

LAPORAN UJIAN AKHIR SEMESTER SAINS DATA GENOM

*Cluster Analysis And Classification Analysis
for GDS3514 Dataset*



DISUSUN OLEH:
Wahyu Dimasdi Putra (2106704736)

Sains Data Genom B

**PROGRAM STUDI STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
2023**

I. Pendahuluan

Liposarkoma merupakan jenis kanker yang berasal dari jaringan lemak. Penyakit ini dapat terjadi di bagian tubuh mana saja. Namun, area yang paling sering terkena adalah area tubuh yang mengandung banyak lemak seperti di perut, lengan, dan tungkai. Penyakit ini juga sering kali sulit untuk diobati. Salah satu pendekatan yang umum digunakan dalam pengobatan liposarkoma adalah penggunaan doxorubicin, yang telah terbukti sebagai agen kemoterapi efektif dalam beberapa kasus. Namun, kendati keberhasilannya dalam beberapa situasi, respons terhadap doxorubicin seringkali menunjukkan tingkat yang rendah.

Pentingnya memahami alasan di balik resistensi liposarkoma terhadap doxorubicin mendorong penelitian yang lebih dalam dalam bidang ini. Melalui analisis berbagai jenis liposarkoma yang diobati dengan doxorubicin secara *in vitro*, penelitian ini bertujuan untuk menggali wawasan yang lebih dalam mengenai dasar molekuler dari resistensi ini.

Penelitian ini menggunakan Dataset GDS3514 sebagai landasan dalam melakukan analisis klaster dan analisis klasifikasi. Dataset ini menyediakan informasi yang berharga mengenai respons liposarkoma terhadap doxorubicin serta karakteristik molekuler yang mungkin mempengaruhi resistensi terhadap agen kemoterapi ini.

Dalam makalah ini, akan dilakukan analisis klaster dan klasifikasi pada Dataset GDS3514 untuk mengidentifikasi pola-pola dan hubungan antara berbagai jenis liposarkoma yang diobati dengan doxorubicin. Hasil dari analisis ini diharapkan dapat memberikan pemahaman yang lebih mendalam mengenai pola respons dan karakteristik molekuler yang mempengaruhi resistensi liposarkoma terhadap doxorubicin. Dengan demikian, penelitian ini diharapkan dapat memberikan sumbangan penting dalam upaya pengembangan strategi pengobatan yang lebih efektif untuk mengatasi resistensi terhadap doxorubicin dalam pengobatan liposarkoma.

II. Metode Analisis

1. Filtering Gene (Penyaringan Gen):

Filtering gene adalah proses seleksi atau penyaringan gen-gen tertentu dari dataset genetik berdasarkan kriteria-kriteria tertentu untuk mengecilkan jumlah gen yang akan dianalisis. Hal ini dilakukan untuk fokus pada gen-gen yang dianggap penting atau relevan dalam konteks penelitian. Tujuan utama dari filtering gene adalah untuk mengurangi kompleksitas data genetik dan memfokuskan analisis pada gen-gen yang dianggap memiliki peran penting atau signifikan dalam fenomena yang sedang dipelajari, seperti respons terhadap pengobatan atau kondisi kesehatan tertentu. Gen-gen ini akan difilter berdasarkan duplikat, variansi yang rendah, ID ENTREZ yang tidak valid, dan lainnya. Nantinya hasil filtering ini akan dianalisis lebih lanjut untuk dicari top genesnya.

2. LIMMA Analysis (untuk 50 Gen Teratas):

LIMMA (Linear Models for Microarray Analysis) adalah metode analisis statistik yang digunakan untuk menganalisis data ekspresi gen dalam studi mikroarray. Teknik ini sering digunakan untuk menemukan perbedaan signifikan dalam ekspresi gen antara dua atau lebih kondisi perlakuan, seperti kelompok pasien yang sehat dan pasien yang sakit, atau kelompok yang menerima perlakuan tertentu dengan kelompok kontrol. Hasil dari

LIMMA akan fokus pada 50 gen teratas yang memiliki perbedaan ekspresi yang paling signifikan antara kedua kelompok tersebut.

3. Analisis Klaster (Clustering Analysis):

a. K-Means Clustering

K-Means adalah salah satu algoritma yang paling umum digunakan dalam analisis klaster (clustering analysis) untuk mengelompokkan data menjadi kelompok-kelompok (klaster) berdasarkan kemiripan atau kesamaan di antara observasi-observasi dalam data. Tujuan utama dari K-Means adalah untuk meminimalkan varians di dalam klaster (varians intra-klaster) dan memaksimalkan varians antara klaster (varians antar-klaster). Hasil dari algoritma K-Means adalah kelompok-kelompok data yang memiliki observasi yang serupa di dalamnya, sedangkan observasi di kelompok yang berbeda memiliki karakteristik yang lebih berbeda satu sama lain.

b. Hierarchical Clustering

Hierarchical Clustering adalah teknik analisis klaster yang mengelompokkan data ke dalam hierarki atau struktur berjenjang. Tujuannya adalah untuk mengorganisir data menjadi serangkaian kelompok yang membentuk struktur pohon atau dendrogram, yang menunjukkan tingkat kemiripan antar observasi dalam data.

c. Principal Component Analysis (PCA) & Biplot

PCA digunakan untuk mereduksi dimensi data dan memvisualisasikan pola-pola utama yang ada dalam data. Biplot adalah representasi grafis dari PCA yang menunjukkan hubungan antara sampel dan variabel (gen) dalam satu gambar.

4. Analisis Klasifikasi (Lasso Classification):

LASSO (Least Absolute Shrinkage and Selection Operator) adalah teknik regresi yang digunakan dalam analisis statistik dan pembelajaran mesin untuk regularisasi dan seleksi fitur. Namun, "lasso classification" mengacu pada penggunaan metode LASSO dalam konteks klasifikasi atau pengelompokan. LASSO classification digunakan untuk membangun model klasifikasi yang memiliki kemampuan untuk melakukan prediksi dan pengelompokan pada data dengan fitur-fitur yang relevan atau penting. Teknik ini terutama bermanfaat ketika dataset memiliki banyak fitur dan ada kebutuhan untuk memilih fitur-fitur yang paling informatif dalam membedakan antara kelas atau kelompok yang berbeda.

III. Hasil dan Pembahasan

1. Dataset

Data yang digunakan adalah ‘GDS3514’ yang berupa data Analisis berbagai jenis liposarkoma yang diobati dengan doxorubicin secara in vitro. Meskipun doxorubicin merupakan kemoterapi yang terbukti untuk mengobati liposarkoma, respons terhadap obat ini rendah. Hasil penelitian memberikan wawasan tentang dasar molekuler dari resistensi liposarkoma terhadap doxorubicin. Berikut beberapa informasi tentang dataset yang digunakan :

- Organisme sampel dalam dataset adalah Homo sapiens / manusia.

- Jumlah gen yang terlampir pada data sebanyak 22283 dan berasal dari 38 sampel.
- Platform eksperimen yang digunakan, yaitu "GPL96."
- Jenis sampel yang digunakan pada data ini adalah RNA.
- Tanggal terakhir pembaruan atau update pada data atau sumber informasi adalah 15 September 2009.
- Sumber data yang digunakan berasal dari web <http://www.ncbi.nlm.nih.gov/geo>.
- Terdapat beberapa atribut yang terdapat pada data, seperti sampel, status penyakit, dan lainnya.

2. Phenotype Data

```
> head(phdtgeo)
#> #> sample specimen agent development.stage disease.state
#> GSM325240 GSM325240 tumor 314 untreated grade 2 atypical, dedifferentiated
#> GSM325241 GSM325241 tumor 314 doxorubicin grade 2 atypical, dedifferentiated
#> GSM325242 GSM325242 tumor 387 untreated grade 2 atypical
#> GSM325243 GSM325243 tumor 387 doxorubicin grade 2 atypical
#> GSM325244 GSM325244 tumor 400 untreated grade 1 atypical
#> GSM325245 GSM325245 tumor 400 doxorubicin grade 1 atypical
#>
#> description
#> GSM325240      Value for GSM325240: Patient no. 1: control; src: Liposarcoma culture from tumor 314, untreated
#> GSM325241      Value for GSM325241: Patient no. 1: treated; src: Liposarcoma culture from tumor 314, doxorubicin treated
#> GSM325242      Value for GSM325242: Patient no. 9: control; src: Liposarcoma culture from tumor 387, untreated
#> GSM325243      Value for GSM325243: Patient no. 9: treated; src: Liposarcoma culture from tumor 387, doxorubicin treated
#> GSM325244      Value for GSM325244: Patient no. 18: control; src: Liposarcoma culture from tumor 400, untreated
```

Didapat beberapa informasi sebagai berikut :

- Terdapat 38 sample yang digunakan yaitu GSM325240, GSM325241, dan lainnya.
- Spesimen tumor yang mungkin berasal dari pasien atau lokasi yang berbeda dalam tubuh seperti tumor 314, tumor 387, tumor 400, dan lainnya.
- Perlakuan atau zat yang diberikan pada sampel ("doxorubicin" menandakan pengobatan dengan obat doxorubicin).
- Tahap perkembangan tumor ada "grade 1", "grade 2", dan kemungkinan lainnya.
- disease.state juga memberi informasi tentang keadaan penyakit yang terkait dengan sampel.

3. Filtering Gene

Selanjutnya dilakukan filtering data dan didapatkan hasil *filtering* sebagai berikut.

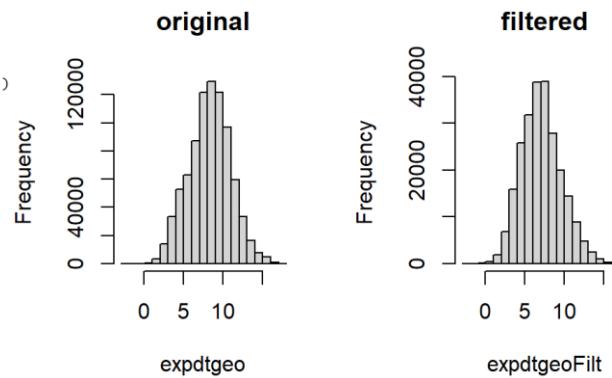
```
> esetFilter
$eset
ExpressionSet (storageMode: lockedEnvironment)
assayData: 6322 features, 38 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: GSM325240 GSM325241 ... GSM325277 (38 total)
  varLabels: sample.specimen ... description (6 total)
  varMetadata: labelDescription
featureData
  featureNames: 203440_at 207078_at ... 205241_at (6322 total)
  fvarLabels: ID Gene title ... GO:Component ID (21 total)
  fvarMetadata: column labelDescription
experimentData: use 'experimentData(object)'
pubMedIds: 18959781
Annotation: hg133a

$filter.log
$filter.log$numDupsRemoved
[1] 7551

$filter.log$numLowVar
[1] 6323

$filter.log$numRemoved.ENTREZID
[1] 2077

$filter.log$feature.exclude
[1] 10
```



Berikut beberapa informasi yang didapat dari gene filtering.

