. preseku redova.

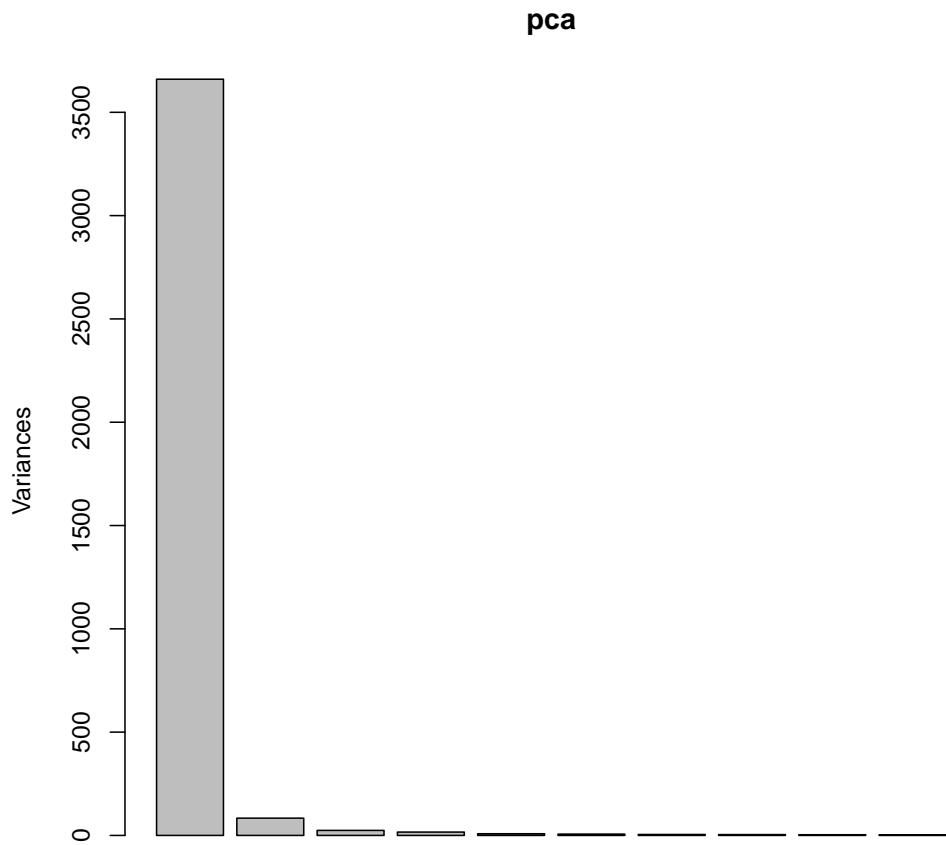
```
G.HESC.CO = merge(G.HESC.CO.1, G.HESC.CO.2, by = "row.names")
G.HESC.CO = column_to_rownames(G.HESC.CO, var = "Row.names")
```

Spojeni podaci se smeštaju u okvir podataka sa oznakom *G.HESC.CO*.

U prethodnoj analizi su razmatrani grafici koji su bili zasnovani na medijanama i srednjim vrednostima gena. Oni opisuju samo jedan aspekt ponašanja gena i ne predstavljaju verodostojnu vizuelizaciju samih gena. Jedan od načina da se približno vizuelizuju geni jeste da se prvo upotrebi metoda *PCA* (eng. *Principal Component Analysis*) za redukciju atributa. Sledeći kod izvršava datu metodu i za rezultat daje komponente (attribute) kao i broj varansi koje one sadrže.

²<https://en.wikipedia.org/wiki/MALAT1>

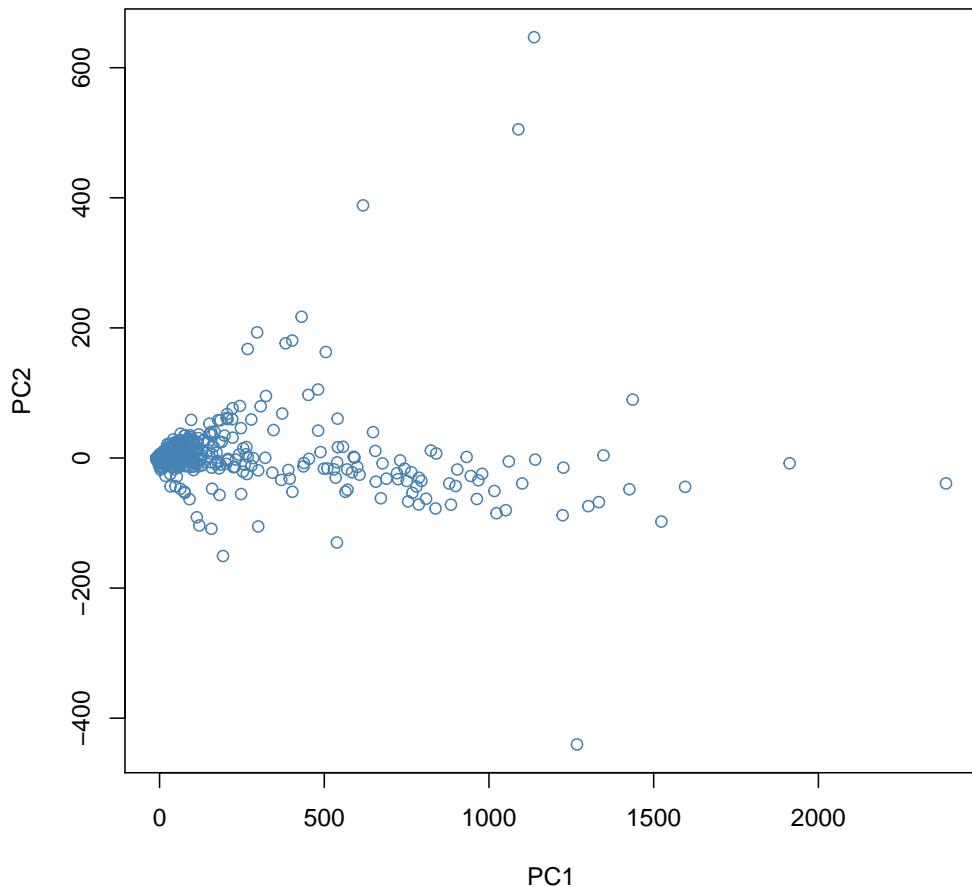
```
pca = prcomp(data.matrix(G.HESC.CO), scale. = TRUE)
plot(pca)
```



Slika 5: Redukcija dimenzija

Na plotu 5 je uočljivo da najviše varijansi sadrži prva komponenta. Već kod druge komponente broj varijansi naglo opada. Uzimaju se prve dve komponente (PC_1, PC_2) za atribute gena i na osnovu tih atributa se predstavljaju na grafiku.

```
plot(as.data.frame(pca$x)[, 1:2], col = "steelblue")
```



Slika 6: Prikaz redukovanih podataka

Kao i u prethodnom načinu analize, gde smo računali medijanu i srednju vrednost za svaki gen, vidi se da je najviše gena grupisano u vrednostima bliskim ili jednakim nuli i da postoji manja grupa gena koja od toga odstupa. Prethodna metoda 3 je takođe identifikovala manju grupu gena kao elemente van granica. Na osnovu nje razdvajamo te dve grupe kako bi ih

zasebno razmatrali. Naglašavamo da se spomenuta metoda koristi samo kao kriterijum za razdvajanje, nikako kao alat za redukciju kolona. Podaci i dalje sadrže sve atribute bez ikakve modifikacije. Geni koji su detektovani kao elementi van granica nadalje zovemo *aktivnija grupa gena*. Tu grupu smeštamo u *OUT.G.HESC.CO* a ostale u *NOTOUT.G.HESC.CO* okvir podataka.

```
OUT.G.HESC.CO = G.HESC.CO %>% rownames_to_column("gene") %>%
  filter(gene_medians$row_number() > outlier_upper) %>% column_to_rownames("gene")
NOTOUT.G.HESC.CO = G.HESC.CO %>% rownames_to_column("gene") %>%
  filter(gene_medians$row_number() <= outlier_upper) %>% column_to_rownames("gene")
```

3.2 Aktivnija grupa gena

Podaci o ekspresiji gena sadrže ključne informacije potrebne za razumevanje vitalnih bioloških procesa. Napredak tehnologije u merenju gena je rezultiralo generisanjem ogromnih podataka. Dokazano je da je grupisanje podataka korisno u poznavanju prirodne strukture opisane ekspresijom gena, razumevanju funkcija gena, ćelijskih procesa, otkrivanja korisnih informacija o samim genima[4]. Kako bi geni bili grupisani, potrebno je odrediti mere koje najbolje opisuju sličnost gena i definisati samu proceduru grupisanja. U terminima računarstva, potrebno je pronaći prikladnu mjeru sličnosti kao i algoritam klasterovanja.

