21301487 applied project

by Muniraghuveer Cirimavilla

Submission date: 22-Oct-2023 03:09AM (UTC+1100)

Submission ID: 2202728038

File name: 395862_Muniraghuveer_Cirimavilla_21301487_applied_project_6744268_1781468607.pdf (915.97K)

Word count: 2629

Character count: 15610

"This is my own work. I have not copied any of it from anyone else."

Name: Muniraghuveer cirimavilla

Student ID: 21301487

From Genes to Prognosis: Uncovering Breast Cancer Diversity through Microarray Analysis

Background:

Breast cancer is one of the most common cancers affecting women around the world, including Australia. Imagine breast cancer as not just one single disease but rather a group of closely related diseases. Each of these "sub-groups" might behave differently and might need different treatments. The challenge is detecting these sub-groups early on.

Now, every person's cancer has a unique pattern of active and inactive genes. By studying these patterns, we can group patients into these potential sub-groups of breast cancer. If we can identify which sub-group a patient belongs to at the beginning of their treatment, doctors might be able to choose the best treatment for them right from the start. This not only could make the treatment more effective but also save on unnecessary medical expenditure.

In this study, we're diving deep into these unique genetic patterns to figure out if there are indeed different "sub-groups" of breast cancer and what they might mean for a patient's treatment and recovery.

Methods:

Data Loading and Preprocessing:

The data is loaded into the environment using readRDS. Before diving into any analysis, it's crucial to preprocess the data. This includes normalization and scaling, which ensure that the data is on a comparable scale and that variations due to technical factors are minimized.

Hierarchical Clustering:

With the use of hclust and the complete linkage method, samples are grouped based on their similarity in gene expression profiles. The complete linkage method considers the maximum distance between samples in two clusters, making it a powerful technique for spotting potential subtypes or patterns in data.

Silhouette Analysis:

Employing the silhouette function allows for the evaluation of the validity of clusters. This method compares the tightness of clusters to their separation, providing insights into the optimal number of clusters, which can hint at potential breast cancer subtypes.

Heatmap Visualization:

Using heatmap.2 offers a comprehensive visualization of gene expression data. By showcasing sample clusters and patterns of gene expression simultaneously, it becomes easier to identify potential subtypes and understand gene activity.

Principal Component Analysis (PCA):

By implementing princomp, the high-dimensional data is transformed into principal components. PCA captures the most variance in the data, offering a clear visualization of the overall data structure and potential clusters, making it easier to identify patterns or groups in the data.

Differential Gene Expression Analysis:

Methods like lmFit, eBayes, and qvalue are used to spot genes with varying expression levels across different conditions or subtypes. Identifying these genes can highlight those that are crucial to each subtype or have been impacted by it.

Gene Ontology (GO) Analysis:

With the help of functions like select, goana, and topGO, it's possible to gain insight into the biological processes, molecular functions, and cellular components associated with a set of genes. This understanding provides a biological narrative for the differentially expressed genes and their potential roles in breast cancer.

Survival Analysis:

Using tools such as coxph, survfit, and Kaplan-Meier plots, the survival time of patients is linked to one or more predictors. This analysis is pivotal in determining if specific gene expression profiles or subtypes have a bearing on patient outcomes, either improving or worsening them.

Visualization:

The functions autoplot, ggsurvplot, and ggplotly are employed to visualize various aspects of the data. Visualization techniques are paramount in bioinformatics, helping researchers and clinicians make sense of complex datasets, especially when trying to understand patient prognosis in relation to gene profiles or potential cancer subtypes.

Results and summary:

Cluster Analysis:

The dendrogram provides a visualization of the relationships and dissimilarities between samples based on their gene expression profiles. At the top, the tree starts with a single cluster containing all samples, which is gradually split into smaller clusters as we move down the tree. The height at which two clusters merge represents the distance (dissimilarity) between them, with taller heights indicating larger distances. The red line suggests a threshold that could be used to determine an optimal number of clusters for the data. By cutting the tree at this height, the number of clusters formed below the line can be taken as a potential clustering solution for the dataset.