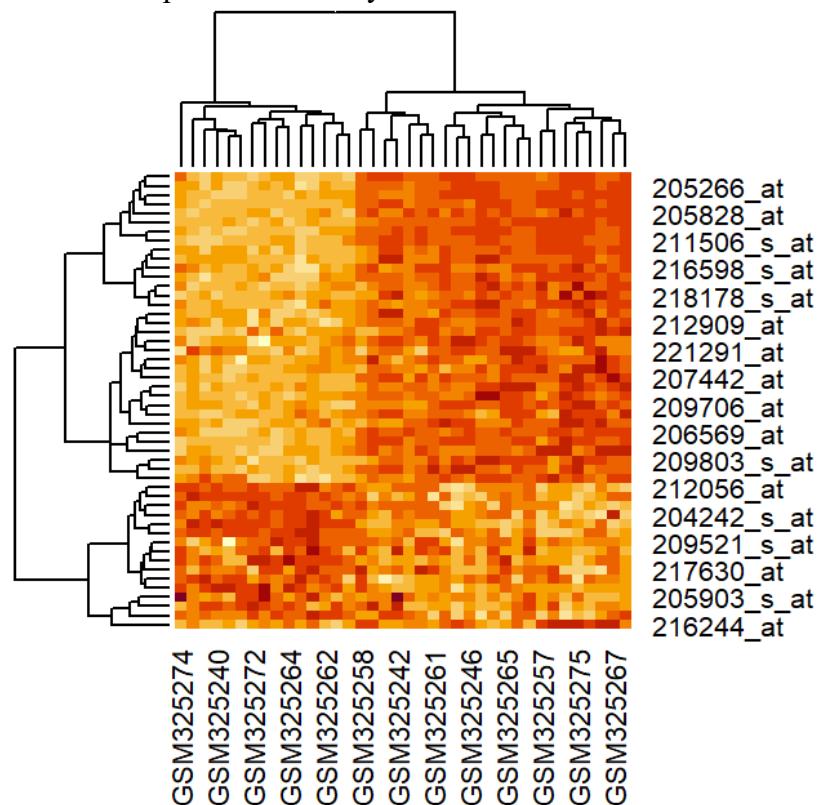
- 7551 fitur dihapus karena merupakan duplikat.
- 6323 fitur yang dihapus karena variasi rendah.
- 2071 fitur yang dihapus karena tidak memiliki ID ENTREZ yang sesuai(tidak valid).
- 10 fitur yang telah dihapus.
- Hasil filtering secara keseluruhan telah mengurangi jumlah fitur dari 22283 menjadi 6322. Gen yang disaring merupakan gen yang memiliki ekspresi rendah seperti yang terlihat pada histogram di atas.

4. LIMMA Analysis

Dari proses filtering gene selanjutnya dipilih top 50 gen berdasarkan variansi yang tinggi untuk dilakukan proses clustering. Top 50 gen tersebut adalah sebagai berikut :

```
> topResult <- topTable(fit, coef=2, number=50)
> rownames(topResult)
[1] "222265_at"    "207442_at"    "209521_s_at" "203691_at"
[5] "207850_at"    "205476_at"    "208075_s_at" "207201_s_at"
[9] "205680_at"    "204242_s_at"  "215101_s_at" "211506_s_at"
[13] "201631_s_at"  "212909_at"    "208343_s_at" "210260_s_at"
[17] "209921_at"    "216244_at"    "216598_s_at" "212056_at"
[21] "205266_at"    "218178_s_at"  "221908_at"   "205289_at"
[25] "206569_at"    "220012_at"    "204470_at"   "36711_at"
[29] "209803_s_at"  "205207_at"    "204621_s_at" "201905_s_at"
[33] "209706_at"    "221291_at"    "217630_at"   "218844_at"
[37] "209294_x_at"  "204827_s_at"  "213468_at"   "206025_s_at"
[41] "220744_s_at"  "204420_at"    "204475_at"   "204224_s_at"
[45] "220054_at"    "201467_s_at"  "205828_at"   "205903_s_at"
[49] "209431_s_at"  "217953_at"
```

Dan berikut adalah heatmap visualizationnya



Dari LIMMA didapat 10 overexpressed gene dengan variansi tertinggi sebagai berikut :

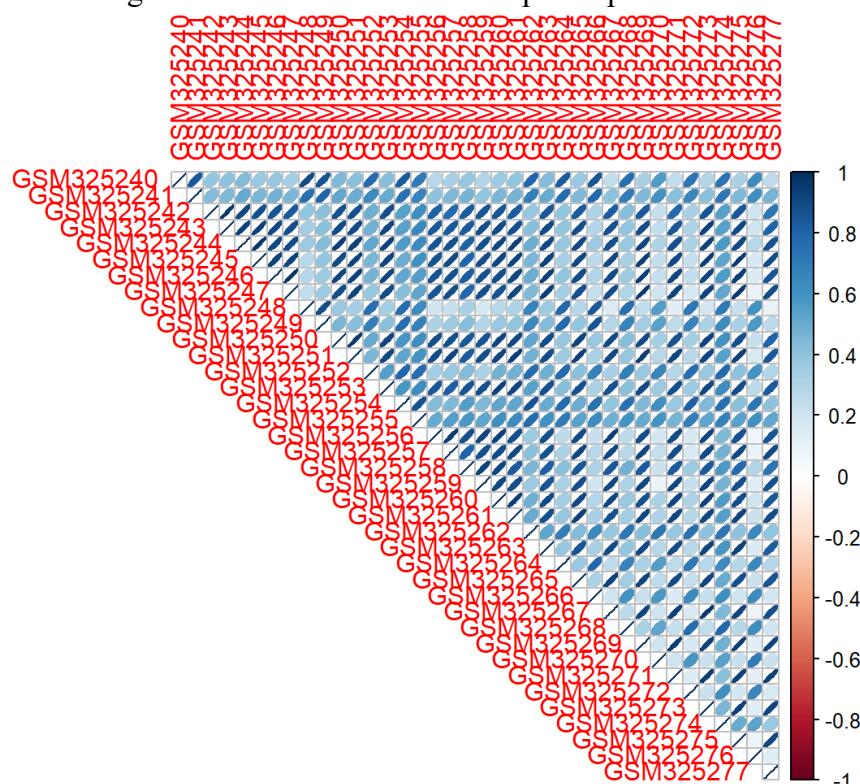
```
> print(top_variance_genes_info)
    logFC    AveExpr      t     P.Value adj.P.Val      B
222265_at    1.111107  6.011796  4.013083 0.0002225606 0.3277337  0.24668130
207442_at   -2.440665  7.032916 -3.891930 0.0003240813 0.3277337 -0.04273088
209521_s_at   1.304958  6.280409  3.843847 0.0003757296 0.3277337 -0.15662656
203691_at   -3.118928  8.937573 -3.818718 0.0004057970 0.3277337 -0.21591947
207850_at   -3.561718 10.685588 -3.812339 0.0004137917 0.3277337 -0.23094580
205476_at   -4.545127 10.403037 -3.797593 0.0004328585 0.3277337 -0.26564133
208075_s_at  -1.817285  7.410949 -3.764525 0.0004787441 0.3277337 -0.34323830
207201_s_at  -1.979558  6.879791 -3.727221 0.0005361346 0.3277337 -0.43042705
205680_at   -3.014890  8.847318 -3.717898 0.0005514815 0.3277337 -0.45215970
204242_s_at   1.428987  6.893529  3.699176 0.0005835848 0.3277337 -0.49572710
```

Didapat juga 10 gene yang paling underexpressed sebagai berikut :

```
> print(topResult_underexpressed)
    logFC    AveExpr      t     P.Value adj.P.Val      B
205476_at   -4.545127 10.403037 -3.797593 0.0004328585 0.3277337 -0.2656413
211506_s_at  -4.045968 12.405636 -3.642189 0.0006927402 0.3277337 -0.6277323
215101_s_at  -3.801075  8.442563 -3.662708 0.0006513515 0.3277337 -0.5803085
207850_at   -3.561718 10.685588 -3.812339 0.0004137917 0.3277337 -0.2309458
206569_at   -3.452725  8.924449 -3.415677 0.0013533595 0.3277337 -1.1427786
203691_at   -3.118928  8.937573 -3.818718 0.0004057970 0.3277337 -0.2159195
209774_x_at  -3.048206 11.741796 -3.063602 0.0036768546 0.3521133 -1.9083733
205207_at   -3.031644 12.878222 -3.367006 0.0015587485 0.3284803 -1.2512868
205680_at   -3.014890  8.847318 -3.717898 0.0005514815 0.3277337 -0.4521597
204475_at   -2.962541 11.537691 -3.202327 0.0024959208 0.3521133 -1.6122353
```

5. Data Visualization

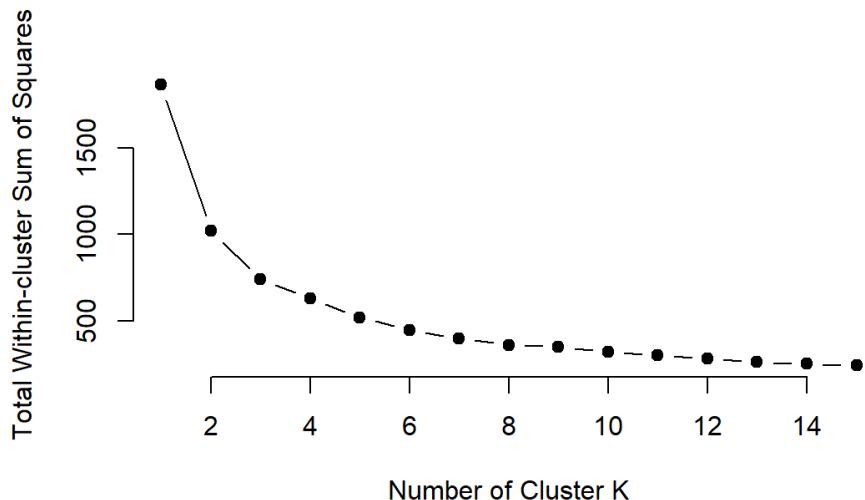
Sebelum melakukan analisis klaster, terlebih dahulu dilakukan visualisasi data. Data set yang digunakan adalah expression data dari 50 gen teratas yang telah diseleksi sebelumnya. Paling sederhana dengan menambahkan correlation plot seperti dibawah ini.



6. Penentuan Cluster yang Optimal

Sebelum melakukan clustering dapat ditentukan jumlah cluster yang diinginkan sesuai dengan kondisi data dengan melihat jumlah cluster yang optimal. Hal ini dapat dilakukan dengan menghitung variasi dari berbagai cluster dan mencari cluster yang menghasilkan jumlah variasi minimum. Berikut ini adalah penerapan within cluster variation pada setiap jumlah klaster yang diinginkan (k). Jumlah cluster terbaik adalah dengan nilai ss within terkecil.

```
> expdtgeoselScale <- scale(expdtgeosel, scale = T)
> set.seed(123)
> k.max <- 15
> wss <- sapply(1:k.max, function(k){kmeans(expdtgeoselScale, k, nstart=10)$tot.withinss})
> plot(1:k.max, wss, type ='b', pch = 19, frame = FALSE, xlab = "Number of Cluster K",
+       ylab = "Total Within-cluster Sum of Squares") #Number Of Cluster = 3
```



Plot di atas merupakan Elbow Plot. yang digunakan untuk mencari nilai K terbaik, yaitu jumlah cluster untuk K-Means Clustering. Nilai K yang terbaik dipilih dengan cara melihat penurunan nilai (Total) Within-Cluster Sum of Squares (WCSS) seiring dengan bertambahnya nilai K . Perhatikan titik elbow di plot, yaitu saat $K = 3$. Elbow pada plot merupakan titik di mana laju penurunan berubah dengan cepat, menentukan jumlah cluster yang optimal, dimana penambahan jumlah klaster lebih lanjut tidak signifikan mengurangi nilai WCSS. Oleh sebab itu, jumlah cluster yang optimal adalah 3.

