UJIAN TENGAH SEMESTER SAINS DATA GENOM

(SCST603107)

ANALISIS PERBEDAAN EKSPRESI GEN PADA SUBTIPE KANKER PAYUDARA

Oleh

Nama : Kamal Muftie Yafi

NPM : 2106725034

PROGRAM STUDI STATISTIKA

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

DEPOK

2023

1. Pendahuluan

Kanker payudara merupakan kanker paling umum yang didiagnosis pada wanita, terhitung lebih dari 1 dari 10 diagnosis kanker baru setiap tahunnya. Penyakit ini merupakan penyebab kematian terbanyak kedua akibat kanker pada wanita di dunia. Kanker payudara berkembang secara diam-diam, dan sebagian besar penyakit ditemukan melalui pemeriksaan rutin. Kanker payudara adalah kelainan molekuler heterogen yang diklasifikasikan menjadi lima subtipe termasuk luminal A, luminal B, basal-like, HER2-enriched dan normal-like. Klasifikasi ini didasarkan pada ada/berlimpahnya reseptor estrogen (ER), reseptor progesteron (PR), HER2 dan Ki67. Identifikasi jaringan tersebut merupakan langkah penting menuju desain terapi yang ditargetkan pada kanker payudara.

Dalam penelitian ini, penulis telah mengambil data dari sebuah dataset eksperimen *microarray* (GSE45827) dari Curated Microarray Database (CuMiDa). Data tersebut berisi spesimen tumor pada saat operasi sebelum perawatan pasien. Total RNA diekstraksi dari semua sampel dan seluruh transkriptome dikuantifikasi dengan platform berbasis chip GPL570 (HG-U133_Plus_2) Affymetrix U133 Plus 2.0. *Package* R digunakan untuk identifikasi gen yang diekspresikan secara berbeda (*diferentially expressed genes*) dan penilaian ontologi gen (*gene ontology*).

2. Metode

2.1 Exploration

Exploratory Data Analysis (EDA) bagaikan inti dari seluruh proses analisa data. Kemampuan untuk melakukan EDA dengan baik merupakan prasyarat esensial untuk seluruh profesi yang berhubungan dengan pengolahan data, baik itu *business intelligence*, data *analyst*, *data scientist*, dan lain sebagainya. EDA juga menjadi tahapan awal dalam kebanyakan proses analisis data, dan secara signifikan memengaruhi kualitas analisis data yang berikutnya.

2.2 Gene Filtering

Gene filtering (penyaringan gen) termasuk ke dalam tahap *pre-pocessing* yang dilakukan untuk mereduksi data berdasarkan kriteria tertentu. Pada proses ini, *package* R yang digunakan adalah `genefilter`. *Package* tersebut dapat mengeluarkan gen-gen yang tidak banyak bervariasi antarsampel, memiliki ekspresi yang kecil di seluruh sampel, dan gen yang tidak memiliki cukup anotasi. Proses ini dilakukan untuk mengurangi waktu proses analisis lanjutan dan mengurangi terjadinya *false positive* yang akan meningkatkan *power* dari studi.

2.3 Gene Expression Data

Setiap sel pada makhluk hidup mempunyai peranannya masing-masing, dan aktivitas setiap sel tersimpan dalam kode genetiknya, yaitu asam deoksiribonukleat (DNA). Dogma Sentral Biologi Molekuler menyebutkan bahwa setiap sel menghasilkan satu mRNA per protein, sehingga tingkat ekspresi gen setiap gen dapat berbeda. Teknologi *microarray* juga sering digunakan untuk mengukur nilai ekspresi gen. Data mentah *microarray* adalah gambar, yang harus diubah menjadi matriks–tabel ekspresi gen di mana baris mewakili gen, kolom mewakili berbagai sampel seperti jaringan atau kondisi eksperimental, dan angka di setiap sel mencirikan tingkat ekspresi gen tertentu dalam kondisi tertentu.

2.4 Differential Gene Expression Analysis

Analisis mRNA dan protein banyak digunakan untuk membandingkan pola ekspresi gen antara sel atau jaringan yang berbeda jenis dan dalam kondisi berbeda; misalnya antara sel normal dan sel kanker. Tujuan dari individu yang mengembangkan metode ini adalah untuk memungkinkan analisis yang lebih cepat, sederhana, lebih sensitif dan sistematis, dan selama beberapa dekade terakhir teknik ini menjadi semakin canggih.

2.5 Gene Ontology

Proyek Gene Ontology (GO) (http://www.geneontology.org/) menyediakan kosakata dan klasifikasi yang terstruktur dan terkontrol yang mencakup beberapa domain biologi molekuler dan seluler dan tersedia secara bebas untuk digunakan komunitas dalam anotasi gen, produk gen dan urutan. Banyak database organisme model dan kelompok anotasi genom menggunakan GO dan menyumbangkan kumpulan anotasinya ke sumber daya GO. Basis data GO mengintegrasikan kosakata dan kontribusi anotasi serta menyediakan akses penuh ke informasi ini dalam beberapa format.

3. Hasil dan Analisis

3.1 Eksplorasi

Exploratory Data Analysis (EDA) dilakukan pada data ekspresi gen dua dimensi dengan data numerik terkait ekspresi dari sub tipe kanker payudara. Data tersebut diunduh dari situs Structural Bioinformatics and Computational Biology (SBCB) Lab pada tautan berikut: https://sbcb.inf.ufrgs.br/data/cumida/Genes/Breast/GSE45827/Breast_GSE45827.csv (diakses pada 14 Oktober 2023). Data ekspresi gen terdiri dari 54.675 gen, yang dibagi menjadi enam kelas—empat sub tipe kanker payudara invasif primer (41 Basal, 30 HER2, 29 Luminal A dan 30 Luminal B) serta 7 sampel jaringan normal dan 14 garis sel.

```
> # Set seed
> set.seed(2106725034)
>
> ## Load the dataset ##
> library(data.table)
> dtgse <- fread("Breast_GSE45827.csv")
>
> dfgse <- as.data.frame(dtgse)
> as.data.frame(table(dfgse$type))
       Var1 Freq
1     basal   41
2 cell_line   14
3       HER   30
4 luminal_A   29
5 luminal_B   30
6    normal    7
```

Selanjutnya, akan diambil sampel acak sebanyak 50% dari total gen, yakni 151 sampel dan 27337 gen.

```
> typesample <- subset(dfgse, select = c(samples, type))
> rdgse <- subset(dfgse, select = -c(type, samples))
>
> rdgse <- rdgse[, sample(ncol(rdgse), ncol(rdgse) * 0.5)]
> rdgse <- rdgse[, order(names(rdgse))]
>
> rdgse <- cbind(typesample, rdgse)
```

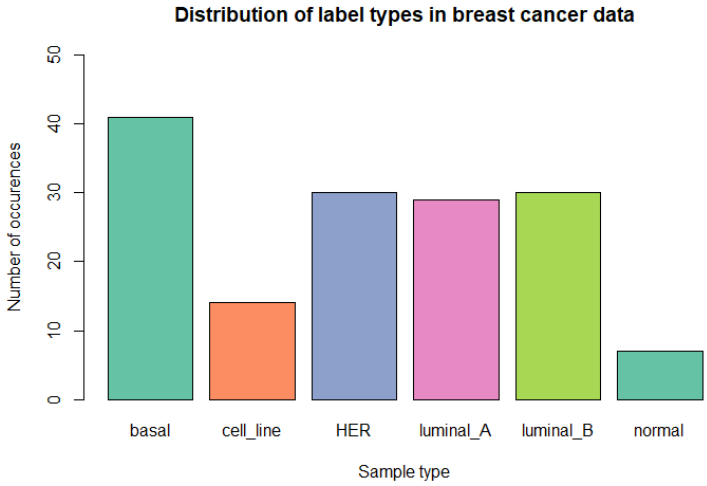
Dari data yang sudah diambil sampel acak, akan dilakukan transformasi menjadi data *expression* dari gen-gennya.

```
> exprdgse <- rdgse
> rownames(exprdgse) <- exprdgse$samples
> types <- exprdgse$type
>
> exprdgse <- subset(exprdgse, select = -c(type, samples))
> exprdgse_t <- transpose(exprdgse)
>
> rownames(exprdgse_t) <- colnames(exprdgse)
> colnames(exprdgse_t) <- rownames(exprdgse)
>
> exprdgse_mat <- as.matrix(exprdgse_t)
```

Setelah diperoleh bentuk data yang sesuai, yakni dalam bentuk *expression*, akan dibentuk visualisasi terhadap distribusi datanya.

