Untitled

2025-05-02

Read the data and inspect the dimension

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
seCount <-  readRDS(file = "Common_pan_cancer_hyper_bins_adjusted_and_normalized_cnt_in_SE.RDS")
dim(seCount)
```

```
## [1] 24418   378
```

```
## Compute Scale of individual sample metric
varList <- apply(seCount, MARGIN = 2, FUN  = var) %>% unlist()
rangeList <- apply(seCount, MARGIN = 2,
                   FUN  = function(x) {
                     paste0("[", paste(round(range(x), 2), collapse = ","), "]")
                   }) %>% unlist()

scaleSmry <- data.frame(
  "var" = varList,
  "range" = rangeList
)

head(scaleSmry)
```

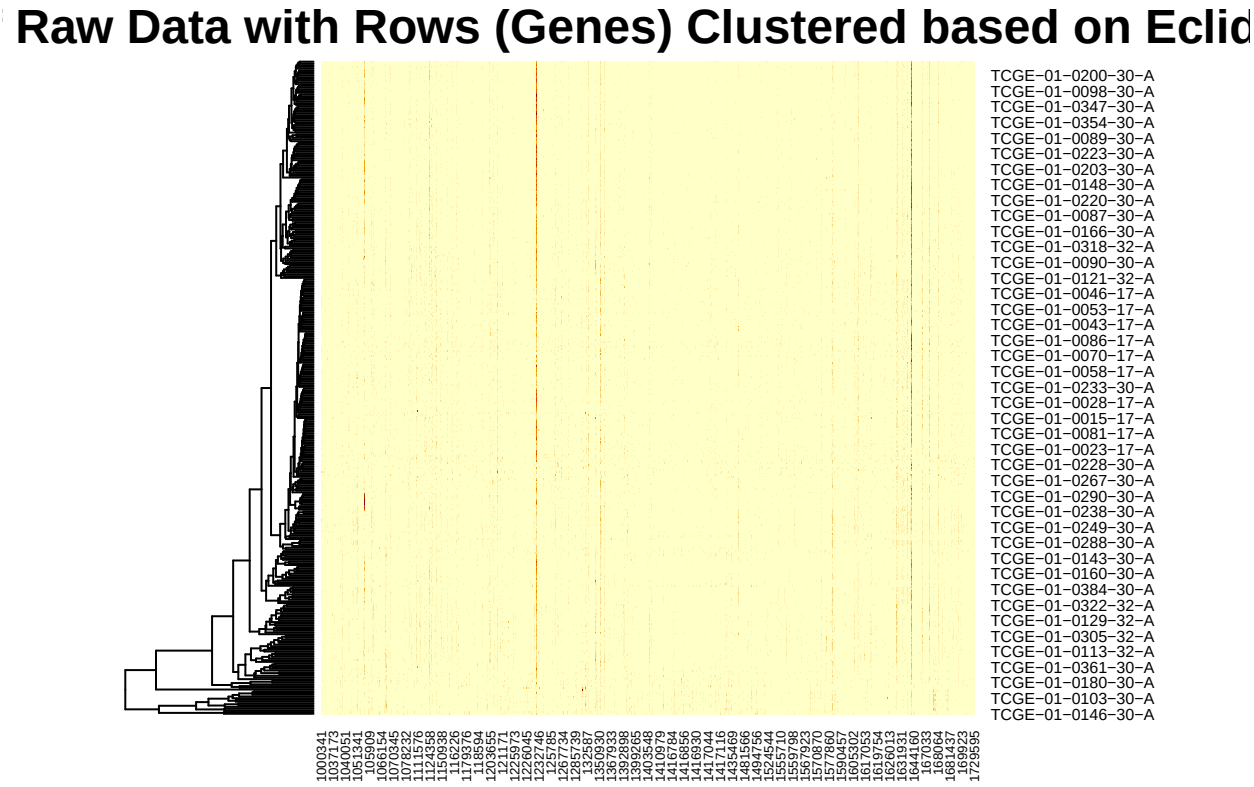
```
##                         var     range
## TCGE-01-0112-32-A 16.43653 [0,81.69]
```