U ovom trenutku su na raspolaganju dva skupa podataka. Prvi skup sadrži gene čija je aktivnost kroz ćelije slabije izražena a koji obuhvata veliku većinu gena. Drugi skup predstavlja aktivniju grupu gena i sadrži 152 gena.

```
dim(OUT.G.HESC.CO)
```

```
## [1] 152 3947
```

Jako teško je dati opšte pravilo za odabir prikladne mere sličnosti. U velikom broju slučajeva, odabir zavisi od konkretnih podataka i ciljeva obrade a izbor mere sličnosti utiče na krajnje rezultate i može da bude presudan za njihov kvalitet[2]. Mali skup podataka dopušta odabir mere sličnosti i metode klasterovanja na osnovu eksperimentalnog iskustva. Takođe, interpretacija rezultata je jednostavnija. Konkretno, odabrano je sakupljajuće hijerarhijsko klasterovanje čijom analizom rezultata, tj. upoređivanjem dendograma, biramo adekvatnu mjeru sličnosti. U narednoj tabeli 3 je dat ispis svih mera

	Formula
Euklidsko rastojanje	$\sqrt{\sum_i (x_i - y_i)^2}$
Menhetn rastojanje	$\sum_i x_i - y_i $
Kosinusno rastojanje	$(a * b) / (a * b)$
Čebiševljevo rastojanje	$\max_i x_i - y_i $
Mahalanobisovo rastojanje	$\sqrt{(x - y) \Sigma^{-1} (x - y)}$
Fazi-Žakardovo rastojanje	$\sum_i (\min x_i, y_i) / \sum_i (\max x_i, y_i)$
Helingerovo rastojanje	$\sqrt{\sum_i (\sqrt{x_i / \sum_i x} - \sqrt{y_i / \sum_i y})^2}$
Sorgelovo rastojanje	$\sum_i x_i - y_i / \sum_i \max x_i, y_i$

Tabela 3: Mere sličnosti (rastojanja)

koje se tom prilikom koriste. Dendogrami su grupisani na osnovu mere. Svaka grupa sadrži četiri dendograma i predstavlja jednu meru sličnosti dok se u okviru grupe koriste različite metode za određivanje rastojanja između klastera (*ward.D*, *ward.D2*, *complete*, *average*).

Ispod sledi kôd koji izračunava matrice rastojanja i izvršava hijerarhijsko klasterovanje.

```

euclidean.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "euclidean")
manhattan.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "manhattan")
cosine.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "cosine")
maximum.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "maximum")
mahalanobis.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "mahalanobis",
    inverted = TRUE)
fJaccard.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "fJaccard")
hellinger.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "hellinger")
soergel.dist = parDist(data.matrix(OUT.G.HESC.CO), method = "soergel")

ward.D.euclidean.hc = hclust(euclidean.dist, method = "ward.D")
ward.D2.euclidean.hc = hclust(euclidean.dist, method = "ward.D2")
complete.euclidean.hc = hclust(euclidean.dist, method = "complete")

```

```

average.euclidean.hc = hclust(euclidean.dist, method = "average")

ward.D.manhattan.hc = hclust(manhattan.dist, method = "ward.D")
ward.D2.manhattan.hc = hclust(manhattan.dist, method = "ward.D2")
complete.manhattan.hc = hclust(manhattan.dist, method = "complete")
average.manhattan.hc = hclust(manhattan.dist, method = "average")

ward.D.cosine.hc = hclust(cosine.dist, method = "ward.D")
ward.D2.cosine.hc = hclust(cosine.dist, method = "ward.D2")
complete.cosine.hc = hclust(cosine.dist, method = "complete")
average.cosine.hc = hclust(cosine.dist, method = "average")

ward.D.maximum.hc = hclust(maximum.dist, method = "ward.D")
ward.D2.maximum.hc = hclust(maximum.dist, method = "ward.D2")
complete.maximum.hc = hclust(maximum.dist, method = "complete")
average.maximum.hc = hclust(maximum.dist, method = "average")

ward.D.mahalanobis.hc = hclust(mahalanobis.dist, method = "ward.D")
ward.D2.mahalanobis.hc = hclust(mahalanobis.dist, method = "ward.D2")
complete.mahalanobis.hc = hclust(mahalanobis.dist, method = "complete")
average.mahalanobis.hc = hclust(mahalanobis.dist, method = "average")

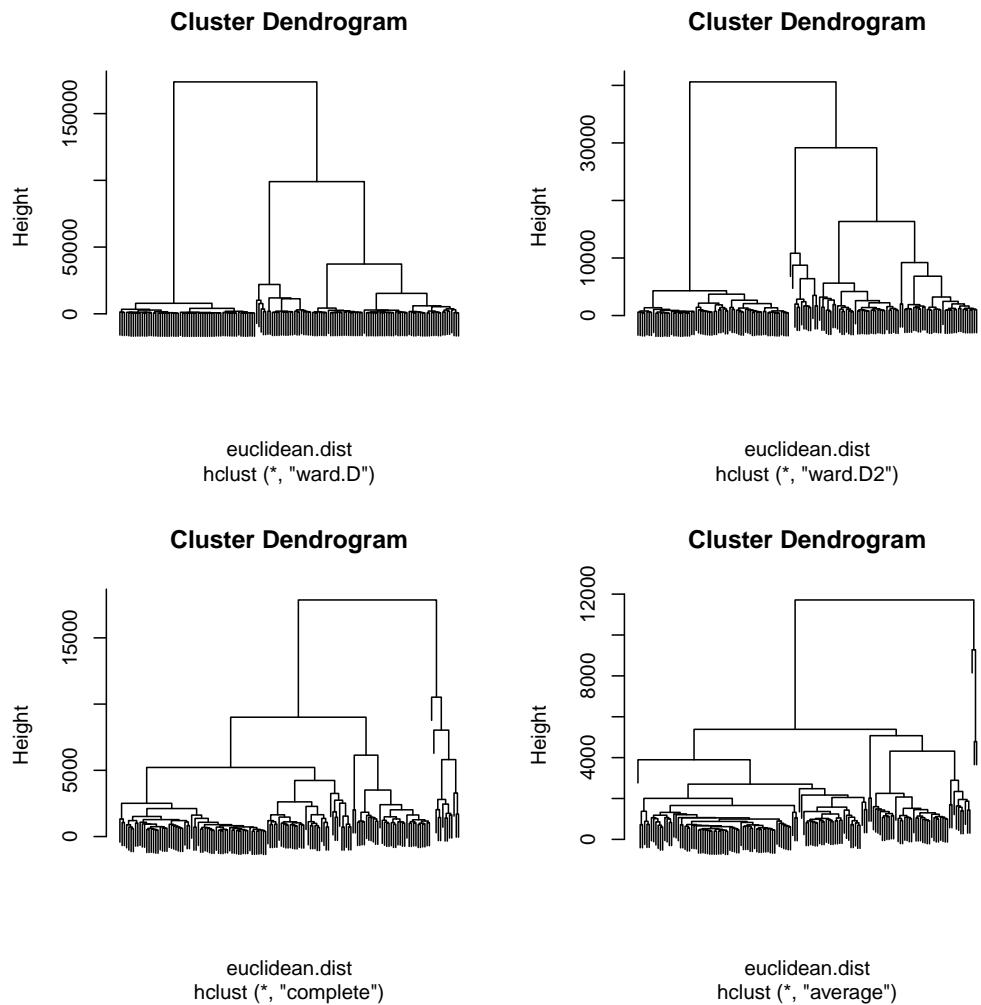
ward.D.fJaccard.hc = hclust(fJaccard.dist, method = "ward.D")
ward.D2.fJaccard.hc = hclust(fJaccard.dist, method = "ward.D2")
complete.fJaccard.hc = hclust(fJaccard.dist, method = "complete")
average.fJaccard.hc = hclust(fJaccard.dist, method = "average")

ward.D.hellinger.hc = hclust(hellinger.dist, method = "ward.D")
ward.D2.hellinger.hc = hclust(hellinger.dist, method = "ward.D2")
complete.hellinger.hc = hclust(hellinger.dist, method = "complete")
average.hellinger.hc = hclust(hellinger.dist, method = "average")

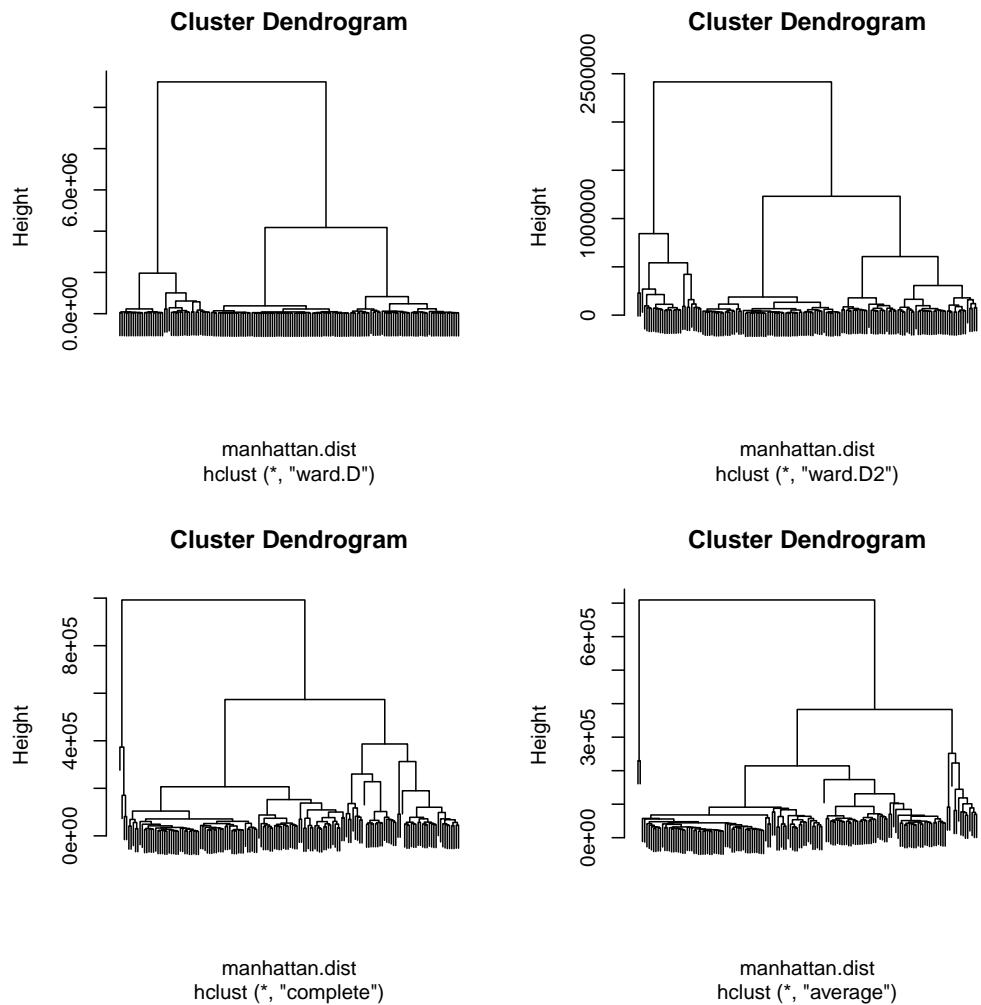
ward.D.soergel.hc = hclust(soergel.dist, method = "ward.D")
ward.D2.soergel.hc = hclust(soergel.dist, method = "ward.D2")
complete.soergel.hc = hclust(soergel.dist, method = "complete")
average.soergel.hc = hclust(soergel.dist, method = "average")