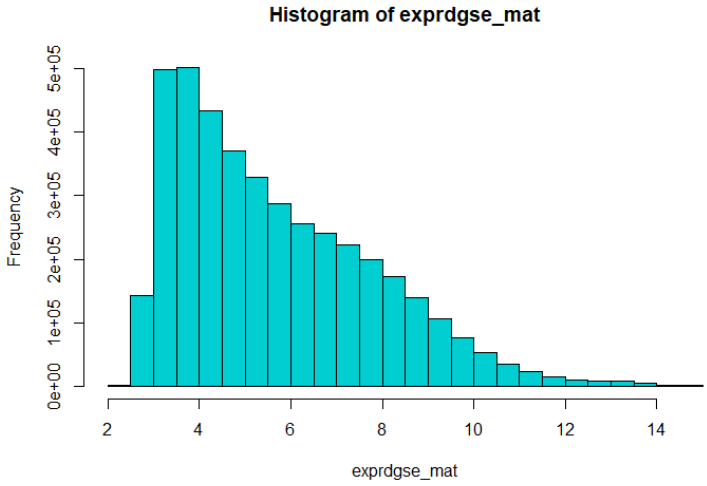
```
> library(RColorBrewer)
> colour <- brewer.pal(5, "Set2")
> barplot(table(rdgse$type),
+         main = "Distribution of label types in breast cancer data",
+         xlab = "Sample type",
```

```
+          ylab = "Number of occurences",
+          ylim = c(0,20),
+          col = colour)
```



Gambar 1. Distribusi Tipe Kanker Payudara

```
> hist(exprdgse_mat, col = 'darkturquoise')
```



Gambar 2. Distribusi Dataset

3.2 Gene Filtering

Berdasarkan hasil pada tahapan sebelumnya, jumlah gen yang kita miliki sangat banyak (lebih dari 50.000) sehingga langkah awal yang diperlukan adalah menyaring gen (*gene filtering*). Proses ini merupakan salah satu proses reduksi data dalam *pre-processing stage*.

Kemudian, terlihat pada Gambar 2 bahwa distribusi datanya menceng kiri. Maka, dapat disimpulkan bahwa terdapat banyak gen yang *underexpressed*. Oleh karena itu, *gene filtering* sangat perlu untuk dilakukan. Dalam penelitian ini, penulis menggunakan package R bernama `genefilter`.

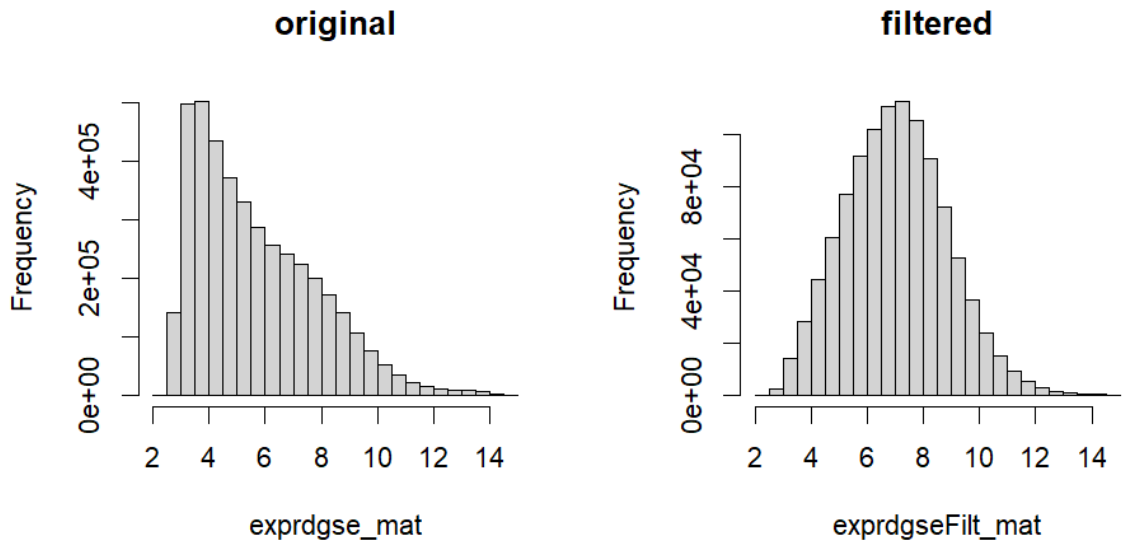
```
> ## Gene Filtering ##
> library(Biobase)
```

```
>
> # Preparing expression set
> pData <- data.frame(type = rdgse$type)
> rownames(pData) <- rdgse$samples
>
> eset <- ExpressionSet(assayData = exprdgse_mat,
+                       phenoData = AnnotatedDataFrame(pData),
+                       annotation = 'hgu133plus2')
```

Sebelum memasuki tahapan *gene filtering*, penulis perlu membentuk sebuah *expression set* agar dapat diproses dengan package `genefilter`. Setelah dibentuk *expression set*, *gene filtering* dapat langsung dilakukan.

```
> # Perform gene filtering
> require(genefilter)
Loading required package: genefilter
> esetFilt <- nsFilter(eset)
>
> # Filtering result
> esetFilt
$eset
ExpressionSet (storageMode: lockedEnvironment)
assayData: 7026 features, 151 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: 84 85 ... 238 (151 total)
  varLabels: type
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133plus2

$filter.log
$filter.log$numDupsRemoved
[1] 7463

$filter.log$numLowVar
[1] 7026

$filter.log$numRemoved.ENTREZID
[1] 5818

$filter.log$feature.exclude
[1] 4

>
> # Extract the Expression of the Filtered Dataset
> exprdgseFilt_mat <- exprs(esetFilt$eset)
```

```
> par(mfrow = c(1,2))
> hist(exprdgse_mat, main = 'original')
> hist(exprdgseFilt_mat, main = 'filtered')
```



Gambar 3. Histogram Data Sebelum dan Sesudah Filtering

Terlihat dari hasil di atas bahwa sebelum dilakukan *filtering*, terdapat 27.337 *features*/gen. Kemudian setelah proses *filtering*, tersisa 7026 gen saja. Gen yang disaring merupakan gen yang memiliki ekspresi rendah.

Agar hasil *filtering* dapat digunakan pada tahapan analisis selanjutnya, diperlukan untuk mengubah tipe data `exprdgseFilt_mat` kembali dari matriks menjadi dataframe.

```
> # Convert to dataframe
> exprdgseFilt <- as.data.frame(exprdgseFilt_mat)
>
> rdgseFilt <- t(exprdgseFilt)
> rdgseFilt <- cbind(typesample, rdgseFilt)
```

3.3 Analisis Ekspresi Gen yang Berbeda

3.3.1 Perbedaan Subtipe Kanker Payudara

Pada bagian pertama ini, penulis tertarik untuk mencari apakah ada gen yang berekspresi berbeda dari seluruh subtipe kelas. Hal yang perlu dilakukan sebelum analisis adalah dengan memuat *library* yang diperlukan.

```
> library(limma)
> library(ggplot2)
> library(stringr)
> library(tibble)
> library(dplyr)
```

Kemudian, karena penulis tertarik untuk membandingkan empat subtipe kelas (basal, HER, Luminal A, dan Luminal B), akan dibentuk dataframe baru dengan mengecualikan gen normal dan garis sel.

```
> # Create dataframe with only cancer genes
> crdgse <- rdgseFilt[!(rdgseFilt$type == "cell_line" | rdgseFilt$type ==
"normal"),]
>
> cexprdgse <- crdgse
```

```
> rownames(cexprdgse) <- cexprdgse$samples
>
> ctypes <- cexprdgse$type
>
> cexprdgse <- subset(cexprdgse, select = -c(type, samples))
> cexprdgse_t <- transpose(cexprdgse)
>
> rownames(cexprdgse_t) <- colnames(cexprdgse)
> colnames(cexprdgse_t) <- rownames(cexprdgse)
```

Selanjutnya, langkah pertama adalah menyiapkan matriks desain, yang menentukan sampel yang terlibat dalam analisis dan termasuk dalam tipe mana. Matriks desain ditentukan menggunakan fungsi `model.matrix` sebagai berikut. Dengan melihat pratinjau desain matriks, kita dapat melihat ada 4 grup yang telah ditentukan, yaitu `basal`, `HER`, `luminal_A`, dan `luminal_B`. Setiap baris dalam matriks desain adalah sampel, dan angka 0 atau 1 menunjukkan kelompok mana sampel tersebut berada.

```
> des_mat <- model.matrix(~ ctypes + 0, data = cexprdgse_t)
> colnames(des_mat) <- str_remove(colnames(des_mat), "ctypes")
> head(des_mat)
  basal HER luminal_A luminal_B
1     1   0         0         0
2     1   0         0         0
3     1   0         0         0
4     1   0         0         0
5     1   0         0         0
6     1   0         0         0
```

Sekarang sudah siap untuk mulai fit model *differential expression* ke data. Untuk mengakomodasi desain yang kali ini memiliki lebih dari 2 grup, perlu dilakukan dalam beberapa langkah.

