Análsiis de datos RNAseq: Análisis de Componentes Principales (PCA)

El objetivo de nuestro análisis es el siguiente: **analizar el perfil trancripcional de los macrófagos después de un oncoentrenamiento en el microambiente tumoral del cáncer de mama.**

Para ello, hemos estado trabajando con datos RNAseq bulk de tipo pareados. Por cuestiones administrativas tenemos datos de dos lotes de secuenciación independientes. El primer lote (Lote 1, *TruSeq RNA sample prep v2 LT*) tiene una calidad adecuada, todas nuestras lecturas (100pb) mostraban una puntuación Phred score mayor a 34 y con un promedio de 28.7 millones de lecturas por biblioteca. Sin embargo, nuestra preocupación viene del segundo lote (lote 2 *NovaSeq de Illumina*), cuyas lecturas (150pb) no tienen la misma calidad, en especial las R2. La puntuación Phred para estas lecturas cae hasta 20. En tanto, el R1, cae hasta 22 en la escala Phred. El promedio de lecturas por biblioteca es de 22.7 millones.

Ante el escenario mencionado decidimos hacer *trimming* sobre el lote 2, las recortamos a 100 pb, así elevamos la puntuación Phred 27 y 21, para la lectura R1&R2, respectivamente, conservando las 22.7 millones de read en promedio por biblioteca.

Nuestro siguiente paso fue alinear las secuencias de forma independiente con STAR. Para el lote 1 obtuvimos un porcentaje de alineamiento único del 92% en promedio para todas las muestras, lo cual nos generó mucha alegría. En tanto el lote 2, obtuvimos un porcentaje de alineamineto único del 83%, lo cual nos tiene un poco consternados. **Consideramos que la calidad de secuenciación del lote 2 es menor a la obtenida en el lote 1.**

Posterior al alineamiento, ensamblamos y cuantificamos con Stringtie. Aquí te mostramos el flujo de trabajo:

El asunto en todo es: ¿los datos provenientes del lote 2 son confiables? ¿el segundo lote de secuenciacióne es viable? ¿puedo homogenizar ambos lotes de secuenciación? Lo que pretendemos con los siguientes análisis es homogenizar los datos de secuenación RNAseq, provenientes de dos eventos distintos de secuenciación, para poder hacer comparativas adecuadas. Para ello haremos un PCA para observar el comportamiento, agrupación de los datos y encontrar posibles efectos por lote u otros artefactos. En caso de existir dicho efecto debemos solucionarlo y verificar su resolución mediante otro PCA, en el cual la distribución de los datos debería ser homogena.

La intención de este análisis es poder homegenizar ambos lotes de secuenciacíon, considerando que la calidad del lote 1 es mayor a la del lote 2, respecto a las lecturas obtenidas.

Lo primero que necesitamos es instalar las librerías necesarias para realizar el PCA:

```
# Librerias
#install.packages("ggplot2")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
#install.packages("ggfortify")
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.3.3
```

A)