```

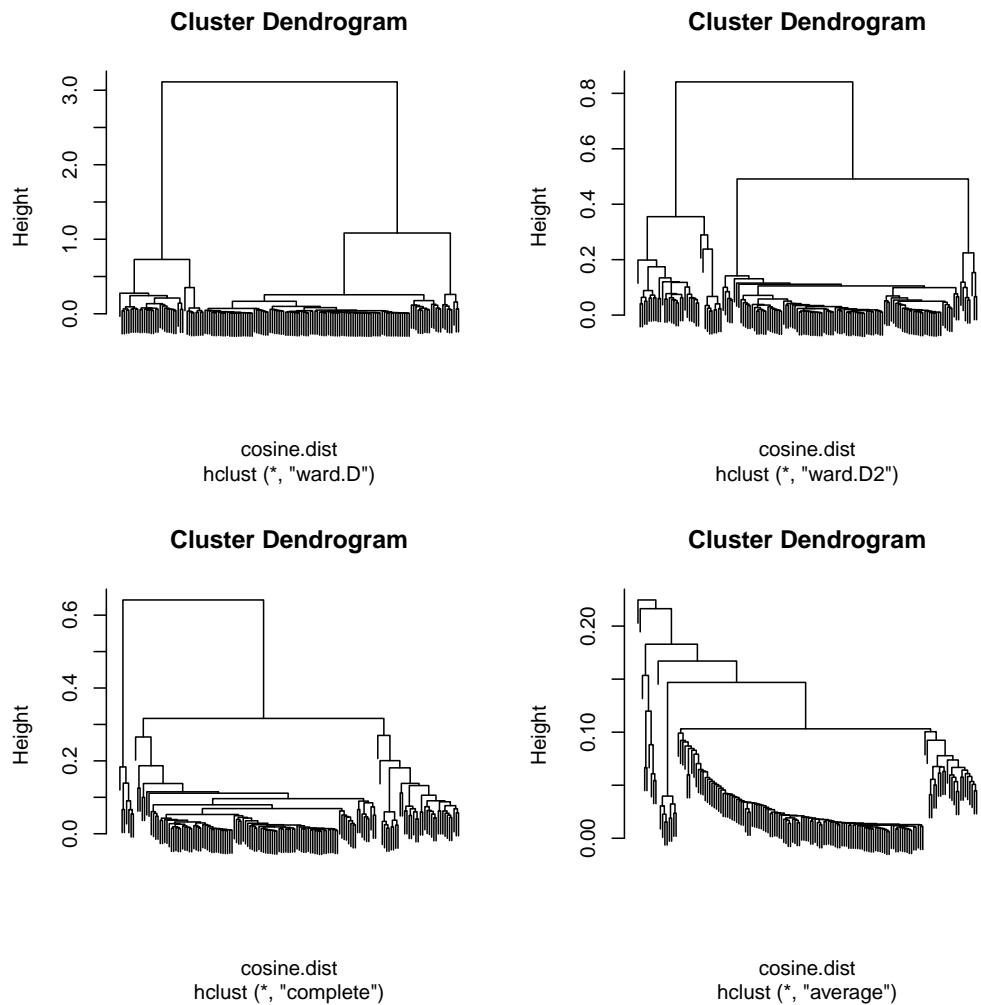
Dendograme analiziramo posmatrajući vertikalne linije. Veća dužina spomenute linije predstavlja bolje razdvojene klastere. Na osnovu vizuelne interpretacije dendograma biramo Mahalanobisovo rastojanje sa *ward.D* meto-



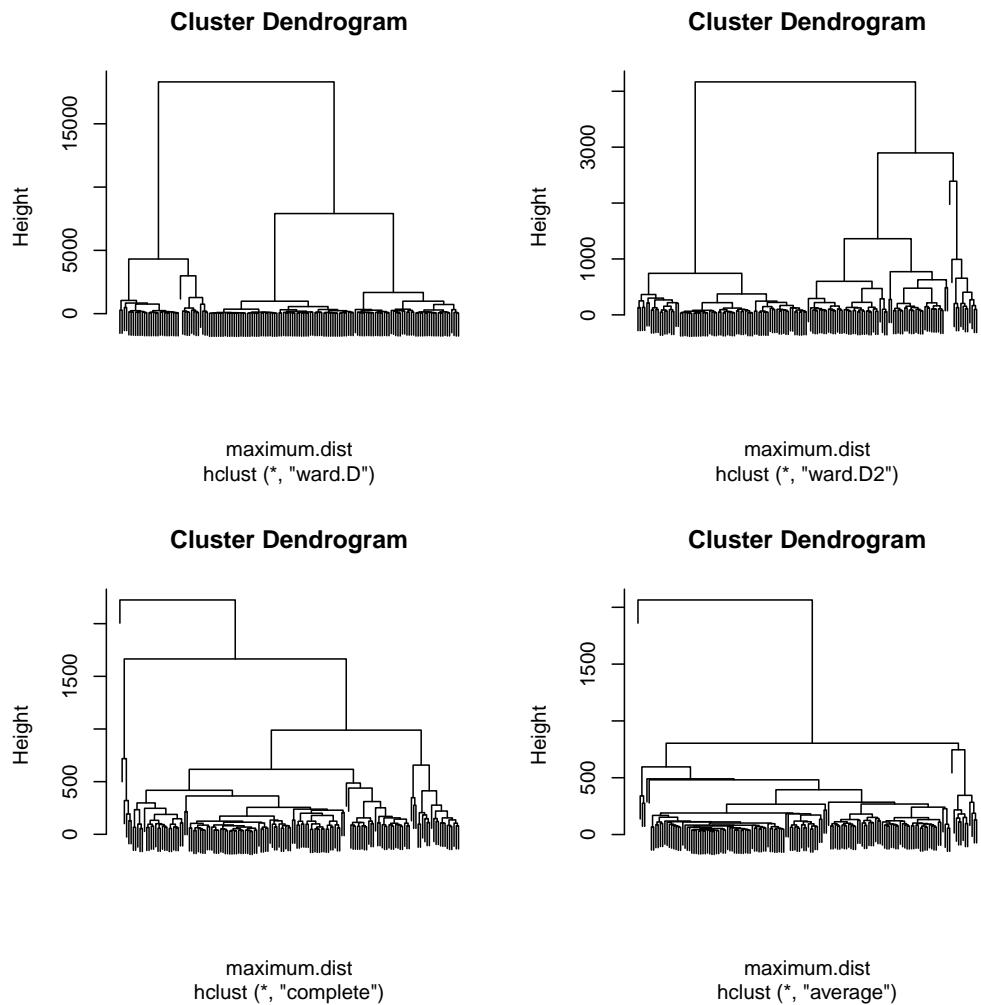
Slika 7: Dendogrami: Euklidsko rastojanje