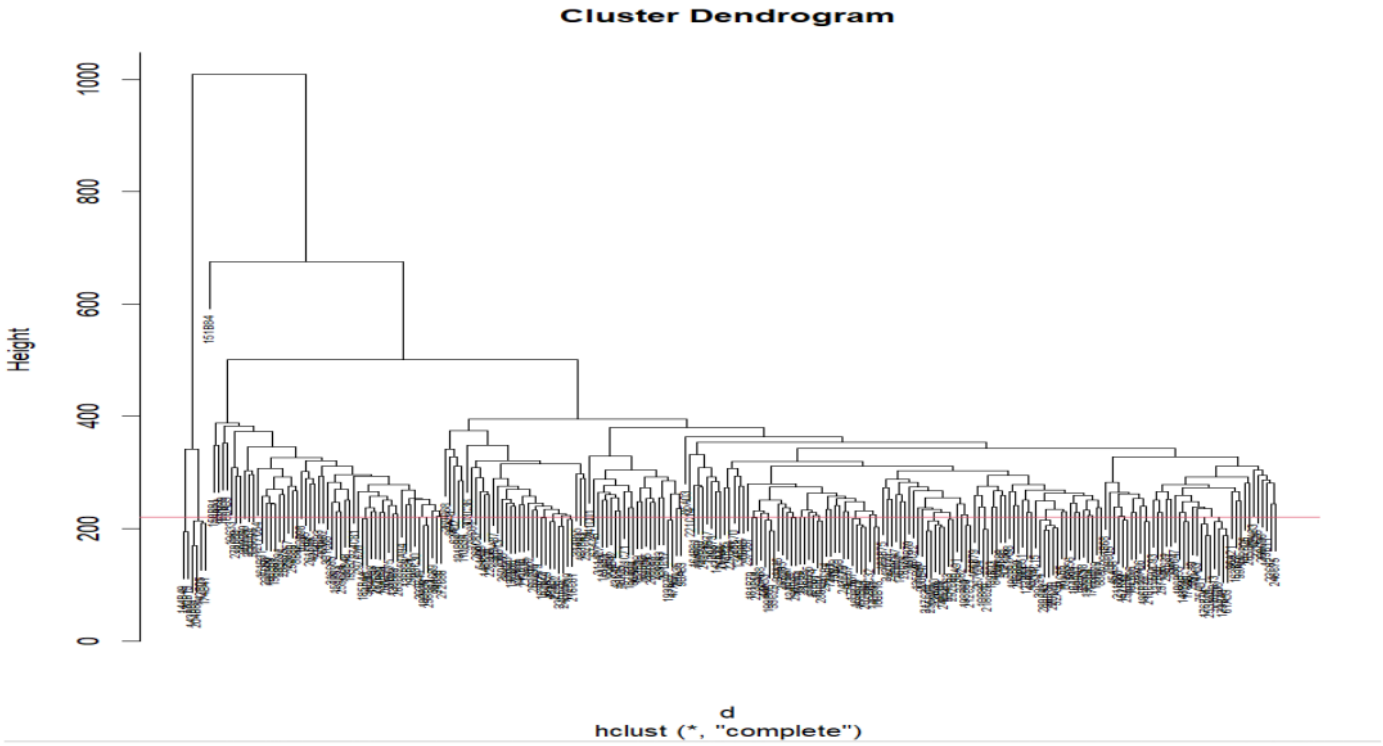


Figure 1: Cluster Dendrogram of Gene Expression Data

The dendrogram represents the hierarchical clustering of gene expression data using complete linkage clustering. The x-axis represents individual samples or clusters of samples, while the y-axis indicates the height (distance) at which samples or clusters are merged. The red horizontal line denotes a specific height threshold.

Selecting the Optimal Number of Clusters:

The graph depicts the relationship between the number of clusters and the median silhouette scores using the complete linkage method. The silhouette score peaks at around 2 clusters, indicating that 2 is the optimal number of clusters for this dataset and almost all the identifiers have been assigned to cluster 1. As the number of clusters increases beyond two, the silhouette score drops significantly, suggesting that increasing the number of clusters may lead to less cohesive and more dispersed clusters.

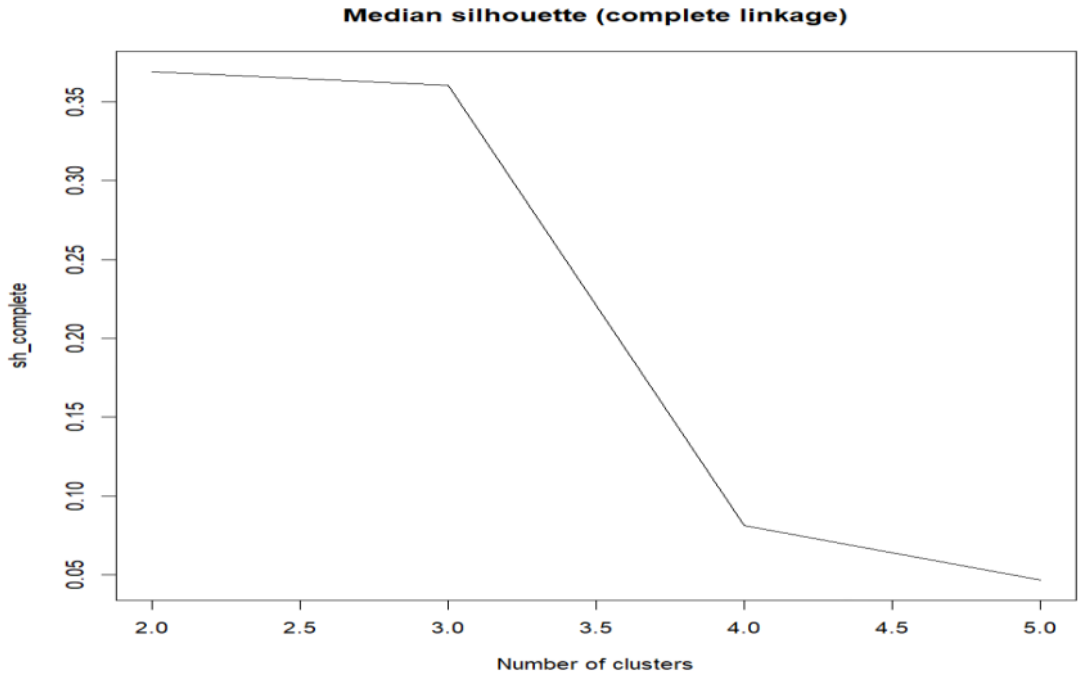


Figure 2: Median silhouette scores for different numbers of clusters using the complete linkage method.

The x-axis represents the number of clusters while the y-axis displays the corresponding median silhouette scores. The peak silhouette score suggests the optimal number of clusters.

