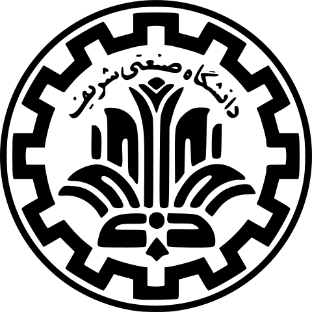
بسم الله الرحمن الرحیم



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

درس: مقدمه بیوانفورماتیک  
استاد درس: دکتر شریفی، دکتر کوهی

پروژه پایانی: تجربه اولیه در تحلیل داده‌های بیوانفورماتیک

دانشجو:

محمدحسین موثقی‌نیا

شماره دانشجویی:

400200919

بهمن 1400

**فهرست**

[**1. مقدمه** 3](#_Toc94626406)

[**1-1 تنظیمات و موارد اولیه** 3](#_Toc94626407)

[**2. کنترل کیفیت داده‌ها** 4](#_Toc94626408)

[**1-2. نرمال‌سازی و نمودار جعبه‌ای** 5](#_Toc94626409)

[**2-2. کاهش ابعاد داده** 7](#_Toc94626410)

[**3-2. بررسی همبستگی بین نمونه‌ها** 12](#_Toc94626411)

[**3. بررسی تمایز در بیان ژن‌ها** 14](#_Toc94626412)

[**4. آنالیز anthology gene و pathway ها** 19](#_Toc94626413)

# **1. مقدمه**

این گزارش شامل توضیحات هر بخش از پروژه به همراه قطعه کد‌ها و نمودار‌ها می‌باشد. در کنار فایل گزارش، کدها، نمودارها و پیوست‌های دیگر قرار داده شده‌اند.

## **1-1 تنظیمات و موارد اولیه**

به منظور تحليل داده‌ها ابتدا مي بايست، داده‌ها را دانلود و در دايركتوري مناسب قرار داده و نمونه هاي نرمال و تست را مشخص نمود. براي اين منظور از طريق قطعه كد زير، محل فعلي دايركتوري را به عنوان مرجع مشخص مي كنيم و سپس براساس آن آدرس دهي هاي بعدي را انجام مي دهيم:

|  |  |
| --- | --- |
|  | curD <- dirname(rstudioapi::getActiveDocumentContext()$path)  setwd(sub(paste0("/",sub("(.+)/","",curD)),"",curD)) |

سپس مي توان شماره سري داده‌ها و پلتفرم آن‌ها را به صورت مجزا تعيين نمود، البته الزامي به اين كار نيست و مي توان داخل كد دستوري قرار داد، پس از آن مي بايست داده‌ها را در دايركتوري مورد دلخواه خود دانلود نمود:

|  |  |
| --- | --- |
|  | series <- "GSE48558"  platform <- "GPL6244"  gset <- getGEO(series, GSEMatrix = TRUE, AnnotGPL = TRUE, destdir = 'data/') |

براي اين منظور از تابعي در كتابخانه GEOquery به منظور دانلود استفاده مي شود.

پس از آن درصورتي كه داده چندين پلتفرم داشته باشد، صرفا پلتفرم خاص تعيين شده توسط سوال را مي خواهيم. براي اين منظور مي بايست يك فيلتر روي داده صورت پذيرد:

|  |  |
| --- | --- |
|  | if (length(gset) > 1) idx <- grep(platform, attr(gset, "names")) else idx <- 1  gset <- gset[[idx]] |

البته درصورتي كه از داده length گرفته شود و صرفا يك پلتفرم موجود باشد، كافيست همان جايگزين شود و ديگر به عنوان ليستي از پلتفرم‌ها نباشد.

در نهايت مي بايست گروه هاي نرمال و بيمار و غيره را مشخص كنيم و گروه هاي به جز بيمار و نرمال را از داده‌ها حذف كنيم:

|  |  |
| --- | --- |
|  | gsms <- paste0("1111111111111XXXXXXXXXXXXXXXXXXXXXXXXXXX0XXX0XXXXX",  "XXXXXXXXXXXXXXXXXX0X0XXX0X0000X0XX00XX00X0X0X0X0X0",  "XXX0XXX0XXXXXXXXXXXXXXXXXXXXXXXXXXXXX0000000110111",  "00000000000000000000")  sml <- strsplit(gsms, split="")[[1]]  ### filter by X  sel <- which(sml != "X")  sml <- sml[sel]  gset <- gset[ ,sel]  gs <- factor(sml)  groups <- make.names(c("normal", "test"))  levels(gs) <- groups  gset$group <- gs |

از طريق قطعه كد بالا مي توان داده‌ها را دسته بندي كرده و يك ستون جديد به داده‌ها افزود و گروه بندي مورد نظر را در آن گنجاند. (البته گروه هاي غيراز سالم و بيمار، از داده‌ها حذف مي شوند.)

# **2. کنترل کیفیت داده‌ها**

این قسمت شامل موارد زیر می‌باشد که با توجه به تعریف صورت سوال پروژه در قسمت‌های مجزایی بررسی خواهند شد:

* نرمال‌سازی دادها در صورت نیاز، به منظور بررسی صحیح داده‌ها، چرا که در صورت عدم نرمال‌سازی ممکن است در صورت‌اندازه گیری نمونه‌ها با کیت‌های مختلف، تاثیرات محیطی و ... دو نمونه که در واقعیت شباهت زیادی به هم داشته‌اند به دلایل نام برده تفاوت فاحشی با هم پیدا کنند.
* کاهش ابعاد به منظور بررسی پیوستگی و شباهت داد‌های مورد بررسی
* بررسی همبستگی بین نمونه‌ها به جهت اطمینان از مرتبط بودن نمونه‌های جمع آوری شده، می‌باشد.

البته بخش‌های بالا صرفا به منظور بررسی کیفیت داده نمی باشد و اطلاعات گسترده دیگری نیز در اختیار قرار می‌دهد که در هر بخش بیشتر بحث خواهد شد.

## **1-2. نرمال‌سازی و نمودار جعبه‌ای**

به منظور بررسی نرمال بودن داده‌ها، یکی از راه‌ها رسم نمودار جعبه‌ای نمونه هاست. درصورتی که نمودار‌های جعبه‌ای حاصل به صورت منظمی در کنار هم و تقریبا مشابه یک دیگر از نظر چارک‌های اول، دوم و سوم باشند؛ می‌توان نتیجه گیری کرد که داده‌ها با هم هماهنگ و نرمال هستند. درغیر این صورت می‌بایست نرمال‌سازی با یکی از روش‌های مناسب، صورت پذیرد. یکی از روش‌های متداول و ساده نرمال‌سازی، نرمال‌سازی گسسته[[1]](#footnote-1) می‌باشد، که به این صورت عمل می‌کند در هر دور اجرای آن ماکسیمم هر سری از داده‌ها را میانگین می‌گیرد و جایگزین تمامی آن‌ها می‌کند و در دور بعدی آن‌ها را در نظر نمی گیرد. به این ترتیب داده‌ها در یک طیف یکسان دسته بندی می‌شوند. البته روش‌های دیگری که تفاوت داده‌ها را بتواند بهتر پوشش دهد نیز وجود دارد.