Penulis akan menggunakan fungsi `lmFit()` dari *package* `limma` untuk menguji setiap gen untuk *differential expression* antara dua kelompok menggunakan model linier. Setelah menyesuaikan data ke model linier, dalam contoh ini penulis menerapkan pemulusan Bayes empiris menggunakan fungsi `eBayes()`.

```
> # Apply linear model to data
> fit <- lmFit(cexprdgse_t, design = des_mat)
>
> # Apply empirical Bayes to smooth standard errors
> fit <- eBayes(fit)
```

Kini setelah model dasar cocok, akan diselidiki perbedaan di antara semua kelompok yang ada. Dalam matriks kontras ini, dibandingkan setiap subtipe dengan semua subtipe lainnya. Penulis membaginya dengan tiga dalam persamaan ini sehingga setiap kelompok dibandingkan dengan rata-rata tiga kelompok lainnya

```
> # Create contrast matrix
> contrast_matrix <- makeContrasts(
+   "basalvsOther" = basal - (HER + luminal_A + luminal_B) / 3,
+   "HERvsOther" = HER - (basal + luminal_A + luminal_B) / 3,
+   "luminal_AvsOther" = luminal_A - (basal + HER + luminal_B) / 3,
+   "luminal_BvsOther" = luminal_B - (basal + HER + luminal_A) / 3,
+   levels = des_mat
+ )
```

Sekarang setelah matriks kontras siap, model dapat di fit kembali dengan `contrast.fit()` dan memuluskannya kembali dengan `eBayes()`.

```
> contrasts_fit <- contrasts.fit(fit, contrast_matrix)
```

```
> contrasts_fit <- eBayes(contrasts_fit)
```

Setelah seluruh proses di atas, mari buat tabel hasil berdasarkan model yang dilengkapi kontras.

```
> # Apply multiple testing correction and obtain stats
> stats_df <- topTable(contrasts_fit, number = nrow(cexprdgse_t)) %>%
+   rownames_to_column("Gene")
> head(stats_df)
        Gene basalvsOther HERvsOther luminal_AvsOther luminal_BvsOther
1 216836_s_at    -2.908985  2.8466385       -1.7911094        1.8534558
2 224447_s_at    -2.006364  2.5224539       -2.0739632        1.5578733
3   200934_at     1.452789  0.1346503       -0.8284887       -0.7589501
4 204508_s_at    -3.123366 -1.5819853        2.5260975        2.1792542
5 219918_s_at     2.230490  0.8524954       -2.9437462       -0.1392393
6   209173_at    -6.110400  1.0391902        2.0262763        3.0449337
    AveExpr        F      P.Value    adj.P.Val
1 11.036157 187.2172 3.063673e-47 2.152537e-43
2 10.678823 171.4732 3.130551e-45 1.099762e-41
3 10.638873 151.9325 1.599776e-42 3.746675e-39
4  7.358736 150.3646 2.709881e-42 4.759905e-39
5  7.981458 145.3622 1.498735e-41 2.106023e-38
6  9.612653 137.5915 2.340503e-40 2.740729e-37
```

Untuk setiap gen, *fold* ekspresi masing-masing kelompok ditampilkan, dibandingkan dengan rata-rata kelompok lain. Secara *default*, hasil diurutkan dari nilai *F* terbesar hingga terkecil, yang berarti gen yang paling berbeda ekspresinya di semua kelompok harus berada di urutan teratas.

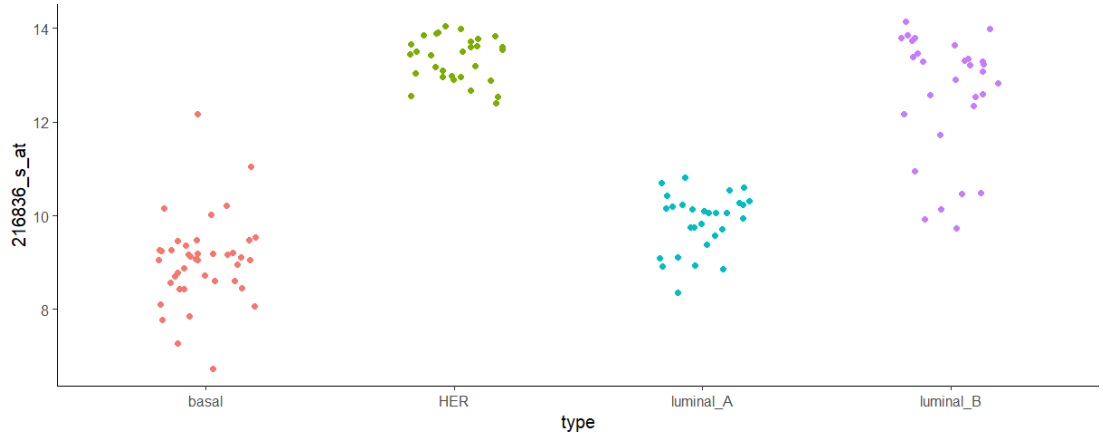
Untuk menguji apakah hasil ini masuk akal, dapat dibuat plot dari salah satu gen teratas. Akan dicoba ekstraksi data untuk `216836_s_at` dan disiapkan *data frame* sendiri untuk pembuatan plot. Berdasarkan hasil di `stats_df`, diperkirakan gen ini lebih rendah pada sampel `luminal_A` dan `basal`, serta lebih tinggi pada sampel `luminal_B` dan `HER`.

Pertama-tama, perlu disiapkan data untuk gen ini dan label subtipenya ke dalam data frame untuk dibuat plot.

```
> # Show top genes
> top_gene_df <- dplyr::select(crdgse, samples, '216836_s_at', type)
> head(top_gene_df)
  samples 216836_s_at  type
1      84    8.723483 basal
2      85    6.728168 basal
3      87    7.859878 basal
4      90    8.617997 basal
5      91    9.468692 basal
6      92    7.274662 basal
```

Sekarang plot data untuk `216836_s_at` menggunakan `top_gene_df`.

```
> ggplot(top_gene_df, aes(x = type, y = `216836_s_at`, color = type)) +
+   geom_jitter(width = 0.2, height = 0) +
+   theme_classic() +
+   theme(legend.position = "none")
```



Gambar 4. Plot Data untuk Gen 216836_s_at

Sesuai dengan prediksi, hasil ini masuk akal. Sampel `basal` dan `luminal_A` memiliki ekspresi `216836_s_at` yang lebih rendah dibandingkan sampel lainnya.

Akan dibuat plot lagi yaitu *volcano plot* dengan menggunakan `ggplot2`. Namun, perlu disiapkan data terlebih dahulu untuk *plotting*, seperti *p-values* dari masing-masing kontras serta perubahan *fold* lognya.

```
> contrast_p_vals_df <- -log10(contrasts_fit$p.value) %>%
+   as.data.frame() %>%
+   tibble::rownames_to_column("Gene") %>%
+   tidyr::pivot_longer(dplyr::contains("vsOther"),
+                       names_to = "contrast",
+                       values_to = "neg_log10_p_val"
+   )
```

Selanjutnya, ekstrak perubahan *fold* log dari `stats_df`.

```
> log_fc_df <- stats_df %>%
+   dplyr::select("Gene", dplyr::contains("vsOther")) %>%
+   tidyr::pivot_longer(dplyr::contains("vsOther"),
+                       names_to = "contrast",
+                       values_to = "logFoldChange"
+   )
```

Dapat dilakukan `inner_join()` dari kedua kumpulan data ini menggunakan kolom `Gene` dan `contrast`.

```
> plot_df <- log_fc_df %>%
+   dplyr::inner_join(contrast_p_vals_df,
+                     by = c("Gene", "contrast"),
+                     suffix = c("_log_fc", "_p_val")
+   )
```

Lihat *preview* dari `plot_df`.

```
> head(plot_df)
# A tibble: 6 × 4
  Gene        contrast          logFoldChange neg_log10_p_val
  <chr>       <chr>                     <dbl>           <dbl>
1 216836_s_at basalvsOther              -2.91            35.3
2 216836_s_at HERvsOther                 2.85            30.6
3 216836_s_at luminal_AvsOther          -1.79            16.1
4 216836_s_at luminal_BvsOther           1.85            17.3
5 224447_s_at basalvsOther              -2.01            26.4
6 224447_s_at HERvsOther                 2.52            30.9
```

Nyatakan apa yang dianggap sebagai level signifikan untuk *fold change* dan -log10 *p-values*. Dengan menyimpannya sebagai variabelnya sendiri, *cutoff* ini hanya perlu diubah di satu tempat jika perlu disesuaikan nanti.

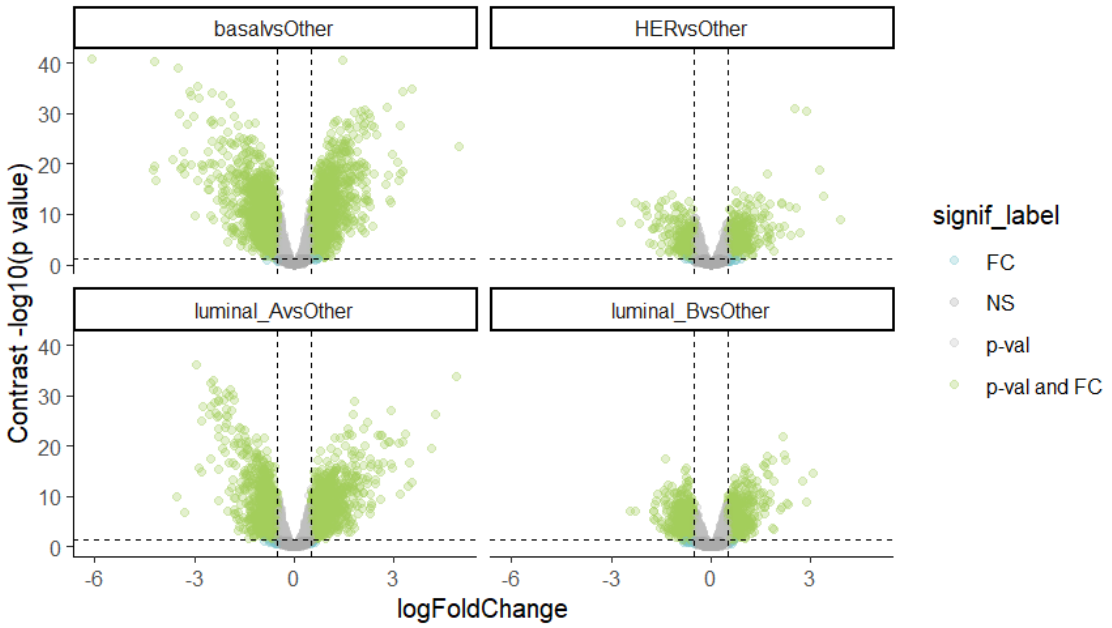
```
> # Convert p value cutoff to negative log 10 scale
> p_val_cutoff <- -log10(0.05)
>
> # Absolute value cutoff for fold changes
> abs_fc_cutoff <- 0.5
```