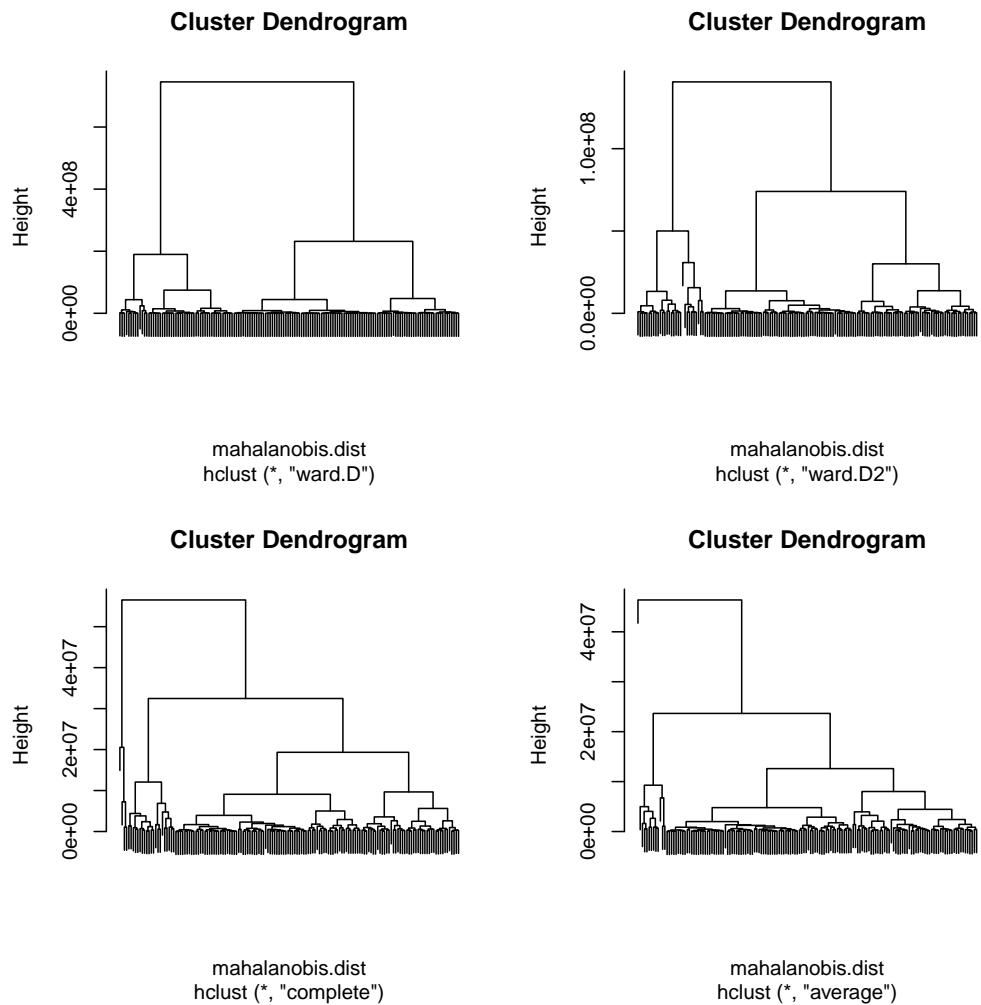
Slika 8: Dendogrami: Menhetn rastojanje



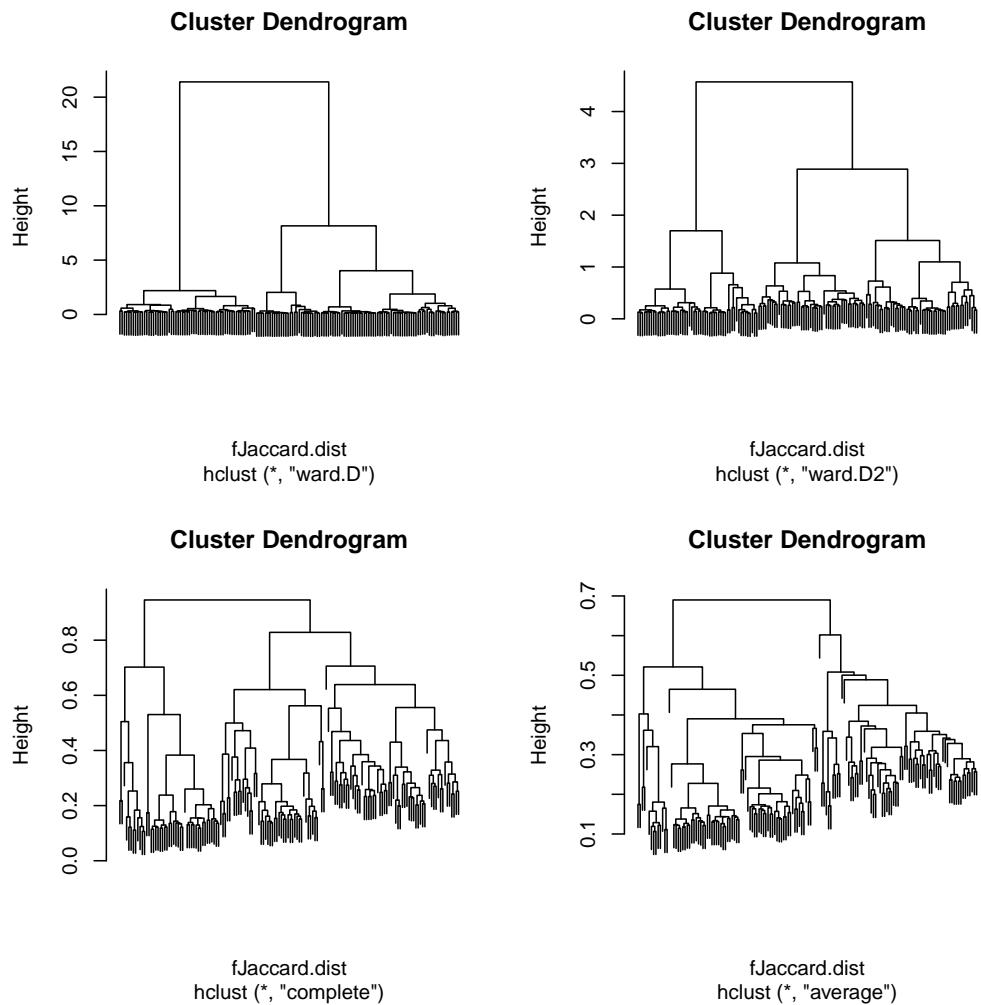
Slika 9: Dendogrami: Kosinusno rastojanje



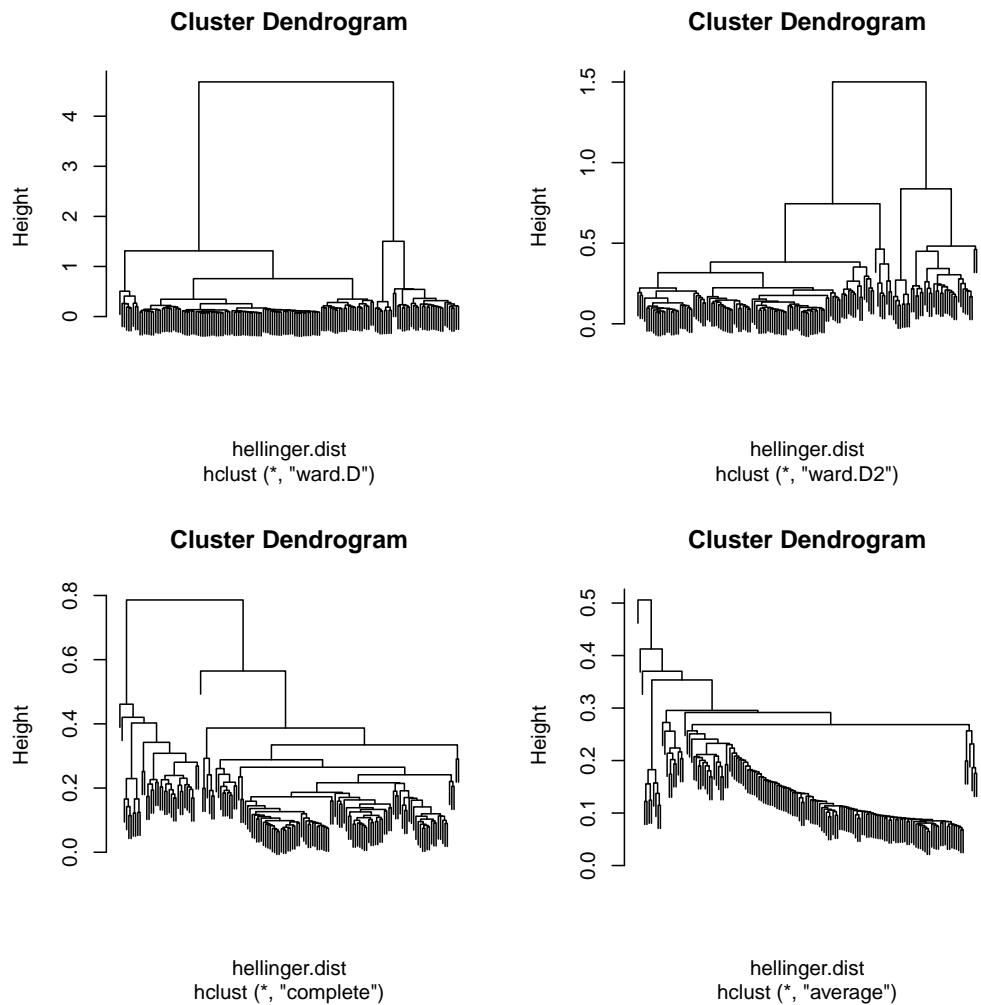
Slika 10: Dendogrami: Cebiševljevo rastojanje