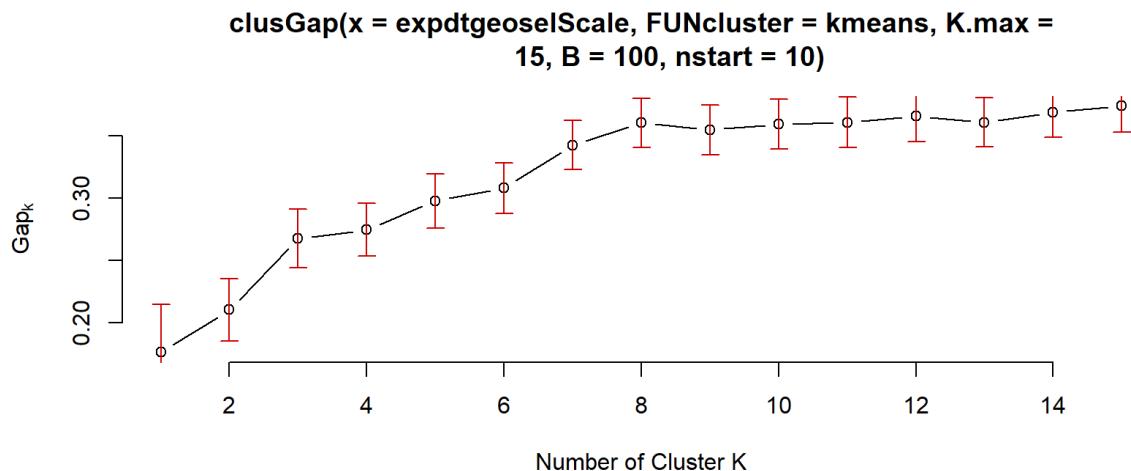
Selain dengan jumlah pendekatan variasi di dalam klaster, dapat juga digunakan metode Gap statistic, yang intinya membandingkan nilai within cluster dari dengan data yang ada dari reference null distribution yang didapat dengan Monte Carlo Sampling. Berikut ini adalah penerapan metode Gap Statistic.

```

> gap_stat <- clusGap(expdtgeoselScale, FUN = kmeans, nstart = 10,
+ K.max = 15, B = 100)
Clustering k = 1,2,..., K.max (= 15): ... done
Bootstrapping, b = 1,2,..., B (= 100) [one "." per sample]:
..... 50
..... 100
> print(gap_stat, method = "firstmax")
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = expdtgeoselScale, FUNcluster = kmeans, K.max = 15, B = 100, nstart = 10)
B=100 simulated reference sets, k = 1..15; spaceH0="scaledPCA"
--> Number of clusters (method 'firstmax'): 8
   logw    E.logw      gap     SE.sim
[1,] 4.595230 4.771932 0.1767016 0.03847725
[2,] 4.304735 4.515322 0.2105873 0.02517434
[3,] 4.139975 4.407746 0.2677713 0.02362145
[4,] 4.046786 4.321409 0.2746234 0.02110255
[5,] 3.953834 4.251733 0.2978992 0.02166970
[6,] 3.888808 4.196788 0.3079806 0.02007255
[7,] 3.805967 4.148449 0.3424811 0.01971589
[8,] 3.743963 4.104069 0.3601060 0.01969489
[9,] 3.706371 4.061003 0.3546320 0.02000852
[10,] 3.660471 4.019619 0.3591485 0.01986284
[11,] 3.619088 3.979693 0.3606050 0.02029492
[12,] 3.574319 3.939734 0.3654143 0.02021701
[13,] 3.539029 3.899674 0.3606447 0.01985848
[14,] 3.491872 3.860513 0.3686407 0.02020498
[15,] 3.446695 3.820355 0.3736595 0.02079137

```

Hasil code menunjukkan bahwa k-means digunakan sebagai algoritma clustering. nstart = 10 digunakan sebagai jumlah percobaan awal dalam k-means untuk setiap jumlah cluster dengan K.max = 15 yaitu jumlah klaster maksimal yang dipertimbangkan (dari 1 hingga 15). Dengan menggunakan Metode firstmax untuk memilih hasil Gap statistic, hasilnya menunjukkan bahwa jumlah cluster yang dipilih adalah 8. Berikut adalah visualisasi dari clusGap-nya.



Grafik tersebut membantu memvisualisasikan tren Gap statistic dan dapat membantu dalam menentukan jumlah cluster yang optimal berdasarkan penurunan dalam nilai Gap statistic. Titik dimana peningkatan Gap statistic melambat atau mendatar dapat dianggap sebagai jumlah cluster optimal. Dalam konteks ini, mungkin perlu mempertimbangkan titik dimana kurva menunjukkan puncak atau di mana nilai Gap statistic mencapai maksimum lokal. Tetapi jika melihat sekilas dari gambar jumlah cluster sebanyak 8 cocok untuk diinterpretasikan jika menggunakan metode ini.

7. K-means Clustering

K-means clustering merupakan metode untuk melakukan pengelompokan objek ke sejumlah K klaster. Nantinya, setiap data akan dikelompokkan pada klaster dengan titik pusat yang terdekat dari data tersebut. Dalam kasus ini, pengelompokan dilakukan menggunakan metode k-means dengan jumlah kelompok yang ditentukan sebanyak tiga. Terlihat untuk masing masing top 50 gen masuk ke kelompok 1, 2, atau 3. Didapat kelompok 1 berisi 16 features, kelompok 2 berisi 21 features, dan kelompok 3 berisi 13 features.

```
> set.seed(123)
> kMres <- kmeans(expdtgeose1$scale ,centers=3)
> kMres$cluster
 201905_s_at 206569_at 212909_at 207442_at 209521_s_at
    1          2          2          2          1
 201467_s_at 213468_at 212056_at 217953_at 209431_s_at
    2          1          1          1          1
 209921_at   36711_at  208343_s_at 210260_s_at 204224_s_at
    2          3          1          3          3
 204470_at   207850_at 216244_at 205207_at 211506_s_at
    3          2          1          3          3
 205903_s_at 205266_at 204475_at 205828_at 205680_at
    1          3          3          2          2
 209706_at   204621_s_at 220054_at 203691_at 220744_s_at
    2          2          2          2          1
 220012_at   218178_s_at 216598_s_at 208075_s_at 205476_at
    2          3          3          2          2
 215101_s_at 205289_at 207201_s_at 206025_s_at 209803_s_at
    2          2          2          3          3
 218844_at   221291_at  204420_at  204242_s_at  221908_at
    1          2          2          1          1
 222265_at   209294_x_at 201631_s_at 204827_s_at 217630_at
    1          2          3          1          1
> kMres$size
[1] 16 21 13
```

Perlu diperhatikan bahwa proses randomisasi pada algoritma ini akan menyebabkan hasil output yang berbeda. Berikut adalah simulasinya hasil dari dua pemanggilan fungsi k-means yang berbeda (kMres1 dan kMres2).

```
> kMres1 <- kmeans(expdtgeose1, centers=3)
> head(kMres1$cluster)
 201905_s_at 206569_at 212909_at 207442_at 209521_s_at 201467_s_at
    3          2          2          2          3          2
> kMres1$size
[1] 15 19 16
> kMres2 <- kmeans(expdtgeose1, centers=3)
> head(kMres2$cluster)
 201905_s_at 206569_at 212909_at 207442_at 209521_s_at 201467_s_at
    1          2          2          2          1          2
> kMres2$size
[1] 16 19 15
```

Dari pengelompokan tersebut didapat informasi pengelompokan gen sebagai berikut.

- Pemanggilan pertama (kMres1): Gen-gene seperti 201905_s_at, 206569_at, dan 212909_at berturut turut termasuk ke dalam kelompok 3, 2, 3, dan seterusnya untuk gen yg lain.
- Pemanggilan kedua (kMres2): Gen-gene seperti 201905_s_at, 206569_at, dan 212909_at berturut turut termasuk ke dalam kelompok 1, 2, 2, dan seterusnya untuk gen yg lain.

Dari pengelompokan tersebut juga didapat informasi ukuran setiap kelompok sebagai berikut.

- Pemanggilan pertama (kMres1): Kelompok 1 memiliki 15 gen, kelompok 2 memiliki 19 gen, dan kelompok 3 memiliki 16 gen.
- Pemanggilan kedua (kMres2): Kelompok 1 memiliki 16 gen, kelompok 2 memiliki 19 gen, dan kelompok 3 memiliki 15 gen.

Selanjutnya akan dibandingkan kesesuaian antara cluster membership dari dua kali run menggunakan matriks kontingensi sebagai berikut.

```
> table(kMres1 $cluster, kMres2 $cluster)
```