Sekarang, *cutoff* ini dapat digunakan untuk membuat variabel baru yang menyatakan gen mana yang dianggap signifikan.

```
> plot_df <- plot_df %>%
+   dplyr::mutate(
+     signif_label = dplyr::case_when(
+       abs(logFoldChange) > abs_fc_cutoff &
+         neg_log10_p_val > p_val_cutoff ~ "p-val and FC",
+       abs(logFoldChange) > abs_fc_cutoff ~ "FC",
+       neg_log10_p_val > p_val_cutoff ~ "p-val",
+       TRUE ~ "NS"
+     )
+   )
```

Tampilkan volcano plot

```
> volcanoes_plot <- ggplot(
+   plot_df,
+   aes(
+     x = logFoldChange,
+     y = neg_log10_p_val,
+     color = signif_label
+   )
+ ) +
+   geom_point(alpha = 0.3) +
+   geom_hline(yintercept = p_val_cutoff, linetype = "dashed") +
+   geom_vline(xintercept = c(-abs_fc_cutoff, abs_fc_cutoff),
+              linetype = "dashed") +
+   scale_colour_manual(values = c("cadetblue3", "darkgray",
+                                  "gray", "darkolivegreen3")) +
+   ylab("Contrast -log10(p value)") +
+   facet_wrap(~contrast) +
+   theme_classic()
>
> volcanoes_plot
```



Gambar 5. Volcano Plot Subtipe Kanker dengan Cut-off: p-value < 0.05 and |log2FC|> 0.5

Dari sini, titik-titik hijau mungkin menarik untuk diteliti lebih lanjut.

Selain melalui *volcano plot* dan *scatter plot*, penulis tertarik juga untuk menguji dengan menggunakan metode heatmap. Maka, akan dipilih sampel 50 gen yang berekspresi berbeda.

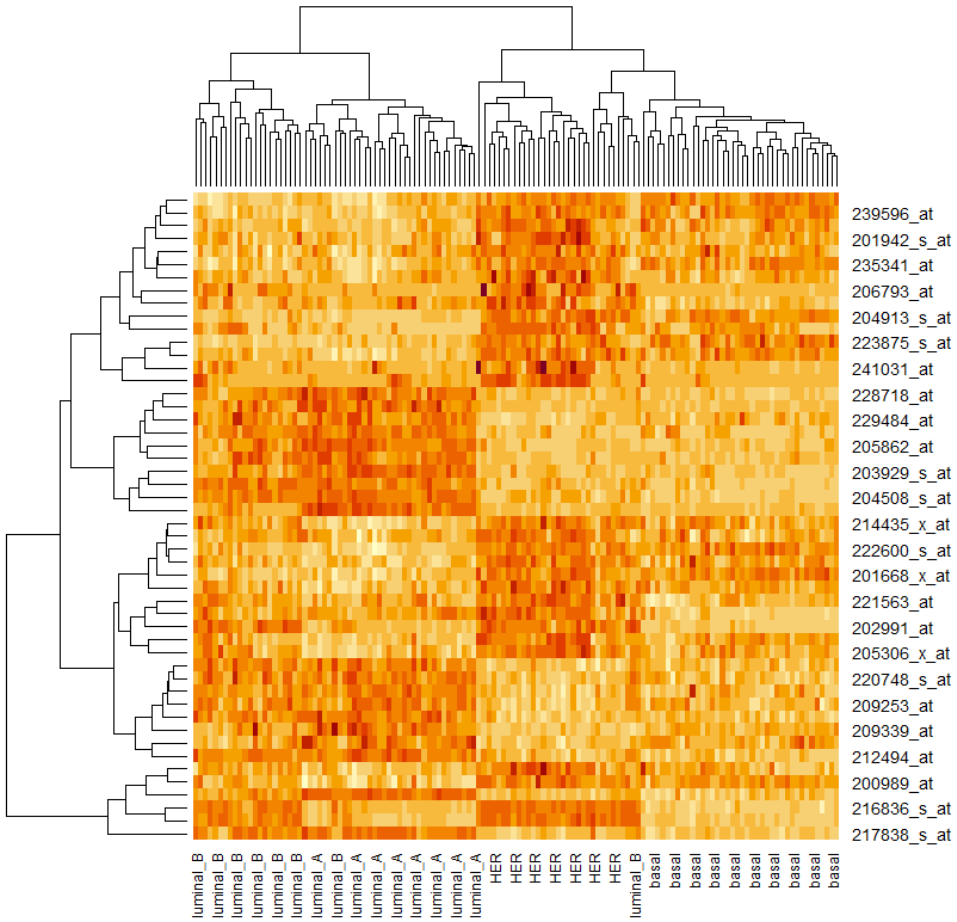
```
> # Select only 50 top genes
> topResult <- topTable(contrasts_fit, coef = 2, number = 50)
>
> # Selected Genes
> rownames(topResult)
 [1] "224447_s_at" "216836_s_at" "204913_s_at" "202991_at"
 [5] "239596_at"   "229484_at"   "226226_at"   "242248_at"
 [9] "217528_at"   "218273_s_at" "203740_at"   "217838_s_at"
[13] "211712_s_at" "209253_at"   "205306_x_at" "200989_at"
[17] "222600_s_at" "228554_at"   "205822_s_at" "227918_s_at"
[21] "241031_at"   "206793_at"   "228718_at"   "221563_at"
[25] "215380_s_at" "225996_at"   "205862_at"   "203929_s_at"
[29] "204497_at"   "217388_s_at" "220581_at"   "242350_s_at"
[33] "204508_s_at" "201942_s_at" "212195_at"   "224674_at"
[37] "201668_x_at" "227286_at"   "205696_s_at" "223875_s_at"
[41] "235341_at"   "212494_at"   "228731_at"   "209339_at"
[45] "220748_s_at" "220326_s_at" "212249_at"   "214435_x_at"
[49] "225412_at"   "1559739_at"
```

Selanjutnya ekstrak informasi dari dataset menjadi expression dengan melakukan modifikasi dataset sebagai berikut.

```
> # Extract selected gene names
> cexprdgse_mat <- as.matrix(cexprdgse_t)
> selected <- rownames(cexprdgse_mat) %in% rownames(topResult)
>
> # Extract the expression of the selected genes
> exprdgsesel <- cexprdgse_mat[selected, ]
> colnames(exprdgsesel) <- ctypes
```

Setelah diperoleh data expression, ekspresi gen dapat langsung di plot.

```
> ## Heatmap of the top genes ##
> heatmap(exprdgsesel)
```



Gambar 6. Heatmap Ekspresi Antar Subtipe Kanker Payudara

Terlihat dari heatmap di atas bahwa sampel dengan subtipe kanker `luminal_A` dan `luminal_B` berekspresi serupa, sedangkan sampel dengan subtipe kanker `HER` dan `basal` berekspresi berbeda. Dan perbedaan antara sampel dengan subtipe kanker `luminal_A` dan `HER` sangat terlihat kontras.

3.3.2 Perbedaan cell sehat dengan cancer

Hal yang ingin penulis lakukan adalah mencari gen dengan ekspresi yang berbeda (*differentially expressed genes*) antara grup normal (orang yang sehat) dan grup kanker (orang yang menderita kanker payudara dari 5 subtipe lainnya).

```
> group <- ifelse(types == "normal", 0, 1)
```

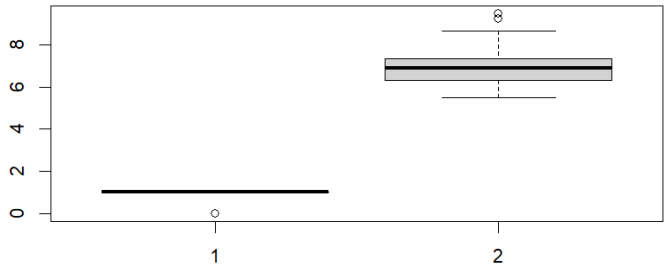
Akan dilihat perbedaan antar grup dengan *t*-test. Perhatikan *t*-test dilakukan untuk baris (ekspresi gen) pertama dari setiap gen.

```
> # Apply t-test
> t.test(group, exprdgseFilt_mat[1,])$p.value
[1] 1.375067e-159
```

Terlihat bahwa *p-value* dari hasil *t-test* < 0.05, maka hipotesis nol ditolak. Jadi, terdapat cukup bukti untuk mengatakan bahwa ada perbedaan ekspresi gen pertama yang signifikan

antara grup normal dan kanker. Selanjutnya, akan dilihat *box plot* ekspresi gen pertama dari kedua grup.

```
> boxplot(group, exprdgseFilt_mat[1,])
```

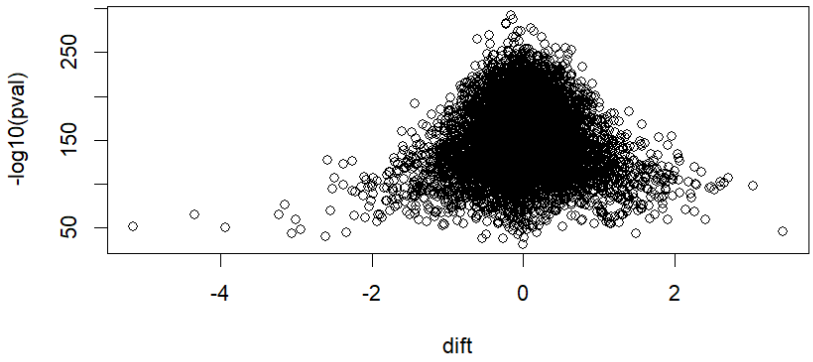


Gambar 7. Box Plot Antara Ekspresi Sampel Sehat dan Kanker

Terlihat dari *box plot* bahwa terdapat perbedaan yang signifikan antara kedua grup.