Creating a Heatmap:

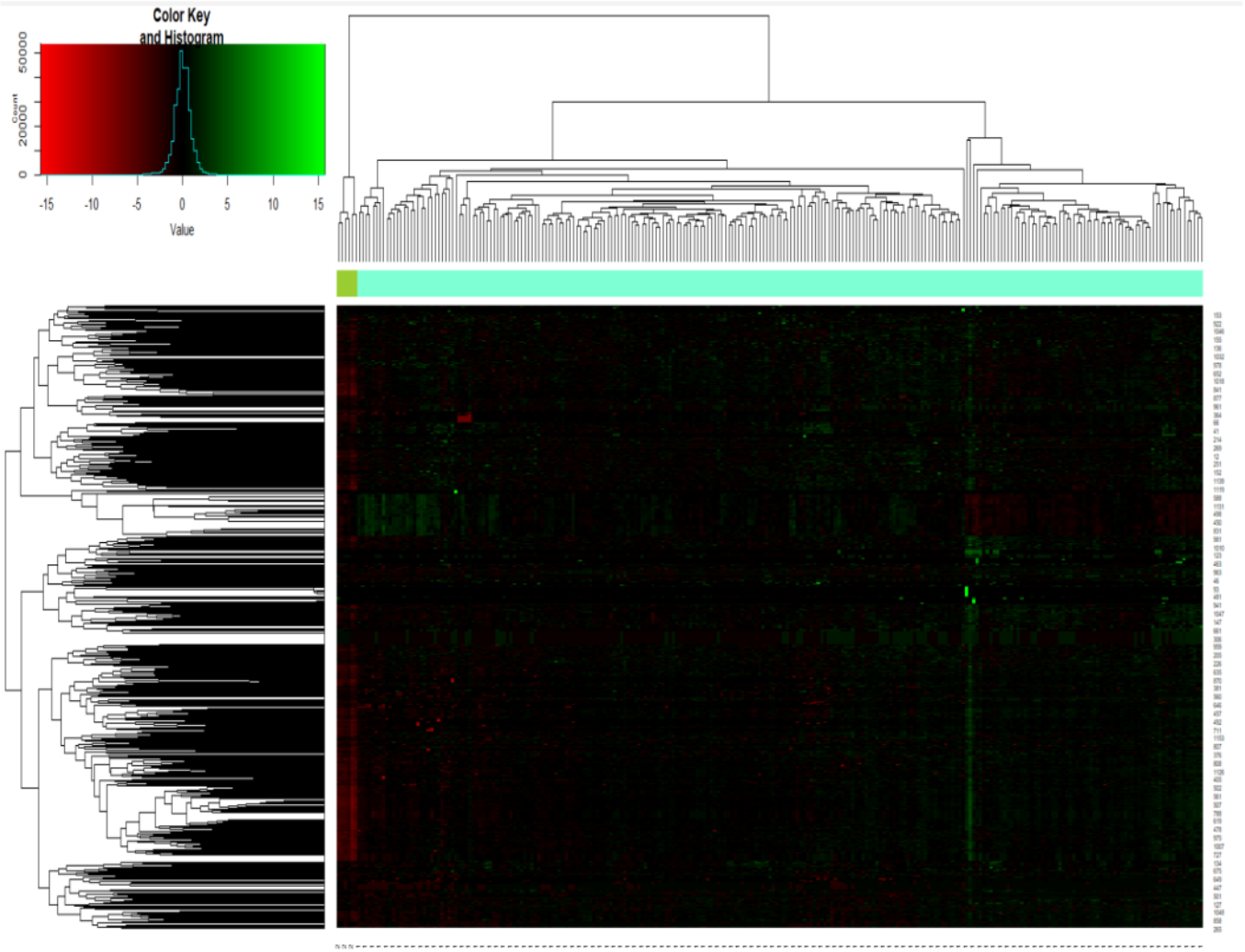


Figure 3: Heatmap of Gene Expression with Hierarchical Clustering.

The heatmap visualizes gene expression data across different samples. Hierarchical clustering is employed to group genes and samples with similar expression profiles, aiding in identifying patterns and potential relationships. The color scale provides a clear representation of upregulated (green), downregulated (red), and neutral (black) genes. The dendrograms on the top and left sides further aid in interpreting the relationships and patterns within the dataset, allowing for insights into co-expressed gene clusters and similarities between samples.

Principal component analysis:

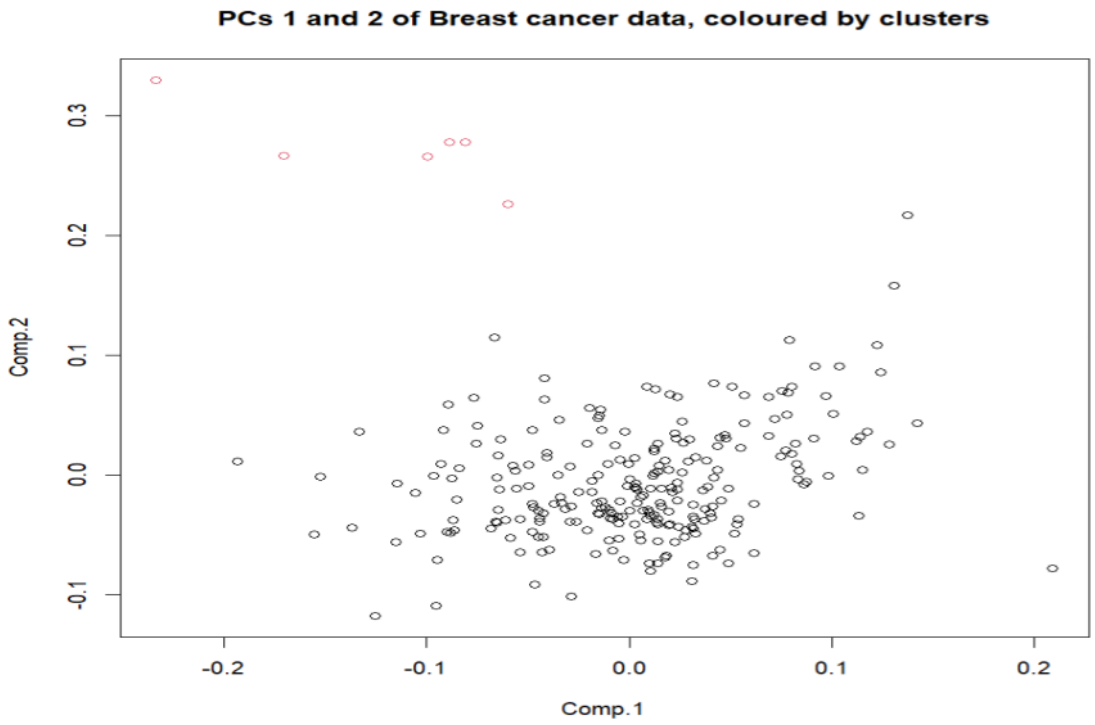


Figure 4: Principal Component Analysis (PCA) of Breast Cancer Data.

Red Circles represent a specific cluster of breast cancer data and the black Circles Indicate another cluster or the general distribution of breast cancer data.

The scatter plot visualizes the PCA of breast cancer data, with data points colored based on identified clusters. Most data points, depicted as black circles, are closely clustered around the origin, indicating similar gene expression patterns. A separate, distinct cluster is represented by the red circles towards the upper part of the graph, suggesting a group with different gene expression characteristics. The two principal components, Comp.1 and Comp.2 capture the maximum variance in the data and allow for differentiation between these clusters.

Gene Expression Analysis:

```
design <- model.matrix(~as.factor(cl))
DE.object <- lmFit(gene_data, design)
DE.object <- eBayes(DE.object)
qval.complete<-qvalue(DE.object$p.value[, 2], fdr.level = 0.05)
sum(qval.complete$significant))
```

[1] 45685

Here, the qvalue function is used to adjust the p-values for multiple testing. The code views the computed q-values and then counts how many genes are deemed significant. The result 45685 indicates that 45,685 genes are differentially expressed at the 5% false discovery rate suggesting they might play a role in the differences between the groups being compared.

Gene ontology test:

	Term	Ont	N	DE	P.DE
GO:0070060	'de novo' actin filament nucleation	BP	1	0	1
GO:0071266	'de novo' L-methionine biosynthetic process	BP	1	0	1
GO:1901737	(R)-mevalonic acid biosynthetic process	BP	1	0	1
GO:1901735	(R)-mevalonic acid metabolic process	BP	1	0	1
GO:1903100	1-phosphatidyl-1D-myo-inositol 3,5-bisphosphate metabolic process	BP	1	0	1
GO:0046360	2-oxobutyrate biosynthetic process	BP	1	0	1
GO:0019606	2-oxobutyrate catabolic process	BP	1	0	1
GO:0006666	3-keto-sphinganine metabolic process	BP	1	0	1
GO:0018960	4-nitrophenol metabolic process	BP	1	0	1
GO:0140484	5-aminolevulinic acid import across plasma membrane	BP	1	0	1
GO:0035998	7,8-dihydroneopterin 3'-triphosphate biosynthetic process	BP	1	0	1
GO:0034463	90S preribosome assembly	BP	1	0	1
GO:0021742	abducens nucleus development	BP	1	0	1
GO:0009738	abscisic acid-activated signaling pathway	BP	1	0	1
GO:0016038	absorption of visible light	BP	1	0	1

>

Figure 5: Overview of Identified Biological Processes using Gene Ontology (GO)

The table showcases a selection of biological processes represented by their specific Gene Ontology (GO) terms. Each term is accompanied by a brief description and is uniformly classified under the Biological Process category. The dataset further breaks down the number of genes that are associated with each GO term(N), the subset of those genes displaying differential expression (DE), and a measure of the significance of said expression(P.DE). Notably, in the data above, each biological process is linked to a single gene that is differentially expressed with a consistency in significance scoring.

Kaplan-Meier Survival analysis:

I can see in your R code that you have done cox regression models, but you have not given results in your report.