```
## TCGE-01-0121-32-A 16.17406 [0,81.94]
## TCGE-01-0122-32-A 22.70644 [0,78.83]
## TCGE-01-0123-32-A 38.56015 [0,83.01]
## TCGE-01-0124-32-A 10.18328 [0,82.82]
## TCGE-01-0125-32-A 28.22627 [0,75.72]
```

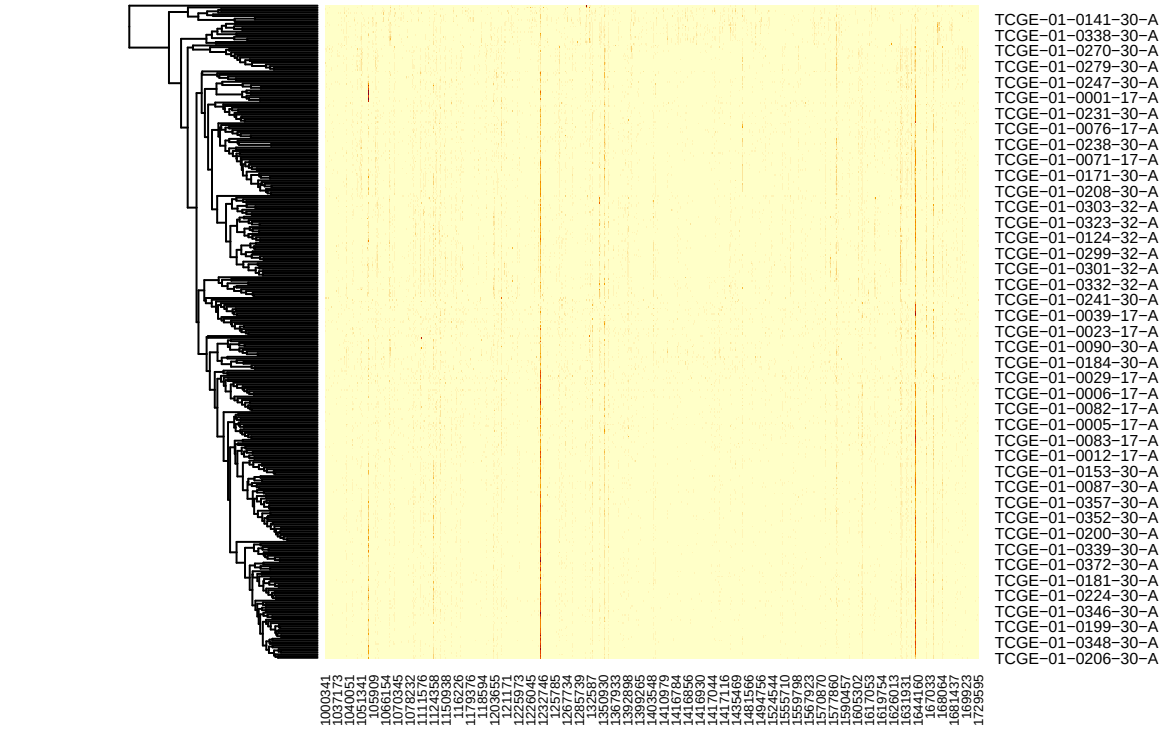
Gene methylation values typically range from 0-80, but the scale (variances) of gene methylation differ across samples. It would be beneficial to re-scale each sample.

Scaling Methylation Data by Individuals

```
heatmap(t(seCount[1:2000,]), Colv = NA,
        main = "Heatmap of Raw Data with Rows (Genes) Clustered based on Eclidean Distance")
```




```
X.Scaled <- scale(seCount, scale = T) %>% t()
heatmap(X.Scaled[, 1:2000], Colv = NA,
        main = "Heatmap of Scaled  Data with Rows (Genes) Clustered based on Eclidean Distance")
```



Sample Data

```
library(tidyverse)
sampleInfo <- read.csv("Common_pan_cancer_hyper_bins_adjusted_cnt_in_SE_samples_info.csv")

sampleInfo %>% group_by(sample_type, vial) %>% summarise("Count" = n()) %>% head()
```

```
## `summarise()` has grouped output by 'sample_type'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3 x 3
## # Groups:   sample_type [3]
##   sample_type vial  Count
##         <int> <chr> <int>
## 1          17 A        85
## 2          30 A       235
## 3          32 A        58
```

```
sampleInfo %>% group_by(cancer_type) %>% summarise("Count" = n())
```

```
## # A tibble: 8 x 2
##   cancer_type       Count
```

```
##   <chr>             <int>
## 1 Bladder Cancer       19
## 2 Blood Cancer         58
## 3 Breast Cancer        25
## 4 Colorectal Cancer    23
## 5 Lung Cancer          77
## 6 Normal               85
## 7 Pancreatic Cancer    71
## 8 Renal Cancer         20
```