داده‌های موجود در این پروژه در صورت رسم نمودار جعبه‌ای نرمال هستند. (شکل 1) به منظور این کار می‌بایست، ابتدا ماتریس بیان ژن‌ها را از داده‌ها به دست آوریم، برای این منظور از طریق قطعه کد زیر این کار را انجام می‌دهیم:

|  |  |
| --- | --- |
|  | ex <- exprs(gset) |

درصورت نیاز به نرمال‌سازی، می‌توان از قطعه کد زیر استفاده نمود:

|  |  |
| --- | --- |
|  | ex <- exprs(gset)  ex <- normalizeQuantiles(ex)  exprs(gset) <- ex |

تابع exprs یک تابع دو طرفه بوده و می‌تواند مقادیر را علاوه بر خروجی گرفتن، مقداردهی نیز انجام دهد. تابع normalizeQuantiles نیز همان نرمال‌سازی گسسته را برای داده‌های بیان ژن انجام می‌دهد.

که البته در داده‌های موجود، با توجه به شکل 1 نیازی به نرمال‌سازی نیست، درصورت اجرای نرمال‌سازی بالا و رسم دوباره نمودار جعبه ای، نمودار حاصل، شکل 2 خواهد بود که کاملا مشخص است که تغییر قابل توجه‌ای رخ نداده است. بنابراین داده‌ها به صورت پیشفرض نرمال بوده‌اند.

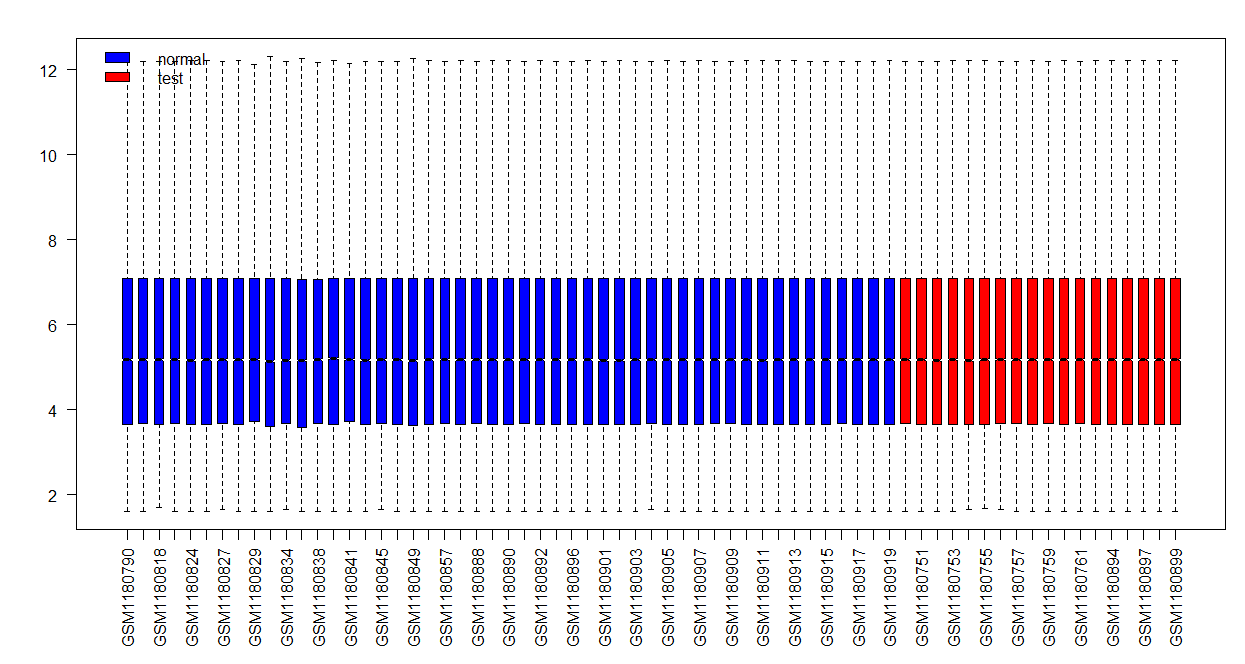
سپس به منظور رسم نمودار جعبه‌ای بر اساس ژن‌های بیان شده از طریق قطعه کد زیر اقدام می‌کنیم:

|  |  |
| --- | --- |
|  | boxplot(ex) |

به منظور ساخت بهتر نمودار جعبه‌اي و ايجاد قابليت تفكيك بين نمونه هاي سالم و بيمار، با استفاده از قطعه كد زير مي توان اين تفاوت را ايجاد نمود:

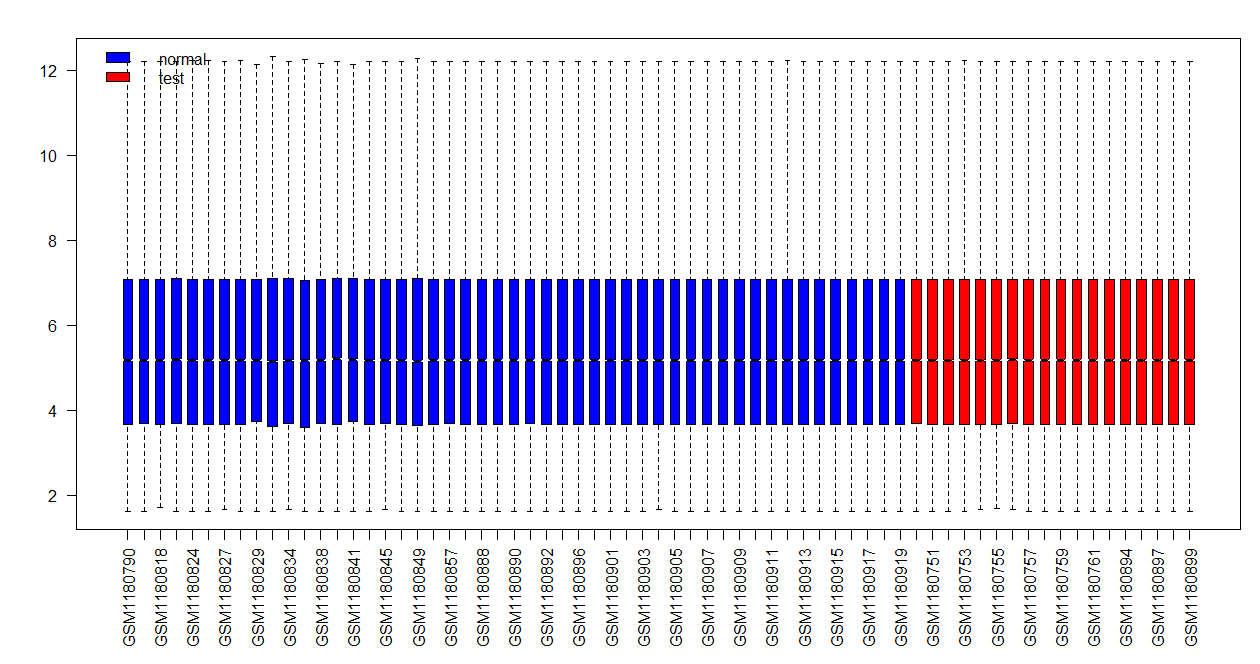
|  |  |
| --- | --- |
|  | dev.new(width=3+ncol(gset)/6, height=5)  ord <- order(gs)  palette(c("blue", "red"))  par(mar=c(7,4,2,1))  boxplot(ex[,ord], boxwex=0.6, notch=T, outline=FALSE, las=2, col=gs[ord])  legend("topleft", groups, fill=palette(), bty="n") |

در كد بالا، ابتدا ابعاد نمودار خروجي مبتني بر نمونه‌ها مشخص مي‌شود، سپس ترتيب نمايش نمونه‌ها بر اساس نرمال به بيمار مرتب مي‌شود و پس از آن رنگ‌بندي نمودار براي جداسازي نرمال از بيار و در نهايت رسم نمودار و تعيين راهنما براي آن (مشخص كردن اين كه هر رنگ مرتبط به كدام نمونه است.)