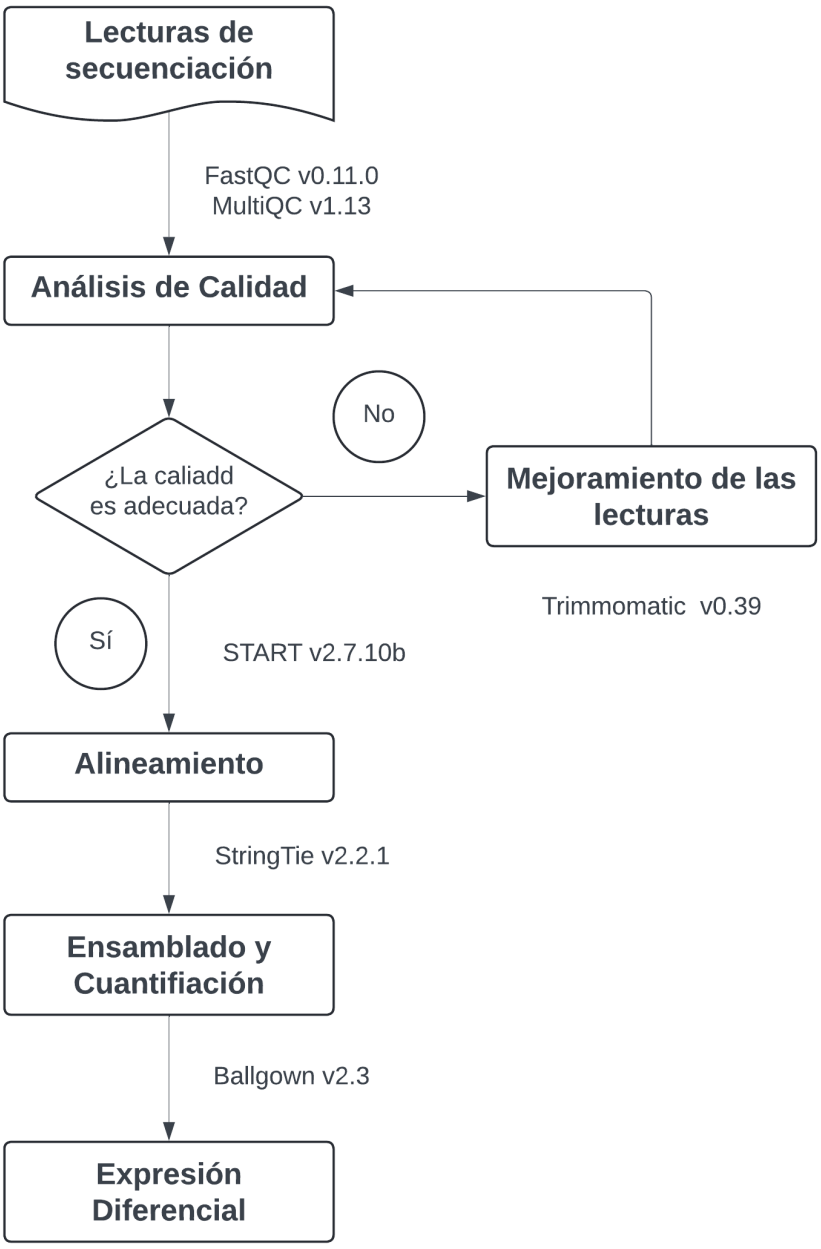


Figure 1: Esquema de nuestro flujo de trabajo

Después es **indispensable establecer nuestro directorio de trabajo**. Aquí deben estar todos los archivos y datos de entrada. Además, será el sitio en el cuál se depositen las salidas, es decir, los resultados de nuestros análisis. Para el PCA vamos a trabajar con los datos de expresión normalizados en FPKM.

```
# Cargar datos
setwd("D:/marval_windows/JR_MARVAL/himfg/maestria/rnaseq_macrophage/DEA_ballgown_5_all_samples/batch/ba
list.files()
```

```
##  [1] "batch_pyjn.ipynb"
##  [2] "batch_pyjn_function.ipynb"
##  [3] "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv"
##  [4] "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv"
##  [5] "fpkm_all_samples_with_genes_wiso_mean_L1.csv"
##  [6] "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv"
##  [7] "fpkm_macs_with_genes_wiso_mean_L1.csv"
##  [8] "fpkm_macs_with_genes_wiso_mean_L1_median.csv"
##  [9] "fpkm_without_gmcsf_with_genes_wiso_mean_L1.csv"
## [10] "heatmap_all_data.R"
## [11] "rna_pca_batc.html"
## [12] "rna_pca_batc.Rmd"
## [13] "work_flow_transcriptome.png"
```

```
data <- read.table(file = "fpkm_all_samples_with_genes_wiso_mean_L1.csv", sep = ",", head=T, row.names =
head(data)
```

```
##             basal_1   basal_2  basal_3  basal_4  basal_5   basal_6  gmcsf_1
## 5_8S_rRNA 0.1742263 0.3309187 0.262425 0.454116 0.382549 0.3626667 3.827592
## A1BG      2.0267650 2.6587460 2.512201 1.548792 2.591755 2.9539550 1.622624
## AAAS      4.6538210 5.8562810 3.204630 6.189765 3.186074 3.5462180 6.643039
## AACS      2.5528370 2.4409970 2.550747 2.657844 2.614494 2.6225150 7.577627
## AAGAB     5.1936000 5.3058360 4.881932 4.786989 4.699411 4.7646865 5.892926
## AAK1      2.9256310 0.0897400 2.770806 1.833814 0.393437 0.4958700 0.852122
##            gmcsf_2  gmcsf_3  gmcsf_4  gmcsf_5   gmcsf_6   mcf7_1   mcf7_2
## 5_8S_rRNA 3.246799 0.212196 0.153785 0.446328 0.2653463 1.732242 1.253446
## A1BG      2.754114 1.344875 2.509614 1.064880 1.5209920 1.164606 2.312324
## AAAS      4.173436 3.675864 6.076991 3.776479 3.7661020 3.980284 6.873029
## AACS      7.647595 7.777211 7.228054 7.330215 6.9851020 4.456494 4.528874
## AAGAB     6.082545 6.050280 6.211525 6.203869 6.4772185 5.068387 4.905913
## AAK1      4.088208 1.624996 0.000000 1.509110 0.0835230 0.346149 2.536555
##             mcf7_3    mcf7_4   mcf7_5   mcf7_6 mda231_1  mda231_2  mda231_3
## 5_8S_rRNA 0.501903 0.5663673 9.051003 9.242948 0.397877 0.2071323 0.6310303
## A1BG      1.958059 1.6275070 5.057527 4.469102 3.316178 4.1614470 5.7911540
## AAAS      6.895456 4.4412290 4.731468 4.724869 3.021399 6.4236060 4.8378600
## AACS      4.018071 4.7185950 4.599300 4.384636 3.456069 3.6143200 3.7622600
## AAGAB     5.151965 5.1348330 4.879101 5.158355 4.818155 4.2662295 4.2388970
## AAK1      0.592022 0.2991000 0.201282 0.266572 0.153466 0.0865630 0.0000000
##            mda231_4  mda231_5 mda231_6    t47d_1    t47d_2   t47d_3    t47d_4
## 5_8S_rRNA 0.5091887 0.8305067 1.022664 0.5078863 0.6955683 0.446838 0.5035913
## A1BG      2.3004480 1.4757640 3.907411 1.3217070 2.4534700 1.058542 0.9737510
## AAAS      2.1481840 5.0204500 2.584269 9.8211310 4.9702260 9.173691 6.0950070
## AACS      3.8246210 3.9647230 3.442420 5.4080130 5.1156640 4.954972 4.9466850
## AAGAB     4.3786485 4.4533655 4.585009 4.9692575 5.3032865 5.571677 5.3863905
## AAK1      4.3496650 0.1920900 0.147194 0.1964440 1.2598560 3.532730 0.2679930
```

```
##              t47d_5   t47d_6   uivc1_1  uivc1_2   uivc1_3  uivc1_4  uivc1_5
## 5_8S_rRNA 0.3971687 0.340262 0.4795253 0.424720 0.5909317 1.788434 1.996841
## A1BG      3.4223130 1.954777 2.0496470 1.278879 1.2388840 3.848390 2.711705
## AAAS      4.3793060 5.467438 4.9289260 3.664525 4.9935900 3.335820 2.542631
## AACS      4.3822240 4.168936 5.0057980 5.007914 5.2681230 5.366636 5.299304
## AAGAB     4.9697050 5.328748 5.8999955 5.137998 5.1787385 5.020115 5.632012
## AAK1      0.9257080 0.268473 0.1062120 0.107368 0.0730990 0.628346 3.264310
##             uivc1_6  uivc4_1  uivc4_2   uivc4_3  uivc4_4  uivc4_5   uivc4_6
## 5_8S_rRNA 0.4192673 0.227280 0.574254 0.2967593 0.375872 0.304031 0.3513897
## A1BG      0.7515560 1.238688 1.809417 2.3650460 3.161502 3.819606 2.6536350
## AAAS      4.9246090 3.523808 4.118847 3.8321880 5.181192 4.855953 3.3672790
## AACS      4.9766960 4.208766 4.144949 3.5441990 3.327990 4.107236 3.9547270
## AAGAB     5.6806400 4.669164 4.283131 4.6562500 4.723536 4.646779 4.5638065
## AAK1      0.0000000 0.000000 1.690735 0.3168510 0.000000 1.081493 0.2646560
```