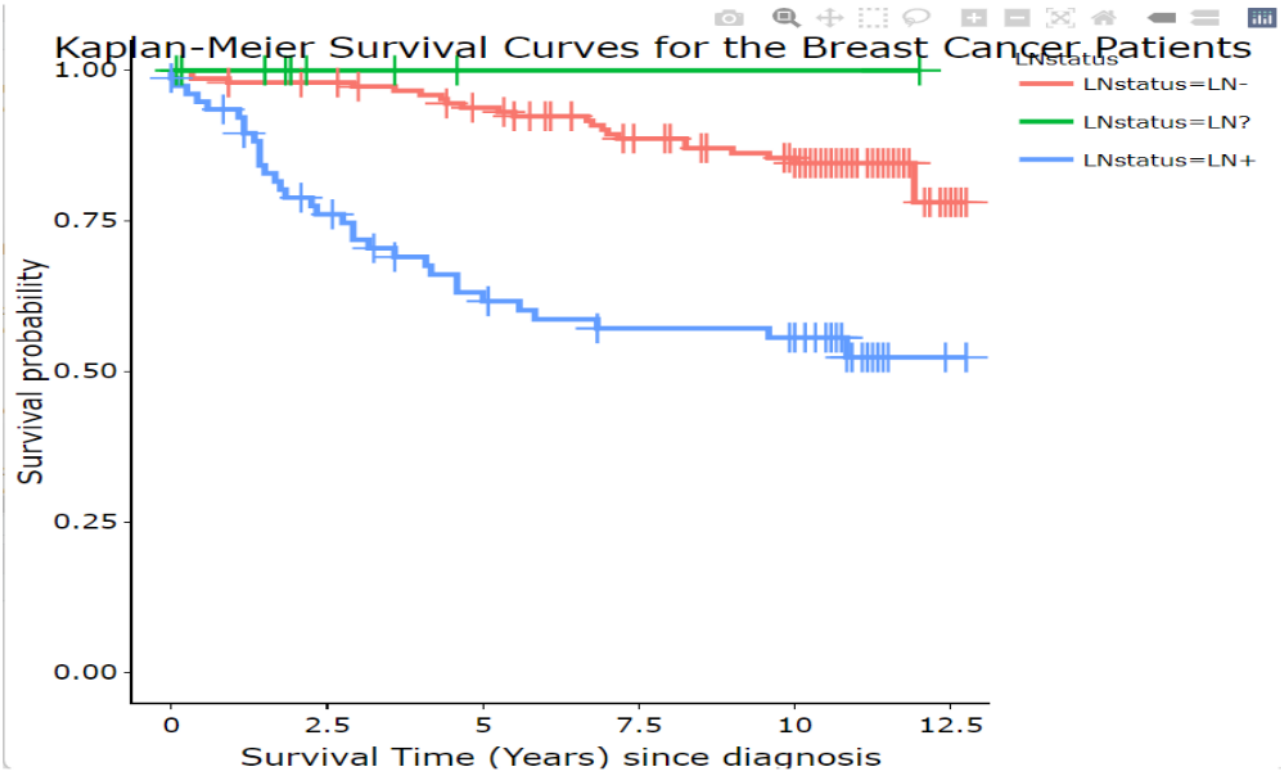
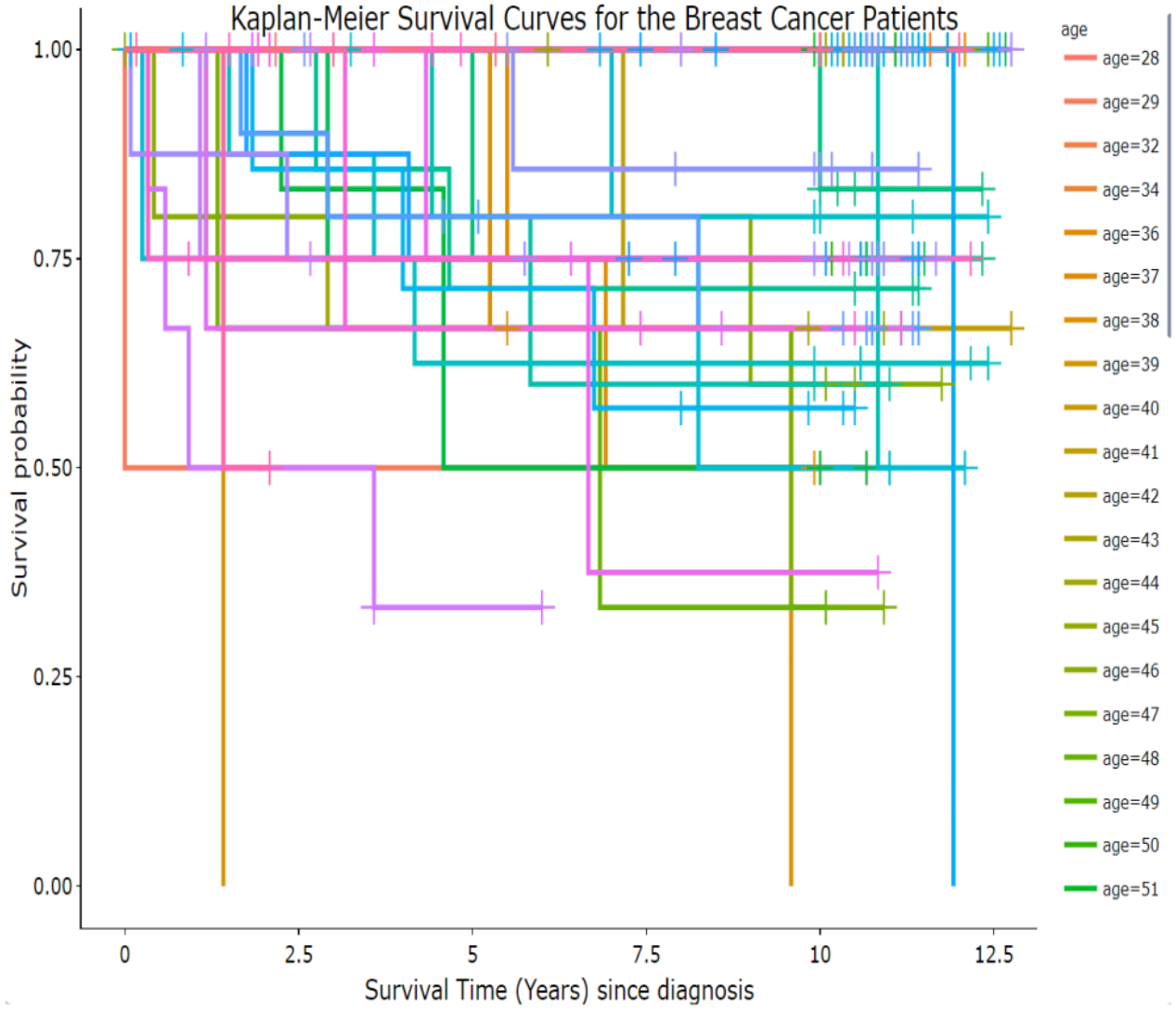


Figure 6: Kaplan-Meier Survival Curves for Breast Cancer Patients Based on Lymph Node (LN) Status.