```
cancerTypes <- sampleInfo$cancer_type[sampleInfo$sample_id %in% colnames(seCount)]
```

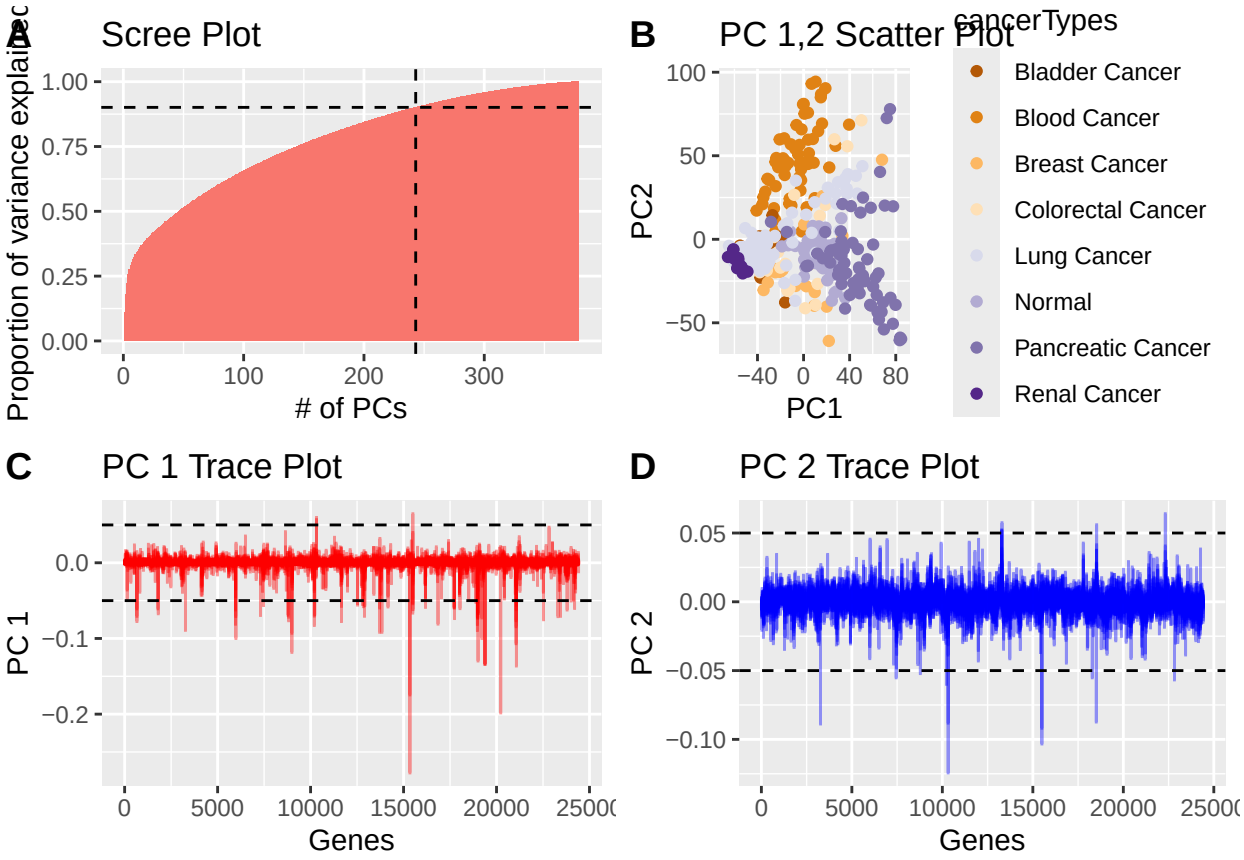
PCA on the Gene Methylation

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
## Centering X
X <- scale(X.Scaled, center = T, scale = F)

PCA <- prcomp(X,  scale = F, rank. = min(dim(X)))

## Compute Scree Plot
totalVar <- sum(PCA$sdev^2)
cumVarExpl <- cumsum(PCA$sdev^2)/totalVar
minPC90 <- which.max(cumVarExpl > 0.9)  # Minimum number of PC that explains 90% of variances
percPCmin <- cumVarExpl[minPC90]
scree <- data.frame(
  "k" = 1:min(dim(seCount)),
  "perc" = cumVarExpl
)

A <- ggplot(scree) +
  geom_bar(aes(x = k, y = perc, fill = adjustcolor("red", alpha = 0.4)), stat = "identity")+
  geom_hline(linetype = "dashed", yintercept = percPCmin) +
  geom_vline(linetype = "dashed", xintercept = minPC90) +
  xlab("# of PCs") +
  ylab("Proportion of variance explained")+
  ggtitle("Scree Plot") +
  theme(legend.position = "none")

## Compute PC1 vs PC 2 Scatter Plot
Z2 <- X %*% PCA$rotation[,1:2]
Z2 <- data.frame(Z2)
```

```r
colnames(Z2) <- c("PC1", "PC2")

Z2$cancerType <- cancerTypes
B <- ggplot(Z2) +
  geom_point(aes(x = PC1, y = PC2, color = cancerTypes))+
  ggtitle("PC 1,2 Scatter Plot")+
  scale_color_brewer(palette = "PuOr")

## PC 1 and PC 2 Trace Plot
V2 <- data.frame(PCA$rotation[,1:2])
colnames(V2) <- c("PC1", "PC2")
V2$k <- 1:dim(V2)[1]

C <- ggplot(V2) +
  geom_bar(aes(x = k, y = PC1), color = adjustcolor("red", alpha = 0.4),  stat = "identity")+
  geom_hline(linetype = "dashed", yintercept = 0.05) +
  geom_hline(linetype = "dashed", yintercept = -0.05) +
  xlab("Genes") +
  ylab("PC 1") +
  theme(legend.position = "none")+
  ggtitle("PC 1 Trace Plot")

D <- ggplot(V2) +
  geom_bar(aes(x = k, y = PC2), color = adjustcolor("blue", alpha = 0.4), stat = "identity")+
  geom_hline(linetype = "dashed", yintercept = 0.05) +
  geom_hline(linetype = "dashed", yintercept = -0.05) +
  xlab("Genes") +
  ylab("PC 2") +
  theme(legend.position = "none") +
  ggtitle("PC 2 Trace Plot")

cowplot::plot_grid(plotlist = list(A,B,C,D),
                   nrow = 2, ncol = 2,
                   labels = c("A", "B", "C", "D"))
```



Observations:

1. Around 140 PCs explain 90% of variances
2. In the space spanned by PC1 and PC2, points are mixed together near 0. Outlier tend to be Lung cancers and colorectal cancers.
3. After sample scaling correction, not all PC 1 components are negative

```
# The proportion of PC 1 components exceed 0.05
mean(abs(PCA$rotation[,1]) > 0.05) *100
```

```
## [1] 0.3112458
```

```
mean(abs(PCA$rotation[,2]) > 0.05) *100
```

```
## [1] 0.06962077
```

```
mean(abs(PCA$rotation[,3]) > 0.05) *100
```