	1	2	3
1	0	0	15
2	0	19	0
3	16	0	0

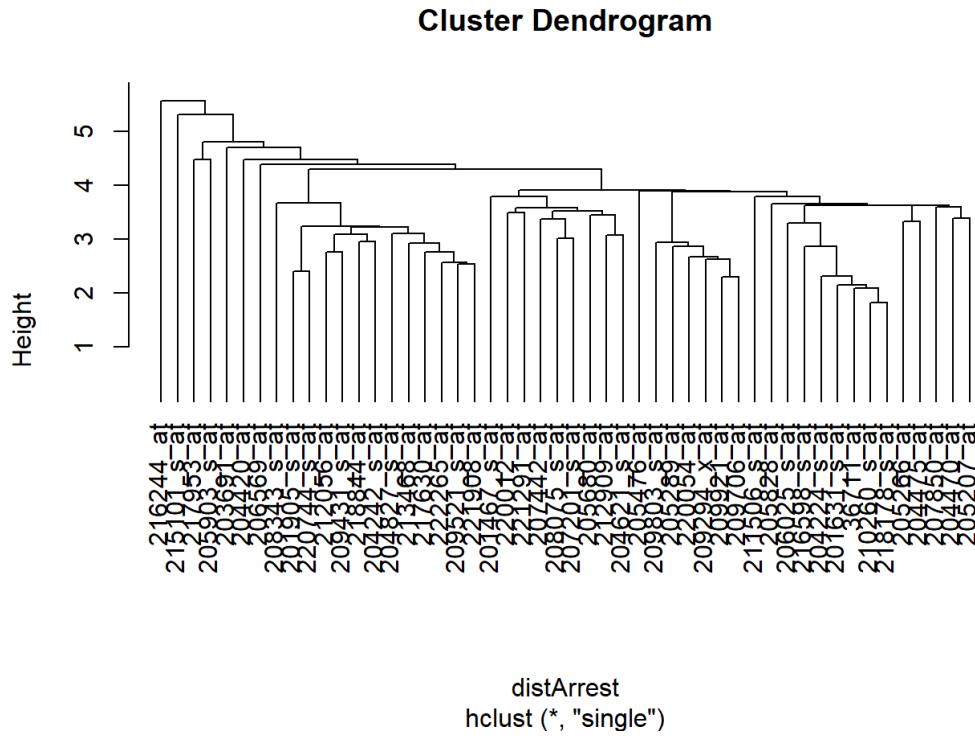
8. Hierarchical Clustering

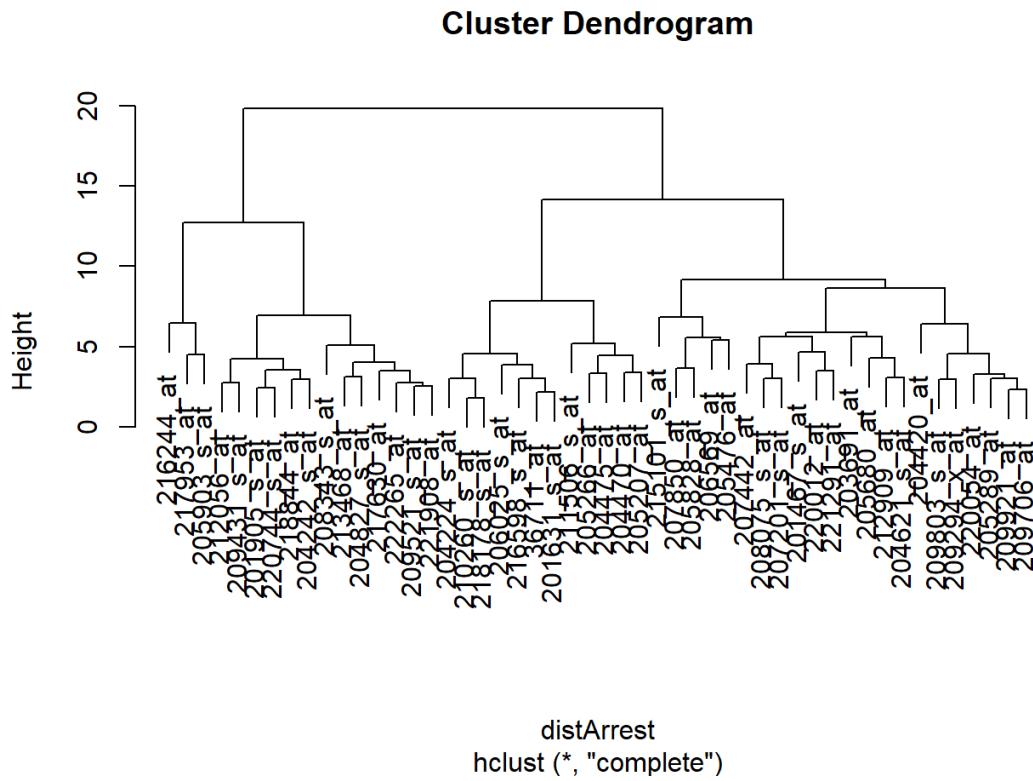
Hierarchical clustering merupakan metode untuk melakukan pengelompokan objek-objek yang mirip pada hierarki yang berdekatan, begitupun sebaliknya sebaliknya. Langkah pertama yang dilakukan pada Hierarchical Clustering adalah membuat matriks jarak menggunakan data yang telah dinormalisasi (expdtgeoselScale) di mana jarak antar setiap pasang pengamatan diukur dengan hasil sebagai berikut.

```
> head(distArrest)
[1] 10.737092 6.365710 8.094022 4.417126 7.046120 3.683050
```

Selanjutnya, dengan jarak diatas, dapat dibentuk sebuah dendogram berdasarkan berbagai macam ukuran kemiripan antarkelompok.

- Hierarchical Clustering dengan Single Linkage Method.
Single linkage menggunakan jarak terkecil antara satu objek dalam klaster dengan satu objek dalam cluster yang lain.

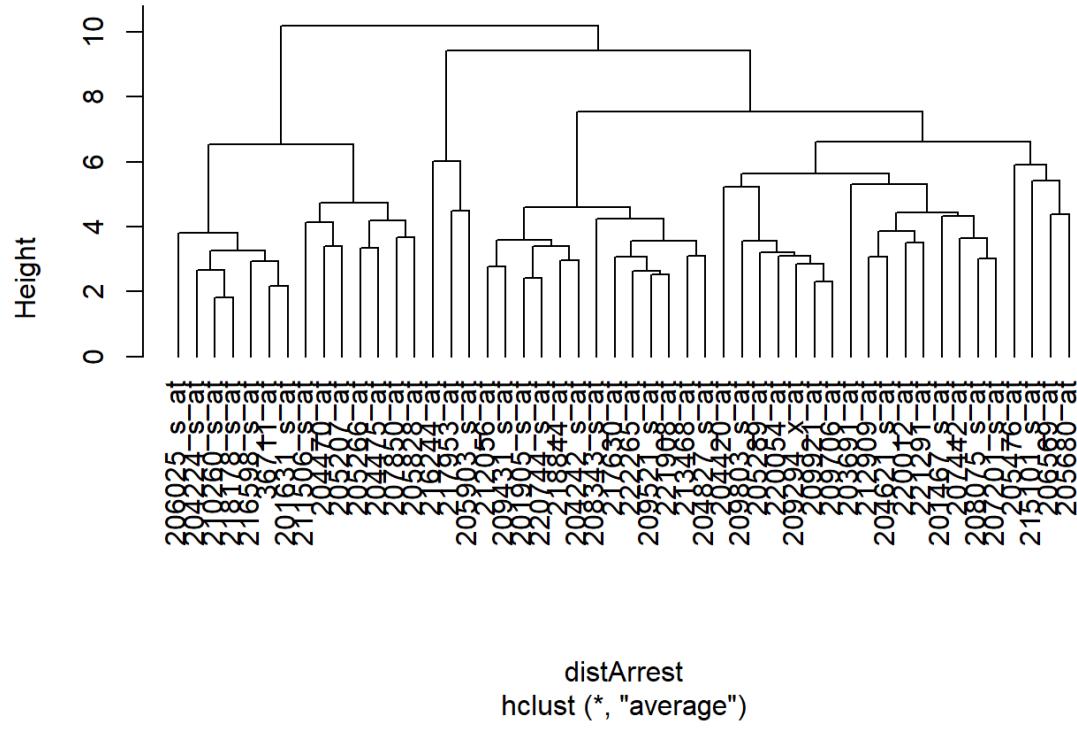




Jika diasumsikan terdapat 3 klaster, maka klaster pertama berisi gen 216244_at, 217953_at, 205903_t_at, 212056_at, 209431_s_at, 201905_5_at, sampai 221908_at, klaster kedua berisi gen 204224_s_at, 210260_s_at, sampai 205207_at, serta gen sisanya masuk ke dalam klaster kedua. Terlihat bahwa metode tersebut cenderung menghasilkan klaster yang lebih seimbang dibandingkan dengan keterkaitan tunggal, karena lebih tangguh dalam menangani outlier karena mempertimbangkan jarak maksimum antar cluster.

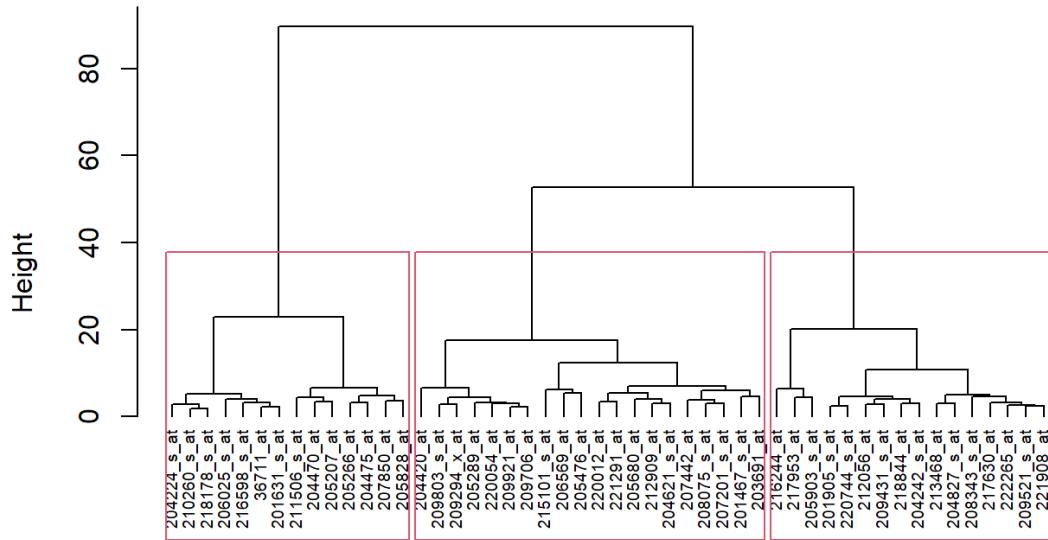
- Hierarchical Clustering dengan Average Linkage Method.
Average linkage menggunakan jarak rata-rata antar objek-objek dalam satu klaster dengan objek-objek dalam klaster lain.

Cluster Dendrogram



- Jika diasumsikan terdapat 3 klaster, maka klaster pertama berisi gen 206025_s_at, 204224_s_at, sampai 205828_at, klaster 2 berisi hanya 216244-at, 217953_at, dan 205903_s_at, serta gen sisanya masuk ke dalam klaster ketiga. Metode ini sering kali dianggap sebagai kompromi antara 2 metode sebelumnya dengan cara menghindari efek berantai yang terlihat pada metode single linkage dan cenderung kurang sensitif terhadap outlier dibandingkan complete linkage.
- Hierarchical Clustering dengan Ward's Method.
Metode Ward memperhatikan keragaman dari klaster tersebut dengan meminimalkan nilai Sum of Squared Error (SSE).

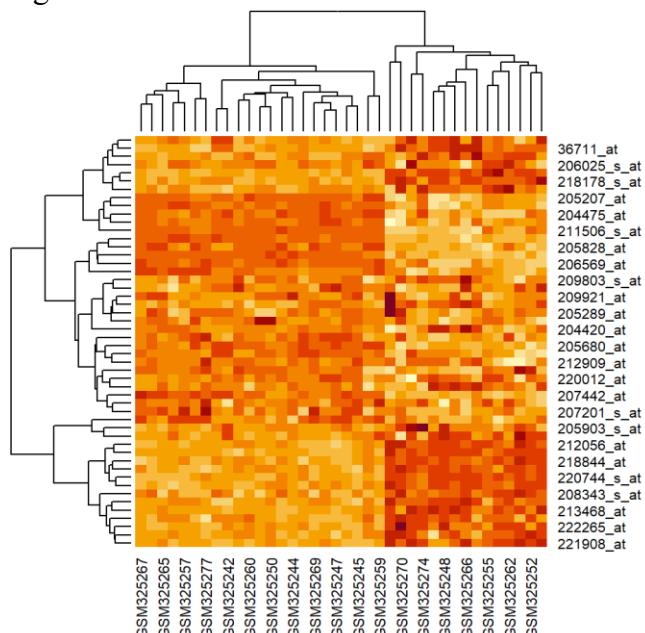
Cluster Dendrogram



distArrest
hclust (*, "ward.D")

Jika diasumsikan terdapat 3 klaster, maka klaster pertama berisi gen yang berada pada kotak merah paling kiri, klaster kedua berada di kotak merah yang tengah, dan klaster ketiga berada di kotak merah paling kanan. Terlihat bahwa metode ini berhasil membuat cluster yang kompak, yaitu jumlah gen pada tiap klaster hampir sama dan seimbang.