Selanjutnya, akan dilihat perbedaan seluruh ekspresi gen dari tiap gen yang ada dengan melihat *p-value* dari t-test.

```
> pval <- apply(exprdgseFilt_mat, 1, function(x) t.test(group, x)$p.value)
> dift <- apply(exprdgseFilt_mat, 1,
+               function(x) diff(t.test(x[1:4], x[5:8])$estimate))
> plot(dift, -log10(pval))
```



Gambar 8. Volcano Plot Ekspresi Gen

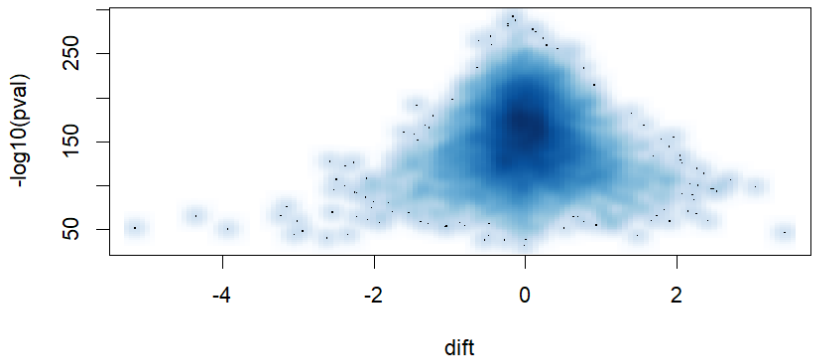
Volcano plot tersebut merupakan *scatter plot* dengan sumbu x menyatakan selisih ekspresi antara kedua grup dan sumbu y adalah -log(pval). Berdasarkan plot tersebut, dapat dilihat bahwa terdapat banyak gen secara statistik signifikan dalam konteks perbedaan ekspresi antara dua grup tersebut.

- Gen yang paling "*upregulated*" (ekspresinya meningkat secara signifikan) terletak di sebelah kanan grafik. Artinya, gen-gen tersebut memiliki nilai ekspresi yang lebih tinggi pada satu grup dibandingkan dengan grup lainnya.

- Gen yang paling "*downregulated*" (ekspresinya menurun secara signifikan) terletak di sebelah kiri grafik. Artinya, gen-gen tersebut memiliki nilai ekspresi yang lebih rendah pada satu grup dibandingkan dengan grup lainnya.
- Gen yang paling "*statistically significant*" (signifikansi statistiknya paling tinggi) terletak di bagian atas grafik, lebih dekat ke sumbu *y* yang lebih tinggi. Gen-gen ini memiliki perbedaan ekspresi yang sangat signifikan antara dua grup, yang dinyatakan dalam nilai statistik yang rendah.

Selanjutnya, akan ditampilkan *smooth scatter plot*.

```
> library(RColorBrewer)
> smoothScatter(dift, -log10(pval))
```



Gambar 9. Smooth Volcano Plot Ekspresi Gen

Bagian tengah plot tersebut menunjukkan bahwa gen memiliki *p-value* tinggi namun secara statistik tidak dapat dikatakan signifikan dalam konteks perbedaan ekspresi antara kedua grup.

Selanjutnya, akan dicari jumlah ekspresi gen yang berbeda secara signifikan dari tiap gen dengan tingkat signifikansi 5% dilihat dari *p-value* t-test.

```
> sum(pval < 0.05)
[1] 7026
```

Terdapat sebanyak 7.026 ekspresi gen dengan *p-value* yang lebih kecil nilainya dari 0.05 sehingga dapat dikatakan sejumlah 7.026 ekspresi gen berbeda secara signifikan. Terlihat bahwa seluruh ekspresi gen yang terfilter dari data sampel adalah signifikan.

Metode selanjutnya adalah metode yang paling banyak digunakan, yaitu LIMMA. LIMMA pada dasarnya merupakan modifikasi model linier yang digunakan untuk analisa ekspresi gen atau protein.

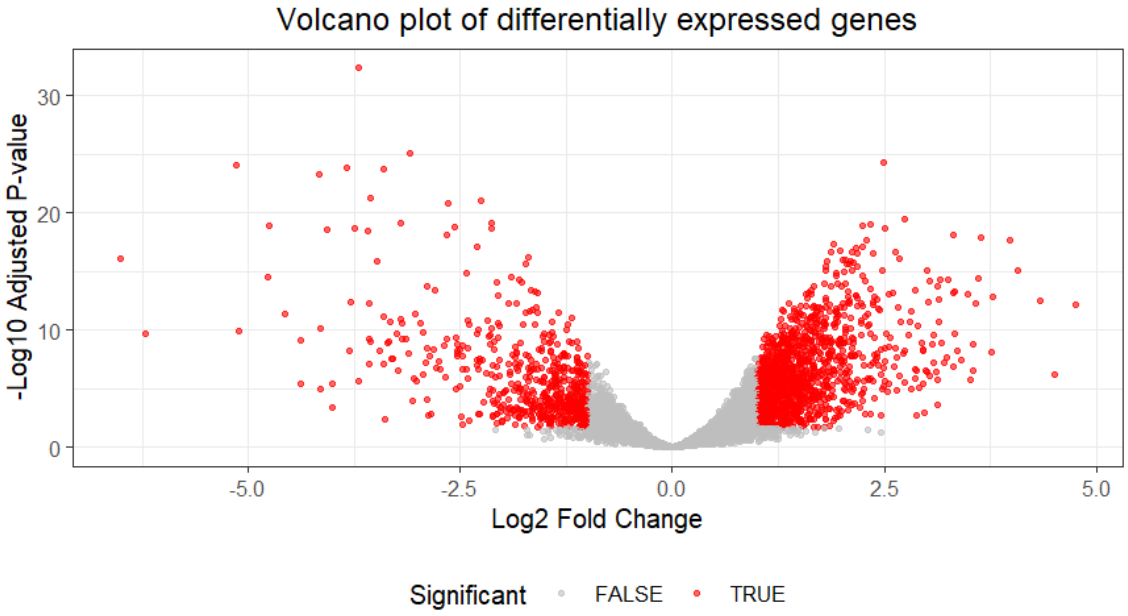
```
> ## LIMMA Analysis ##
> design <- model.matrix(~group)
>
> # Apply linear model to data
> fit <- lmFit(exprdgseFilt_mat, design)
> fit
An object of class "MArrayLM"
```

```
$coefficients
            (Intercept)      group
204639_at      6.223349  0.7029917
212607_at      7.197352  0.1221336
207078_at      6.535787 -0.5216947
209027_s_at    7.188457  1.9954922
227647_at      6.939071 -0.2210058
7021 more rows ...

$rank
[1] 2

$assign
[1] 0 1

$qr
$qr
   (Intercept)       group
1 -12.28820573 -11.7185538
2   0.08137885  -2.5836983
3   0.08137885   0.0165921
4   0.08137885   0.0165921
5   0.08137885   0.0165921
146 more rows ...

$qraux
[1] 1.081379 1.016592

$pivot
[1] 1 2

$tol
[1] 1e-07

$rank
[1] 2


$df.residual
[1] 149 149 149 149 149
7021 more elements ...

$sigma
  204639_at   212607_at   207078_at 209027_s_at   227647_at
  0.6727945   1.1203804   0.6139466   0.6061347   0.9207112
7021 more elements ...

$cov.coefficients
            (Intercept)      group
(Intercept)   0.1428571 -0.1428571
group        -0.1428571  0.1498016

$stdev.unscaled
            (Intercept)     group
204639_at     0.3779645 0.3870421
212607_at     0.3779645 0.3870421
207078_at     0.3779645 0.3870421
209027_s_at   0.3779645 0.3870421
227647_at     0.3779645 0.3870421
7021 more rows ...

$pivot
[1] 1 2

$Amean
  204639_at   212607_at   207078_at 209027_s_at   227647_at
   6.893752    7.313823    6.038277    9.091443    6.728311
7021 more elements ...
```

```
$method
[1] "ls"

$design
  (Intercept) group
1           1     1
2           1     1
3           1     1
4           1     1
5           1     1
146 more rows ...
```

Program telah menampilkan informasi mengenai statistik (*F*) dan *p-value* dari masing-masing gen yang dianalisis. Semakin tinggi nilai statistiknya, maka gen tersebut semakin berbeda di antara kedua grup. Berikutnya akan dibuat *volcano plot* dengan sumbu *x* menyatakan perbedaan antara level-level dari ekspresi tiap gen dan sumbu *y* adalah -log(pval) (signifikansi level dari tiap gen).

```
> fitted.ebayes <- eBayes(fit)
> toptab <- topTable(fitted.ebayes, coef=2, n = Inf)
>
> toptab <- toptab %>%
+   mutate(Significant = ifelse(
+     abs(logFC) > 1 & adj.P.Val < 0.05,
+     TRUE, FALSE))
>
> ggplot(toptab, aes(x = logFC,
+                    y = -log10(P.Value),
+                    colour = Significant)) +
+   geom_point(size = 1, alpha = 0.6) +
+   scale_color_manual(values = c("grey", "red")) +
+   theme_bw() +
+   theme(legend.position = "bottom") +
+   labs(x = "Log2 Fold Change", y = "-Log10 Adjusted P-value") +
+   ggtitle("Volcano plot of differentially expressed genes") +
+   theme(plot.title = element_text(hjust = 0.5))
```