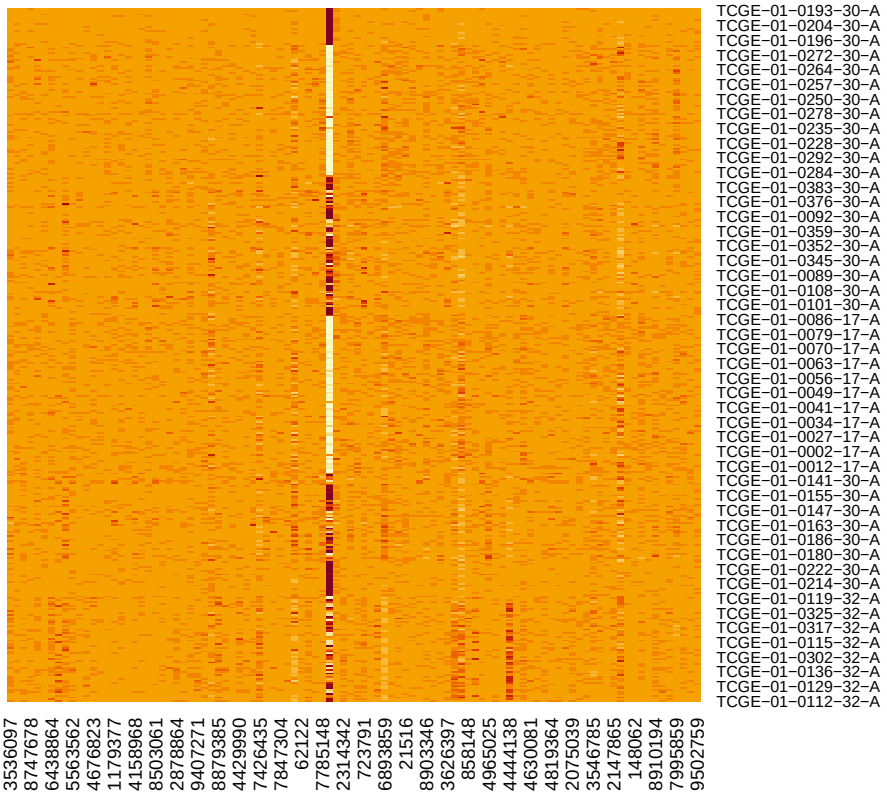
```
## [1] 0.08190679
```

```
mean(abs(PCA$rotation[,4]) > 0.05) *100
```

```
## [1] 0.05323941
```

Around 0.31% of PC 1 components significantly differ from 0. This proportion plummets to 0.05 for the 4th PC as the order of the PC increases.

```
## Filter out genes with top 100 PC1 values
orderGenes <- order(abs(V2[,1]))
indexGenes <- match(1:100, orderGenes)
impGenes <- colnames(X.Scaled)[indexGenes]

heatmap(X[,impGenes], Rowv = NA, Colv = NA)
```



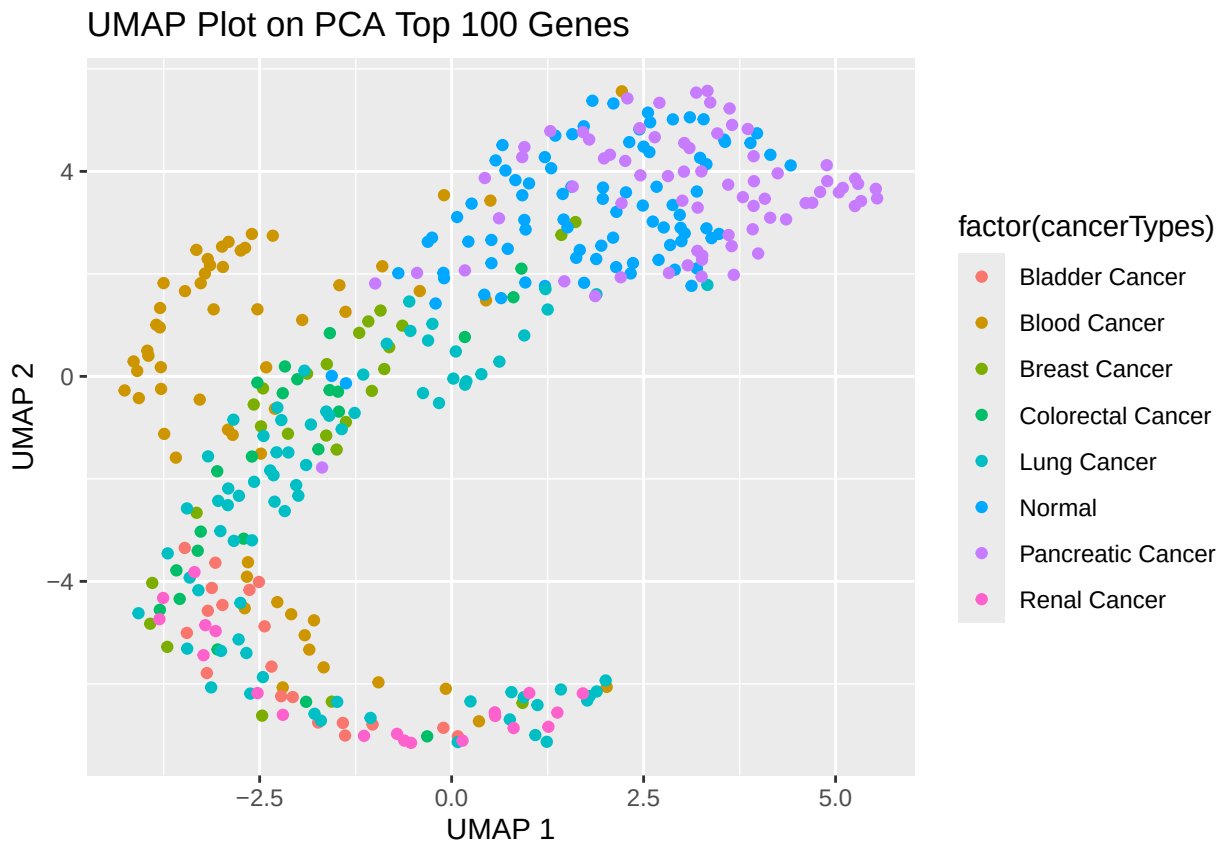
UMAP on the Selected important genes

```
X_imp <- X[,impGenes]

library(umap)
```

```
## Warning: package 'umap' was built under R version 4.4.3
```

```
custom.config <- umap.defaults
custom.config$n_neighbors = 15
custom.config$min_dist = 0.5

UMAP <- umap(X_imp, config = custom.config)
X_hat <- data.frame(UMAP$layout)
colnames(X_hat) <- c("PC1", "PC2")
X_hat$cancerTypes <- cancerTypes
```

```r
ggplot(data = X_hat) +
    geom_point(aes(x = PC1, y = PC2, color = factor(cancerTypes)))+
    labs(x = "UMAP 1",
         y = "UMAP 2",
         title = "UMAP Plot on PCA Top 100 Genes")
```