- Heatmap expresi gen



Heatmap tersebut masih terlihat kurang begitu jelas untuk menunjukkan kelompok kelompok yang berkorelasi sama / mendekati untuk itu dapat dilakukan analisis lebih lanjut terkait hal ini.

9. Principal Component Analysis (PCA)

Analisis komponen utama (PCA) adalah pendekatan reduksi dimensi klasik. Ini membangun kombinasi linier ekspresi gen, yang disebut komponen utama (PC). PC-PC tersebut ortogonal satu sama lain, dapat secara efektif menjelaskan variasi ekspresi gen, dan mungkin memiliki dimensi yang jauh lebih rendah.

```
> pca
Standard deviations (1, ... , p=38):
 [1] 4.92146408 2.75750467 1.96521041 0.87873175 0.76070600 0.73191254 0.66856527 0.61548730 0.56334976 0.53705548
 [11] 0.51067139 0.46957002 0.45466998 0.41297313 0.38834407 0.35929924 0.34887181 0.30564449 0.29842749 0.28553112
 [21] 0.25758456 0.23826987 0.21563705 0.20201311 0.19246374 0.16599903 0.16338836 0.16052199 0.12231691 0.11156659
 [31] 0.10982547 0.08991564 0.08194339 0.07151031 0.07084482 0.06020183 0.05678153 0.05170687

Rotation (n x k) = (38 x 38):
   PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
GSM325240 -0.11996468 0.26073758 0.226162542 -0.0593086599 0.079986425 -0.036212887 0.141814027 -0.019425126
GSM325241 -0.13329240 0.20348292 0.171783235 -0.2736646632 -0.039389139 -0.340163441 0.054315173 -0.171371596
GSM325242 -0.18047889 -0.05731685 0.092258234 0.2849071798 -0.060520054 0.248774061 -0.240125738 -0.152302143
GSM325243 -0.18743288 -0.07103759 0.045485507 0.1719293335 -0.023891232 0.225014775 -0.132978155 -0.174252483
GSM325244 -0.19201904 -0.06127874 -0.006466789 0.0052476232 0.005925085 0.151516542 -0.140178596 -0.136797083
GSM325245 -0.18891970 -0.09344570 0.002215878 -0.0368988422 0.088626816 0.132234371 -0.014366754 -0.184213940
GSM325246 -0.18879496 -0.08553694 0.026167571 0.0080539466 0.148676769 -0.011041969 0.081465470 -0.173810639
GSM325247 -0.18442968 -0.10893818 0.057759180 -0.0818216311 0.187288764 -0.041954674 0.093503219 -0.216324815
GSM325248 -0.10151200 0.27477473 0.320679655 0.0571356242 0.104646749 0.066842907 -0.214503887 0.005528092
GSM325249 -0.11808273 0.23446863 0.369393633 -0.1494499114 -0.091022407 0.073561940 -0.092012105 0.080346287
GSM325250 -0.19155777 -0.05427472 0.004276529 0.0437331584 0.046838917 0.011016609 -0.159702760 0.166085539
GSM325251 -0.19117370 -0.07946366 0.043734140 -0.0044539669 0.066074421 -0.005074637 -0.076734627 0.081296965
GSM325252 -0.12823262 0.21902334 -0.120457556 0.1333027929 0.306706725 0.292965277 0.007129213 0.095653174
GSM325253 -0.19073417 -0.05100745 -0.108150791 0.0796945921 0.053305056 0.055005655 -0.067063393 -0.102649101
GSM325254 -0.14301040 0.22096942 -0.112329716 0.0989732413 -0.151159343 0.122938206 0.180322165 0.063187674
GSM325255 -0.15152046 0.15777776 -0.170527387 0.0066032726 -0.097373507 0.400457220 0.226492548 0.050585950
GSM325256 -0.17889052 -0.12143270 0.038669424 -0.0089067297 -0.121732392 -0.073077946 0.010433854 0.404677374
GSM325257 -0.18115410 -0.10624337 0.043894153 -0.1603208577 -0.002088429 -0.037783305 0.069146741 0.294603969
GSM325258 -0.17896588 -0.05022109 0.004986605 0.2591999039 -0.334373417 -0.190956796 -0.052979794 0.169064422
GSM325259 -0.18007698 -0.11877652 0.027372949 0.1354937248 -0.266838254 -0.089119208 0.133643423 -0.021151615
GSM325260 -0.18905325 -0.07970281 -0.094423327 0.0711530440 -0.062568810 -0.005967351 -0.088450160 0.225746249
GSM325261 -0.18964862 -0.07547406 -0.052691782 -0.0622071895 0.016618493 0.064216245 -0.021439762 0.058619093
GSM325262 -0.13205411 0.20847669 -0.294667587 -0.0747997053 0.199836430 -0.099230034 0.321812608 0.128541356
GSM325263 -0.18820101 -0.06744184 0.059173076 -0.0009154709 0.186653417 -0.002172034 0.041697736 -0.042337639
GSM325264 -0.11390089 0.24313876 0.151635093 -0.0748619654 -0.142462308 0.096556782 0.240707907 -0.149913161
GSM325265 -0.18619193 -0.08734155 0.032724671 -0.0415694595 -0.117842210 -0.033354374 -0.133661494 0.107585845

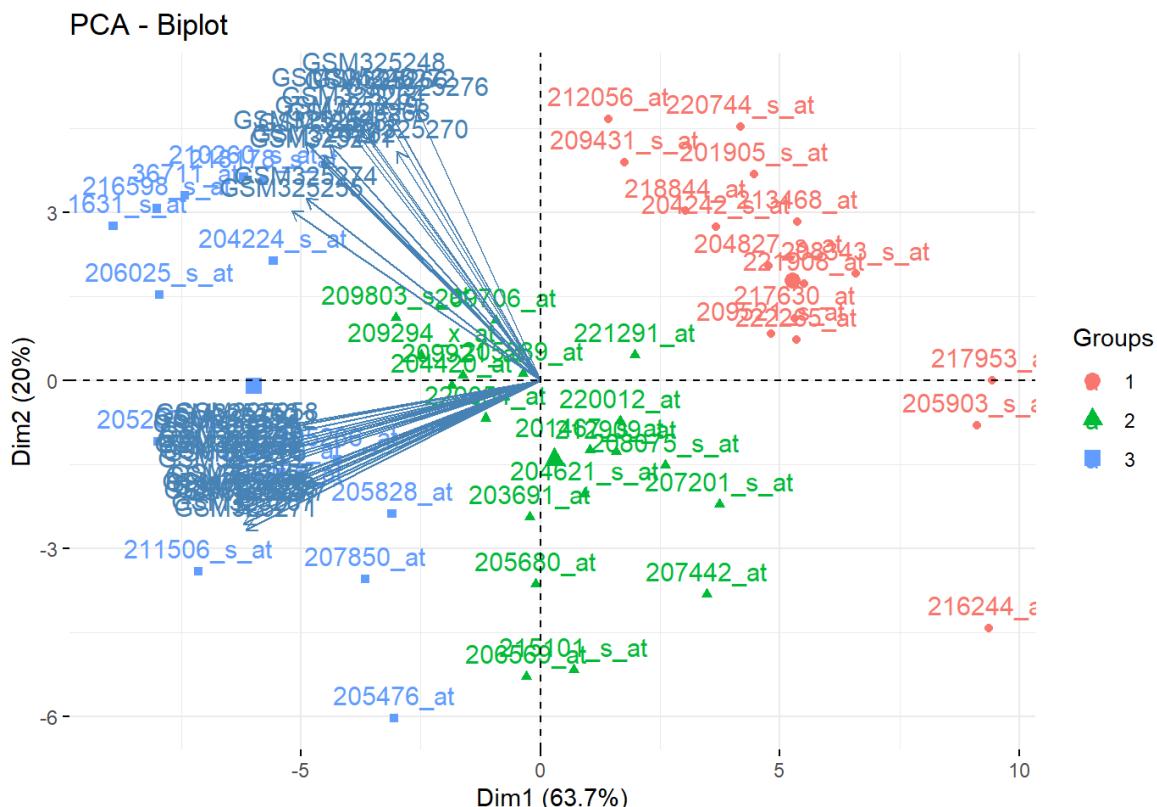
   PC9      PC10     PC11     PC12     PC13     PC14     PC15     PC16
GSM325240 0.048192100 -0.125134297 0.092601499 -0.052879503 0.053040942 0.088343823 -0.111687408 -0.543174943
GSM325241 0.151229614 0.158517159 -0.433130535 -0.146644911 0.110675350 -0.102887588 0.169986044 -0.086371714
GSM325242 0.109034658 0.133859684 -0.099618343 0.128384311 0.001079887 -0.261208687 0.088215181 -0.136992874
GSM325243 0.095694975 0.061938358 -0.112698157 0.156201670 -0.002142471 -0.205900697 -0.092349875 -0.184390781
GSM325244 0.090514412 -0.112004585 0.038965214 -0.232565043 0.063715494 0.061727800 -0.141137250 -0.106707877
GSM325245 0.141902105 -0.083244836 -0.024331828 -0.120417379 -0.074716231 0.101218449 -0.246481872 -0.021266800
GSM325246 -0.153840755 -0.041679700 0.1582823279 0.157009378 0.199021569 -0.199352623 0.139918480 0.211167388
GSM325247 -0.054116686 0.009230501 0.099099413 0.147612105 0.048923599 0.011521766 0.210214228 0.241868540
GSM325248 0.107140424 0.038099570 0.157850891 0.067810206 -0.072306750 0.140905355 -0.016509816 0.037019456
GSM325249 0.239955982 0.082960798 0.044904158 0.119280337 -0.017239685 0.410161115 0.073369080 0.239788055
GSM325250 0.083355483 -0.048332877 0.151560688 -0.118975094 0.110889502 -0.194244115 0.280043020 0.076603304
GSM325251 0.114858566 -0.016942554 0.173922004 -0.138110433 -0.176201651 -0.047597266 0.324845467 0.028283648
GSM325252 -0.375240916 0.169520018 -0.030895354 -0.219917418 -0.094165531 0.151628191 -0.249099347 0.170431496
GSM325253 -0.103735794 0.106646930 0.014431339 -0.137999149 0.251230315 0.115319749 0.263035300 -0.125650518
GSM325254 0.168330343 -0.288700469 0.059800193 -0.0188666820 0.128482384 -0.198232344 0.098228869 0.087311041
GSM325255 0.142959092 -0.249927050 -0.394780415 0.243080710 -0.240807685 0.053897822 0.220123560 0.026087269
GSM325256 -0.019105033 0.019521149 -0.147758099 0.066367228 -0.004880937 -0.089647095 -0.138346580 0.047242773
GSM325257 -0.031071599 -0.108859291 -0.103439567 0.003188021 -0.385947464 -0.114895587 -0.106635100 0.178046972
GSM325258 -0.201367444 0.028990884 0.006633350 0.225312716 0.106170723 0.309208677 0.114237827 0.063768613
GSM325259 -0.074796965 -0.016962984 -0.087424408 0.003115066 0.032949254 0.080049372 0.019218075 -0.126116005
GSM325260 -0.078139458 0.034191929 -0.120144821 -0.034061346 0.136455753 0.276679934 0.003332512 -0.223827701
GSM325261 -0.102528518 -0.127019576 0.113504037 -0.217469030 -0.170405897 0.224485655 0.149608683 -0.137622533
GSM325262 0.082855190 -0.237077139 0.0333005765 0.223495504 0.169044877 -0.005576550 -0.034314201 -0.184600399
GSM325263 0.034347230 0.003709463 -0.040192172 0.378464936 0.201553443 0.115510702 -0.339015733 0.104847243
GSM325264 -0.566511813 0.283991584 0.053155569 0.170423432 -0.193362751 -0.175991500 0.083587848 -0.153152915
GSM325265 -0.084336105 -0.063188904 0.259989714 -0.187543466 -0.243982006 -0.145562601 -0.066130978 -0.150037747
```