Gambar 10. Volcano Plot Gen yang Berekspresi Berbeda

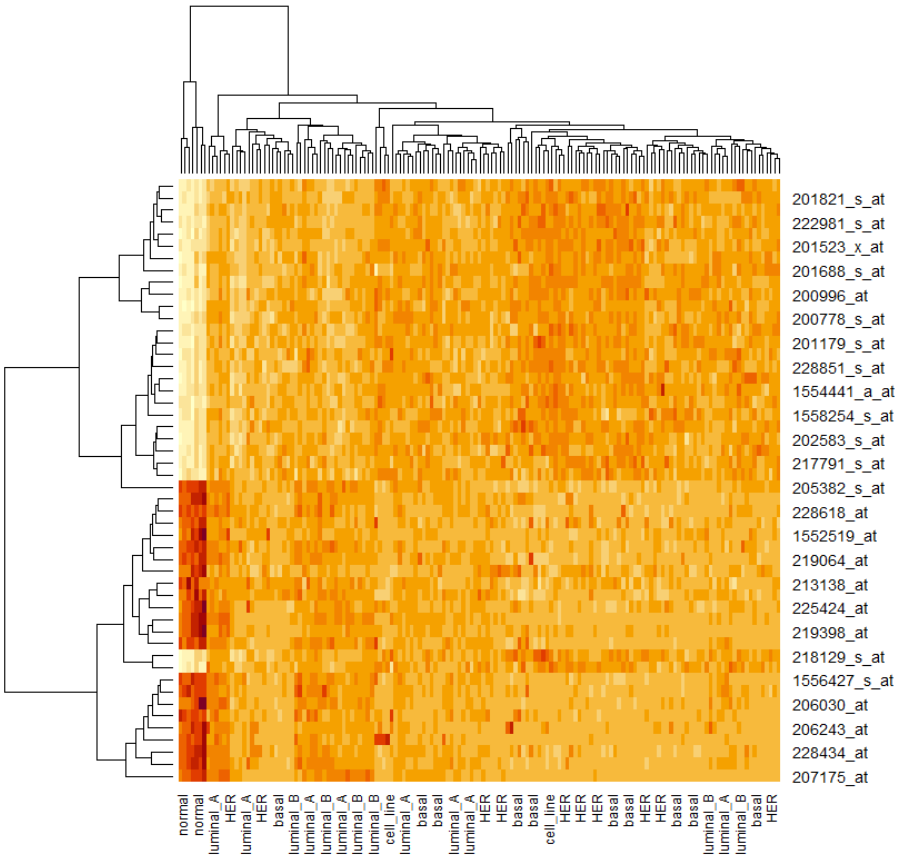
Terlihat dari plot bahwa ekspresi gen di kiri dan di kanan yang berwarna merah merupakan gen yang signifikan.

Selanjutnya, misal akan dicari gen-gen yang paling berbeda di antara kedua grup tersebut (dilihat dari nilai statistiknya). Berikut adalah 50 gen dengan ekspresi paling berbeda.

```
> # Show only top 50 genes
> topResult <- topTable(fitted.ebayes, coef = 2, number = 50)
>
> # Selected Genes
> rownames(topResult)
 [1] "219398_at"    "216331_at"    "201096_s_at"  "205913_at"
 [5] "219689_at"    "1552519_at"   "228434_at"    "225424_at"
 [9] "204134_at"    "207016_s_at"  "1555797_a_at" "219064_at"
[13] "206030_at"    "202583_s_at"  "205382_s_at"  "201821_s_at"
[17] "1556427_s_at" "201461_s_at"  "206243_at"    "205824_at"
[21] "222717_at"    "227419_x_at"  "226303_at"    "1564494_s_at"
[25] "1557910_at"   "201688_s_at"  "217140_s_at"  "200996_at"
[29] "228618_at"    "1558254_s_at" "201179_s_at"  "217791_s_at"
[33] "202543_s_at"  "223289_s_at"  "201411_s_at"  "202166_s_at"
[37] "218302_at"    "213138_at"    "207175_at"    "228851_s_at"
[41] "202955_s_at"  "222981_s_at"  "215695_s_at"  "1555278_a_at"
[45] "1554441_a_at" "218129_s_at"  "201523_x_at"  "200744_s_at"
[49] "1564525_at"   "200778_s_at"
```

Kemudian, akan ditampilkan pola ekspresi dari ke-50 gen tersebut menggunakan *heatmap* dan *box plot*.

```
> # Extract selected gene names
> selected <- rownames(exprdgseFilt_mat) %in% rownames(topResult)
>
> # Extract the expression of the selected genes
> exprdgsesel <- exprdgseFilt_mat[selected, ]
> colnames(exprdgsesel) <- types
>
> ## Heatmap of the top genes ##
> heatmap(exprdgsesel)
```

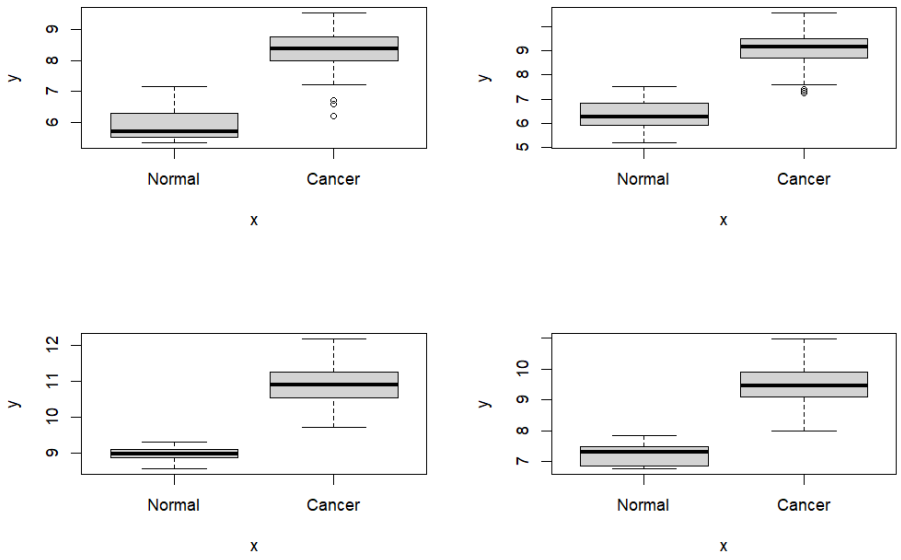


Gambar 11. Heatmap Ekspresi Gen Kanker dengan Normal

Heatmap di atas menggambarkan ekspresi dari tiap gen (baris) dan tiap sampel (kolom). Warna merah menandakan gen tersebut memiliki ekspresi lebih tinggi (*over expression*). Terlihat pengelompokan yang jelas antara sampel grup normal (yang muncul di *heatmap* sejumlah 7 sampel di kiri) dan sampel grup cancer (sisanya di kanan).

Selanjutnya, akan dilihat box plot dari 4 gen dengan nilai statistik terbesar.

```
> ### Boxplot for the top 4 genes ###
> group2 <- as.factor(group)
> levels(group2) <- c('Normal', 'Cancer')
>
> par(mfrow = c(2,2))
> for (i in 1:4) plot(group2, exprdgsesel[i,],
+                     main = rownames(exprdgsesel[i]))
```



Gambar 12. Box Plot Antara Sampel Normal dan Kanker

Dari box plot tersebut, terlihat bahwa gen dari masing-masing kelompok (Normal dan Kanker) memiliki ekspresi yang cukup berbeda.

3.4 Hasil Analisis Ontologi Gen

Analisis ontologi gen dilakukan untuk memperoleh informasi biologis dari data yang dianalisis. Analisis ontologi gen dilakukan pada 50 data gen yang paling berbeda. Analisis dilakukan dengan bantuan library `GO.db`.

```
> GeneSelected <- select(hgu133plus2.db, rownames(topResult),
+                        c("SYMBOL", "ENTREZID", "GENENAME"))
'select()' returned 1:1 mapping between keys and columns
> GeneSelected
        PROBEID    SYMBOL ENTREZID
1     219398_at     CIDEC    63924
2     216331_at     ITGA7     3679
3   201096_s_at      ARF4      378
4     205913_at     PLIN1     5346
5     219689_at    SEMA3G    56920
6    1552519_at    ACVR1C   130399
7     228434_at     BTNL9   153579
8     225424_at      GPAM    57678
9     204134_at     PDE2A     5138
10  207016_s_at   ALDH1A2     8854
11 1555797_a_at     ARPC5    10092
12    219064_at     ITIH5    80760
13    206030_at      ASPA      443
14  202583_s_at    RANBP9    10048
15  205382_s_at       CFD     1675
16  201821_s_at   TIMM17A    10440
17 1556427_s_at   LRRN4CL   221091
18  201461_s_at MAPKAPK2     9261
19    206243_at     TIMP4     7079
20    205824_at     HSPB2     3316
21    222717_at    CAVIN2     8436
22  227419_x_at     PLAC9   219348
23    226303_at      PGM5     5239
```