Sparse PCA

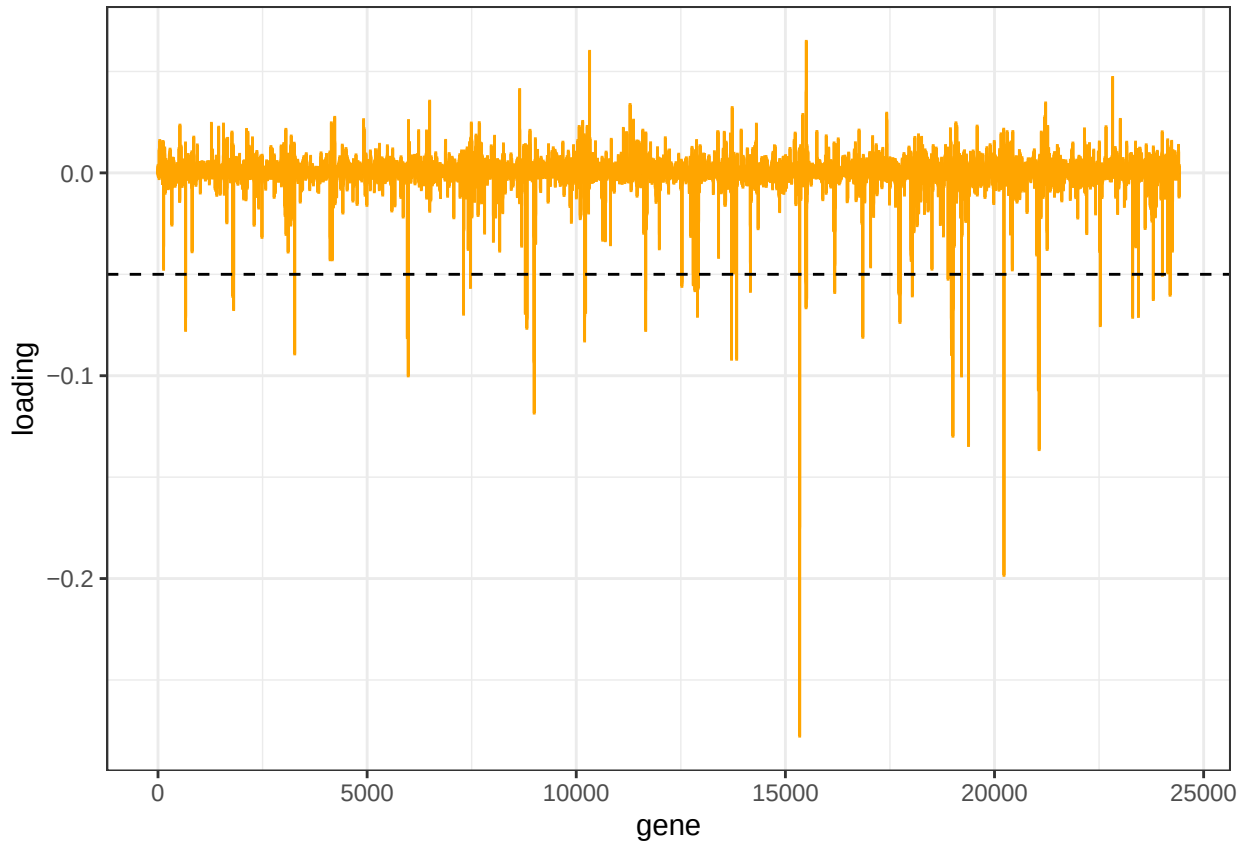
```r
library(PMA)

FrobeniusNorm <- function(X, Xhat) {
  sum((X - Xhat)^2)
}
#
# X <- t(seCount) ## gene as columns, samples as rows
# C_min = 11
# C_max = 100
#
# tuningSmry <- data.frame(
#   "C" = c(),
#   "Frobenius Norm" = c(),
#   "Time" = c()
# )
```

```r
# for (C in seq(C_min, C_max, length.out = 10)) {
#   startTime <- Sys.time()
#   SPCA <- SPC(X, sumabsv = C, K = 2, orth = T, center = T, niter = 100)
#   endTime <- Sys.time()
#   diffTime <- endTime - startTime
#
#   Xhat <- SPCA$u %*% diag(SPCA$d) %*% t(SPCA$v)
#   Fnorm <- FrobeniusNorm(X, Xhat)
#
#   tuningSmry = rbind(tuningSmry, c(C, Fnorm, diffTime))
# }
#
# colnames(tuningSmry) <- c("C", "Frobenius Norm", "Time")
# ggplot(tuningSmry, aes(C, `Frobenius Norm`)) +
#   geom_line(color = "orange")+
#   theme_bw()+
#   theme(legend.position = "none")
```

```r
## Pick the Cost using the elbow method
## Pick C that result in 10, 100,
# C_optimal <- tuningSmry$C[which.min(tuningSmry$`Frobenius Norm`)]
C_optimal <- 60
SPCA <- SPC(X, sumabsv = C_optimal, K = 10, center = T, orth = T, niter = 100)
```

```
## 1
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444
## 12345678910111213141516171819202122232425262728293031323334353637383940414243
## 12345678910111213141516171819202122232425262728293031323334353637383940
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505051525354
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455
## 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455
```

```r
df = data.frame("loading" = SPCA$v[,1], "gene" = 1:dim(X)[2])

ggplot(df, aes(gene, loading)) +
  geom_bar(stat = "identity", color = "orange")+
  theme_bw()+
  geom_hline(linetype = "dashed", yintercept = -0.05) +
  theme(legend.position = "none")
```




```
# The proportion of PC 1 components exceed 0.05
V10 <- SPCA$v[, 1:10]
mean(abs(V10[,1]) > 0.05) *100
```

```
## [1] 0.3112458
```

```
mean(abs(V10[,2]) > 0.05) *100
```

```
## [1] 0.1720043
```

```
mean(abs(V10[,3]) > 0.05) *100
```