**شکل 1**- نمودار جعبه‌ای نمونه‌های سالم و بیمار. نمونه‌های بیمار با رنگ قرمز و نمونه‌های سالم با رنگ آبی مشخص شده‌اند. نمودار جعبه‌ای نشان دهنده نرمال بودن داده‌ها و عدم نیاز آن‌ها به نرمال‌سازی است. (به دلیل هماهنگی کامل داده‌ها با هم از نظر چارک اول، میانه و چارک سوم و همچنین ماکسیمم و مینیمم ها.

**2-2. کاهش ابعاد داده**

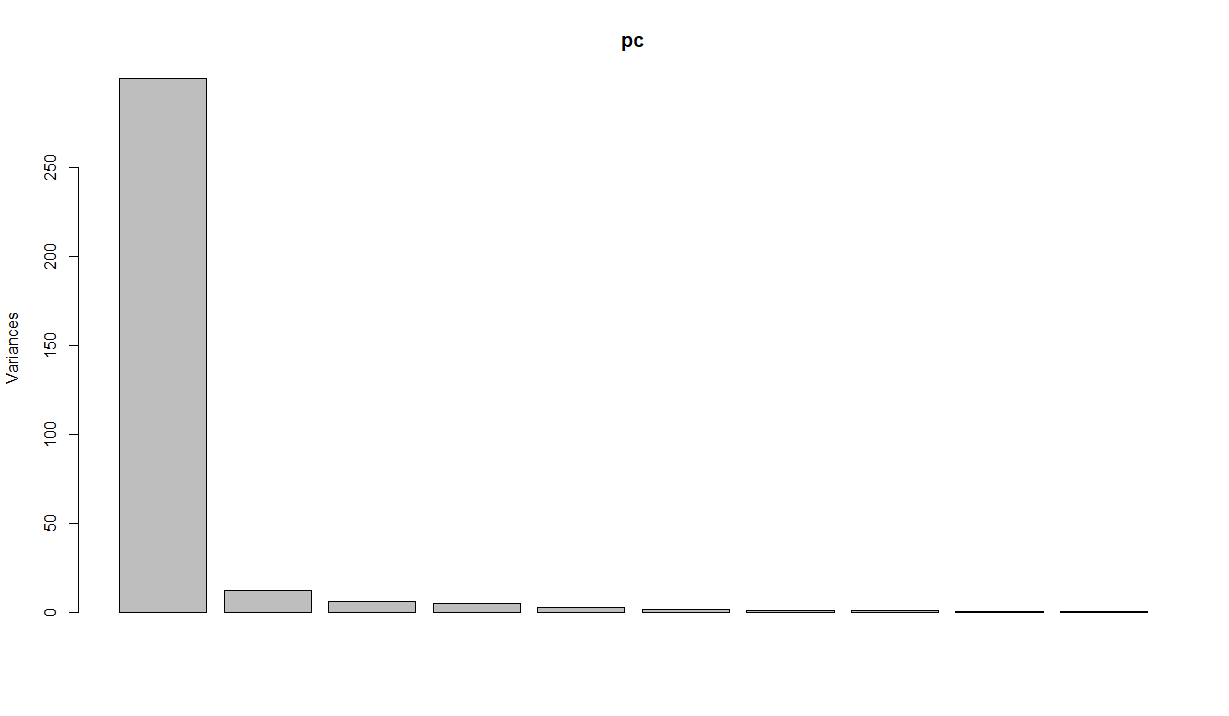


**شکل 2**- نمودار جعبه‌ای نمونه‌های سالم و بیمار. نمونه‌های بیمار با رنگ قرمز و نمونه‌های سالم با رنگ آبی مشخص شده‌اند. نمودار جعبه‌ای پس از اعمال نرمال‌سازی گسسته نشان دهنده این است که داده‌ها قبل از آن نیز نرمال بودنه‌اند چرا که در مقایسه با شکل1 تغییر محسوسی مشاهده نمی‌شود. از طرف دیگر شکل1 خود کاملا مشخص است که نرمال می‌باشد.

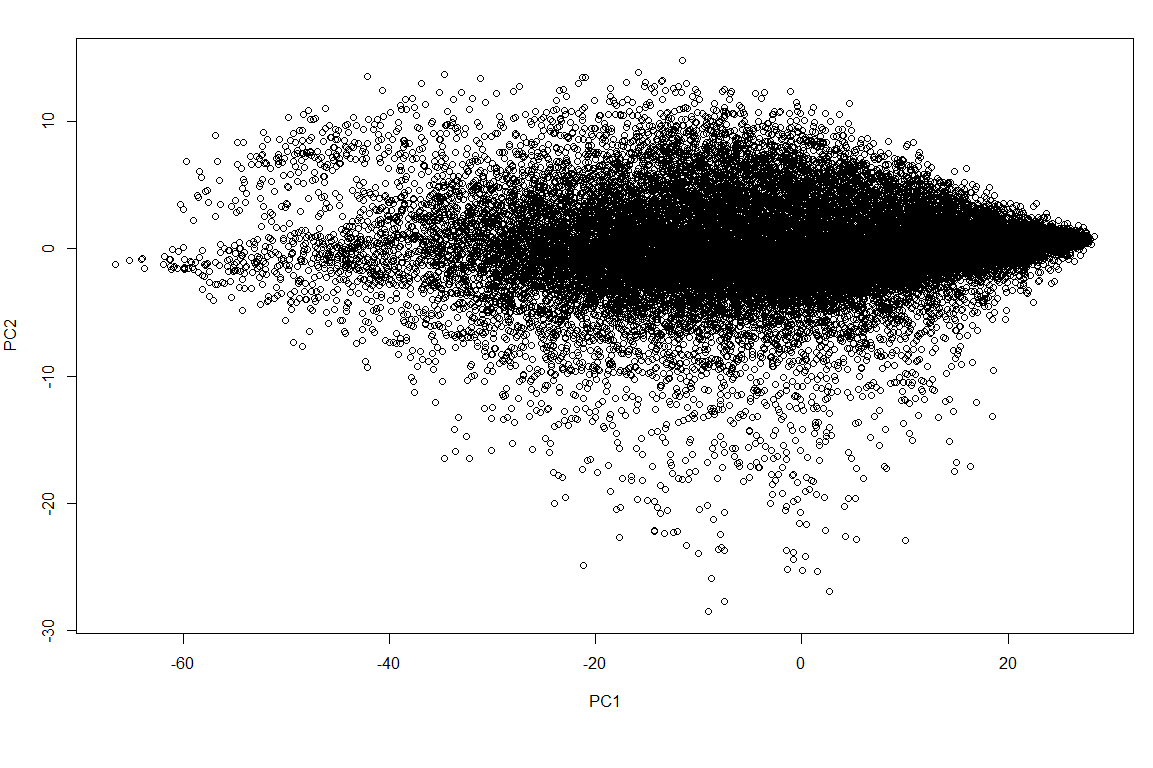
به منظور کاهش ابعاد روش‌های مختلفی وجود دارد، یکی از این روش‌ها PCA[[2]](#footnote-2) می‌باشد. در این روش، چندین جهت مختلف برای محور‌های بررسی داده در نظر گرفته می‌شود و در هر کدام از این روش‌ها مقدار تفکیک بین داده‌ها مورد بررسی قرار می‌گیرد. برای این منظور می‌توان از قطعه کد زیر برای بیان ژن‌ها استفاده کرد:

|  |  |
| --- | --- |
|  | pc <- prcomp(ex) |

داخل متغیر pc تمامی جهت‌هایی که می‌تواند داده را تفکیک کند به ترتیب میزان تمایز مرتب شده‌اند و در صورت رسم نمودار آن (شکل3) می‌توان تفاوت‌های میزان تفکیک در هر کدام از این مولفه‌ها را مشاهده کرد. همانگونه که در شکل3 مشخص است میزان تمایز در مولفه اول از همه بیشتر و به ترتیب کاهش می‌یابد.



**شکل 3**- بررسی هرکدام از مولفه‌های خروجی تابع prcomp نشان دهنده میزان واریانس (تمایز) ایجاد شده میان ژن‌ها در هرکدام از این مولفه هاست. به ترتیب میزان تمایز از اولین به آخرین مولفه مرتب شده است. بنابراین بهترین گزینه برای رسم نمودار کاهش ابعاد مولفه اول و دوم می‌باشد. البته درصورت نیاز به رسم نمودار سه بعدی می‌توان از مولفه سوم نیز کمک گرفت.



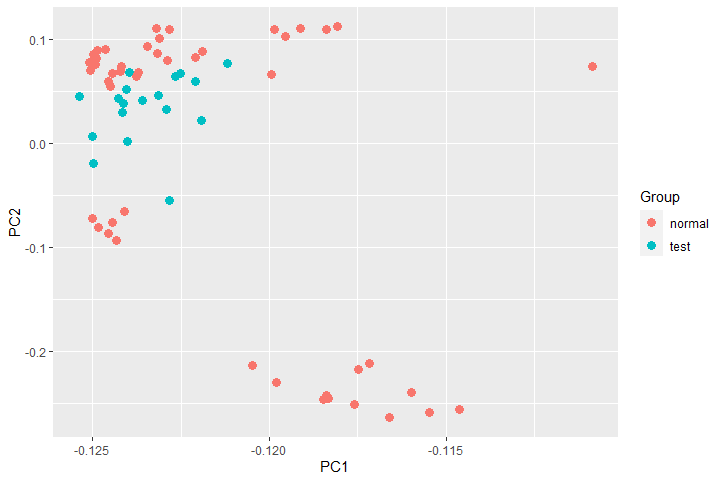
**شکل 4**- رسم داده‌ها مبتنی بر مولفه اول و دوم روش PCA. همانطور که در نمودار مشخص است تفاوت‌های موجود در مولفه اول به نوعی صرفا نشان می‌دهد که بعضی از ژن‌ها خیلی کم بیان شده‌اند یا اصلا بیان نشده‌اند و در مقابل بعضی دیگر بسیار زیاد بیان شده‌اند. این به نوعی داده‌ای ناکارآمد محسوب می‌شود.

درصورت رسم نمودار دو بعدی براساس مولفه اول و دوم که بیشترین تفکیک را ایجاد می‌کنند، مشاهده می‌شود که یک توزیع افقی گسترده برای ژن‌ها وجود دارد.(شکل4) مولفه اول در شکل4 نشان دهده میزان بیان ژن‌های مختلف در نمونه می‌باشد. همانگونه که درصورت رسم PCA برای نمونه‌ها مطابق شکل5، در این نمودار نیز این نمونه‌ها قابل جداسازی از یکدیگر نیستند و این نشان دهنده وجود مشکل در روش تحیلی می‌باشد که باید اصلاح شود. به نوعی این خروجی ارزش زیادی ندارد چرا که هدف یافتن تفاوت بیان هاست، برای رفع این مشکل با تغییر ابعاد داده‌ها تاثیر عدم بیان یا بیان بالای ژن‌ها را از بین برد. برای این منظور بیان ژن‌ها از میانگین بیان خودشان کسر می‌شوند (به عبارت دیگر میانگین بیان تمامی ژن‌ها صفر شوند.) تا اثر بیان دائمی یک ژن یا عدم بیان آن از بین برود و فقط تفاوت‌ها ارزشمند شود و مولفه‌های PCA موارد بهتری را نمایش دهند. برای این منظور از طریق کد زیر این تغییر ابعاد انجام می‌شود:

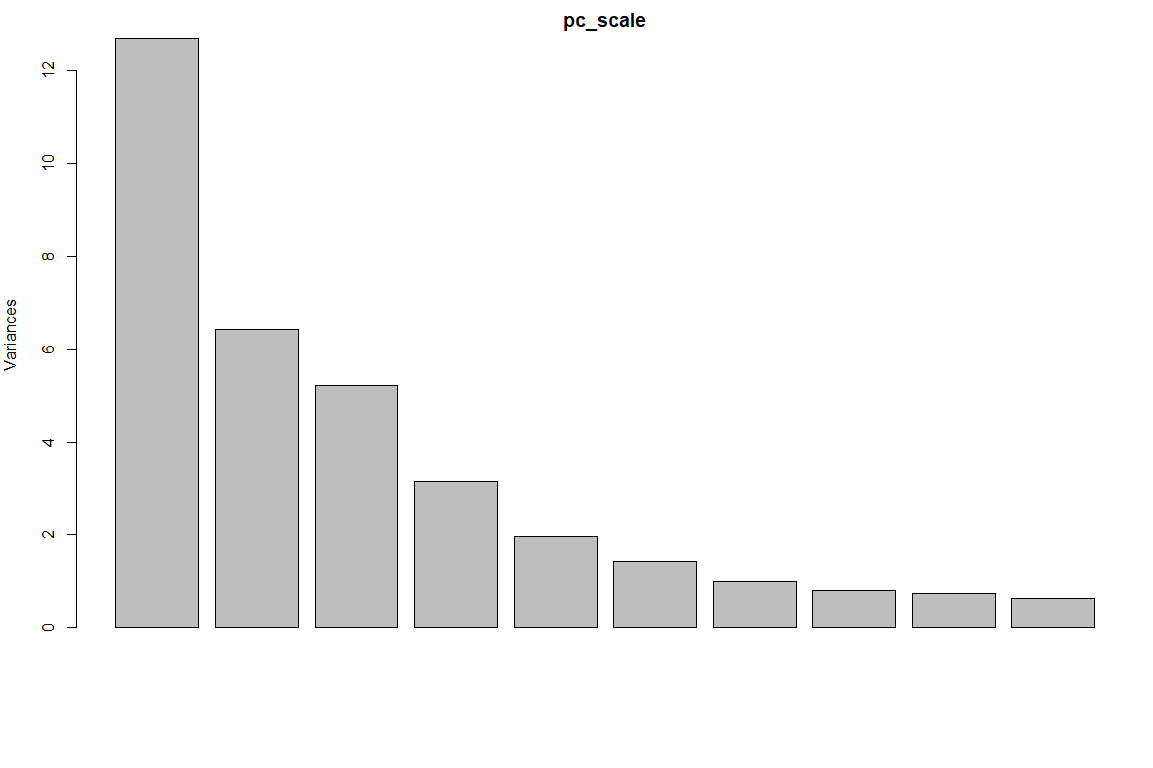
|  |  |
| --- | --- |
|  | ex.scale <- t(scale(t(ex), scale = F)) |