Ahora debemos trabajar un poco los datos. Primero habrá que remover los genes que no se encuentren expresados en ninguna de las condiciones. Para ello, si la suma de la fila es igual a 0, entonces se elimina dicha fila (en transcriptoma no hay problema con eliminar transcritos). También debemos transponer el dataframe, para que sea compatible con la librería.

```
# Eliminar filas cuya suma sea 0
data <- data[!(rowSums(data[,]) == 0), ]
# Transponer dataframe
df_tras = data.frame(t(data[,]))
```

Ahora seguimos propiamente con el **Análisis de Componentes Principales**. En primer lugar, analizamos los datos del lote 1. Estos datos se ensamblaron y cuantificaron de forma independiente al lote 2. Es importante considerar que parte de los argumentos para contruir el PCA, es centrar y escalar los datos, lo cual afecta la forma en que se comporta la varianza.

```
# PCA
df <- prcomp(df_tras[,c(1:7652)], center = T, scale. = T)
summary(df)
```

```
## Importance of components:
##                            PC1     PC2      PC3      PC4      PC5      PC6
## Standard deviation     54.2571 28.7192 23.07500 20.99764 15.73607 11.07281
## Proportion of Variance  0.3847  0.1078  0.06958  0.05762  0.03236  0.01602
## Cumulative Proportion   0.3847  0.4925  0.56209  0.61971  0.65207  0.66809
##                            PC7      PC8     PC9    PC10    PC11   PC12    PC13
## Standard deviation     10.67084 10.34272 9.75256 9.35024 9.24852 9.1335 9.03469
## Proportion of Variance  0.01488  0.01398 0.01243 0.01143 0.01118 0.0109 0.01067
## Cumulative Proportion   0.68297  0.69695 0.70938 0.72080 0.73198 0.7429 0.75355
##                           PC14    PC15    PC16    PC17    PC18    PC19    PC20
## Standard deviation     8.84893 8.83180 8.74509 8.72478 8.67474 8.61934 8.55614
## Proportion of Variance 0.01023 0.01019 0.00999 0.00995 0.00983 0.00971 0.00957
## Cumulative Proportion  0.76378 0.77398 0.78397 0.79392 0.80375 0.81346 0.82303
##                           PC21    PC22    PC23    PC24   PC25    PC26    PC27
## Standard deviation     8.51524 8.49482 8.48752 8.40582 8.3923 8.26995 8.21395
## Proportion of Variance 0.00948 0.00943 0.00941 0.00923 0.0092 0.00894 0.00882
## Cumulative Proportion  0.83251 0.84194 0.85135 0.86059 0.8698 0.87873 0.88754
##                           PC28    PC29    PC30    PC31    PC32    PC33    PC34
## Standard deviation     8.2069 8.18529 8.15101 8.09801 8.03455 8.00685 7.90143
```

```
## Proportion of Variance 0.0088 0.00876 0.00868 0.00857 0.00844 0.00838 0.00816
## Cumulative Proportion  0.8963 0.90510 0.91378 0.92235 0.93079 0.93917 0.94733
##                           PC35   PC36    PC37    PC38    PC39    PC40   PC41
## Standard deviation     7.80510 7.7251 7.62187 7.58711 7.53168 7.46795 7.3684
## Proportion of Variance 0.00796 0.0078 0.00759 0.00752 0.00741 0.00729 0.0071
## Cumulative Proportion  0.95529 0.9631 0.97068 0.97820 0.98562 0.99290 1.0000
##                             PC42
## Standard deviation     3.391e-14
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

El resultado del PCA podemos guardarlo como un dataframe:

```
# Realizar el análisis de componentes principales
pca_result <- prcomp(df_tras[, c(1:7652)], center = T, scale. = T)
# Crear un data frame con los resultados del PCA
pca_data <- as.data.frame(pca_result$x)
head(pca_data)
```

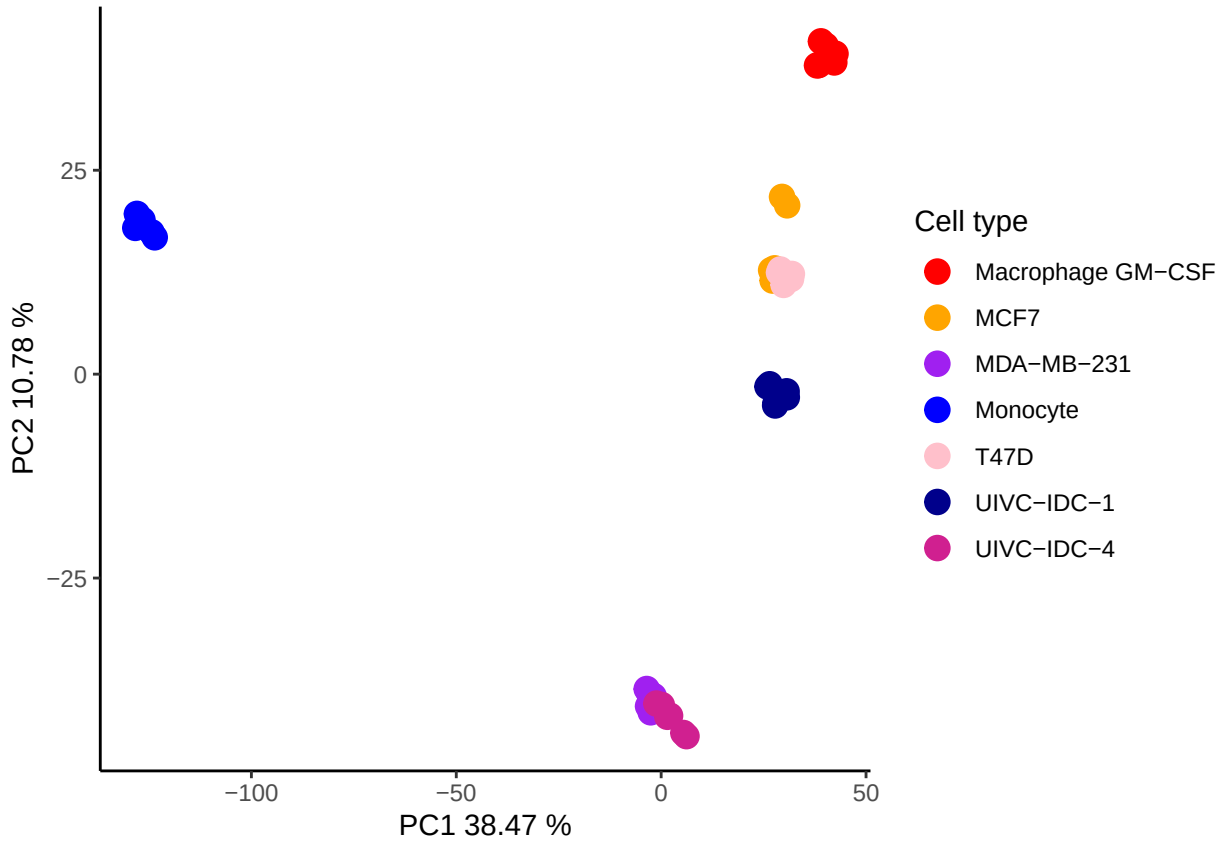
```
##               PC1      PC2       PC3       PC4       PC5       PC6       PC7
## basal_1 -127.9351 19.64761  2.235515  4.327044 -2.026739 -5.066487  3.067181
## basal_2 -126.5887 18.91237  2.088124  4.765684 -2.663429 -5.261878  2.196386
## basal_3 -124.3598 17.39107  4.253869  7.348091 -2.758668 -7.551558  3.179325
## basal_4 -123.5940 16.77512  5.468866  8.503291 -3.623411 -5.342424  2.831882
## basal_5 -127.3771 18.52577 -6.896302 -2.481440  3.931883 12.494017 -8.063154
## basal_6 -128.3339 17.93184 -5.933464 -2.703876  3.996935 12.099387 -4.091782
##                PC8        PC9        PC10       PC11      PC12       PC13
## basal_1 -8.8099672  5.6019370 -12.5806420  33.259217 -9.042866   1.371226
## basal_2 -8.5438619  7.8118337   1.6432622  17.533868 -3.681497  19.612390
## basal_3 -5.0371993 -7.3167283   2.9073631 -20.196501  8.951021  11.723997
## basal_4  0.2594069 -5.7369699  -1.6517610 -17.563640  3.779890  14.727629
## basal_5 10.1105926  2.1413875   0.1710834 -12.513120 -4.199037 -24.101285
## basal_6 10.2767659 -0.4060753   8.5503331  -2.099937  4.430034 -23.115586
##               PC14       PC15      PC16       PC17       PC18       PC19
## basal_1  7.2654981   4.256735 -1.269376  -7.699412 -0.7253641 -17.577164
## basal_2 -0.6897751   7.292905 -6.569837   2.413444  9.3065309  16.422292
## basal_3 -2.0986628  -7.060945  9.004338 -12.385832  5.4291885  -1.461061
## basal_4 -8.6997668 -20.440405  3.930607  17.097467 -4.0011536  -6.264422
## basal_5  0.1045565  10.124739 -3.338014  -4.737484 -5.4274088  18.372661
## basal_6  2.8697589   4.610297 -1.307252   5.339439 -4.9686586  -9.128527
##               PC20       PC21      PC22       PC23       PC24         PC25
## basal_1  12.119479   6.342874  1.491217  5.0721031 -16.624692  -3.93276201
## basal_2 -20.811937  -9.162038 -1.064907 -1.4802722  18.428507  -0.02209895
## basal_3  10.703179  16.694275 -7.810677 11.9764617  -3.931047  13.66968657
## basal_4  -5.145396  -3.978813 10.709041 -8.4315332  -8.519422 -10.28631986
## basal_5   1.306641  18.642051 -1.885976 -6.9334517   6.420448 -10.74699665
## basal_6   2.009497 -28.835712 -1.607300 -0.4694198   3.914182  10.98009428
##              PC26       PC27       PC28       PC29       PC30       PC31
## basal_1 -2.122805  -0.6313936  -4.266664  2.574832  6.9773665   9.609358
## basal_2 -3.605398   3.2642017   7.290682 -6.258738 -4.0927076  -3.502473
## basal_3 -4.929178   6.4734842   2.918069  3.971142 -2.0525191 -17.664479
## basal_4 16.032332  -5.6179189  -1.492536 -2.196187  0.4174916  12.567607
## basal_5 -7.631834 -12.4148435   6.472191 -2.249440 -1.7785214  12.436069
```