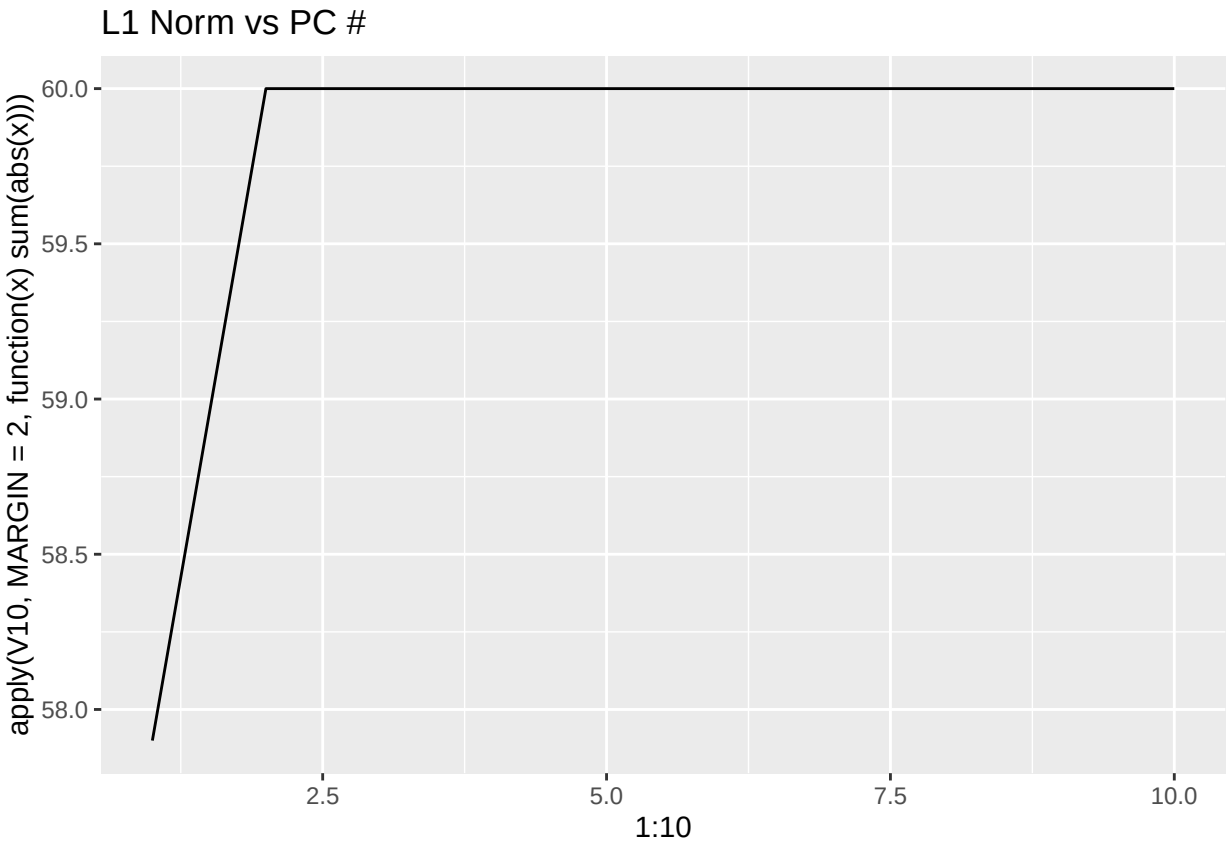
```
## [1] 0.1638136
```

```
mean(abs(V10[,4]) > 0.05) *100
```