به دلیل این که تابع scale ستون‌ها را تغییر ابعاد می‌دهد، می‌بایست ماتریس بیان را ترانهاده کرده و سپس تغییر ابعاد اعمال شود و سپس دوباره به حالت ترانهاده اولیه برگردد. به منظور این که تابع scale به صورت پیشفرض داده‌ها را تقسیم بر انحراف معیار نیز می‌کند، می‌بایست این عملیات را از طریق False کردن متغیر scale در تابع، لغو نمود.

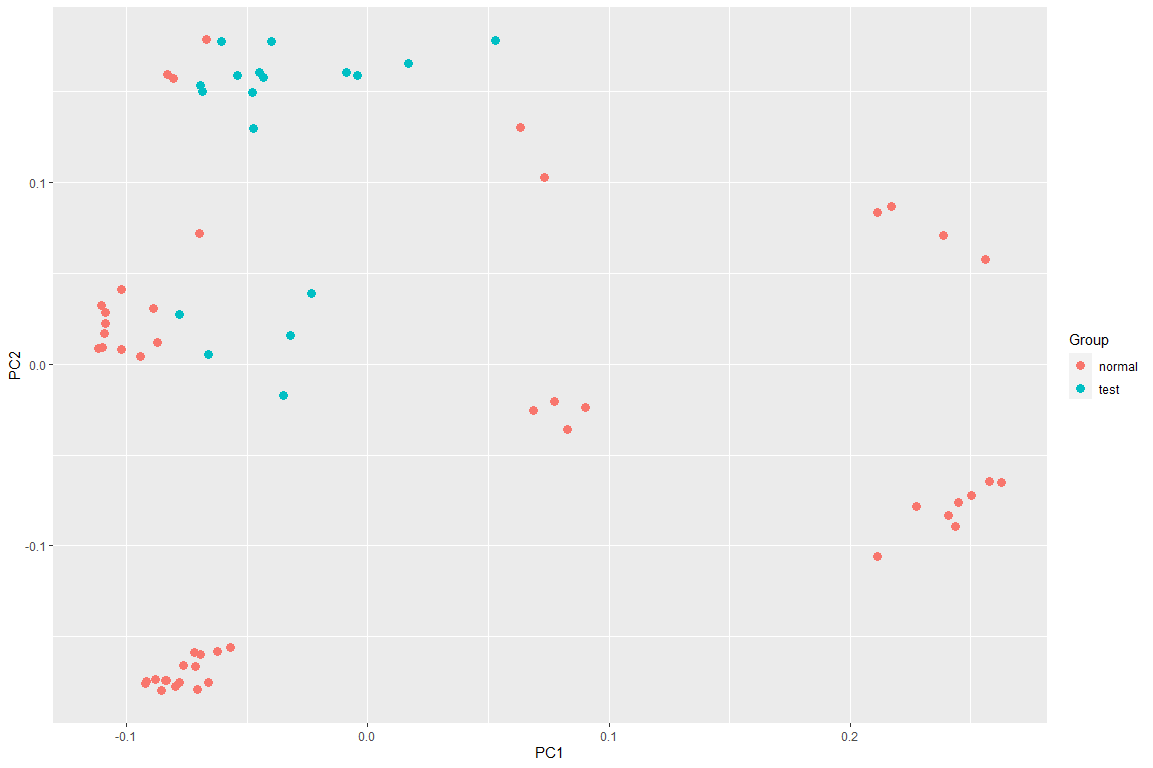
درصورتی که براساس مقادیر جدید، مولفه‌های مختلف روش PCA مورد بررسی قرار داده شود، شکل6 حاصل می‌شود که تمایز‌ها در مولفه‌های مختلف معقول تر شده‌اند و از طرفی در صورت رسم نمونه‌ها مطابق شکل7 ، می‌توان نتایج را مشاهده نمود که دیگر مولفه اول به سمت بیان کامل یا عدم بیان کامل گرایش ندارد و داده‌ها به صورت متفاوتی قرار گرفته‌اند. به منظور بررسی نمونه‌ها نیز می‌توان نمودار را مطابق روش قبلی برای نمونه‌ها رسم کرد (شکل8) به این ترتیب می‌توان مشاهده کرد که نمونه‌های مختلف تقریبا در کلاستر‌های مختلفی قابل جداسازی از یکدیگر هستند و مولفه اول و دوم روش PCA پس از اعمال تغییر ابعاد داده ها، به خوبی توانسته است تمایز میان آن‌ها را نمایش دهد. البته بعضی از نمونه‌های سرطانی شباهت زیادی به نمونه‌های سالم دارند ولی با این حال اکثر آن‌ها قابل تمایز هستند.