	PC17	PC18	PC19	PC20	PC21	PC22	PC23	PC24
GSM325240	0.138894488	0.133841259	0.074774334	-0.274097089	-0.282434519	0.011150854	0.075268648	0.150919037
GSM325241	0.061197529	0.276396600	-0.001925896	-0.059901093	0.138522467	-0.089324436	0.223262981	-0.232182580
GSM325242	-0.075310843	-0.029079805	0.254752186	0.008475616	0.021337247	-0.061154721	0.120488092	-0.187953348
GSM325243	0.061557450	-0.174831169	-0.128244118	-0.060575682	-0.015931676	0.107525672	0.117828928	-0.143804733
GSM325244	-0.016584208	0.201685697	0.129190209	0.283401213	0.144916722	-0.359412519	-0.071887857	0.114967921
GSM325245	0.041933422	-0.126774441	-0.152417379	0.103613744	-0.123295168	0.048401053	0.127468935	0.125872394
GSM325246	-0.011968517	0.091752881	0.112054762	-0.002181889	-0.033014603	0.199449441	0.179760544	0.080629678
GSM325247	-0.105518752	-0.053234764	0.014190660	0.046089190	-0.028859136	-0.200671556	0.206757958	-0.127278268
GSM325248	0.021382368	-0.112414094	0.336273868	-0.086666558	0.358799735	-0.122145948	-0.249059346	-0.132454723
GSM325249	0.211942610	-0.162264132	-0.328009355	0.171896597	-0.032713690	0.022765791	0.098574620	0.069439270
GSM325250	0.028852396	0.3647235898	-0.194007610	0.048512062	-0.123005226	-0.188100049	-0.329135129	0.163261281
GSM325251	0.006965453	0.232692765	-0.052301644	-0.242675675	-0.136692506	0.056565054	0.081979914	0.021877133
GSM325252	0.230440931	0.289715230	0.030652940	-0.021597086	-0.205357874	0.043529140	0.022197410	-0.299258639
GSM325253	0.258189971	-0.146314791	-0.065596591	0.352256448	-0.002217599	0.176155269	0.163180741	0.253487403
GSM325254	0.275854193	-0.099439061	0.125792297	-0.049834591	0.157492128	0.387185456	-0.047269995	-0.028754356
GSM325255	-0.171791501	0.132611906	-0.185539182	0.003487765	-0.030275392	-0.054018985	-0.199171511	0.002499856
GSM325256	0.212819619	0.154086979	0.315921240	0.054080997	0.152206270	0.004653424	0.108293489	0.268761809
GSM325257	0.107211024	-0.106230166	0.195259968	0.023091950	0.021164325	-0.017169641	0.270732628	0.095253800
GSM325258	0.055049733	-0.002844862	0.058923226	-0.036456327	-0.161192002	-0.088435174	0.006888588	-0.305330388
GSM325259	0.027931207	-0.255276137	0.163012888	-0.097695253	-0.376257075	-0.309635504	-0.120957968	0.207965689
GSM325260	-0.190288622	0.133677606	-0.068080717	0.045626748	0.161292233	0.287930993	0.075355235	-0.100901678
GSM325261	-0.432752822	-0.107445315	0.119458172	0.105490235	0.104143495	-0.013149606	0.148872641	-0.088398270
GSM325262	0.033025770	-0.088669555	-0.018410333	0.084698786	0.077488690	-0.259479869	0.139169887	-0.240934658
GSM325263	-0.231371327	0.223626431	-0.037632226	-0.031584971	0.178488992	-0.018716224	-0.006315210	0.309058446
GSM325264	-0.001064445	-0.005979521	-0.137816178	0.069449028	0.193627417	-0.052053410	-0.107013864	0.21255594
GSM325265	-0.010559855	-0.126797206	-0.357328032	-0.210595402	0.292236879	-0.001216597	0.075336240	-0.024039194
	PC25	PC26	PC27	PC28	PC29	PC30	PC31	PC32
GSM325240	0.1320153589	-0.007977149	-0.116755888	-0.104414858	-0.301517240	6.287125e-02	0.06312273	0.189837280
GSM325241	-0.0009557655	0.043232606	0.031866631	0.222516821	0.103958047	-2.868170e-02	-0.17379339	-0.056042600
GSM325242	-0.0136896551	-0.311099953	-0.123595342	-0.228002346	0.262965984	-1.851493e-02	0.06173836	0.017524529
GSM325243	0.2180292447	0.156119008	-0.180923091	-0.038991047	-0.221710765	-1.419500e-01	0.17947446	-0.098170055
GSM325244	0.0671721412	-0.087231424	0.030572631	0.114785124	0.179049973	3.826444e-01	0.04882712	0.033821607
GSM325245	0.0375079703	0.211201191	0.303038327	0.258573099	0.152319414	-3.891131e-01	0.10084432	-0.302741495
GSM325246	0.0788107935	0.045186340	0.045900925	0.135157394	-0.072442441	-1.223266e-01	-0.13897112	-0.027217784
GSM325247	0.0386118484	-0.050493227	-0.113801993	0.075882074	-0.320158867	3.333736e-02	-0.10488218	-0.008489016
GSM325248	-0.2965319041	0.258520380	0.195700162	-0.009665897	-0.256912752	-6.316539e-02	-0.00942414	-0.168418862
GSM325249	0.130800228	-0.289580060	-0.091541322	-0.157385081	0.133828391	-8.221510e-02	-0.03006815	0.050768885
GSM325250	0.2358744589	-0.041201454	0.099654846	-0.179361079	-0.224400769	-8.047179e-02	-0.08650266	-0.313937270
GSM325251	-0.1459879625	0.143026730	0.086689451	-0.062883438	0.369335946	-7.139239e-02	0.33549231	0.140505504
GSM325252	-0.0798623380	-0.188796109	-0.105574232	0.104084430	-0.015112417	-7.346625e-02	-0.07018251	-0.054800521
GSM325253	-0.3955311866	0.207375843	-0.145951566	-0.092499252	-0.085560320	2.260974e-01	-0.07261619	0.046413162
GSM325254	0.0927095176	-0.324914268	0.351260588	0.188733666	-0.038233262	1.416547e-01	-0.02511549	0.071119258
GSM325255	-0.2036283030	0.172049678	-0.173574059	0.095737353	-0.073908155	2.418457e-02	-0.10400094	0.070562114
GSM325256	-0.1839604295	0.017124497	0.142196154	-0.014694721	0.040002289	-5.355163e-01	-0.08161134	0.091187858
GSM325257	0.2635121662	0.152821092	-0.090443586	-0.222148530	-0.063720881	2.902368e-01	-0.02563605	-0.203559754
GSM325258	0.1820013528	0.138179504	0.053298648	0.192883329	0.049160542	2.21866e-01	0.12612866	-0.096776878
GSM325259	-0.1603211788	-0.199179785	0.078133500	0.193019149	0.010130902	-1.393707e-01	-0.13073868	-0.114955819
GSM325260	0.1334362508	0.191150929	0.208499958	-0.056756010	-0.057555705	2.698163e-02	0.10621570	0.012260491
GSM325261	0.1389923225	-0.172727835	0.133522763	-0.078897322	-0.088628993	-2.679568e-01	-0.28222116	0.303318479
GSM325262	-0.0891671957	0.060453064	-0.087713130	-0.173858092	0.288297430	-8.667129e-02	0.03748432	-0.193330934
GSM325263	0.0480028385	-0.132967127	-0.027740865	0.181914772	0.009474672	1.038485e-01	0.19630772	0.136644337
GSM325264	0.1081068286	0.043225972	0.232429323	-0.120194104	0.185707133	-1.685298e-05	0.02320541	-0.016961904
GSM325265	-0.1355134120	-0.147706388	-0.268811882	0.402395533	-0.041878263	1.547190e-01	-0.06680835	-0.128449856
	PC33	PC34	PC35	PC36	PC37	PC38		
GSM325240	0.1466912867	-0.173761354	-0.053237724	-0.060504538	-0.104589605	-0.1181687441		
GSM325241	-0.1783588470	0.006686961	0.068164935	-0.024945895	0.146217792	0.038599300		
GSM325242	0.1728793461	-0.089717203	0.164231429	-0.224415024	0.037898117	-0.3114157013		
GSM325243	-0.2992003207	0.135245759	-0.068334783	0.192848871	0.061842200	0.4714394291		
GSM325244	0.0745754438	-0.016979219	-0.435304068	0.015091527	0.012497937	0.2682321907		
GSM325245	0.1896378960	-0.340680325	0.009385298	-0.055156903	0.108410484	-0.161398187		
GSM325246	-0.1454441025	0.135037507	-0.407882841	-0.383531506	-0.380430121	-0.1063201681		
GSM325247	0.5496896034	0.029295166	0.065590388	0.380291627	0.068206128	0.0387080644		
GSM325248	-0.0860263953	0.056654366	-0.038302461	0.040451492	-0.060284029	-0.1076704077		
GSM325249	-0.0001210042	0.135969062	-0.077496150	-0.102375563	-0.082135915	0.0551069563		
GSM325250	-0.0734834512	-0.040603528	0.256121220	-0.127978181	0.099531798	0.0412765775		
GSM325251	-0.0615828301	0.131149599	-0.083077761	0.432379260	-0.204942850	-0.0852437697		
GSM325252	-0.0809339452	0.078123938	0.013143873	0.056238145	0.076478358	-0.0209264186		
GSM325253	-0.1685800281	-0.112767028	0.155783940	0.047720853	0.085387508	-0.1037629928		
GSM325254	0.0346026478	-0.046083077	0.046288975	0.192940376	0.137048410	0.0349065351		
GSM325255	0.0413762736	-0.063427183	-0.101331411	-0.090154093	-0.075674875	-0.0659217830		
GSM325256	0.2302528589	-0.090183478	0.113046119	-0.011869858	-0.158330060	0.3681989339		
GSM325257	-0.1554011060	0.005809686	-0.132065544	0.050408742	0.221125305	-0.2969512613		
GSM325258	-0.0625423757	-0.380729285	0.056842771	-0.013172900	-0.206972435	0.0563681553		
GSM325259	-0.1314943446	0.433574360	-0.050629024	0.084984881	0.132321116	-0.1051787296		
GSM325260	0.3239931044	0.497484472	-0.005982329	-0.168991792	0.161438826	-0.0619385084		
GSM325261	-0.3086019856	-0.192922571	0.069952713	0.058067377	-0.001662876	0.0732693231		
GSM325262	-0.0653773877	0.094657659	0.082185103	-0.098654772	0.063882203	0.0952726619		
GSM325263	-0.2828965279	0.025311052	0.311133626	0.152985536	0.025414928	-0.1871956453		
GSM325264	0.0774564129	-0.040742564	0.013071187	0.063216713	0.034237270	0.0775039626		
GSM325265	0.0593616127	0.072014916	0.158014383	-0.146981534	-0.175370634	-0.0453673779		

[reached getoption("max.print") -- omitted 12 rows]

Hal ini menunjukkan bahwa dua PC pertama memiliki kontribusi yang dominan terhadap variabilitas data. Sementara PC1 (Principal Component 1) mewakili arah dengan variabilitas tertinggi dalam data. Gen-gen dengan loadings positif pada PC1 memberikan kontribusi positif terhadap variasi data yang terutama ditunjukkan oleh PC1. PC2 (Principal Component 2) Mewakili arah variabilitas kedua tertinggi setelah PC1. Gen-gen dengan loadings positif pada PC2 memberikan kontribusi positif terhadap variasi data yang terutama ditunjukkan oleh PC2, dan seterusnya sampai PC38.