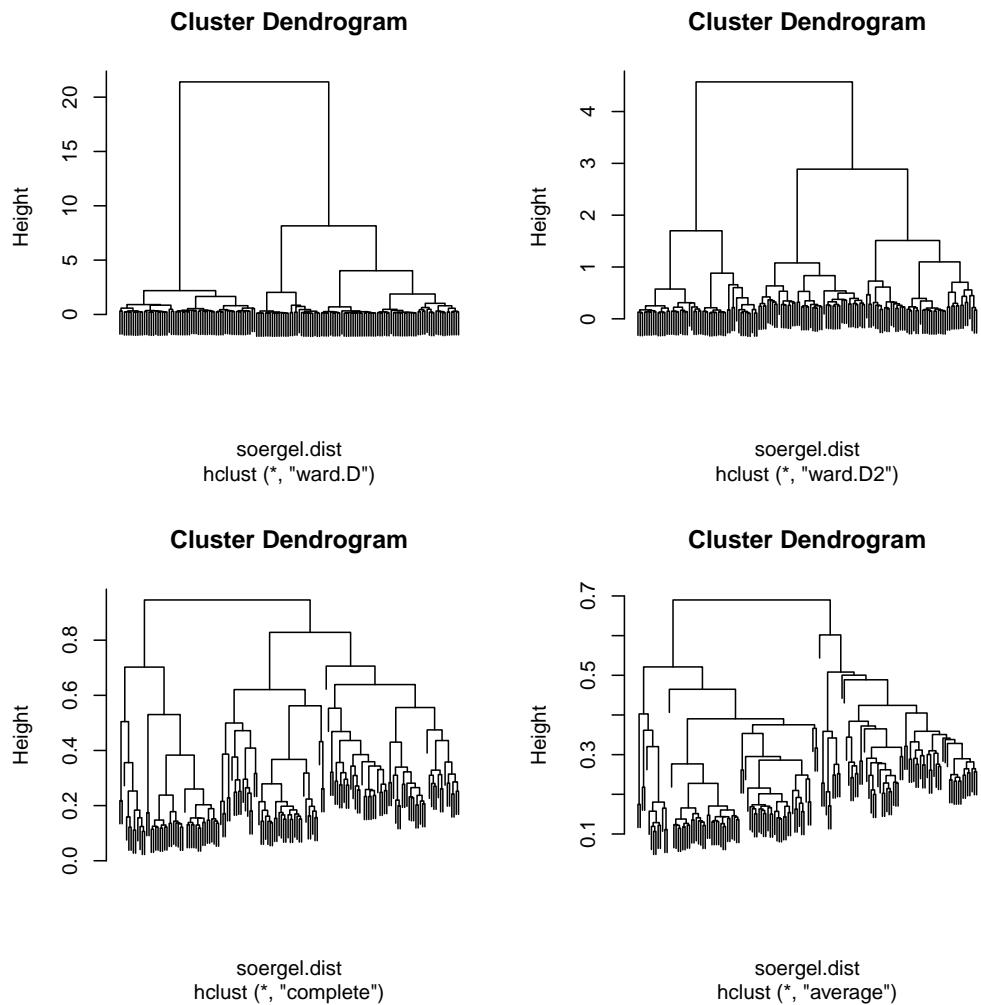
Slika 11: Dendogrami: Mahalanobisovo rastojanje



Slika 12: Dendogrami: Fazi-Žakardovo rastojanje



Slika 13: Dendogrami: Helingrovo rastojanje



Slika 14: Dendogrami: Sorgelovo rastojanje

dom gde raspoznaјemo osam klastera. U kodu ispod ispisujemo nazine gena i njima pripadajući klaster za izabranu meru:

```
cutree(ward.D.mahalanobis.hc, k=8)

##    hg38_ACTB    hg38_ACTG1    hg38_ATP5MC2    hg38_BTF3
##    1            1            2            2
##    hg38_CCNI    hg38_CFL1    hg38_CIRBP    hg38_COX4I1
##    2            3            3            3
##    hg38_COX7C    hg38_EEF1A1    hg38_EEF1B2    hg38_EEF2
##    3            3            2            1
##    hg38{EIF1    hg38{EIF3E    hg38{EIF3L    hg38ENO1
##    3            3            3            2
##    hg38_FAU    hg38_FTH1    hg38_FTL    hg38_GAPDH
##    2            1            4            5
##    hg38GSTP1    hg38H2AFZ    hg38H3F3B    hg38_HINT1
##    3            2            2            3
##    hg38_HMGA1    hg38_HMGB1    hg38_HMGN1    hg38_HMGN2
##    1            2            3            2
##    hg38_HNRNPA1    hg38_HNRNPA2B1    hg38_HNRNPC    hg38_HSP90AA1
##    4            3            3            3
##    hg38_HSP90AB1    hg38_LDHA    hg38_LDHB    hg38_MARCKSL1
##    2            2            2            2
##    hg38_MDK    hg38_MORF4L1    hg38_MT-ATP6    hg38_MT-C01
##    2            3            2            6
##    hg38_MT-C02    hg38_MT-C03    hg38_MT-CYB    hg38_MT-ND1
##    4            6            2            1
##    hg38_MT-ND2    hg38_MT-ND4    hg38_NACA    hg38_NAP1L1
##    1            1            1            2
##    hg38_NDUFA4    hg38_NNAT    hg38_NPM1    hg38_NUCKS1
##    3            3            1            2
##    hg38_PABC1    hg38_PKM    hg38_PPIA    hg38_PRDX2
##    2            3            3            3
##    hg38_PTMA    hg38_PTN    hg38_RACK1    hg38_RAN
##    7            3            7            3
##    hg38_RBmx    hg38_RPL10    hg38_RPL10A    hg38_RPL11
##    3            6            6            6
##    hg38_RPL12    hg38_RPL13    hg38_RPL13A    hg38_RPL14
##    7            5            6            4
##    hg38_RPL15    hg38_RPL18    hg38_RPL18A    hg38_RPL19
##    5            7            6            7
##    hg38_RPL21    hg38_RPL23    hg38_RPL23A    hg38_RPL24
##    7            1            4            4
##    hg38_RPL26    hg38_RPL27    hg38_RPL27A    hg38_RPL28
##    7            1            7            4
##    hg38_RPL29    hg38_RPL3    hg38_RPL30    hg38_RPL31
##    6            5            4            4
##    hg38_RPL32    hg38_RPL34    hg38_RPL35    hg38_RPL35A
##    6            6            1            7
##    hg38_RPL37    hg38_RPL37A    hg38_RPL38    hg38_RPL39
##    1            7            3            4
##    hg38_RPL4    hg38_RPL5    hg38_RPL6    hg38_RPL7
##    1            7            4            4
##    hg38_RPL7A    hg38_RPL8    hg38_RPL9    hg38_RPLP0
##    6            7            7            7
##    hg38_RPLP1    hg38_RPLP2    hg38_RPS10    hg38_RPS11
##    5            4            2            1
##    hg38_RPS12    hg38_RPS13    hg38_RPS14    hg38_RPS15
##    5            7            5            7
##    hg38_RPS15A    hg38_RPS16    hg38_RPS17    hg38_RPS18
##    6            7            6            8
##    hg38_RPS19    hg38_RPS2    hg38_RPS20    hg38_RPS21
##    8            8            7            2
##    hg38_RPS23    hg38_RPS24    hg38_RPS25    hg38_RPS27
##    6            4            2            4
##    hg38_RPS27A    hg38_RPS28    hg38_RPS29    hg38_RPS3
##    7            2            3            5
##    hg38_RPS3A    hg38_RPS4X    hg38_RPS5    hg38_RPS6
##    6            8            4            5
##    hg38_RPS7    hg38_RPS8    hg38_RPS9    hg38_RPSA
##    6            6            7            7
##    hg38_SERBP1    hg38_SET    hg38_SFRP1    hg38_SLC25A3
##    3            3            3            3
##    hg38_SOX2    hg38_SRPI4    hg38_STMN1    hg38_SUMO2
```

```

##      3      3      2      3
## hg38_TMSB10 hg38_TMSB4X hg38_TPI1 hg38_TPT1
##      2      1      2      4
## hg38_TUBA1A hg38_TUBA1B hg38_TUBB hg38_UBA52
##      2      1      4      2
## hg38_UBB hg38_UQCRH hg38_VIM hg38_VBX1
##      2      3      3      1

```

Prilikom provere rezultata kao smernica se može uzeti činjenica da nazivi gena mogu imati isti prefiks. To nam sugerije da postoji veza između njih. Prilikom čitanja tabele obratićemo pažnju na tu činjenicu ali je nećemo uzeti za pravilo.