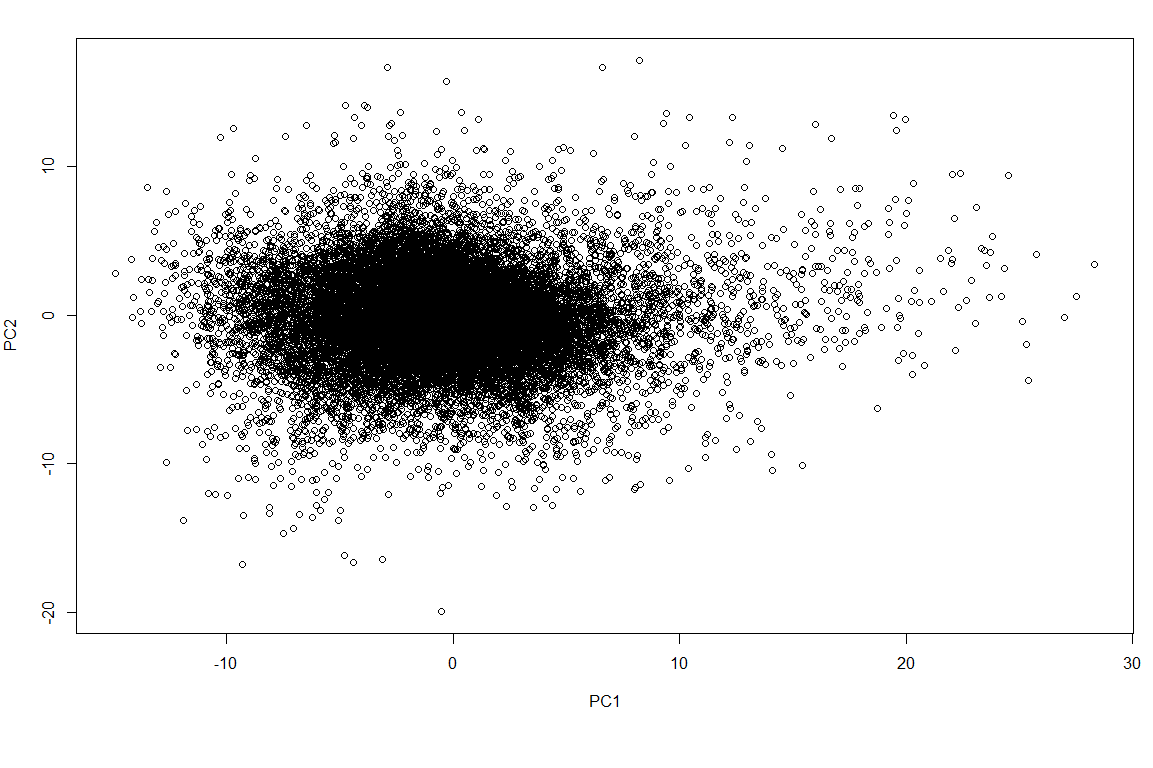
**شکل 5** - تفاوت نمونه‌ها مبتنی بر مولفه‌های اول و دوم به دست آمده از روش PCA، پیش از اعمال تغییر ابعاد.



**شکل 6**- مولفه‌های مختلف به دست آمده مبتنی بر روش PCA پس از اعمال تغییر ابعاد. همانگونه که مشخص است دیگر تفاوت فاحش میان مولفه اول و دیگر مولفه‌ها نیست و با اعمال تغییر ابعاد، امکان این فراهم شد تا تاثیر بسیار بزرگی که ژن‌هایی که در تمامی نمونه‌ها بیان شده بودند و همچنین ژن‌هایی که اصلا بیان نشده بودند از بین برود و این روش بتواند تمایز بهتری متناسب با تفاوت بیان ژن‌ها ارائه بکند.



**شکل 7**- تفاوت نمونه‌ها مبتنی بر مولفه‌های اول و دوم به دست آمده از روش PCA، پس از اعمال تغییر ابعاد.



**شکل 8**- تفاوت بیان ژن‌های مختلف نمونه‌ها مبتنی بر مولفه اول و دوم روش PCA پس از تغییر ابعاد داده. همانگونه که در شکل مشهود است، با تغییر ابعاد داده و از بین بردن تاثیر بیان ژن‌ها و یا عدم بیان آن‌ها در اکثر نمونه ها، می‌توان به خروجی بهتری مبتنی بر تفاوت بیان ژن‌ها دست یافت.

به منظور رسم نمودار‌های 3 و 6 قطعه کد زیر مورد استفاده قرار می‌گیرد:

|  |  |
| --- | --- |
|  | plot(pc) |

به منظور رسم نمودار‌های 4 و 7 قطعه کد زیر مورد استفاده قرار می‌گیرد:

|  |  |
| --- | --- |
|  | plot(pc$x[,1:2]) |

در قطعه کد بالا، x نشان دهنده ژن‌ها می‌باشد که بر اساس 2 مولفه اول برای رسم ارسال شده است.

به منظور رسم نمودار‌های 5 و 8 از کتابخانه ggplot2 استفاده شده است و قطعه کد زیر مورد استفاده قرار گرفته است:

|  |  |
| --- | --- |
|  | pcr <- data.frame(pc$r[,1:3], Group = gset$group)  ggplot(pcr, aes(PC1, PC2, color=Group)) + geom\_point(size=3) |

در قطعه کد بالا، مبتنی بر rotation در pc که همان نمونه‌های مختلف هستند، یک دیتافریم به همراه گروه‌های آن‌ها ساخته شده‌است و سپس به ggplot داده شده تا مبتنی بر مولفه اول و دوم و همچنین رنگ‌بندی براساس گروه ها، نمودار را رسم نماید.

## **3-2. بررسی همبستگی بین نمونه‌ها**

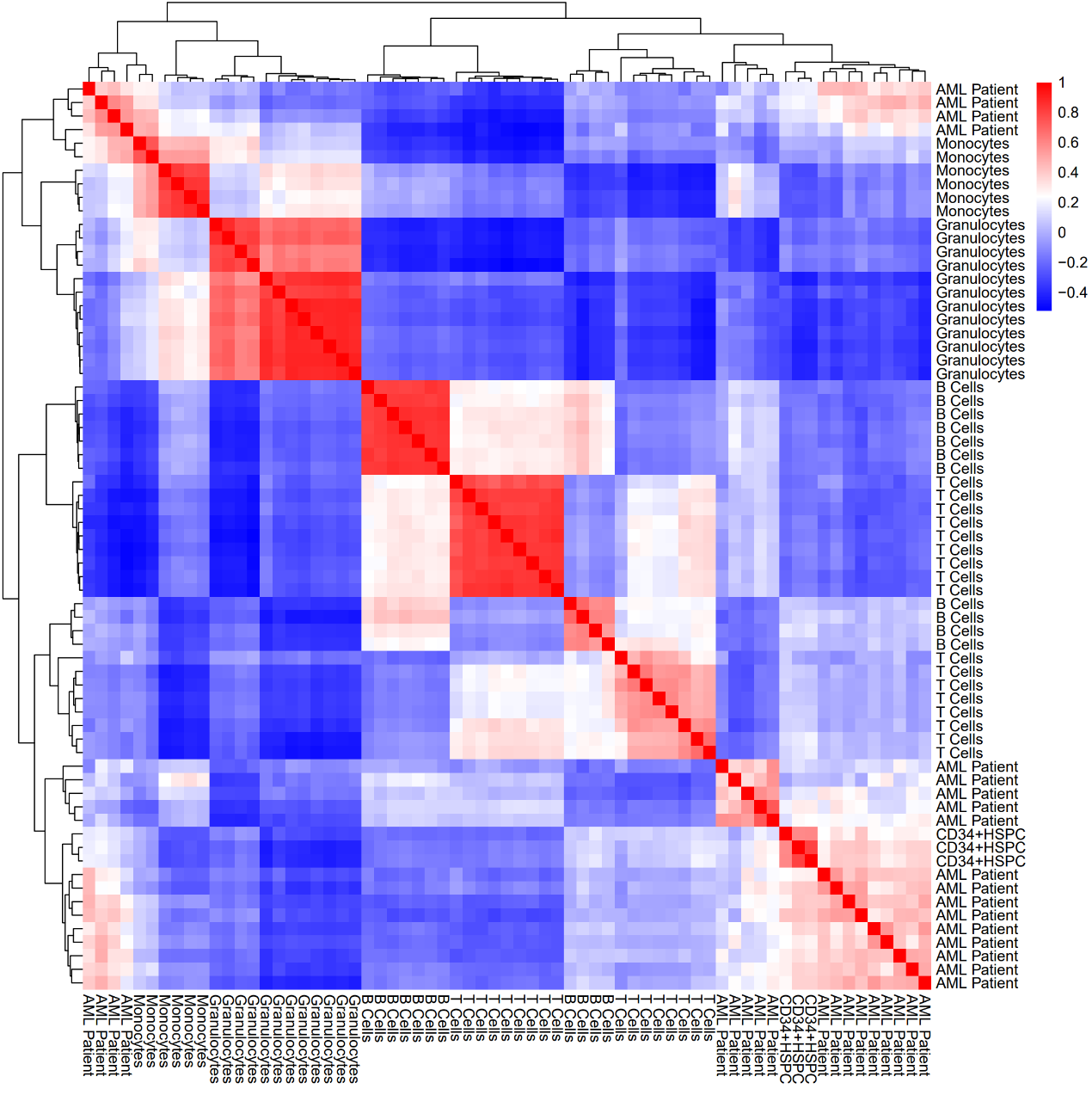
بررسی همبستگی دو به دو نمونه‌ها با یکدیگر با هدف یافتن میزان ارتباط بین نمونه‌های مختلف با یکدیگر می‌باشد. برای این منظور می‌توان از Heatmap استفاده نمود. با توجه به تغییر ابعاد انجام شده در بخش قبل، مبتنی بر خروجی‌های آن اقدام به رسم Heatmap می‌شود. ابتدا ماتریس همبستگی دو به دو برای نمونه‌ها مشخص می‌شود. سپس Heatmap براساس این همبستگی رسم می‌شود.