24	1564494_s_at	P4HB	5034	
25	1557910_at	HSP90AB1	3326	
26	201688_s_at	TPD52	7163	
27	217140_s_at	VDAC1	7416	
28	200996_at	ACTR3	10096	
29	228618_at	PEAR1	375033	
30	1558254_s_at	SRPK2	6733	
31	201179_s_at	GNAI3	2773	
32	217791_s_at	ALDH18A1	5832	
33	202543_s_at	GMFB	2764	
34	223289_s_at	USP38	84640	
35	201411_s_at	PLEKHB2	55041	
36	202166_s_at	PPP1R2	5504	
37	218302_at	PSENEN	55851	
38	213138_at	ARID5A	10865	
39	207175_at	ADIPOQ	9370	
40	228851_s_at	ENSA	2029	
41	202955_s_at	ARFGEF1	10565	
42	222981_s_at	RAB10	10890	
43	215695_s_at	GYG2	8908	
44	1555278_a_at	CKAP5	9793	
45	1554441_a_at	WAPL	23063	
46	218129_s_at	NFYB	4801	
47	201523_x_at	UBE2N	7334	
48	200744_s_at	GNB1	2782	
49	1564525_at	GSN	2934	
50	200778_s_at	SEPTIN2	4735	

	GENENAME
1	cell death inducing DFFA like effector c
2	integrin subunit alpha 7
3	ADP ribosylation factor 4
4	perilipin 1
5	semaphorin 3G
6	activin A receptor type 1C
7	butyrophilin like 9
8	glycerol-3-phosphate acyltransferase, mitochondrial
9	phosphodiesterase 2A
10	aldehyde dehydrogenase 1 family member A2
11	actin related protein 2/3 complex subunit 5
12	inter-alpha-trypsin inhibitor heavy chain 5
13	aspartoacylase
14	RAN binding protein 9
15	complement factor D
16	translocase of inner mitochondrial membrane 17A
17	LRRN4 C-terminal like
18	MAPK activated protein kinase 2
19	TIMP metallopeptidase inhibitor 4
20	heat shock protein family B (small) member 2
21	caveolae associated protein 2
22	placenta associated 9
23	phosphoglucomutase 5
24	prolyl 4-hydroxylase subunit beta
25	heat shock protein 90 alpha family class B member 1
26	tumor protein D52
27	voltage dependent anion channel 1
28	actin related protein 3
29	platelet endothelial aggregation receptor 1
30	SRSF protein kinase 2
31	G protein subunit alpha i3
32	aldehyde dehydrogenase 18 family member A1
33	glia maturation factor beta
34	ubiquitin specific peptidase 38
35	pleckstrin homology domain containing B2
36	protein phosphatase 1 regulatory inhibitor subunit 2
37	presenilin enhancer, gamma-secretase subunit
38	AT-rich interaction domain 5A
39	adiponectin, C1Q and collagen domain containing
40	endosulfine alpha
41	ADP ribosylation factor guanine nucleotide exchange factor 1

```
42                                RAB10, member RAS oncogene family
43                                                   glycogenin 2
44                                cytoskeleton associated protein 5
45                                      WAPL cohesin release factor
46                     nuclear transcription factor Y subunit beta
47                                ubiquitin conjugating enzyme E2 N
48                                        G protein subunit beta 1
49                                                         gelsolin
50                                                         septin 2
```

Setelah mengetahui nama gen, selanjutnya akan dicari model dan fungsi dari gen tersebut dengan menggunakan Gene Ontology.

```
> ids <- rownames(topResult)
> GeneSelected <- select(hgu133plus2.db, ids,
+                        c("SYMBOL", "ENTREZID", "GENENAME", "GO"))
'select()' returned 1:many mapping between keys and columns
>
> ### Gene ontology for the top genes ###
> library(GO.db)
> GOselected <- select(GO.db, GeneSelected$GO, c("TERM", "GOID"))
'select()' returned many:1 mapping between keys and columns
> head(GOselected)
       GOID                  TERM
1 GO:0003674    molecular_function
2 GO:0005515       protein binding
3 GO:0005634               nucleus
4 GO:0005783 endoplasmic reticulum
5 GO:0005811         lipid droplet
6 GO:0005829               cytosol
```

Informasi dari beberapa proses sebelumnya dapat digabung dan di-*export* dalam bentuk *comma separated values* (.csv) seperti berikut.

```
> ### Combine the result ###
> finalres <- cbind(GeneSelected, GOselected)
>
> ### Convert to csv ###
> write.csv2(finalres, file = "GEOres_UTS Sains Data Genom.csv")
```

4. Kesimpulan

Dalam analisis gen dari sampel orang normal dan orang yang menderita kanker payudara dari 5 subtipe (HER, Basal, Luminal A, dan Luminal B, cell line) menggunakan metode LIMMA, hasil yang diperoleh menunjukkan perbedaan signifikan dalam ekspresi gen antara sampel individu yang normal dengan individu yang memiliki kanker. Hal ini dapat disimpulkan berdasarkan visualisasi data melalui *heatmap* dan *box plot* dari 50 gen dengan ekspresi yang paling signifikan. Data ini memberikan wawasan yang kuat tentang perbedaan ekspresi gen antara kedua kelompok sampel, menyoroti potensi keterlibatan gen-gen tertentu dalam kanker payudara.

Kemudian, dalam analisis dengan menggunakan LIMMA untuk *multiple comparisons* terhadap 4 subtipe kanker (HER, Basal, Luminal A, dan Luminal B) pada gen paling signifikan `216836_s_at` ditunjukkan bahwa sampel dengan subtipe kanker Basal dan Luminal A berekspresi rendah dan serupa dibandingkan dengan sampel dengan subtipe kanker HER dan Luminal B yang juga berekspresi serupa. Dan ekspresi antara sampel dengan subtipe kanker Luminal A dan HER sangat berbeda secara kontras.

Referensi

Alkabban, F. M., & Ferguson, T. (2023). *Breast Cancer.* StatPearls Publishing.

Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics, 20*(16), 2493-2503. https://doi.org/10.1093/bioinformatics/bth283

Bhalla, D. (n.d.). *How to Efficiently Read Large CSV Files in R*. ListenData: https://www.listendata.com/2016/01/reading-large-csv-file-with-r.html

Brazma, A., & Vilo, J. (2000). Gene expression data analysis. *FEBS Letters, 480*(1), 17-24. https://doi.org/10.1016/S0014-5793(00)01772-5

Gene Ontology Consortium. (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research, 32*(suppl_1), D258–D261. https://doi.org/10.1093/nar/gkh036

Hin, N. (2021, June 28). *Differential Gene Expression Analysis*. BULK RNA-SEQ: https://biocellgen-public.svi.edu.au/sahmri-bulk-rnaseq/de.html

Liang, P., & Pardee, A. B. (2003). Timeline: Analysing differential gene expression in cancer. *Nature Reviews Cancer, 3*(11), 869-876. https://doi.org/10.1038/nrc1214

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., Quackenbush, J. F., Stijleman, I. J., Palazzo, J., Marron, J., Nobel, A. B., & Mard. (2009). Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes. *Journal of Clinical Oncology, 27*(8), 1160-1167. https://doi.org/10.1200/jco.2008.18.1370

Prat, A., Ellis, M. J., & Perou, C. M. (2011). Practical implications of gene-expression-based assays for breast oncologists. *Nat Rev Clin Oncol, 9*(1), 48-57. https://doi.org/10.1038/nrclinonc.2011.178