The Kaplan-Meier survival analysis shown illustrates the survival probabilities of breast cancer patients based on their lymph node status over a span of approximately 12.5 years since diagnosis. Patients with an unknown lymph node status (LN?) demonstrate the highest survival probability, whereas those with a positive lymph node status (LN+) exhibit the lowest. The group with negative lymph node status (LN-) presents an intermediate survival probability. The data underscores the potential impact of lymph node status on survival outcomes in breast cancer patients.

We would like to see K-M survival curves separated for your identified clusters as well



It is better you categorise age into fewer groups and do K-M curves so it is easy to interpret the figure

Figure 6: Kaplan-Meier Survival Curves representing the survival probabilities of breast cancer patients

Over a span of 12.5 years post-diagnosis stratified by age ranging from 28 to 93 years old, the graph displays a diverse range of survival probabilities across different age groups. It's observable that some age groups tend to have higher or lower survival probabilities, although a direct age-to-survival relationship is not immediately clear due to the overlap and convergence of various curves.

It is better you categorise tumour size into fewer groups and do K-M curves so it is easy to interpret the figure

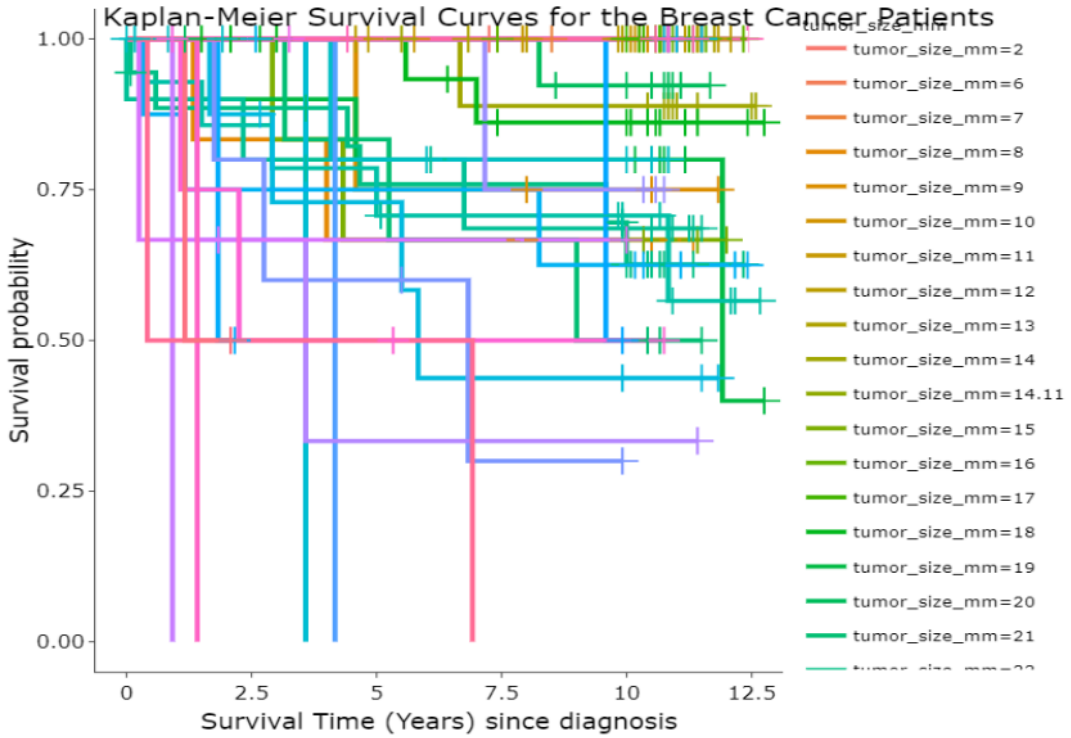


Figure 7: Kaplan-Meier Survival Curves illustrating the survival probabilities of breast cancer patients over approximately 12.5 years since diagnosis, stratified by tumor size. Each colored curve corresponds to a specific tumor size, given in millimeters (mm), ranging from 2mm to 65mm.

Over the course of roughly 12.5 years post-diagnosis, there's a noticeable variation in survival probabilities for different tumor sizes. While some curves overlap, suggesting similar survival rates for certain tumor sizes, other curves diverge significantly, indicating a potential correlation between tumor size and survival rate. This graph underscores the potential impact of tumor size on the prognosis of breast cancer patients.

Conclusion:

The comprehensive microarray analysis of gene expression patterns undertaken in this study and Key findings from the study underscored the existence of distinct “sub-groups” of breast cancer, identifiable through gene expression profiles. Hierarchical clustering and silhouette analysis pinpointed an optimal division into two main clusters, lending credibility to the notion that breast cancer is a spectrum of diseases rather than a single entity. The heatmap visualization and the PCA further solidified these findings, emphasizing clear genetic divergences within the broader diagnosis of breast cancer.

The differential gene expression analysis highlighted a staggering 45,685 genes that exhibited variable expression at a 5% false discovery rate. This sheds light on the breadth of genes that might be involved in the unique progressions and characteristics of different breast cancer subtypes. Moreover, the gene ontology results painted a vivid picture of the biological processes that these genes are implicated in.

