Prinipal component analysis

2023-11-05

#This script performs the following steps:

#1.Loads the required libraries. #2.Reads RNA-seq data and transposes it. #3. Conducts PCA on the transposed data. #4. Visualizes the PCA results in several ways: basic scatter plot, scatter plot with ellipses, and scatter plot with sample labels. #5.Exports the final PCA plot as both PNG and PDF files for sharing or publication.

#adapted on the following tutorial: https://bioinformatics.ccr.cancer.gov/docs/data-visualization-with-r/Lesson3_plotcustomization/

Load necessary libraries for plotting and data manipulation

```
library(ggrepel)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Read the data file from an RNA-seq analysis into a matrix

```
mat<-read.csv("/Users/jeffreyreina/Documents/Salk/RNAseq MDA-MB-231 results/03.Result_X202SC23073852-Z0
```

Transpose the matrix to make samples the rows and genes the columns

```
tranmat <- t(mat)
```

Perform Principal Component Analysis (PCA) on the transposed matrix

```
pca <- prcomp(tranmat, scale = TRUE)
```

Optional: View the structure of the PCA object to understand its components

```
str(pca)
```

```
## List of 5
##  $ sdev    : num [1:6] 75.4 52.4 34.4 28.1 25.4 ...
##  $ rotation: num [1:11056, 1:6] -0.0124 0.0127 0.013 0.0131 -0.0132 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11056] "PMEPA1" "LHPP" "LY6E" "CEMIP" ...
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:11056] 59.48 1.33 73.47 22.8 14.19 ...
##   ..- attr(*, "names")= chr [1:11056] "PMEPA1" "LHPP" "LY6E" "CEMIP" ...
##  $ scale   : Named num [1:11056] 46.71 1.08 55.58 15.93 14.47 ...
##   ..- attr(*, "names")= chr [1:11056] "PMEPA1" "LHPP" "LY6E" "CEMIP" ...
##  $ x       : num [1:6, 1:6] -92.2 -45.3 -64.4 61.1 64.3 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:6] "KO1" "KO2" "KO3" "WT1" ...
##   .. ..$ : chr [1:6] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

Convert the first two principal components (PC1 and PC2) into a data frame for plotting

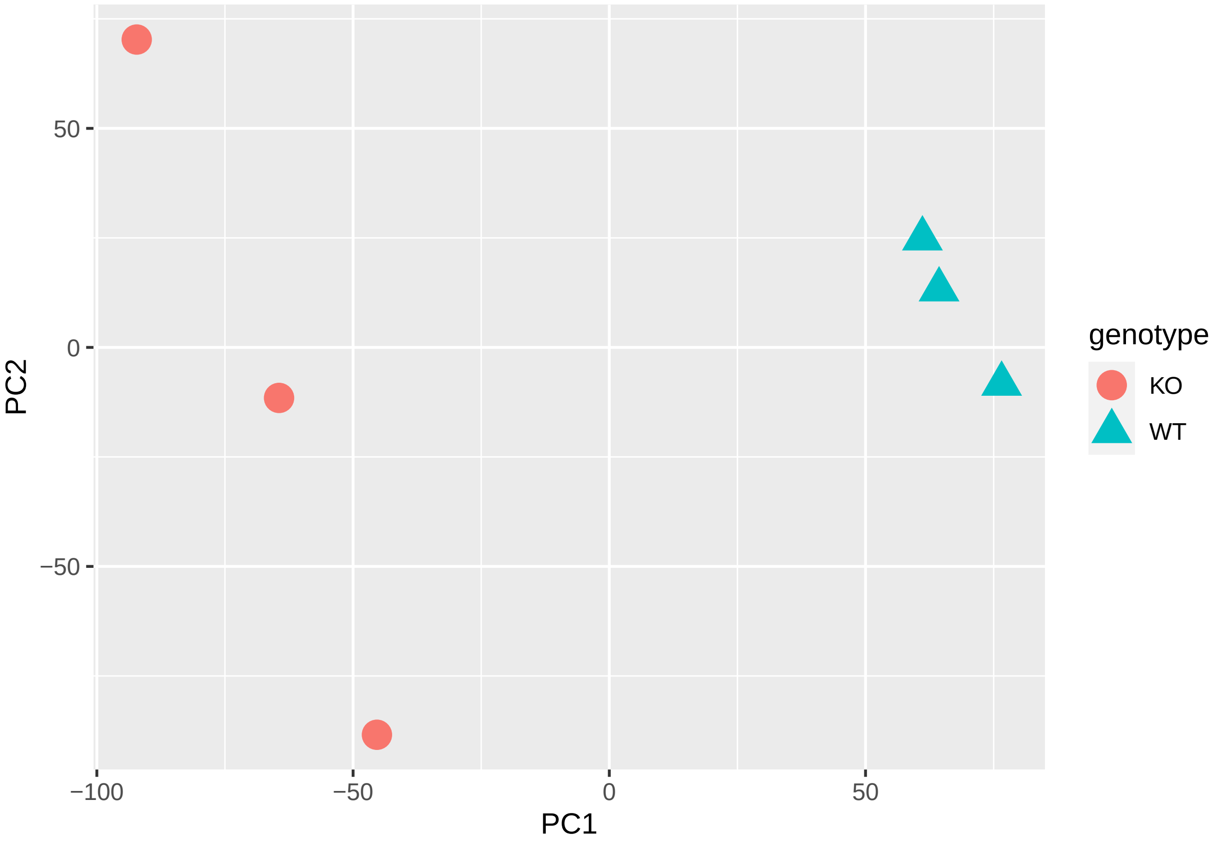
```
pcaData <- as.data.frame(pca$x[, 1:2])
```