Lampiran

Script R

```
## Initialization ##
# Set seed
set.seed(2106725034)

# Set working directory
setwd("D:/OneDrive - UNIVERSITAS INDONESIA/Kuliah/Semester 5/[SCST603107]
Sains Data Genom/UTS")

## Load the dataset ##
library(data.table)
dtgse <- fread("Breast_GSE45827.csv")

# Change to dataframe
dfgse <- as.data.frame(dtgse)
as.data.frame(table(dfgse$type))

# Randomly select 50% of the genes from the dataframe
typesample <- subset(dfgse, select = c(samples, type))
rdgse <- subset(dfgse, select = -c(type, samples))

rdgse <- rdgse[, sample(ncol(rdgse), ncol(rdgse) * 0.5)]
rdgse <- rdgse[, order(names(rdgse))]

rdgse <- cbind(typesample, rdgse)

## Extract Expression (modify dataframe) ##
exprdgse <- rdgse
rownames(exprdgse) <- exprdgse$samples
types <- exprdgse$type

exprdgse <- subset(exprdgse, select = -c(type, samples))
exprdgse_t <- transpose(exprdgse)

rownames(exprdgse_t) <- colnames(exprdgse)
colnames(exprdgse_t) <- rownames(exprdgse)

exprdgse_mat <- as.matrix(exprdgse_t)

#### Nomor 1 ####
## Exploration ##
as.data.frame(table(rdgse$type))

library(RColorBrewer)
colour <- brewer.pal(5, "Set2")
barplot(table(rdgse$type),
        main = "Distribution of label types in breast cancer data",
        xlab = "Sample type",
        ylab = "Number of occurences",
        ylim = c(0,50),
        col = colour)

hist(exprdgse_mat, col = 'darkturquoise')

## Gene Filtering ##
library(Biobase)
```

```
# Preparing expression set
pData <- data.frame(type = rdgse$type)
rownames(pData) <- rdgse$samples

eset <- ExpressionSet(assayData = exprdgse_mat,
                      phenoData = AnnotatedDataFrame(pData),
                      annotation = 'hgu133plus2')

# Perform gene filtering
require(genefilter)
esetFilt <- nsFilter(eset)

# Filtering result
esetFilt

# Extract the Expression of the Filtered Dataset
exprdgseFilt_mat <- exprs(esetFilt$eset)

# Plot the original and filtered data
par(mfrow = c(1,2))
hist(exprdgse_mat, main = 'original')
hist(exprdgseFilt_mat, main = 'filtered')

# Convert to dataframe
exprdgseFilt <- as.data.frame(exprdgseFilt_mat)

rdgseFilt <- t(exprdgseFilt)
rdgseFilt <- cbind(typesample, rdgseFilt)

#### Nomor 2 ####
## LIMMA Multiple Analysis ##
library(limma)
library(ggplot2)
library(stringr)
library(tibble)
library(dplyr)

# Create dataframe with only cancer genes
crdgse <- rdgseFilt[!(rdgseFilt$type == "cell_line" | rdgseFilt$type ==
"normal"),]

cexprdgse <- crdgse
rownames(cexprdgse) <- cexprdgse$samples

ctypes <- cexprdgse$type

cexprdgse <- subset(cexprdgse, select = -c(type, samples))
cexprdgse_t <- transpose(cexprdgse)

rownames(cexprdgse_t) <- colnames(cexprdgse)
colnames(cexprdgse_t) <- rownames(cexprdgse)

des_mat <- model.matrix(~ ctypes + 0, data = cexprdgse_t)
colnames(des_mat) <- str_remove(colnames(des_mat), "ctypes")
head(des_mat)

# Apply linear model to data
fit <- lmFit(cexprdgse_t, design = des_mat)

# Apply empirical Bayes to smooth standard errors
```

```
fit <- eBayes(fit)

# Create contrast matrix
contrast_matrix <- makeContrasts(
  "basalvsOther" = basal - (HER + luminal_A + luminal_B) / 3,
  "HERvsOther" = HER - (basal + luminal_A + luminal_B) / 3,
  "luminal_AvsOther" = luminal_A - (basal + HER + luminal_B) / 3,
  "luminal_BvsOther" = luminal_B - (basal + HER + luminal_A) / 3,
  levels = des_mat
)

# Fit the model according to the contrasts matrix
contrasts_fit <- contrasts.fit(fit, contrast_matrix)

# Re-smooth the Bayes
contrasts_fit <- eBayes(contrasts_fit)

# Apply multiple testing correction and obtain stats
stats_df <- topTable(contrasts_fit, number = nrow(cexprdgse_t)) %>%
  rownames_to_column("Gene")
head(stats_df)

# Show top genes
top_gene_df <- dplyr::select(crdgse, samples, '216836_s_at', type)
head(top_gene_df)

ggplot(top_gene_df, aes(x = type, y = `216836_s_at`, color = type)) +
  geom_jitter(width = 0.2, height = 0) +
  theme_classic() +
  theme(legend.position = "none")

# Let's extract the contrast p values for each and transform them with -
log10()
contrast_p_vals_df <- -log10(contrasts_fit$p.value) %>%
  # Make this into a data frame
  as.data.frame() %>%
  # Store genes as their own column
  tibble::rownames_to_column("Gene") %>%
  # Make this into long format
  tidyr::pivot_longer(dplyr::contains("vsOther"),
                      names_to = "contrast",
                      values_to = "neg_log10_p_val"
  )

# Let's extract the fold changes from `stats_df`
log_fc_df <- stats_df %>%
  # We only want to keep the `Gene` column as well
  dplyr::select("Gene", dplyr::contains("vsOther")) %>%
  # Make this a longer format
  tidyr::pivot_longer(dplyr::contains("vsOther"),
                      names_to = "contrast",
                      values_to = "logFoldChange"
  )

plot_df <- log_fc_df %>%
  dplyr::inner_join(contrast_p_vals_df,
                    by = c("Gene", "contrast"),
                    # This argument will add the given suffixes to the
column names
                    # from the respective data frames, helping us keep
```

```
track of which columns
                    # hold which types of values
                    suffix = c("_log_fc", "_p_val")
  )

# Print out what this looks like
head(plot_df)

# Convert p value cutoff to negative log 10 scale
p_val_cutoff <- -log10(0.05)

# Absolute value cutoff for fold changes
abs_fc_cutoff <- 0.5

plot_df <- plot_df %>%
  dplyr::mutate(
    signif_label = dplyr::case_when(
      abs(logFoldChange) > abs_fc_cutoff & neg_log10_p_val > p_val_cutoff
~ "p-val and FC",
      abs(logFoldChange) > abs_fc_cutoff ~ "FC",
      neg_log10_p_val > p_val_cutoff ~ "p-val",
      TRUE ~ "NS"
    )
  )

volcanoes_plot <- ggplot(
  plot_df,
  aes(
    x = logFoldChange, # Fold change as x value
    y = neg_log10_p_val, # -log10(p value) for the contrasts
    color = signif_label # Color code by significance cutoffs variable we
made
  )
) +
  # Make a scatter plot with points that are 30% opaque using `alpha`
  geom_point(alpha = 0.3) +
  # Draw our `p_val_cutoff` for line here
  geom_hline(yintercept = p_val_cutoff, linetype = "dashed") +
  # Using our `abs_fc_cutoff` for our lines here
  geom_vline(xintercept = c(-abs_fc_cutoff, abs_fc_cutoff),
             linetype = "dashed") +
  # The default colors aren't great, we'll specify our own here
  scale_colour_manual(values = c("cadetblue3", "darkgray", "gray",
"darkolivegreen3")) +
  # Let's be more specific about what this p value is in our y axis label
  ylab("Contrast -log10(p value)") +
  # This makes separate plots for each contrast!
  facet_wrap(~contrast) +
  # Just for making it prettier!
  theme_classic()

# Print out the plot!
volcanoes_plot

# Select only 50 top genes
topResult <- topTable(contrasts_fit, coef = 2, number = 50)

# Selected Genes
rownames(topResult)
```

```
# Extract selected gene names
cexprdgse_mat <- as.matrix(cexprdgse_t)
selected <- rownames(cexprdgse_mat) %in% rownames(topResult)

# Extract the expression of the selected genes
exprdgsesel <- cexprdgse_mat[selected, ]
colnames(exprdgsesel) <- ctypes

## Heatmap of the top genes ##
heatmap(exprdgsesel)

#### Nomor 3 ####
# Assign Group Code
group <- ifelse(types == "normal", 0, 1)

# Apply t-test
t.test(group, exprdgseFilt_mat[1,])$p.value

# Create box plot
boxplot(group, exprdgseFilt_mat[1,])

# Volcano plot
pval <- apply(exprdgseFilt_mat, 1, function(x) t.test(group, x)$p.value)
dift <- apply(exprdgseFilt_mat, 1,
              function(x) diff(t.test(x[1:4], x[5:8])$estimate))
plot(dift, -log10(pval))

library(RColorBrewer)
smoothScatter(dift, -log10(pval))

# Calculate significants
sum(pval < 0.05)

pvalBonf <- p.adjust(pval, method = "bonferroni" )
pvalHolm <- p.adjust(pval, method = "holm" )

sum(pvalBonf < 0.05)
sum(pvalHolm < 0.05)

## LIMMA Analysis ##
design <- model.matrix(~group)

# Apply linear model to data
fit <- lmFit(exprdgseFilt_mat, design)
fit

# Apply empirical Bayes to smooth standard errors
fitted.ebayes <- eBayes(fit)
toptab <- topTable(fitted.ebayes, coef=2, n = Inf)

# Visualize the volcano plot
toptab <- toptab %>%
  mutate(Significant = ifelse(
    abs(logFC) > 1 & adj.P.Val < 0.05,
    TRUE, FALSE))

ggplot(toptab, aes(x = logFC,
                   y = -log10(P.Value),
                   colour = Significant)) +
  geom_point(size = 1, alpha = 0.6) +
```

```
  scale_color_manual(values = c("grey", "red")) +
  theme_bw() +
  theme(legend.position = "bottom") +
  labs(x = "Log2 Fold Change", y = "-Log10 Adjusted P-value") +
  ggtitle("Volcano plot of differentially expressed genes") +
  theme(plot.title = element_text(hjust = 0.5))

# Show only top 50 genes
topResult <- topTable(fitted.ebayes, coef = 2, number = 50)

# Selected Genes
rownames(topResult)

# Extract selected gene names
selected <- rownames(exprdgseFilt_mat) %in% rownames(topResult)

# Extract the expression of the selected genes
exprdgsesel <- exprdgseFilt_mat[selected, ]
colnames(exprdgsesel) <- types

## Heatmap of the top genes ##
heatmap(exprdgsesel)

## Boxplot for the top 4 genes ##
group2 <- as.factor(group)
levels(group2) <- c('Normal', 'Cancer')

par(mfrow = c(2,2))
for (i in 1:4) plot(group2, exprdgsesel[i,],
                    main = rownames(exprdgsesel[i]))

## See gene name and description ##
library(annotate)
library(hgu133plus2.db)

GeneSelected <- select(hgu133plus2.db, rownames(topResult),
                       c("SYMBOL", "ENTREZID", "GENENAME"))
GeneSelected

ids <- rownames(topResult)
GeneSelected <- select(hgu133plus2.db, ids,
                       c("SYMBOL", "ENTREZID", "GENENAME", "GO"))

## Gene ontology for the top genes ##
library(GO.db)
GOselected <- select(GO.db, GeneSelected$GO, c("TERM", "GOID"))
head(GOselected)

# Combine the result
finalres <- cbind(GeneSelected, GOselected)

# Convert to csv
write.csv2(finalres, file = "GEOres_UTS Sains Data Genom.csv")
```