```
## basal_6  2.351605   9.2330459 -10.826790  4.537287 -0.1512665 -13.775656
##              PC32       PC33       PC34       PC35       PC36       PC37
## basal_1 -3.125131 -1.9503224 -5.7831643   1.824039  0.2916899  1.975169
## basal_2  7.142368  8.5724510  6.8937634  -6.604875  1.7735011 -1.949015
## basal_3 -3.986914  1.9166356 -4.6802253 -10.570693 -2.0085212 -3.706334
## basal_4  3.439296 -3.8228816  5.1562429   9.153246 -2.0780114  5.663696
## basal_5 -6.382155  0.1093631 -0.6047811  -1.083738 -0.7144105  1.208039
## basal_6  2.696986 -4.7356411 -0.9204846   7.237148  2.6055127 -3.235631
##               PC38       PC39       PC40       PC41          PC42
## basal_1 -2.1295341 -3.8806396  0.4560645 -2.1144916 -2.060505e-14
## basal_2  2.6677472  0.3533988 -1.6732810  1.8125098  4.404203e-14
## basal_3  0.3827317  3.1273450 -1.1186094  6.9695115  1.085156e-14
## basal_4  1.8874715  4.8655599  1.2570267 -3.5013978 -8.545682e-14
## basal_5 -2.6101341 -1.1640338  3.0458331 -3.6092795  5.254721e-15
## basal_6 -0.1087289 -3.3309917 -1.8682330  0.4783924 -1.204228e-13
```

Ahora vamos a representar gráficamente nuestros resultado. Para ello nombraremos nuestras muestras por condición, excluyendo su número de replica para que sea más fácil la unificación.

```r
# Funcion para asignar las etiquetas
assign_label <- function(cond_name) {
  if (grepl("^basal_", cond_name)) {
    return("Monocyte")
  } else if (grepl("^gmcsf_", cond_name)) {
    return("Macrophage GM-CSF")
  } else if (grepl("^uivc_hs", cond_name)) {
    return("HS578T")
  } else if (grepl("^mcf7_", cond_name)) {
    return("MCF7")
  } else if (grepl("^mda231_", cond_name)) {
    return("MDA-MB-231")
  } else if (grepl("^uivc_p16_", cond_name)) {
    return("MBCDF-16")
  } else if (grepl("^t47d_", cond_name)) {
    return("T47D")
  } else if (grepl("^uivc_160_", cond_name)) {
    return("UIVC-IDC-2")
  } else if (grepl("^uivc_169_", cond_name)) {
    return("UIVC-IDC-3")
  } else if (grepl("^uivc_172_", cond_name)) {
    return("UIVC-IDC-1b")
  } else if (grepl("^uivc_183_", cond_name)) {
    return("UIVC-IDC-9")
  } else if (grepl("^uivc1_", cond_name)) {
    return("UIVC-IDC-1")
  } else if (grepl("^uivc4_", cond_name)) {
    return("UIVC-IDC-4")
  } else {
    return("Other")  # Añadir un caso para cualquier otra condición
  }
}


# Agregar la condición desde los nombres de las columnas
```

```r
pca_data$Condicion <- sapply(rownames(df_tras), assign_label)



# Agregar la condición desde los nombres de las columnas
pca_data$Condicion <- sapply(rownames(df_tras), function(cond_name) {
  assign_label(cond_name)
})

# Graficar
ggplot(pca_data, aes(x = PC1, y = PC2, color = Condicion)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2,1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2,2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Cell type",
                     values = c("Monocyte" = "blue",
                                "Macrophage GM-CSF" = "red",
                                "HS578T" = "green",
                                "MCF7" = "orange",
                                "MDA-MB-231" = "purple",
                                "MBCDF-16" = "brown",
                                "T47D" = "pink",
                                "UIVC-IDC-2" = "cyan",
                                "UIVC-IDC-3" = "magenta",
                                "UIVC-IDC-1b" = "yellow",
                                "UIVC-IDC-9" = "salmon",
                                "UIVC-IDC-1" = "darkblue",
                                "UIVC-IDC-4" = "violetred"))
```



Como podemos ver, los dos primeros componentes no explican ni el 50% de la varianza de los datos, esto seguramente porque centramos y escalamos los datos.Sin embargo, no es necesario marcar estos parametros pues todos los datos son valores de expresión que se normalizaron juntos, es decir, son la misma unidad en la misma magnitud, entonces **no es necesario centrar ni escalar los datos.**

```
# PCA
df <- prcomp(df_tras[,c(1:7652)], center = F, scale. = F)
summary(df)
```

```
## Importance of components:
##                             PC1       PC2       PC3       PC4      PC5
## Standard deviation     1.807e+04 2863.3097 1.119e+03 762.48581 5.37e+02
## Proportion of Variance 9.684e-01    0.0243 3.710e-03   0.00172 8.50e-04
## Cumulative Proportion  9.684e-01    0.9927 9.964e-01   0.99818 9.99e-01
##                             PC6       PC7       PC8       PC9      PC10
## Standard deviation     351.02119 263.53192 174.52615 153.14518 104.59928
## Proportion of Variance   0.00037   0.00021   0.00009   0.00007   0.00003
## Cumulative Proportion    0.99940   0.99960   0.99969   0.99976   0.99980
##                            PC11     PC12     PC13     PC14     PC15     PC16
## Standard deviation     98.47392 87.85458 75.55117 67.27279 64.64171 54.53804
## Proportion of Variance  0.00003  0.00002  0.00002  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99982  0.99985  0.99986  0.99988  0.99989  0.99990
##                            PC17     PC18     PC19     PC20     PC21     PC22
## Standard deviation     52.52335 50.48433 48.47566 47.37159 45.17772 41.42564
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99991  0.99991  0.99992  0.99993  0.99993  0.99994
```