Pregledom tabele vidimo da geni sa prefiksom *hg38_RP* ne pripadaju istom klaster iako imaju isti prefiks. Vraćamo se opet analizi dendograma i ovaj put biramo kosinusno rastojanje sa *ward.D* metodom.

```

cutree(ward.D.cosine.hc, k=8)

##      hg38_ACTB    hg38_ACTG1   hg38_ATP5MC2   hg38_BTF3
##      1           2           3           3
##      hg38_CCNI    hg38_CFL1    hg38_CIRBP   hg38_COX4I1
##      4           1           4           4
##      hg38_COX7C   hg38_EEF1A1   hg38_EEF1B2   hg38_EEF2
##      3           3           3           3
##      hg38{EIF1     hg38{EIF3E   hg38{EIF3L   hg38_ENO1
##      2           3           3           5
##      hg38_FAU     hg38_FTH1    hg38_FTL    hg38_GAPDH
##      3           1           2           2
##      hg38_GSTP1   hg38_H2AFZ   hg38_H3F3B   hg38_HINT1
##      1           1           4           4
##      hg38_HMGA1   hg38_HMGB1   hg38_HMGN1   hg38_HMGN2
##      3           1           1           1
##      hg38_HNRNPA1 hg38_HNRNPA2B1 hg38_HNRNPC  hg38_HSP90AA1
##      3           1           4           1
##      hg38_HSP90AB1 hg38_LDHA    hg38_LDHB   hg38_MARCKSL1
##      4           5           4           1
##      hg38_MDK     hg38_MORF4L1 hg38_MT-ATP6  hg38_MT-C01
##      1           1           6           6
##      hg38_MT-C02   hg38_MT-C03   hg38_MT-CYB  hg38_MT-ND1
##      6           6           6           6
##      hg38_MT-ND2   hg38_MT-ND4   hg38_NACA   hg38_NAP1L1
##      6           6           3           3
##      hg38_NDUFA4   hg38_NNAT    hg38_NPM1   hg38_NUCKS1
##      4           7           3           1
##      hg38_PABPC1   hg38_PKM     hg38_PPIA   hg38_PRDX2
##      3           4           1           1
##      hg38_PTMA     hg38_PTN     hg38_RACK1  hg38_RAN
##      4           8           3           1
##      hg38_RBMX     hg38_RPL10   hg38_RPL10A  hg38_RPL11
##      4           3           3           3
##      hg38_RPL12   hg38_RPL13   hg38_RPL13A  hg38_RPL14
##      3           3           3           3
##      hg38_RPL15   hg38_RPL18   hg38_RPL18A  hg38_RPL19
##      3           3           3           3
##      hg38_RPL21   hg38_RPL23   hg38_RPL23A  hg38_RPL24
##      3           3           3           3
##      hg38_RPL26   hg38_RPL27   hg38_RPL27A  hg38_RPL28
##      3           3           3           3
##      hg38_RPL29   hg38_RPL3    hg38_RPL30   hg38_RPL31
##      3           3           3           3
##      hg38_RPL32   hg38_RPL34   hg38_RPL35   hg38_RPL35A
##      3           3           3           3
##      hg38_RPL37   hg38_RPL37A  hg38_RPL38   hg38_RPL39
##      3           3           3           3
##      hg38_RPL4    hg38_RPL5    hg38_RPL6    hg38_RPL7
##      3           3           3           3
##      hg38_RPL7A   hg38_RPL8   hg38_RPL9   hg38_RPLP0

```

```

##      3      3      3      3
## hg38_RPLP1 hg38_RPLP2 hg38_RPS10 hg38_RPS11
##      3      3      3      3
## hg38_RPS12 hg38_RPS13 hg38_RPS14 hg38_RPS15
##      3      3      3      3
## hg38_RPS15A hg38_RPS16 hg38_RPS17 hg38_RPS18
##      3      3      3      3
## hg38_RPS19 hg38_RPS2 hg38_RPS20 hg38_RPS21
##      3      3      3      3
## hg38_RPS23 hg38_RPS24 hg38_RPS25 hg38_RPS27
##      3      3      3      3
## hg38_RPS27A hg38_RPS28 hg38_RPS29 hg38_RPS3
##      3      3      3      3
## hg38_RPS3A hg38_RPS4X hg38_RPS5 hg38_RPS6
##      3      3      3      3
## hg38_RPS7 hg38_RPS8 hg38_RPS9 hg38_RPSA
##      3      3      3      3
## hg38_SERBP1 hg38_SET hg38_SFRP1 hg38_SLC25A3
##      1      1      4      3
## hg38_SOX2 hg38_SRPI4 hg38_STMN1 hg38_SUMO2
##      4      1      1      3
## hg38_TMSB10 hg38_TMSB4X hg38_TPI1 hg38_TPT1
##      1      1      5      3
## hg38_TUBA1A hg38_TUBA1B hg38_TUBB hg38_UBA52
##      1      1      1      3
## hg38_UBB hg38_UQCRH hg38_VIM hg38_YBX1
##      1      3      2      1

```

Ovde se primećuju znatno bolji rezultati. Geni sa prefiksom *hg38_RP* su raspoređeni u jedan klaster, sa oznakom 3, i to bez izuzetka. Takođe, svi geni sa prefiksom *hg38_MT* su raspoređeni u jedan klaster sa oznakom 6. Zaključujemo da kosinusno rastojanje jako dobro opisuje sličnost gena. Opet, uz napomenu da upoređivanje prefiksa gena koristimo kao smernicu ali ne kao i pravilo. Zadovoljnićemo se ovim rezultatom ne garantujući da je najoptimalniji.

3.3 Geni sa manjim stepenom ekspresije

Prethodno klasterovanje je izvršeno na znatno manjem skupu. Preostaje drugi obimniji skup. Ovaj skup sadrži 19566 gena i kao takav zahteva iscrpna izračunavanja i složenije analiziranje rezultata. Kako je eksperimentisanje sa ovim skupom znatno skuplje, metode za klasterovanje kao i mere sličnosti se biraju na osnovu preporuka zasnovanih na naučnim tvrdjenjima i rezultatima. Patrik u svom radu "Kako funkcioniše klasterovanje ekspresija gena" (eng. *How does gene expression clustering work?*) predstavlja popularne i često korišćene metode i daje smernice i dobru praksu za rad sa njima[1]. U nastavku je data lista preporučenih metoda koja će se u većini slučajeva slediti pri klasterovanju ovih podataka.

- Hjerarhijsko klasterovanje
- Algoritam K-sredine

- Samoorganizuće mape

Za računanje sličnosti između gena preporučuje sledeće mere:

- Euklidsko rastojanje
- Pirsonov koeficijent korelacije

Uz ove mere dodatno je uvrštena i mera zasnovana na Spirmanovom koeficijentu korelacije koja se smatra za adekvatnu pri radu sa ekspresijom gena[3]. Takođe, uzimamo i kosinusnu mjeru koja se dobro pokazala na prethodnom skupu podataka.

Sledeći kôd izračunava gore navedene matrice sličnosti.

```
cosine.dist.2 = parDist(data.matrix(NOTOUT.G.HESC.CO), method = "cosine")
euclidean.dist.2 = parDist(data.matrix(NOTOUT.G.HESC.CO), method = "euclidean")
```

```
pearson.dist.2 <- get_dist(NOTOUT.G.HESC.CO, method = "pearson")
```

```
spearman.dist.2 <- get_dist(NOTOUT.G.HESC.CO, method = "spearman")
```

Sada, kada su izračunate matrice sličnosti, ostaju otvorena dva pitanja. Prvo, na koji način odabratи broj klastera i drugo, koji kriterijum koristiti za testiranje kvaliteta samih klastera? Predlaže se senka koeficijent (eng. *Silhouette coefficient*) kao interna mera kvaliteta klastera. Štaviše, data mera se može koristiti kao kriterijum za odabir optimalnog broja klastera. Pоказано је да ова mera jako dobro odražava kvalitet klastera[8]. Prema gore spomenutim preporukama izvršavamo hijerarhijsko klasterovanje za sve četiri mere sličnosti. Za određivanje rastojanja između dva klastera biramo *ward.D* metodu.

```
ward.D.cosine.hc.2 = hclust(cosine.dist.2, method = "ward.D")
ward.D.euclidean.hc.2 = hclust(euclidean.dist.2, method = "ward.D")
ward.D.pearson.hc.2 = hclust(pearson.dist.2, method = "ward.D")
ward.D.spearman.hc.2 = hclust(spearman.dist.2, method = "ward.D")
```