The Kaplan-Meier survival analyses provided important clinical insights, demonstrating that factors like lymph node status, patient age, and tumor size have consequential bearings on patient outcomes. Particularly, the lymph node status emerged as a critical determinant of survival probability, reinforcing its significance in clinical settings. Similarly, tumor size and age showcased varying survival outcomes, emphasizing the need for personalized care strategies tailored to these specific factors.

Appendix: R code

```
getwd()

setwd("C:\\Users\\raghu\\OneDrive\\Desktop\\MB
assignment\\applied project")


if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
#BiocManager::install(version = "3.17")
#BiocManager::install("genefilter")
#BiocManager::install("limma")
#BiocManager::install("qvalue")
library(genefilter)
library(limma)
library(qvalue)



#install.packages(c("cluster", "gplots", "RColorBrewer",
"survival"))
library(cluster)
library(gplots)
library(RColorBrewer)
library(survival)



#BiocManager::install("edgeR")
install.packages("edgeR", force = TRUE)


library(edgeR)
```

```
getwd()

setwd("C:\\Users\\raghu\\OneDrive\\Desktop\\MB
assignment\\applied project")


Bc_clinical <- readRDS("bc_clinical.rds")

Bc_expression <- readRDS("BC_data_with_gene_info.rds")


#cluster analysis

dim(Bc_expression)

head(Bc_expression)

gene_data <- data.matrix(Bc_expression[, -(1:3)])

head(gene_data)



#scaling

scale(gene_data,center = TRUE)

scaled.gene_data<-t(scale(t(gene_data),center = TRUE))

dim(scaled.gene_data)


# Normalizing between arrays

normalise_Gene        <-        normalizeBetweenArrays(gene_data,
method="scale")

par(mfrow=c(1,2))
```

You should normalise your data between arrays and then scale it for cluster analysis. You have used un normalised scaled data for your cluster analysis.

```
#complete linkage cluster analysis
d <- dist(t(scaled.gene_data))
hc <- hclust(d, method="complete")
plot(hc, cex=0.5)
abline(h=220, col=2)



#selecting the optimal no of clusters
K<-2:5
sh_complete<-NULL
for (i in K) {
  sh_complete<- c(sh_complete, median(silhouette(cutree(hc, k =
i), dist = d)[, 3], na.rm = T))
}
plot(K,sh_complete,type = "l", main = "Median silhouette (complete
linkage)", xlab = "Number of clusters")
cl <- cutree(hc, k = K[which.max(sh_complete)])
cl

11
install.packages("BiocManager")
BiocManager::install("genefilter")
library(genefilter)
install.packages("gplots")
library(gplots)

1
#heat map
rv <- rowVars(scaled.gene_data)
idx<-order(-rv)[1:1200]
```

```
cols <- colors()[seq(8, length(colors()), len = length(unique(cl)))]
head(cbind(colnames(E), cols), n = 3)
heatmap.2(scaled.gene_data[idx,],labCol = cl,trace = "none", ColSideColors = cols[cl], col = redgreen(100))



#Principal component analysis
par(bg = "white")
pc <- princomp(scaled.gene_data[idx, ])
plot(pc$load[, 1:2], col = cl)
title("PCs 1 and 2 of Breast cancer data, coloured by clusters")


library(qvalue)


#gene expression analysis
design <- model.matrix(~as.factor(cl))
DE.object <- lmFit(gene_data, design)
DE.object <- eBayes(DE.object)
qval.complete<-qvalue(DE.object$p.value[, 2], fdr.level = 0.05)
sum(qval.complete$significant)


#gene ontology test
BiocManager::install("org.Hs.eg.db")
BiocManager::install("GO.db")
library(org.Hs.eg.db)
library(GO.db)
BiocManager::install("clusterProfiler")
library(clusterProfiler)
```