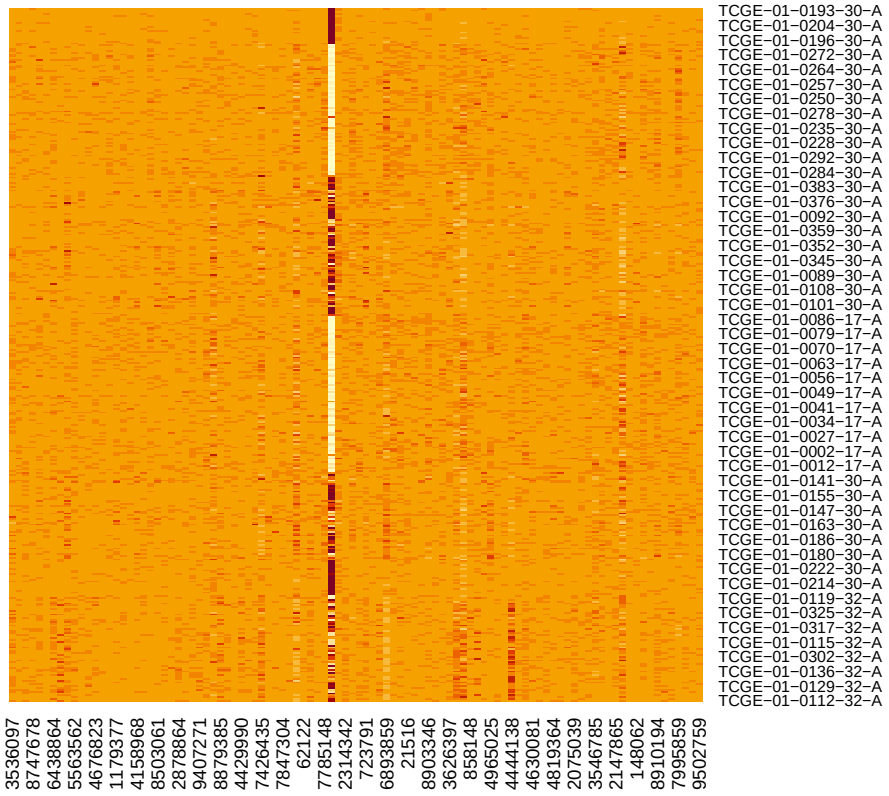
```
## [1] 0.1556229
```

```
# Compare L1 Norm of the first 10 PCs
qplot(x = 1:10, y = apply(V10, MARGIN = 2, function(x) sum(abs(x))),
      geom = "line", main = "L1 Norm vs PC #")
```

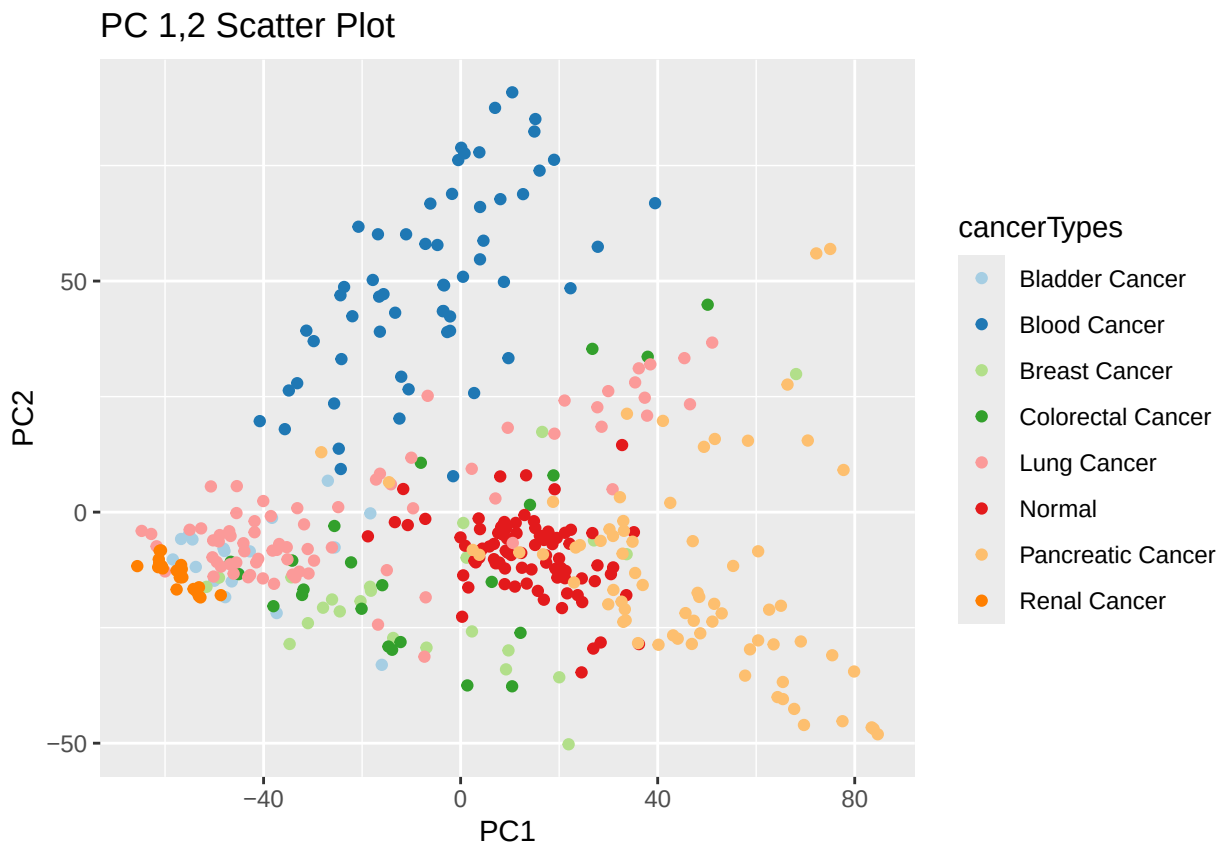
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
## important genes
orderGenes <- order(abs(V10[,1]))
indexGenes <- match(1:100, orderGenes)
impGenes <- colnames(X.Scaled)[indexGenes]