Dari hasil penguraian oleh PCA, dapat dibentuk sebuah biplot yang menampilkan gabungan antara observasi, dengan variabel dari objek tersebut. Semakin dekat titiknya, makin mirip karakteristiknya. Di sisi lain, semakin dekat anak panah, makin besar korelasi dari variabel yang direpresentasikan oleh panah tersebut. Berikut ditampilkan hasil dari grafik biplot.

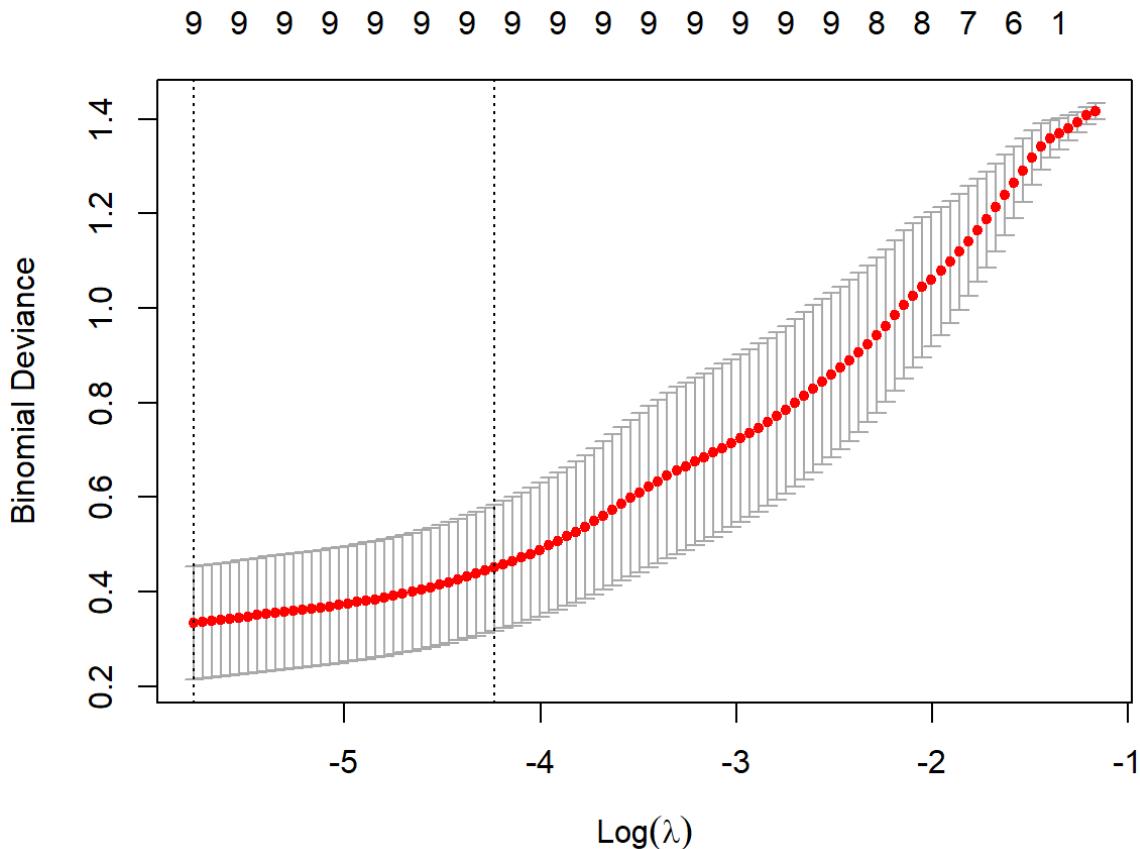


Dari biplot di atas, bisa dilihat bahwa ketiga cluster bisa dikatakan cukup terpisah dengan jelas. Akan tetapi, tampilannya yang kecil membuat data saling tumpang tindih dan sulit diinterpretasikan.

10. LASSO Classification

Akan dibuat model regresi dengan variabel independen nilai dari expression gene yang telah diseleksi (expdtgeoscale) terhadap variabel dependen pengelompokan pasien (group). Karena output dari regresi bernilai biner (1 untuk jenis liposarkoma yang tidak diobati dengan doxorubicin dan 0 untuk yang diobati dengan doxorubicin) dan akan digunakan

regresi lasso, digunakan function `glmnet` dengan family binomial dan alpha = 1. Catatan: digunakan alpha = 0 untuk ridge, sedangkan alpha yang bernilai antara 0 dan 1 untuk elastic net. Dan berikut adalah plot hasil regresi lassonya.



Plot tersebut menampilkan cross-validation error berdasarkan nilai $\log(\lambda)$. Garis vertikal putus-putus kiri menunjukkan bahwa $\log \lambda$ optimal adalah sekitar -5, yang meminimalkan kesalahan prediksi. Nilai λ ini akan memberikan model yang paling akurat. Secara spesifik, nilai λ yang didapat adalah 0.003116811.

```
> optimal_lambda <- lasso_model$lambda.min
> optimal_lambda
[1] 0.003116811
```

Dengan menggunakan nilai λ yang paling akurat, didapat koefisien sebagai berikut dengan nilai titik menunjukkan bahwa variabel itu tidak signifikan secara statistik dalam menentukan kelompok pasien.

```

> lasso_coefficients <- coef(lasso_model, s = optimal_lambda)
> print(lasso_coefficients)
51 x 1 sparse Matrix of class "dgCMatrix"
  s1
(Intercept) -36.675323921
201905_s_at   .
206569_at     .
212909_at     .
207442_at     .
209521_s_at   3.612142615
201467_s_at   .
213468_at     .
212056_at     .
217953_at     0.007997295
209431_s_at   .
209921_at     .
36711_at      .
208343_s_at   0.853024353
210260_s_at   .
204224_s_at   .
204470_at     .
207850_at     .
216244_at     .
205207_at     .
211506_s_at   .
205903_s_at   0.576612524
205266_at     .
204475_at     .
205828_at     .
205680_at     .

209706_at     .
204621_s_at   .
220054_at     .
203691_at     .
220744_s_at   .
220012_at     .
218178_s_at   .
216598_s_at   .
208075_s_at   -1.177163540
205476_at     .
215101_s_at   .
205289_at     -0.086438995
207201_s_at   .
206025_s_at   .
209803_s_at   .
218844_at     .
221291_at     -0.438969443
204420_at     .
204242_s_at   .
221908_at     .
222265_at     1.289532932
209294_x_at   .
201631_s_at   .
204827_s_at   1.870012840
217630_at     .

```

Terlihat bahwa variabel yang berpengaruh signifikan dalam pengelompokan pasien adalah 209521_s_at, 217953_at, 208343_s_at, 205903_s_at, 208075_s_at, 205289_at, 221291_at, 222265_at, dan 204827_s_at.

Fungsi cv.glmnet() juga menemukan nilai lambda yang memberikan model paling sederhana tetapi juga berada dalam satu kesalahan standar dari nilai lambda optimal. Dengan nilai ini yang disebut lambda.1se, didapat nilai lambda sebesar 0.01446696.

```

> optimal_lambda2 <- lasso_model$lambda.1se
> optimal_lambda2
[1] 0.01446696

```

Didapat nilai koefisien sebagai berikut.

```
> lasso_coefficients2 <- coef(lasso_model, s = optimal_lambda2)
> print(lasso_coefficients2)
51 x 1 sparse Matrix of class "dgCMatrix"
  s1
(Intercept) -21.48211740
201905_s_at   .
206569_at     .
212909_at     .
207442_at     .
209521_s_at   2.07202264
201467_s_at   .
213468_at     .
212056_at     .
217953_at    0.06931857
209431_s_at   .
209921_at     .
36711_at      .
208343_s_at   0.42187577
210260_s_at   .
204224_s_at   .
204470_at     .
207850_at     .
216244_at     .
205207_at     .
211506_s_at   .
205903_s_at   0.30822467
205266_at     .
204475_at     .
205828_at     .
205680_at     .

                                         209706_at   .
                                         204621_s_at   .
                                         220054_at   .
                                         203691_at   .
                                         220744_s_at   .
                                         220012_at   .
                                         218178_s_at   .
                                         216598_s_at   .
                                         208075_s_at  -0.70134230
                                         205476_at   .
                                         215101_s_at   .
                                         205289_at  -0.16676364
                                         207201_s_at   .
                                         206025_s_at   .
                                         209803_s_at   .
                                         218844_at     .
                                         221291_at  -0.16621603
                                         204420_at     .
                                         204242_s_at   .
                                         221908_at     .
                                         222265_at   1.00226841
                                         209294_x_at   .
                                         201631_s_at   .
                                         204827_s_at   1.01376972
                                         217630_at     .
```

Terlihat bahwa variabel yang berpengaruh signifikan dalam pengelompokan pasien bukannya menjadi lebih sederhana tetapi menjadi lebih kompleks, yaitu gen 209512_s_at, 217953_at, 208343_at, 205903_s_at, 208075_s_at, 205289_at, 221291_at, 222265_at, dan 204827_s_at. Hal ini dapat menghasilkan model yang lebih kompleks dengan lambda optimal, yang mungkin menunjukkan lebih banyak koefisien non-nol (dalam Lasso) atau pengaruh yang lebih besar pada beberapa variabel (dalam Ridge), tetapi masih mencegah overfitting pada data uji yang baru.

IV. Kesimpulan

- Data yang digunakan berasal dari 38 sampel manusia (homo sapiens), 19 dengan sel yang diberi perlakuan pemberian doxorubicin dan 19 yang tanpa diberi perlakuan tersebut, dan data terdiri dari 22283 fitur gen yang akhirnya diseleksi (gene filtering) sehingga tersisa 6,322 fitur gen saja.
- LIMMA membantu mengidentifikasi gen yang memiliki perbedaan ekspresi yang signifikan, seperti yang terlihat dalam pemilihan top 50 gen dengan variansi tinggi. Dari LIMMA juga didapat masing masing 10 gene yang overexpressed dan underexpressed.
- Penentuan jumlah cluster yang optimal dapat menggunakan analisis Elbow Plot dan Gap Statistic. Dengan Elbow Plot, jumlah klaster optimal pada data ini adalah sebanyak 3. Sedangkan dengan Gap Statistic, metode firstmax menghasilkan 8 klaster optimal, sehingga untuk memudahkan interpretasi dipilih 3 cluster sebagai acuan analisis selanjutnya.
- Metode K-means clustering berhasil mengelompokkan top 50 gen ke dalam tiga kelompok (K=3). Distribusi gen menunjukkan variasi antar kelompok, dengan kelompok 1 memiliki 16 fitur, kelompok 2 memiliki 21 fitur, dan kelompok 3 memiliki 13 fitur.