|  |  |
| --- | --- |
|  | ex.scale.cor <- cor(ex.scale)  pheatmap(ex.scale.cor,  labels\_row = gset$source\_name\_ch1,  labels\_col = gset$source\_name\_ch1,  color = bluered(255), border\_color = NA) |

مطابق کد بالا، ابتدا همبستگی نمونه ها محاسبه شده و سپس با استفاده از کتاب‌خانه pheatmap اقدام به رسم Heatmap مبتنی بر همبستگی نمونه ها می شود. رنگ‌بندی نمودار به صورت آبی و قرمز بوده و نامگذاری براساس source\_name نمونه هاست. نتیجه این نمودار در شکل9 قابل مشاهده است.

به منظور نمایش بهتر خروجی، Heatmap خروجی در فایل heatmap-gene.scale.pdf در دایرکتوری results ذخیره شده است.

همانگونه که در شکل9 مشخص است میزان همبستگی در نمونه های AML Patient و نمونه های سالم، به صورت کلی چهار نمونه AML Patient و 6 نمونه Monocytes و 12 نمونه Granulocytes از دیگر نمونه ها مجزا هستند و همبستگی کمتری به دیگر اعضا دارد، از طرفی در بین این گروه ارتباط AML Patient با یک دیگر نسبتا بالاست و همچنین ارتباط بعضی از Monocytes ها با یکدیگر بسیار بالاست و همچنین از بین Granulocytes ها بعضی ارتباط بالایی با یکدیگر دارند. در مقابل در دسته پایین، ارتباط T Cell ها با خودشان و B Cell ها با خودشان بسیار زیاد است و همچنین ارتباط AML Patient ها با خودشان و همچنین CD34 ها. به صورت کلی این ارتباطات بالا و در نقطه مقابل عدم ارتباط با دیگر سلول ها، نشان دهنده نمونه برداری مناسب و هماهنگ بودن نمونه ها با یکدیگر است که این موضوع به نوعی جزئی از کنترل کیفیت داده ها محسوب می شود.



**شکل 9**- بررسی میزان همبستگی نمونه ها با یکدیگر از طریق Heatmap.

# **3. بررسی تمایز در بیان ژن‌ها**

به منظور یافتن سلول های سالمی که همبستگی بیشتری با سلول های بیمار دارند، می بایست ماتریس نام سطر ها و ستون های ماتریس بیان را به نام های کلی سلول ها (Source Name آن ها) تغییر داد و سپس از ستون ها تمامی AML Patient ها را حذف نمود و همچنین از سطر ها به جز AML Patient تمامی دیگر سلول ها را حذف نمود. در این حالت در سطر ها فقط AML Patient باقی می ماند و در ستون ها دیگر سلول ها و از این طریق با ماکسیمم گیری سطر ها می توان بیشترین همبستگی ها را معین نمود و مرتب کرد. با توجه به توضیحات بالا، کد مد نظر برای اجرای این عملیات به شرح زیر است:

|  |  |
| --- | --- |
|  | df <- data.frame(ex.scale.cor)  a <- t(df)  colnames(a) <- gset$source\_name\_ch1  a <- t(a)  colnames(a) <- gset$source\_name\_ch1  cols <- gset$source\_name\_ch1  cols <- cols[cols != "AML Patient"]  b <- subset(a, select=cols)  b <- t(b)  cols <- gset$source\_name\_ch1  cols <- cols[cols == "AML Patient"]  b <- subset(b, select=cols)  b <- t(b)  bB <- b  maxCor <- data.frame(Genes = unique(colnames(bB)[max.col(bB,ties.method="first")]),  CorWithAML = unique(rowMax(bB)))  for (x in 1:4) {  cols <- colnames(bB)  cols <- cols[cols != maxCor[[1]][x]]  bB <- subset(bB, select=cols)  maxCor <- rbind(maxCor,  c(gene = unique(colnames(bB)[max.col(bB,ties.method="first")]),  cor = unique(rowMax(bB))))  } |

پس از اجرای کد بالا، خروجی یک جدول خواهد بود که بر اساس نام سلول ها و میزان همبستگی آن ها با AML Patient مرتب سازی شده است. خروجی جدول1 قابل مشاهده است.

|  |  |  |
| --- | --- | --- |
|  | نوع سلول | میزان همبستگی با AML Patient (عدد بین -1 تا 1) |
| 1 | Monocytes | 0.31974 |
| 2 | CD34+HSPC | 0.19717 |
| 3 | B Cells | 0.02744 |
| 4 | Granulocytes | 0.02437- |
| 5 | T Cells | 0.06474- |

**جدول 1**- مرتب سازی میزان همبستگی هرکدام از سلول ها با AML Patient. میزان همبستگی بین -1 تا 1 می باشد و هرچه به 1 نزدیک تر باشد نشان دهنده همبستگی بیشتر است.

به منظور بررسی میزان تمایز بیان ژن ها ابتدا می بایست داده ها را مشخص کنیم که از کدام گروه هستند و سپس یک مدل خطی به آن ها فیت کنیم. این مدل خطی با استفاده از پکیج limma می باشد و بسیاری از تفاوت ها بین نمونه ها را مشخص می کند. پس از آن باید مشخص شود که تصمیم بر این است که تفاوت میان کدام گروه ها به دست آید و در نهایت یک مدل بیز به آن نسبت می دهد. براساس خروجی های این مدل، جدول میزان تمایز بیان ژن ها براساس آماره B که مبتنی بر مدل های فیت شده از طریق پکیج limma به دست می آید مرتب سازی می کند. همچنین برای بررسی عدم اتفاق افتادن خطا های نوع اول و دوم، از روش بنجامینی هاچبرگ[[3]](#footnote-3) استفاده می شود. جدول خروجی با توجه به داده های اولیه ای که وجود دارد شامل ستون های متعددی است که انواع کد های ژن و توضیحات آن را شامل می شود. به منظور ساده سازی جدول، صرفا بعضی از این پارامتر ها نگهداری می شود تا جدول ساده شود و سپس آن ها در یک فایل ذخیره سازی می شوند.