Add a genotype column to the data frame to label samples by genotype (KO or WT)

```
pcaData$genotype <- c("KO", "KO", "KO", "WT", "WT", "WT")
```

Basic scatter plot of PC1 vs PC2, with points colored and shaped by genotype

```
ggplot(pcaData) + aes(PC1, PC2, color=genotype, shape=genotype) + geom_point(size = 5)
```

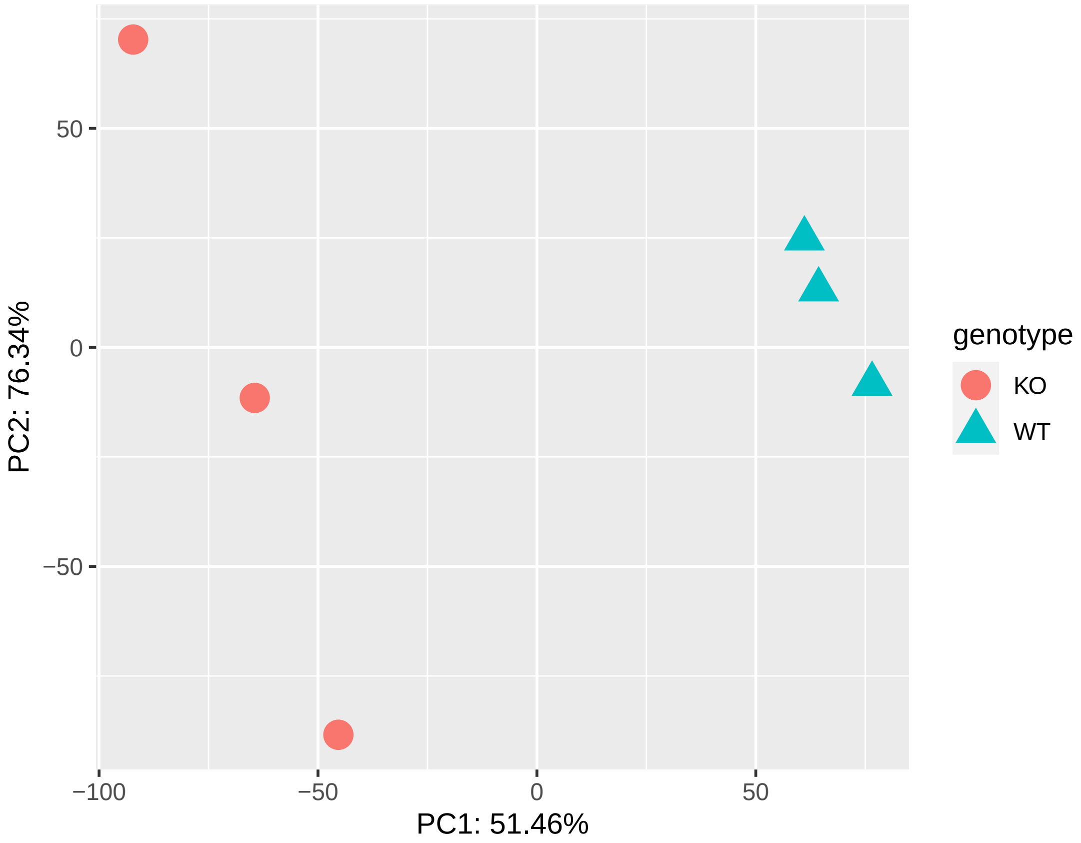


```r
# Extract the variance explained by each principal component
# Useful for understanding the importance of each PC in representing the data
summary(pca)
```

```
## Importance of components:
##                            PC1     PC2     PC3      PC4      PC5       PC6
## Standard deviation     75.4280 52.4438 34.4172 28.06535 25.37834 1.662e-13
## Proportion of Variance  0.5146  0.2488  0.1071  0.07124  0.05825 0.000e+00
## Cumulative Proportion   0.5146  0.7634  0.8705  0.94175  1.00000 1.000e+00
```