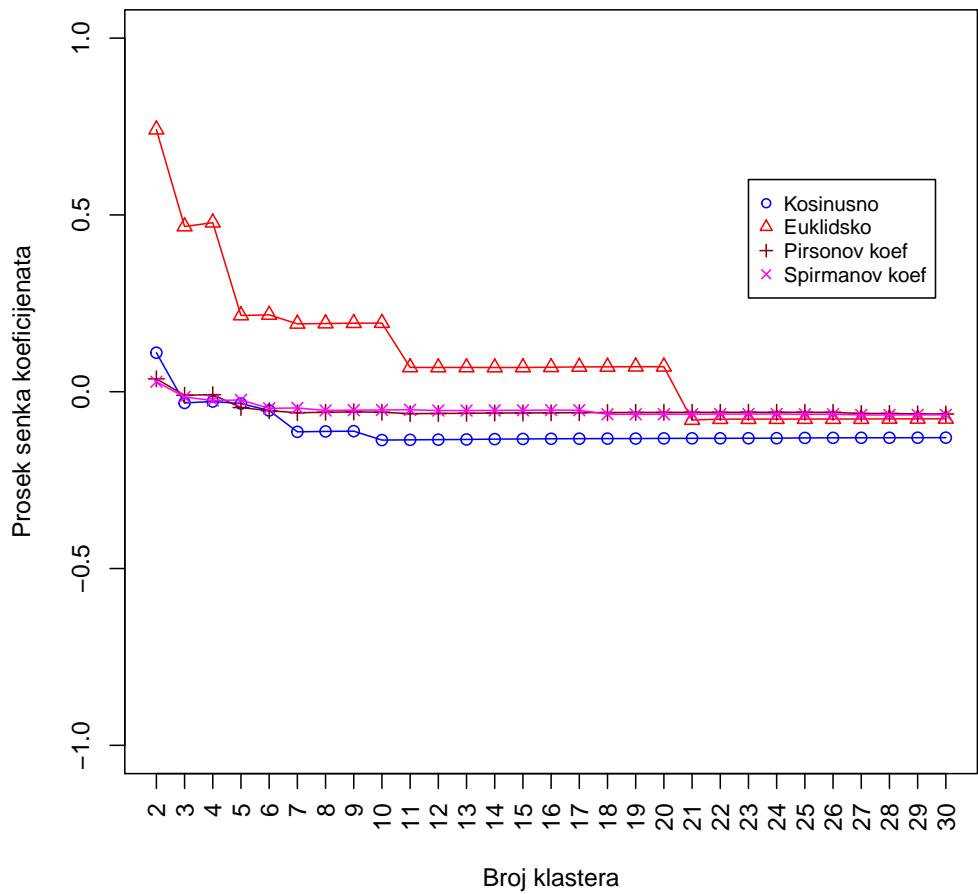
Preostaje da se implementira funkcija koja na osnovu datog broja klastera računa senka koeficijent. Ovaj koeficijent predstavlja mjeru koja opisuje koliko je objekat uskladen sa klasterom kojem pripada u poređenju sa drugim

klasterima. Uzima vrednosti u opsegu $[-1, 1]$ gde visoka vrednost ukazuje na dobru usklađenost sa sopstvenim klasterom i lošu sa ostalim klasterima. Naredna funkcija uzima broj klastera iz intervala $[2, 30]$ i za svaku vrednost računa prosek senka koeficijenata.

```
silh.clusters <- function(hc, dist) {
  x <- c()
  for (i in seq(2, 30)) {
    x[i - 1] <- summary(silhouette(cutree(hc, i), dist))$avg.width
  }
  return(x)
}

cosine.silh <- silh.clusters(ward.D.cosine.hc.2, cosine.dist.2)
euclidean.silh <- silh.clusters(ward.D.euclidean.hc.2, euclidean.dist.2)
pearson.silh <- silh.clusters(ward.D.pearson.hc.2, pearson.dist.2)
spearman.silh <- silh.clusters(ward.D.spearman.hc.2, spearman.dist.2)
```

Dobijene prosečne vrednosti senka koeficijenata iscrtavamo na grafiku pri čemu različite boje linija označavaju različite mere rastojanja koje su korišćene tom prilikom.



Analizom grafika se vidi da kosinusno rastojanje, koje se u prethodnom slučaju pokazalo kao dobro, sada daje najlošije rezultate. Mere zasnovane na korelaciji su blago uspešnije. Euiklidsko rastojanje sve do 20 klastera daje bolji prosečni senka koeficijent, nakon čega se približava Pirsonovom i Spirmanovom koeficijentu. Vršimo još jednu dodatnu proveru. Naime, dešava se da neki algoritmi neravnomerno raspoređuju elemente po klasterima, odnosno može se javiti jedan klaster koji obuhvata veliki broj elemenata i veći broj manjih klastera koji sadrže mali broj elemenata (neki čak i po jedan elemenat). Takvo klasterovanje bi prouzrokovala veliku vrednost senka koeficijenta i dalo lažnu sliku.

```



```

Vidimo da su elementi po klasterima prihvatljivo raspoređeni. Ukoliko bi umesto $ward.D$ metode koristili metodu zasnovanu na centroidu videli bi gore spomenuti efekat. Dobili bismo jedan klaster koji sadrži skoro sve gene i njih 29 koji sadrže po jedan ili dva gena dok bi prosečni senka koeficijent u tom slučaju bio blizak jedinici. Kako to ovde nije slučaj, biramo euklidsko rastojanje i rezultat zapisujemo u datoteci *klasterovanje_gena.txt*.

```
write.hclust(ward.D.euclidean.hc.2, file = "klasterovanje_gena.txt", prefix= "", k = 30)
```

4 Klasterovanje ćelija

U prethodnom poglavlju su geni posmatrani kao objekti. Analiza i klasterovanje je vršeno na osnovu vrednosti po ćelijama. U ovom poglavlju stvari posmatramo iz drugog ugla. Naime, kolone, tj. će se posmatrati kao objekte dok gene kao atributi. U skladu sa tim vršimo pretprocesiranje, kako bi format podataka doveli u standardni oblik. Najpre se podaci iz date dve datoteke objedinjuju. Kako je broj gena isti, podatke spajamo na osnovu njihovih imena. Potom se transponuje matrica tako da redovi predstavljaju ćelije a kolone gene. Najzad, uklanjaju se svi nula redovi, tj. one ćelije koje ne sadže gensku ekspresiju. Opisani proces je implementiran sledećim kodom:

```
C.HESC.CO = as.data.frame(t(as.matrix(merge(HESC.CO.1, HESC.CO.2,
  by = 0, all = TRUE))))
names(C.HESC.CO) <- as.matrix(C.HESC.CO[1, ])
C.HESC.CO <- C.HESC.CO[-1, ]
C.HESC.CO[] <- lapply(C.HESC.CO, function(x) type.convert(as.character(x)))
C.HESC.CO = C.HESC.CO[apply(C.HESC.CO, 1, function(row) any(row !=
  0)), ]
C.HESC.CO[1:10, 1:4]

##      hg38_A1BG hg38_A1BG-AS1 hg38_A1CF hg38_A2M
## X1.x          0          0          0          0
## X2.x          0          0          0          0
## X3.x          0          0          0          0
## X4.x          0          1          0          0
## X5.x          0          0          0          0
```

```

## X6.x      0      0      0      0
## X7.x      0      0      0      0
## X8.x      0      0      0      0
## X9.x      0      0      0      0
## X10.x     0      1      0      0

```

Koristimo istu metodologiju kao i kod klasterovanja gena. Biramo mere sličnosti i algoritme klasterovanja za koje računamo prosek senka koeficijenata. Na osnovu toga biramo broj klastera.

Zadržavamo hijerarhijsko klasterovanje sa euklidskom merom sličnosti i dodajemo dve nove metode. U pitanju su *PAM* i *fuzzy k-means* koji su preporučljivi za klasterovanje ovakvih podataka. U nastavku je dat i dijagram 15 sa metodama koje neće biti korišćene u ovom radu ali koji imaju primenu u praksi[4].