```
##                            PC23  PC24  PC25  PC26  PC27 PC28  PC29  PC30 PC31
## Standard deviation     39.7618 39.16 37.32 36.77 36.45 34.2 33.84 33.62 32.9
## Proportion of Variance  0.0000  0.00  0.00  0.00  0.00  0.0  0.00  0.00  0.0
## Cumulative Proportion   0.9999  1.00  1.00  1.00  1.00  1.0  1.00  1.00  1.0
##                            PC32  PC33 PC34  PC35  PC36  PC37 PC38 PC39  PC40  PC41
## Standard deviation     32.05 31.22 29.9 29.43 29.15 28.42 27.8 27.2 26.04 25.32
## Proportion of Variance  0.00  0.00  0.0  0.00  0.00  0.00  0.0  0.0  0.00  0.00
## Cumulative Proportion   1.00  1.00  1.0  1.00  1.00  1.00  1.0  1.0  1.00  1.00
##                            PC42
## Standard deviation     24.94
## Proportion of Variance  0.00
## Cumulative Proportion   1.00
```

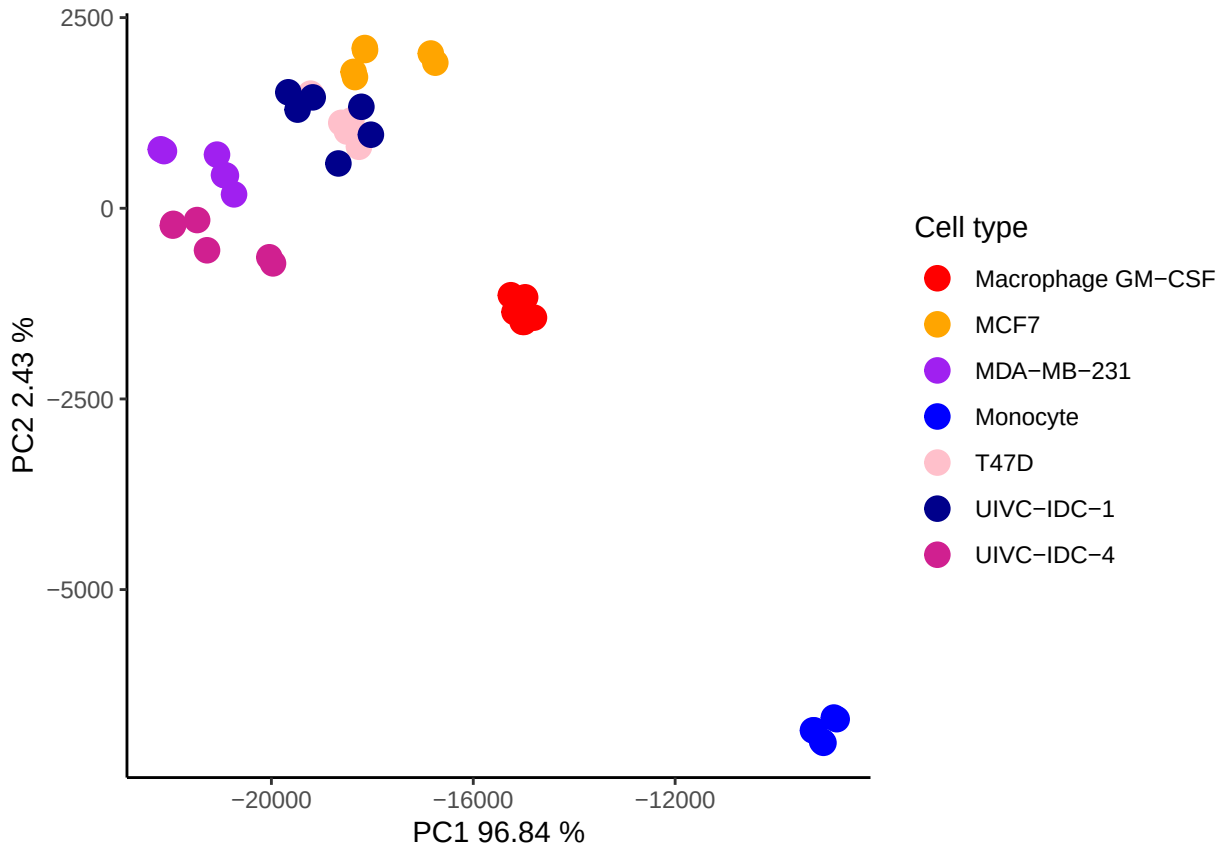
```r
# Realizar el análisis de componentes principales
pca_result <- prcomp(df_tras[, c(1:7652)], center = F, scale. = F)
# Crear un data frame con los resultados del PCA
pca_data <- as.data.frame(pca_result$x)
head(pca_data)
```

```
##                PC1       PC2       PC3       PC4       PC5       PC6       PC7
## basal_1 -8854.673 -6677.818 -990.2740 124.5364 236.38729 197.92817  79.66080
## basal_2 -8798.155 -6699.564 -992.7456 111.7685 222.45279 182.73774  70.36194
## basal_3 -9088.222 -6990.695 -908.6853 156.7312  40.83208  13.95913 -74.08103
## basal_4 -9060.653 -7009.506 -920.8419 161.3632  40.87921  24.43536 -69.78812
## basal_5 -9268.712 -6846.503 -874.1837 215.9163 125.50748  86.87101 -35.66184
## basal_6 -9242.860 -6846.008 -863.8249 198.5315  96.52211  76.82031 -41.39120
##                PC8       PC9       PC10        PC11       PC12       PC13
## basal_1  137.53502  24.61722 -66.771597  160.080230  -52.58346  160.18731
## basal_2  106.19570  68.07961 -88.609077  233.396727 -158.43731   34.34552
## basal_3   80.03376  14.98269  -5.199348 -127.999030  173.62072   34.66484
## basal_4   99.01709 -11.70960   5.656497 -136.087412  134.29535  -24.50145
## basal_5 -227.55322 -53.73361  83.526060 -106.315780   15.62015  -52.60772
## basal_6 -253.14843 -43.32531  83.368832   -6.069872 -102.43212 -145.27604
##               PC14        PC15      PC16      PC17      PC18      PC19
## basal_1 -45.21823  133.619875 -27.07243 105.83547  18.10834  41.13327
## basal_2 -67.72672  -72.158765 -21.57910 -83.99305  45.89184 -32.59013
## basal_3 -86.48875    9.445315  69.23963 -45.68808 -21.60789 -20.61542
## basal_4 -93.79528  -19.845090  79.13754 -63.48616 -70.81794  84.78132
## basal_5 153.71651   65.559676 -28.69696 140.91165  11.45049 -13.82683
## basal_6 141.15394 -120.008100 -68.66167 -54.23871  16.58875 -57.24467
##                PC20        PC21       PC22       PC23      PC24       PC25
## basal_1 -37.523832 -73.1954320  30.200714   2.814236  47.59900   18.55727
## basal_2  14.039732 -10.2342450  25.191826  -5.688183  18.34907   -3.77121
## basal_3  79.764280   0.7527539 -16.114235   3.631964 -10.76021   97.07012
## basal_4   7.683600  20.7802445 -30.459199 -23.825234 -39.00223 -105.35139
## basal_5 -56.189976  32.0294420  -3.092719   5.959375 -21.03597  -26.46368
## basal_6  -8.253236  28.4509414  -6.032395  17.562211   5.73815   21.14734
##                PC26        PC27       PC28       PC29       PC30       PC31
## basal_1  55.89574212  -0.1131289 -25.851336 -21.377350 -14.04225 -24.00845
## basal_2 -51.33807596 -28.7027889  10.285182  52.718788  44.19842  19.57766
## basal_3  17.57995396  -6.6308761   8.010486 -19.019312 -70.50867  46.63596
## basal_4  -0.01471474   7.4885260  10.294426   4.753353  40.09599 -33.57781
## basal_5 -37.01513473  35.9722958 -32.004506 -37.840987  22.43807  14.08329
## basal_6  14.26099689  -7.8758358  29.328611  21.601655 -23.33792 -23.12506
```