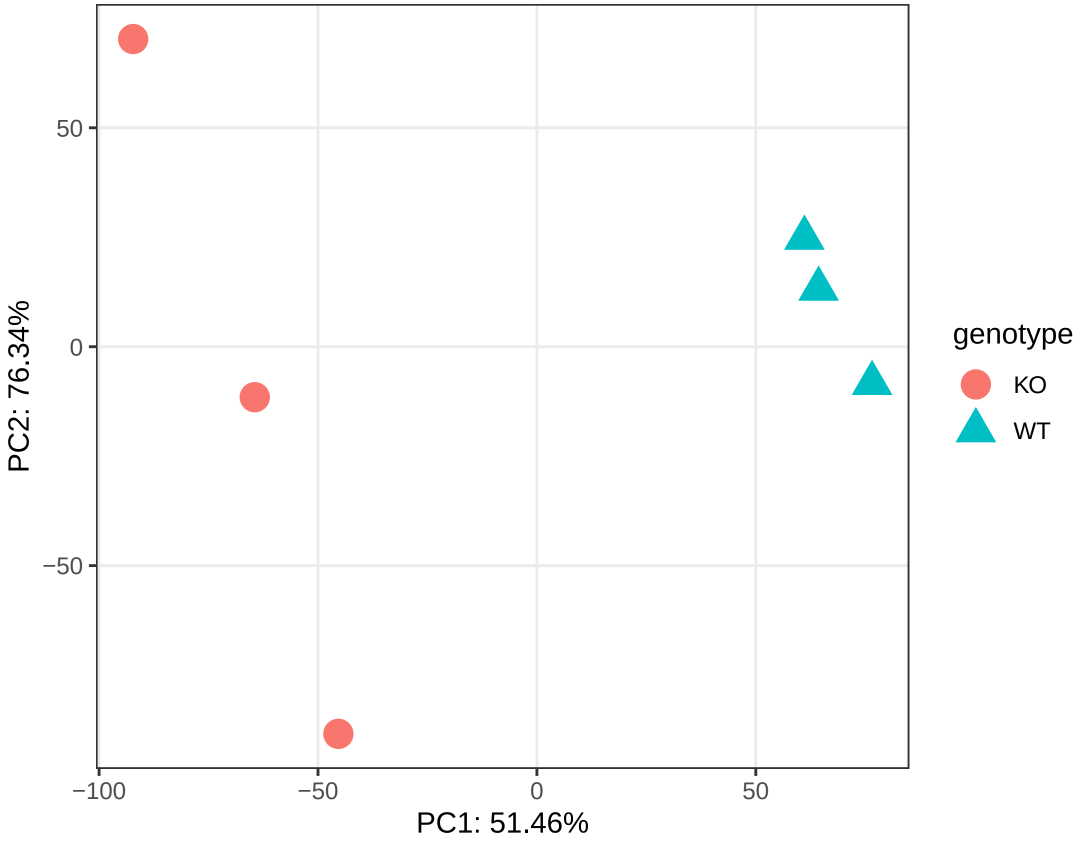
Improved scatter plot of PC1 vs PC2 with fixed aspect ratio and labeled axes

```r
ggplot(pcaData) + aes(PC1, PC2, color = genotype, shape = genotype) +
  geom_point(size = 5) +
  coord_fixed() +
  xlab("PC1: 51.46%")+ # Label for the x-axis, showing variance explained
  ylab("PC2: 76.34%")  # Label for the y-axis, showing variance explained
```