Analogno, kao i u prošlom poglavljju, računamo matricu sličnosti, izvršavamo spomenute algoritme i analiziramo grafik.

```

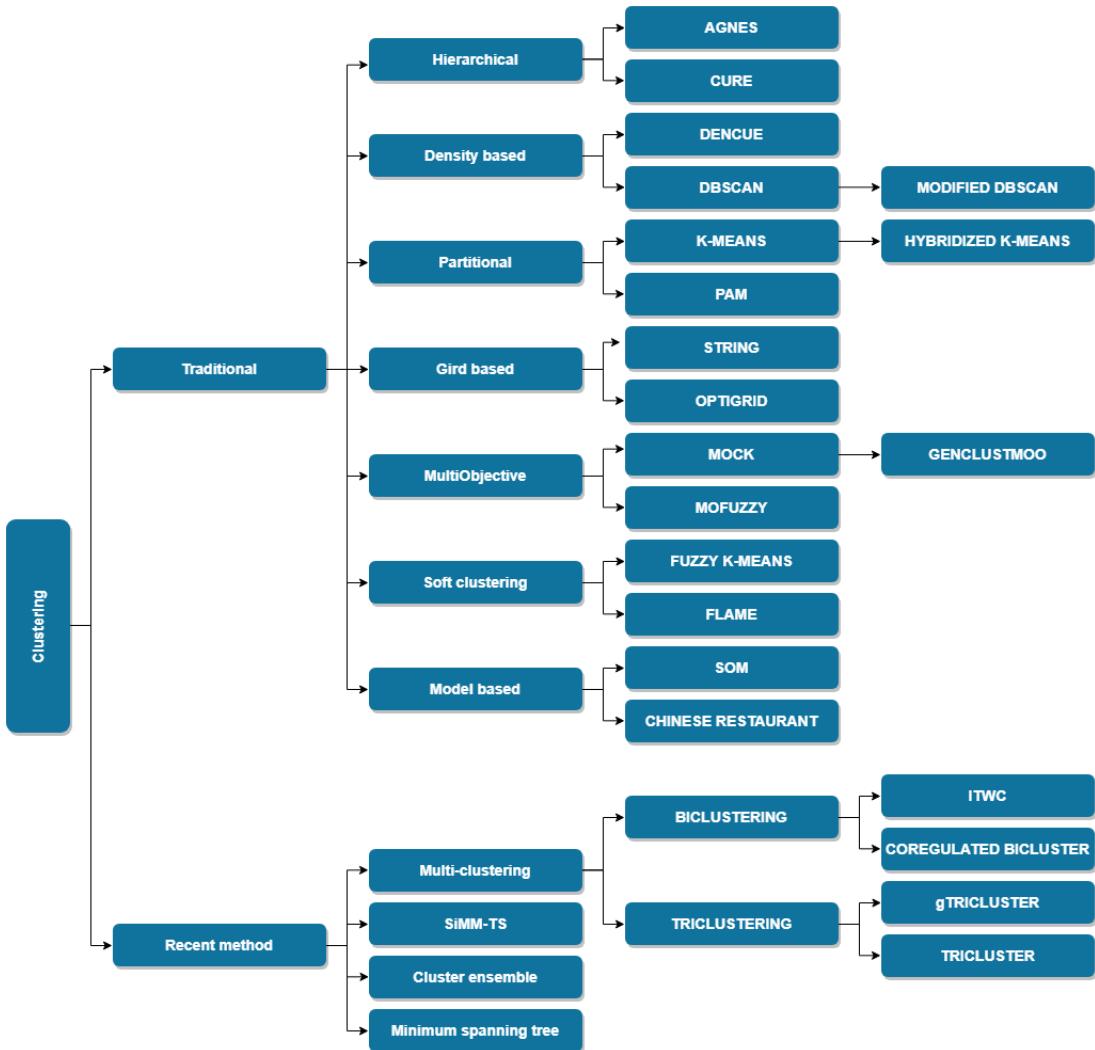
euclidean.dist.c = parDist(data.matrix(C.HESC.CO), method = "euclidean")

ward.D.euclidean.hc.c = hclust(euclidean.dist.c, method = "ward.D")
euclidean.silh.c <- c()
for (i in seq(2, 30)) {
    euclidean.silh.c[i - 1] <- summary(silhouette(cutree(ward.D.euclidean.hc.c,
        i), euclidean.dist.c))$avg.width
}

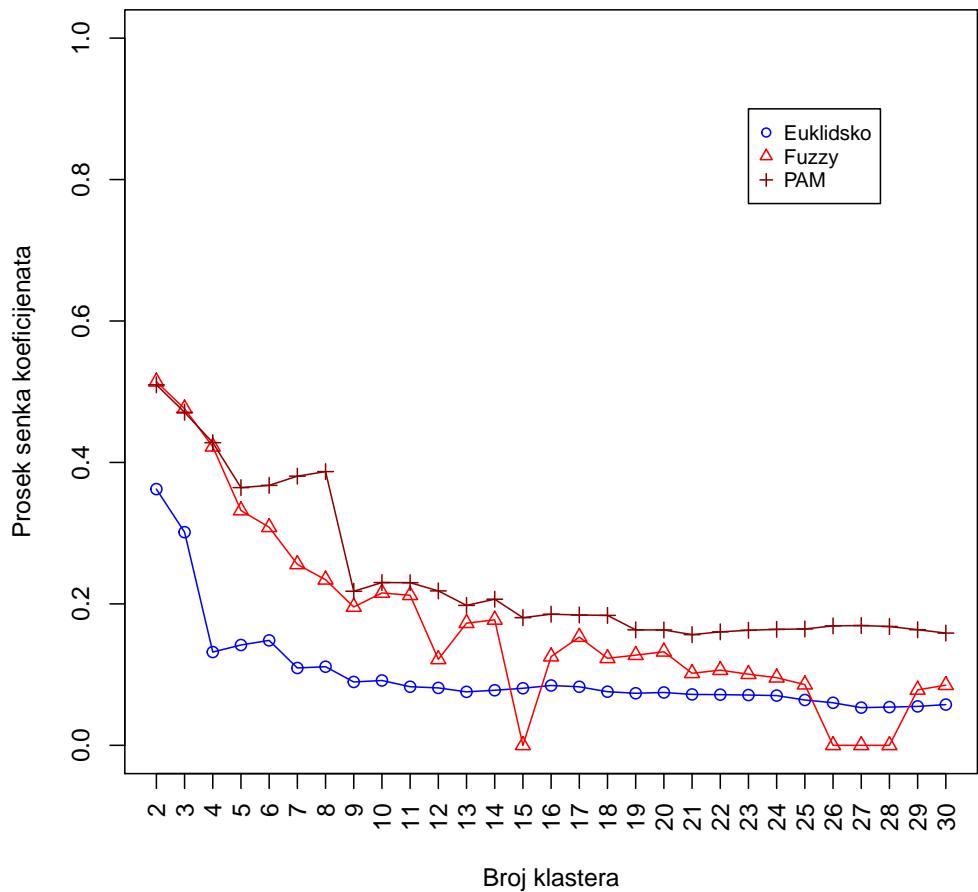
fuzzy.silh.c <- c()
for (i in seq(2, 30)) {
    fanny.c <- fanny(euclidean.dist, k = i, memb.exp = 1.1)
    fuzzy.silh.c[i - 1] <- summary(silhouette(fanny.c, euclidean.dist.c))$avg.width
}

pam.silh.c <- c()
for (i in seq(2, 30)) {
    pam.c <- pam(euclidean.dist, k = i)
    pam.silh.c[i - 1] <- summary(silhouette(pam.c, euclidean.dist.c))$avg.width
}

```



Slika 15: Dijagram metoda klasterovanja



Pregledom grafika se vidi da *PAM* metoda daje najbolje prosečne senka koeficijente. Zanimljiva pojava je variranje vrednosti kod *fuzzy k-means* metode. Ova metoda zahteva *membership exponent* parametar koji je ovde izabran eksperimentalnim putem. Postoje istraživanja koja pronalaze optimalnu vrednost ovog parametra[6]. Odlučujemo se za 18 klastera i rezultate dobijene *PAM* metodom zapisujemo u datoteku *klasterovanje_celija.txt*.

```
pam.c <- pam(euclidean.dist.c, 18)
write.table(pam.c$clustering, file = "klasterovanje_celija.txt", sep=",")
```

5 Zaključak

Ovim radom smo predstavili neke osnovne metode i koncepte koji se koriste za analizu ekspresija gena. Kombinovanjem eksperimentalnog pristupa i prakse zasnovane na naučnoj osnovi, identifikovane su tehnike klasterovanja sa ciljem da se njihovom primenom dode do smislenih grupa gena ili ćelija. Takođe, razmatrane su različite mere sličnosti i njihov uticaj na krajni rezultat.

Analiza ekspresija gena pronalazi svoju konkretnu primenu u raspoznavanju i proučavanju raznih bolesti, kao što su tumor, malarija, astma, tuberkuloza i kao takva predstavlja važnu disciplinu čija dostignuća mogu biti praktično i brzo primenjena.

Literatura

- [1] Patrik D'haeseleer. How does gene expression clustering work? *Nature Biotechnology*, 2005.
- [2] Pablo A Jaskowiak, Ricardo JGB Campello, and Ivan G Costa. On the selection of appropriate distances for gene expression data clustering. *BMC Bioinformatics*, 2014.
- [3] Gang-Guo Li and Zheng-Zhi Wang. Evaluation of similarity measures for gene expression data and their correspondent combined measures. *Interdisciplinary Sciences: Computational Life Sciences*, 2009.
- [4] Jelili Oyelade, Itunuoluwa Isewon, Funke Oladipupo, Olufemi Aromolaran, Efosa Uwoghiren, Faridah Ameh, Moses Achas, and Ezekiel Adebiyi. Clustering algorithms: Their application to gene expression data. *Bioinformatics and Biology Insights*, 2016.
- [5] Tina Hesman Saey. A recount of human genes ups the number to at least 46,831.
- [6] Veit Schwämmle and Ole Nørregaard Jensen. A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. *Bioinformatics*, 2010.
- [7] David Tritchler, Elena Parkhomenko, and Joseph Beyene. Filtering genes for cluster and network analysis. *BMC Bioinformatics*, 2009.

- [8] Chunmei Yang, Baikun Wan, and Xiaofeng Gao. Effectivity of internal validation techniques for gene clustering. *Biological and Medical Data Analysis*, 2006.