You should use normalised data for DE analysis. i.e. use Normalise_gene matrix in lmFit()

```
homo_sapiens <- org.Hs.eg.db
my_symbols <- as.character(sig_genes)
conversion_tab <- AnnotationDbi::select(homo_sapiens,
                                        keys = my_symbols,
                                        columns = c("ENTREZID",
"SYMBOL"),
                                        keytype = "SYMBOL")



DE_genes <- conversion_tab$ENTREZID[conversion_tab$SYMBOL %in%
rownames(DE.object)[which(qval.complete$significant)]]
library(limma)
GO_terms_present <- goana(DE_genes, species='Hs')
GO_top_terms <- topGO(GO_terms_present, n=200)


library(knitr)
GO_BP_terms <- GO_top_terms[GO_top_terms$Ont=='BP',][1:15,]
knitr::kable(GO_BP_terms)




install.packages("survival")
library(survival)

#survival analysis
Bc_clinical<- readRDS("bc_clinical.rds")
```

```
gene.score<- colSums(gene_data[qval.complete$sig, ])

gene.score<- scale(gene.score)

cox.model <- coxph(Surv(Bc_clinical$Surv_time, Bc_clinical$event)
~ gene.score, data = Bc_clinical)

summary(cox.model)


fit_surv <- survfit(cox.model)

plot(fit_surv,      main="Survival      Curves",      xlab="Time",
ylab="Survival Probability")









#head(Bc_clinical$Surv_Time)

#colnames(Bc_clinical)




#Kaplan-Meier Survival Curves
Y <- Surv(Bc_clinical$Surv_time, Bc_clinical$event == TRUE)

kmfit <- survfit(Y ~Bc_clinical$age)

plot(kmfit, lty = c("solid", "dashed"), col = c("blue", "orange"),

     xlab   =   "Survival   Time   In   Days",   ylab   =   "Survival
Probabilities")


Y <- Surv(Bc_clinical$Surv_time, Bc_clinical$event == TRUE)

kmfit <- survfit(Y ~Bc_clinical$tumor_size_mm)

plot(kmfit, lty = c("solid", "dashed"), col = c("green", "red"),
```

```
     xlab = "Survival Time In Days", ylab = "Survival
Probabilities")


Y <- Surv(Bc_clinical$Surv_time, Bc_clinical$event == TRUE)
kmfit <- survfit(Y ~Bc_clinical$LNstatus)
plot(kmfit, lty = c("solid", "dashed"), col = c("red", "black"),
     xlab = "Survival Time In Days", ylab = "Survival
Probabilities")



install.packages(c("ggplot2", "ggfortify"))
library(ggplot2)
library(ggfortify)



autoplot(kmfit) +
  labs(x = "\n Survival Time (Years) since diagnosis ", y =
"Survival Probabilities \n",
       title = "K-M Survival Curves for the Breast Cancer Patients
\n")



install.packages(c("plotly", "survminer"))
library(plotly)
library(survminer)

model_fit <- survfit(Surv(time = Surv_time, event = event) ~
LNstatus, data =Bc_clinical)
p1 <- ggsurvplot(model_fit)
```

```
plotly::ggplotly(p1[[1]])%>%
  layout(title = "\n Kaplan-Meier Survival Curves for the Breast Cancer Patients",
         xaxis = list(title = "Survival Time (Years) since diagnosis"),
         legend=list(title=list(text='LNstatus')))


model_fit <- survfit(Surv(time = Surv_time, event = event) ~ age, data =Bc_clinical)
p1 <- ggsurvplot(model_fit)
plotly::ggplotly(p1[[1]])%>%
  layout(title = "\n Kaplan-Meier Survival Curves for the Breast Cancer Patients",
         xaxis = list(title = "Survival Time (Years) since diagnosis"),
         legend=list(title=list(text='age')))


model_fit <- survfit(Surv(time = Surv_time, event = event) ~ tumor_size_mm, data =Bc_clinical)
p1 <- ggsurvplot(model_fit)
plotly::ggplotly(p1[[1]])%>%
  layout(title = "\n Kaplan-Meier Survival Curves for the Breast Cancer Patients",
         xaxis = list(title = "Survival Time (Years) since diagnosis"),
         legend=list(title=list(text='tumor_size_mm')))
```

21301487 applied project

ORIGINALITY REPORT

32%	5%	3%	29%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to La Trobe University Student Paper	27%
2	"30th Annual San Antonio Breast Cancer Symposium – December 13–16, 2007", Breast Cancer Research and Treatment, 2007 Publication	1%
3	Submitted to University of North Carolina, Greensboro Student Paper	1%
4	Submitted to Associatie K.U.Leuven Student Paper	1%
5	www.iccs-meeting.org Internet Source	1%
6	Submitted to University of Derby Student Paper	<1%
7	serval.unil.ch Internet Source	<1%
8	www.spandidos-publications.com Internet Source	<1%

9	pubmed.ncbi.nlm.nih.gov Internet Source	<1%
10	W. H. KWON. "Optimizing the Number of Clusters in Multi-Hop Wireless Sensor Networks", IEICE Transactions on Communications, 01/01/2008 Publication	<1%
11	tdx.cat Internet Source	<1%
12	www.ajmb.org Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

21301487 applied project

GRADEMARK REPORT

FINAL GRADE

76 /100

GENERAL COMMENTS

Great work Muniraghuveer!

In your report, no need to provide general descriptions in methods. It is sufficient if you just explain the steps and methods you have used in the analysis and why the methods are chosen.

Two mistakes made: One is that you have used un normalised scaled data for your cluster analysis. The other is that you have used original data without any between array normalisation for your DE analysis. You should normalise your data between arrays and then scale it for cluster analysis. See the comments made on your R-code.

You could try “ward.D2” method as well for clustering. We did not discuss this at the lab session. However it is another available option in hclust function: hclust(dist_bc, method = "ward.D2"). Apparently, this also gives very nice clustering for this data.

I can see in your R code that you have done cox regression models, but you have not given results in your report.

It is better if you categorise age and tumour size into fewer groups and do K-M curves so it is easy to interpret the figures. We would like to see K-M survival curves separated for your identified clusters as well.

Presentation of the report is nice. Include references at the end of the report.

Criteria:

Demonstrate specialised theoretical and technical skills relating to Bioinformatics:

24 out of 30

Use specialised cognitive and technical skills to critically analyse statistical issues in Bioinformatics:

23 out of 30

Exhibit autonomy and expert judgement in the application of established theories relevant to statistical issues in Bioinformatics:

15 out of 20

Use advanced communication skills to summarise and transmit statistical results of Bioinformatics analyses using an appropriate presentation structure:

14 out of 20

Total Mark: 76/100

PAGE 6

PAGE 7

Text Comment. I can see in your R code that you have done cox regression models, but you have not given results in your report.

PAGE 8

Text Comment. We would like to see K-M survival curves separated for your identified clusters as well

PAGE 9

Text Comment. It is better you categorise age into fewer groups and do K-M curves so it is easy to interpret the figure

PAGE 10

Text Comment. It is better you categorise tumour size into fewer groups and do K-M curves so it is easy to interpret the figure

PAGE 11

PAGE 12

PAGE 13

Text Comment. You should normalise your data between arrays and then scale it for cluster analysis. You have used un normalised scaled data for your cluster analysis.

PAGE 14

PAGE 15

Text Comment. You should use normalised data for DE analysis. i.e. use Normalise_gene matrix in lmFit()

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20