```
##               PC32       PC33       PC34       PC35       PC36        PC37
## basal_1  -1.443850  -6.339746  -1.947644  42.529169  18.643865  13.9671464
## basal_2   8.603182  22.417041 -39.444831 -63.778621  -3.123667 -15.0083973
## basal_3  37.999993 -30.637825  21.779685  -5.349617  -4.033257  -0.1808938
## basal_4 -48.415320  26.389907  -6.336288  16.849022   2.265683  -3.0287909
## basal_5  11.952708  48.650991  11.929122 -42.828391 -16.298793 -11.9276487
## basal_6  -9.083045 -60.132503  13.664477  52.082126   2.809521  15.7708625
##                 PC38      PC39       PC40       PC41      PC42
## basal_1   1.084870477  46.19465  -1.502284 -10.991819  -8.40051
## basal_2  11.871355894 -35.66361   1.319744   1.236422  11.26129
## basal_3 -15.723941072 -34.79714 -14.413312  24.119023  18.77877
## basal_4  15.696251815  23.35797  15.332240 -16.523662 -21.01769
## basal_5  -0.007591546 -35.19129  -2.338933  27.811536  17.70644
## basal_6 -13.145414893  36.50419   1.855902 -26.431251 -17.68376
```

```r
# Funcion para asignar las etiquetas
assign_label <- function(cond_name) {
  if (grepl("^basal_", cond_name)) {
    return("Monocyte")
  } else if (grepl("^gmcsf_", cond_name)) {
    return("Macrophage GM-CSF")
  } else if (grepl("^uivc_hs", cond_name)) {
    return("HS578T")
  } else if (grepl("^mcf7_", cond_name)) {
    return("MCF7")
  } else if (grepl("^mda231_", cond_name)) {
    return("MDA-MB-231")
  } else if (grepl("^uivc_p16_", cond_name)) {
    return("MBCDF-16")
  } else if (grepl("^t47d_", cond_name)) {
    return("T47D")
  } else if (grepl("^uivc_160_", cond_name)) {
    return("UIVC-IDC-2")
  } else if (grepl("^uivc_169_", cond_name)) {
    return("UIVC-IDC-3")
  } else if (grepl("^uivc_172_", cond_name)) {
    return("UIVC-IDC-1b")
  } else if (grepl("^uivc_183_", cond_name)) {
    return("UIVC-IDC-9")
  } else if (grepl("^uivc1_", cond_name)) {
    return("UIVC-IDC-1")
  } else if (grepl("^uivc4_", cond_name)) {
    return("UIVC-IDC-4")
  } else {
    return("Other")  # Añadir un caso para cualquier otra condición
  }
}


# Agregar la condición desde los nombres de las columnas
pca_data$Condicion <- sapply(rownames(df_tras), assign_label)
```

```r
# Agregar la condición desde los nombres de las columnas
pca_data$Condicion <- sapply(rownames(df_tras), function(cond_name) {
  assign_label(cond_name)
})

# Graficar
ggplot(pca_data, aes(x = PC1, y = PC2, color = Condicion)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2,1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2,2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Cell type",
                     values = c("Monocyte" = "blue",
                                "Macrophage GM-CSF" = "red",
                                "HS578T" = "green",
                                "MCF7" = "orange",
                                "MDA-MB-231" = "purple",
                                "MBCDF-16" = "brown",
                                "T47D" = "pink",
                                "UIVC-IDC-2" = "cyan",
                                "UIVC-IDC-3" = "magenta",
                                "UIVC-IDC-1b" = "yellow",
                                "UIVC-IDC-9" = "salmon",
                                "UIVC-IDC-1" = "darkblue",
                                "UIVC-IDC-4" = "violetred"))
```



Como podemos observar la varianza explicada por el PC1 es muy alta, más del 90%. Si bien observamos la distribución esperada de las muestras, creo que este porcentaje es muy alto, incluso considerando que estamos trabajando con los monocitos lo cuál es una variable biológica importante a considerar **¿Cómo se ve el análisis si excluimos a los monocitos?**

```
# Cargar datos
setwd("D:/marval_windows/JR_MARVAL/himfg/maestria/rnaseq_macrophage/DEA_ballgown_5_all_samples/batch/ba
list.files()
```