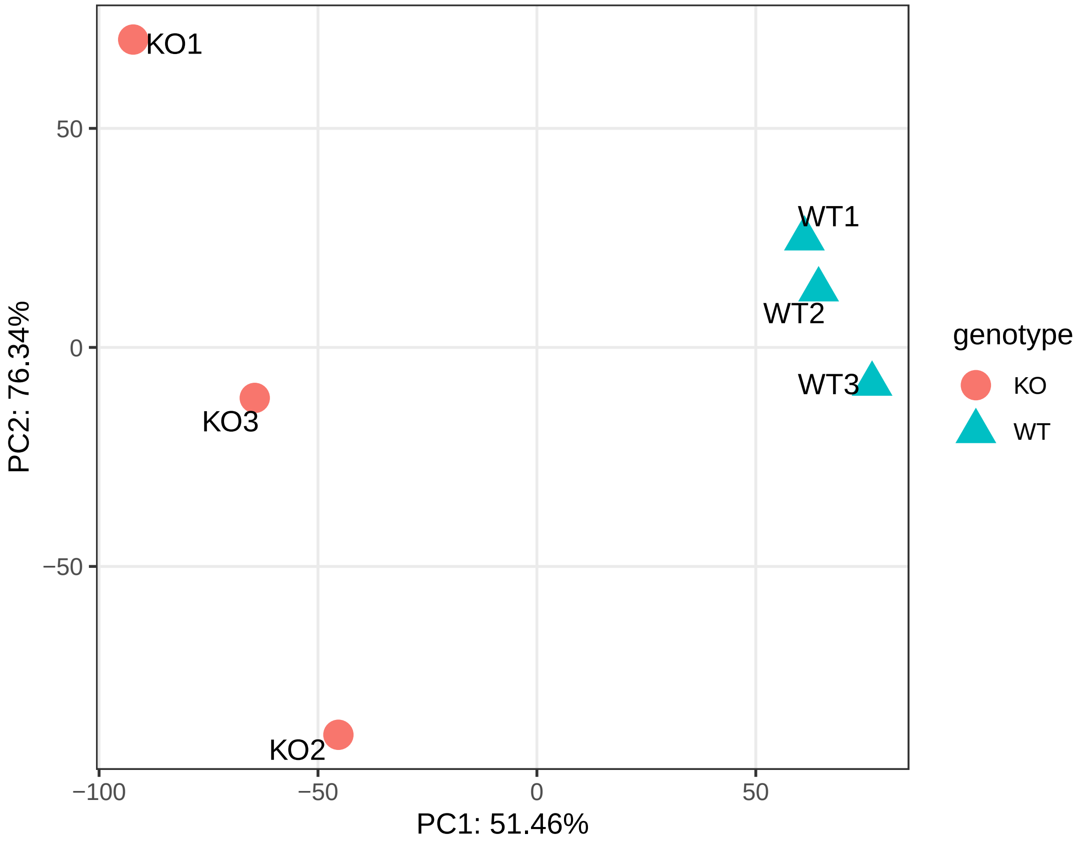
Remove minor grid lines from the plot for a cleaner look

```
ggplot(pcaData) + aes(PC1, PC2, color = genotype, shape = genotype) +
  geom_point(size = 5) +
  coord_fixed() +
  xlab("PC1: 51.46%")+
  ylab("PC2: 76.34%")+
  theme_bw() +
  theme(panel.grid.minor = element_blank())
```



Add sample labels to points with ggplot2, using ggrepel to avoid label overlap

```
ggplot(pcaData, aes(PC1, PC2, color = genotype, shape = genotype)) +
  geom_point(size = 5) +
  coord_fixed() +
  xlab("PC1: 51.46%")+
  ylab("PC2: 76.34%")+
  theme_bw() +
  theme(panel.grid.minor = element_blank()) +
  geom_text_repel(aes(label=rownames(pcaData)), col="black")
```



Save the plot as a PNG file

```
png("PCAWTKO.png", res=300, width=7, height=4.5, unit="in")

ggplot(pcaData, aes(PC1, PC2, color = genotype, shape = genotype)) +
  geom_point(size = 5) +
  coord_fixed() +
  xlab("PC1: 51.46%")+ #x axis label text
  ylab("PC2: 76.34%")+ # y axis label text
  theme_bw() +
  theme(panel.grid.minor = element_blank()) +
  geom_text_repel(aes(label=rownames(pcaData)), col="black")
```

Save the plot as a PDF file

```
pdf("PCAWTKO.pdf", width=7, height=4.5)

ggplot(pcaData, aes(PC1, PC2, color = genotype, shape = genotype)) +
  geom_point(size = 5) +
  coord_fixed() +
  xlab("PC1: 51.46%")+ #x axis label text
  ylab("PC2: 76.34%")+ # y axis label text
  theme_bw() +
  theme(panel.grid.minor = element_blank()) +
  geom_text_repel(aes(label=rownames(pcaData)), col="black")
dev.off
```

```
## function (which = dev.cur())
## {
##     if (which == 1)
##         stop("cannot shut down device 1 (the null device)")
##     .External(C_devoff, as.integer(which))
##     dev.cur()
## }
## <bytecode: 0x1383d06a8>
## <environment: namespace:grDevices>
```

“‘