heatmap(X[,impGenes], Rowv = NA, Colv = NA)
```

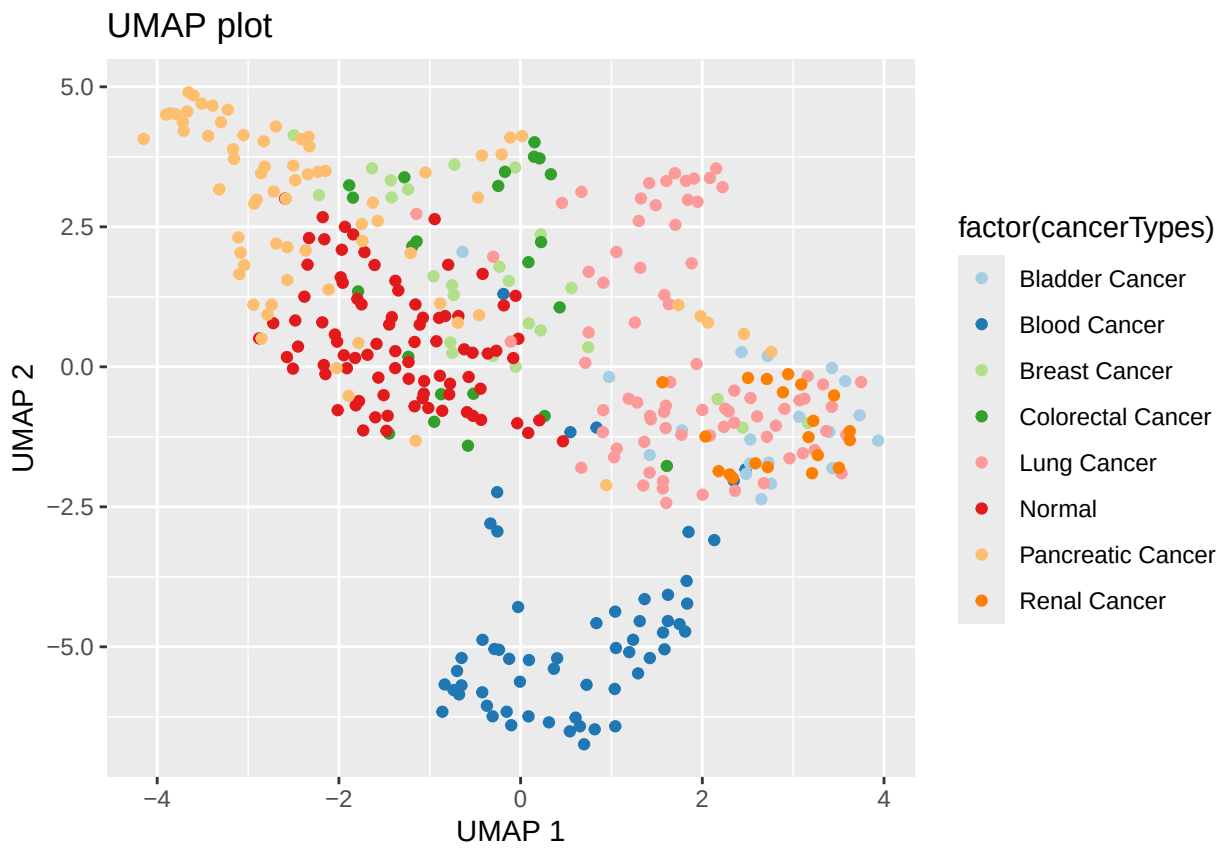



```r
Z2 <- X %*% SPCA$v[,1:2]
Z2 <- data.frame(Z2)
colnames(Z2) <- c("PC1", "PC2")

Z2$cancerType <- cancerTypes

ggplot(Z2) +
  geom_point(aes(x = PC1, y = PC2, color = cancerTypes))+
  xlab("PC1") + ylab("PC2") +
  ggtitle("PC 1,2 Scatter Plot")+
  scale_color_brewer(palette = "Paired")
```



UMAP on the Sparse PCA

```
X_imp <- X[,impGenes]
custom.config <- umap.defaults
custom.config$n_neighbors = 15
custom.config$min_dist = 0.5

UMAP <- umap(X, config = custom.config)
X_hat <- data.frame(UMAP$layout)
colnames(X_hat) <- c("PC1", "PC2")
X_hat$cancerTypes <- cancerTypes

ggplot(data = X_hat) +
    geom_point(aes(x = PC1, y = PC2, color = factor(cancerTypes)))+
    labs(x = "UMAP 1",
         y = "UMAP 2",
         title = "UMAP plot")+
  scale_color_brewer(palette = "Paired")
```



Observations: 1. Blood cancer, Pancreatic cancer seem to be very separated from other types of cancers. 2. Breast cancer, colorectal cancer, renal cancer seem to be highly similar.