پس از این می بایست ژن هایی که در نمونه اولیه نسبت به نمونه دوم بیان بالایی داشته و همچنین به صورت عکس در نمونه دوم نسبت به نمونه اول بیان بالایی داشته است را به دست آورد. برای این منظور با توجه به حد تفاوت معنی دار 0.05 که برای adj.P.Val در نظر گرفته می شود، علاوه بر این می بایست محدودیت دیگری روی logFC گذاشته شود تا این تفاوت را پوشش دهد یعنی بالاتر بودن بیان ویا پایین تر بودن بیان در نقطه مقابل، که برای این حد نیز عدد 1 و در نقطه مقابل عدد -1 در نظر گرفته شده است. به صورت کلی برای ژن هایی که در نمونه AML Patient بیان بیشتری دارند اصطلاح up را به کار می بریم و برای ژن هایی که بیان کم یا صفر دارند down را به کار می بریم. این دو گونه ژن، عامل های اصلی ایجاد بیماری خواهند بود که در قسمت های بعد مورد بررسی قرار خواهند گرفت.

با توجه به جدول1، با سه مدل مختلف عملیات بالا انجام شده است تا خروجی های مختلف را مقایسه کنیم. ابتدا بین کلیه مدل های test یعنی AML Patient و تمامی مدل های سالم بررسی صورت گرفته است و ژن هایی که بیان بالایی داشته یا بیان نداشته اند به دست آمده اند. سپس به ترتیب بین AML Patient و Monocytes و سپس بین AML Patient و CD34+HSPC مقایسه صورت گرفته است.

برای مقایسه کلی بین سلول های بیمار و سالم:

|  |  |
| --- | --- |
|  | ##### based on all test-normal #####  design <- model.matrix(~group + 0, gset)  colnames(design) <- levels(gs)  ## fitting linear model to data  fit <- lmFit(gset, design)  cont.matrix <- makeContrasts(test-normal, levels=design)  fit2 <- contrasts.fit(fit, cont.matrix)  fit2 <- eBayes(fit2, 0.01)  ## adjust by: fault-discovery-rate or Benjamini-hochberg  ## sort by adj.P.Val  tT <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)  tT <- subset(tT, select=c("Gene.symbol", "Gene.ID","adj.P.Val","logFC", "B"))  write.table(tT, "result/dea/dea\_test-normal\_B.txt", row.names=F, sep="\t", quote=F)  ### Top Gene Expression mining  aml.up <- subset(tT, logFC > 1 & adj.P.Val > 0.05)  aml.up.genes <- unique(as.character(strsplit2(unique(aml.up$Gene.symbol), "///")))  write.table(aml.up.genes, "result/dea/dea\_test-normal\_Up.txt",  quote=F, row.names=F, col.names=F)  aml.down <- subset(tT, logFC < -1 & adj.P.Val > 0.05)  aml.down.genes <- unique(as.character(strsplit2(unique(aml.down$Gene.symbol), "///")))  write.table(aml.down.genes, "result/dea/dea\_test-normal\_Down.txt",  quote=F, row.names=F, col.names=F) |

برای مقایسه بین نمونه های بیمار و به صورت مجزا با هر نوع سلول، که برای دو نوع Monocytes و CD34+HSPC که میزان همبستگی بیشتری با نمونه های بیمار داشتند، صورت گرفته است:

|  |  |
| --- | --- |
|  | ##### based on top correlated cells #####  design <- model.matrix(~source\_name\_ch1 + 0, gset)  sfl <- factor(sname.gs)  colnames(design) <- levels(sfl)  ## fitting linear model to data  fit <- lmFit(gset, design)  ###### AMLPatient-Monocytes ######  cont.matrix <- makeContrasts(AMLPatient-Monocytes, levels=design)  fit2 <- contrasts.fit(fit, cont.matrix)  fit2 <- eBayes(fit2, 0.01)  ## adjust by: fault-discovery-rate or Benjamini-hochberg  ## sort by adj.P.Val  tT <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)  tT <- subset(tT, select=c("Gene.symbol", "Gene.ID","adj.P.Val","logFC", "B"))  write.table(tT, "result/dea/dea\_AMLPatient-Monocytes\_B.txt",  row.names=F, sep="\t", quote=F)  ### Top Gene Expression mining  aml.up <- subset(tT, logFC > 1 & adj.P.Val > 0.05)  aml.up.genes <- unique(as.character(strsplit2(unique(aml.up$Gene.symbol), "///")))  write.table(aml.up.genes, "result/dea/dea\_AMLPatient-Monocytes\_Up.txt",  quote=F, row.names=F, col.names=F)  aml.down <- subset(tT, logFC < -1 & adj.P.Val > 0.05)  aml.down.genes <- unique(as.character(strsplit2(unique(aml.down$Gene.symbol), "///")))  write.table(aml.down.genes, "result/dea/dea\_AMLPatient-Monocytes\_Down.txt",  quote=F, row.names=F, col.names=F)  ###### AMLPatient-CD34+HSPC ######  cont.matrix <- makeContrasts(AMLPatient-CD34pHSPC, levels=design)  fit2 <- contrasts.fit(fit, cont.matrix)  fit2 <- eBayes(fit2, 0.01)  ## adjust by: fault-discovery-rate or Benjamini-hochberg  ## sort by adj.P.Val  tT <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)  tT <- subset(tT, select=c("Gene.symbol", "Gene.ID","adj.P.Val","logFC", "B"))  write.table(tT, "result/dea/dea\_AMLPatient-CD34pHSPC\_B.txt",  row.names=F, sep="\t", quote=F)  ### Top Gene Expression mining  aml.up <- subset(tT, logFC > 1 & adj.P.Val > 0.05)  aml.up.genes <- unique(as.character(strsplit2(unique(aml.up$Gene.symbol), "///")))  write.table(aml.up.genes, "result/dea/dea\_AMLPatient-CD34pHSPC\_Up.txt",  quote=F, row.names=F, col.names=F)  aml.down <- subset(tT, logFC < -1 & adj.P.Val > 0.05)  aml.down.genes <- unique(as.character(strsplit2(unique(aml.down$Gene.symbol), "///")))  write.table(aml.down.genes, "result/dea/dea\_AMLPatient-CD34pHSPC\_Down.txt",  quote=F, row.names=F, col.names=F) |

خروجی قطعه کد های بالا در فایل های مجزا در دایرکتوری result/dea/ ذخیره شده است.

# **4. آنالیز anthology gene و pathway ها**

1. Q[uantile Normalization](https://www.biostars.org/p/296992/) [↑](#footnote-ref-1)
2. PCA: Principal Component Analysis [↑](#footnote-ref-2)
3. Benjamini-hochberg [↑](#footnote-ref-3)