- Digunakan juga metode hierarchical clustering dengan berbagai pendekatan seperti single linkage, complete linkage, average linkage, dan Ward's method. Hasil clustering bergantung pada pendekatan yang digunakan dengan Ward's method menunjukkan hasil terbaik untuk pengelompokan kedalam 3 kelompok.
- Dilakukan juga Principal Component Analysis (PCA) dan didapatkan hasil PC1 dan PC2 yang berkontribusi dominan untuk menjelaskan variabilitas pada data. Didapatkan pula visualiasinya berupa biplot yang dibagi menjadi 3 kelompok.
- Lasso regression pada model klasifikasi menggunakan ekspresi gen sebagai variabel independen. Regresi lasso membentuk model yang dapat mengidentifikasi pengelompokan variabel gen pasien secara parsimonious, tetapi variabel yang berpengaruh signifikan dalam pengelompokan pasien menjadi lebih kompleks, yaitu gen 209512_s_at, 217953_at, 208343_at, 205903_s_at, 208075_s_at, 205289_at, 221291_at, 222265_at, dan 204827_s_at .

Daftar Pustaka

- [1.] Halodoc. (2022, May 30). *Liposarkoma - Gejala, Penyebab, Dan Pengobatan* | Halodoc. halodoc. <https://www.halodoc.com/kesehatan/liposarkoma>
- [2.] Akalin, A. (2020, September 30). *4.1 clustering: Grouping samples based on their similarity* | Computational genomics with R. Site not found · GitHub Pages. <https://compgenomr.github.io/book/clustering-grouping-samples-based-on-their-similarity.html>
- [3.] Handhayani, T., & Hiryanto, L. (2015). *Intelligent kernel K-Means for clustering gene expression*. *Procedia Computer Science*, 59, 171–177. <https://doi.org/10.1016/j.procs.2015.07.544>
- [4.] Dumitrescu, D. (1997). *Fuzzy hierarchical classification methods in analytical chemistry*. In Elsevier eBooks (pp. 321–356). <https://doi.org/10.1016/b978-012598910-7/50011-1>
- [5.] Ma S, Dai Y. *Principal component analysis based methods in bioinformatics studies*. *Brief Bioinform*. 2011 Nov;12(6):714-22. doi: 10.1093/bib/bbq090. Epub 2011 Jan 17. PMID: 21242203; PMCID: PMC3220871.
- [6.] Ghosh D, Chinnaiyan AM. *Classification and selection of biomarkers in genomic data using LASSO*. *J Biomed Biotechnol*. 2005 Jun 30;2005(2):147-54. doi: 10.1155/JBB.2005.147. PMID: 16046820; PMCID: PMC1184048.
- [7.] Dumitrescu, D. (1997). *Fuzzy hierarchical classification methods in analytical chemistry*. In Elsevier eBooks (pp. 321–356). <https://doi.org/10.1016/b978-012598910-7/50011-1>
- [8.] Saji, B. (2023, September 20). *Elbow method for finding the optimal number of clusters in K-Means*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/>
- [9.] Löhr, T. (2023, August 21). *K-Means Clustering and the GAp-Statistics - towards Data science*. Medium. <https://towardsdatascience.com/k-means-clustering-and-the-gap-statistics-4c5d414acd29>

Lampiran: R code yang digunakan

```
## EXPRESSION GENE ##

library(Bioconductor)
library(GEOquery)

dtgeo <- getGEO('GDS3514') #Load dataset
dtgeo

eset <- GDS2eSet(dtgeo,do.log2 = TRUE) #membentuk dataset menjadi expression set
eset

phdtgeo<-pData(eset) #melihat phenotype data
phdtgeo
head(phdtgeo)

expdtgeo <- exprs(eset)
dim(expdtgeo)
head(expdtgeo)

Meta(dtgeo)$platform

#BiocManager::install("hgu133a.db", force = TRUE)
annotation(eset) <- "hgu133a" #melihat annotation yang digunakan untuk melakukan filtering
library(hgu133a.db) #didapat annotationnya hgu133a.db

## GENE FILTERING ##

require(genefilter)
esetFilt = nsFilter(eset)
esetFilt

expdtgeoFilt <- exprs(esetFilt$eset)
dim(expdtgeoFilt)
par(mfrow=c(1,2))
hist(expdtgeo, main ='original') #plot sebelum di filter
hist(expdtgeoFilt,main="filtered") #plot setelah filtering

vargrp <- phdtgeo[,3] #melihat kelompok untuk selanjutnya dilakukan encoding
table(vargrp)
group <- ifelse(vargrp == "doxorubicin", 0, 1) #didapat 2 perlakuan sebagai berikut
group
```

SELEKSI GEN BERDASARKAN LIMMA ANALYSIS

```
library(limma)
design <- model.matrix(~group)
design
fit <- eBayes(lmFit(expdtgeoFilt,design))
fit

topResult <- topTable(fit, coef=2, number=50)
rownames(topResult)

selected <- rownames(expdtgeoFilt) %in% rownames(topResult)
expdtgeosel <- expdtgeoFilt [selected, ]
heatmap(expdtgeosel)

par(mfrow=c(2,2))
for(i in 1:4) plot(vargrp, expdtgeosel[i,],
                   main= rownames(expdtgeosel)[i])

#OverExpressed Gene
highest_var_gene <- rownames(topResult)[which.max(topResult$AveExpr)]
# Menampilkan informasi gen dengan varians tertinggi
highest_var_gene_info <- topResult[which(rownames(topResult) == highest_var_gene), ]
print("Gene with the highest variance:")
print(highest_var_gene_info)
top_variance_genes <- rownames(topTable(fit, coef=2, number=10))
# Menampilkan informasi tentang 10 gen dengan varians tertinggi
print("Top 10 genes with the highest variance:")
top_variance_genes_info <- topResult[which(rownames(topResult) %in%
top_variance_genes), ]
print(top_variance_genes_info)

#Underexpressed gene
topResult_underexpressed <- topTable(fit, coef=2, number=50, sort.by="logFC")
head(topResult_underexpressed)
top_underexpressed_gene <- rownames(topResult_underexpressed)[1]
top_underexpressed_gene_info <-
topResult_underexpressed[which(rownames(topResult_underexpressed) ==
top_underexpressed_gene), ] == 
print("Underexpressed gene:")
print(top_underexpressed_gene_info)
topResult_underexpressed <- topTable(fit, coef = 2, number = 10, sort.by = "logFC")
# Menampilkan 10 gen teratas yang dianggap underexpressed pada kelompok 'doxorubicin'
```

```

print("Top 10 underexpressed genes:")
print(topResult_underexpressed)

## GENE ONTOLOGY ##

library("annotate")
GeneSelected <- select(hgu133a.db, rownames(topResult), c("SYMBOL", "ENTREZID",
"GENENAME"))
GeneSelected

ids <- rownames(topResult)
GeneSelected <- select(hgu133a.db, ids, c("SYMBOL", "ENTREZID", "GENENAME",
"GO"))
GeneSelected

library(GO.db)
GOSelected <- select(GO.db, GeneSelected$GO, c("TERM","GOID"))
head(GOSelected)

finalres <- cbind(GeneSelected,GOSelected)
head(finalres)

#### Visualisasi Awal
pairs(expdtgeosel)
library(corrplot)
corrplot(cor(expdtgeosel), method="ellipse", type="upper")

## PENENTUAN KLASTER YANG OPTIMAL ##

# Within Cluster Variaton: Didapat 3 Kluster #
expdtgeoselScale <- scale(expdtgeosel, scale = T)
set.seed(123)
k.max <- 15
wss <- sapply(1:k.max, function(k){kmeans(expdtgeoselScale, k, nstart=10)$tot.withinss})

plot(1:k.max, wss, type ='b', pch = 19, frame = FALSE, xlab = "Number of Cluster K",
     ylab = "Total Within-cluster Sum of Squares") #Number Of Cluster = 3

# Gap Statistic: Didapat 8 Kluster #
library(cluster)
gap_stat <- clusGap(expdtgeoselScale, FUN = kmeans, nstart =10,
                     K.max = 15, B = 100)
print(gap_stat, method = "firstmax")

```

```

plot(gap_stat, frame = FALSE, xlab = "Number of Cluster K") #Number of cluster 3

library(cluster)
gap_stat <- clusGap(expdtgeoselScale, FUN = kmeans, nstart =10,
                     K.max = 15, B = 100)
print(gap_stat, method = "firstmax")
print(gap_stat, method = "firstSEmax")
plot(gap_stat, frame = FALSE, xlab = "Number of Cluster K") #Number of cluster 3

## VISUALISASI KLASTER ##

#KMEANS
set.seed(123)
kMres <- kmeans(expdtgeoselScale ,centers=3)
kMres$cluster
kMres$size
pairs(expdtgeoselScale, col=(1:3)[kMres$cluster], pch=16)

set.seed(2106704736)
kMres1 <- kmeans(expdtgeosel, centers=3)
head(kMres1$cluster)
kMres1$size

kMres2 <- kmeans(expdtgeosel, centers=3)
head(kMres2$cluster)
kMres2$size

table(kMres1 $cluster, kMres2 $cluster)

#HIERARIRCAL
set.seed(2106704736)
distArrest <- dist(expdtgeoselScale)
head(distArrest)

res1 <- hclust(distArrest, method="single")
plot(res1, hang = -2, cex = 1)

res2 <- hclust(distArrest, method="complete")
plot(res2)

res3 <- hclust(distArrest, method="average")
plot(res3, hang = -2)

res4 <- hclust(distArrest, method="ward.D")

```

```

plot(res4, hang = -1, cex = 0.6)
rect.hclust(res4, k = 3)

cutree(res4, k=3)
res4.a <- as.dendrogram(res4)
plot(res4.a, xlab ="Height", horiz =TRUE)

heatmap(expdtgeoselScale)

## GRAFIK BI PLOT HASIL PCA ##
pca <- prcomp(expdtgeosel, scale=T)
pca
library(factoextra)
fviz_pca_biplot(pca, habillage = kMres2$cluster) #Metode BiPlot

```

SELEKSI GEN BERDASARKAN REGRESI LASSO

```

par(mfrow=c(1,1))
library(glmnet)
lasso_model <- cv.glmnet(t(expdtgeosel), group, family = "binomial", alpha = 1)
plot(lasso_model)
optimal_lambda <- lasso_model$lambda.min
optimal_lambda
lasso_coefficients <- coef(lasso_model, s = optimal_lambda)
print(lasso_coefficients)

optimal_lambda2 <- lasso_model$lambda.1se
optimal_lambda2
lasso_coefficients2 <- coef(lasso_model, s = optimal_lambda2)
print(lasso_coefficients2)

```