```
##  [1] "batch_pyjn.ipynb"
##  [2] "batch_pyjn_function.ipynb"
##  [3] "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv"
##  [4] "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv"
##  [5] "fpkm_all_samples_with_genes_wiso_mean_L1.csv"
##  [6] "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv"
##  [7] "fpkm_macs_with_genes_wiso_mean_L1.csv"
##  [8] "fpkm_macs_with_genes_wiso_mean_L1_median.csv"
##  [9] "fpkm_without_gmcsf_with_genes_wiso_mean_L1.csv"
## [10] "heatmap_all_data.R"
## [11] "rna_pca_batc.html"
## [12] "rna_pca_batc.Rmd"
## [13] "rna_pca_batc_files"
## [14] "work_flow_transcriptome.png"
```

```
data <- read.table(file = "fpkm_macs_with_genes_wiso_mean_L1.csv", sep = ",", head=T, row.names = 1)
head(data)
```

```
##            gmcsf_1  gmcsf_2  gmcsf_3  gmcsf_4  gmcsf_5   gmcsf_6   mcf7_1
## 5_8S_rRNA 3.827592 3.246799 0.212196 0.153785 0.446328 0.2653463 1.732242
## A1BG      1.622624 2.754114 1.344875 2.509614 1.064880 1.5209920 1.164606
## AAAS      6.643039 4.173436 3.675864 6.076991 3.776479 3.7661020 3.980284
## AACS      7.577627 7.647595 7.777211 7.228054 7.330215 6.9851020 4.456494
## AAGAB     5.892926 6.082545 6.050280 6.211525 6.203869 6.4772185 5.068387
## AAK1      0.852122 4.088208 1.624996 0.000000 1.509110 0.0835230 0.346149
##             mcf7_2   mcf7_3    mcf7_4   mcf7_5   mcf7_6 mda231_1  mda231_2
## 5_8S_rRNA 1.253446 0.501903 0.5663673 9.051003 9.242948 0.397877 0.2071323
## A1BG      2.312324 1.958059 1.6275070 5.057527 4.469102 3.316178 4.1614470
## AAAS      6.873029 6.895456 4.4412290 4.731468 4.724869 3.021399 6.4236060
## AACS      4.528874 4.018071 4.7185950 4.599300 4.384636 3.456069 3.6143200
## AAGAB     4.905913 5.151965 5.1348330 4.879101 5.158355 4.818155 4.2662295
## AAK1      2.536555 0.592022 0.2991000 0.201282 0.266572 0.153466 0.0865630
##            mda231_3  mda231_4  mda231_5 mda231_6    t47d_1    t47d_2   t47d_3
## 5_8S_rRNA 0.6310303 0.5091887 0.8305067 1.022664 0.5078863 0.6955683 0.446838
## A1BG      5.7911540 2.3004480 1.4757640 3.907411 1.3217070 2.4534700 1.058542
## AAAS      4.8378600 2.1481840 5.0204500 2.584269 9.8211310 4.9702260 9.173691
## AACS      3.7622600 3.8246210 3.9647230 3.442420 5.4080130 5.1156640 4.954972
## AAGAB     4.2388970 4.3786485 4.4533655 4.585009 4.9692575 5.3032865 5.571677
## AAK1      0.0000000 4.3496650 0.1920900 0.147194 0.1964440 1.2598560 3.532730
##             t47d_4    t47d_5   t47d_6   uivc1_1  uivc1_2   uivc1_3  uivc1_4
## 5_8S_rRNA 0.5035913 0.3971687 0.340262 0.4795253 0.424720 0.5909317 1.788434
## A1BG      0.9737510 3.4223130 1.954777 2.0496470 1.278879 1.2388840 3.848390
## AAAS      6.0950070 4.3793060 5.467438 4.9289260 3.664525 4.9935900 3.335820
## AACS      4.9466850 4.3822240 4.168936 5.0057980 5.007914 5.2681230 5.366636
```

```
## AAGAB     5.3863905 4.9697050 5.328748 5.8999955 5.137998 5.1787385 5.020115
## AAK1      0.2679930 0.9257080 0.268473 0.1062120 0.107368 0.0730990 0.628346
##             uivc1_5   uivc1_6  uivc4_1  uivc4_2   uivc4_3  uivc4_4  uivc4_5
## 5_8S_rRNA 1.996841 0.4192673 0.227280 0.574254 0.2967593 0.375872 0.304031
## A1BG      2.711705 0.7515560 1.238688 1.809417 2.3650460 3.161502 3.819606
## AAAS      2.542631 4.9246090 3.523808 4.118847 3.8321880 5.181192 4.855953
## AACS      5.299304 4.9766960 4.208766 4.144949 3.5441990 3.327990 4.107236
## AAGAB     5.632012 5.6806400 4.669164 4.283131 4.6562500 4.723536 4.646779
## AAK1      3.264310 0.0000000 0.000000 1.690735 0.3168510 0.000000 1.081493
##             uivc4_6
## 5_8S_rRNA 0.3513897
## A1BG      2.6536350
## AAAS      3.3672790
## AACS      3.9547270
## AAGAB     4.5638065
## AAK1      0.2646560
```

```
# Eliminar filas cuya suma sea 0
data <- data[!(rowSums(data[,]) == 0), ]
# Transponer dataframe
df_tras = data.frame(t(data[,]))
```

```
# PCA
df <- prcomp(df_tras[,c(1:7640)], center = F, scale. = F)
summary(df)
```

```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     1.921e+04 1.559e+03 846.38484 622.89817 438.66696
## Proportion of Variance 9.894e-01 6.510e-03   0.00192   0.00104   0.00052
## Cumulative Proportion  9.894e-01 9.959e-01   0.99786   0.99890   0.99941
##                             PC6       PC7       PC8       PC9      PC10
## Standard deviation     285.34339 182.82987 164.80592 113.71123 100.41110
## Proportion of Variance   0.00022   0.00009   0.00007   0.00003   0.00003
## Cumulative Proportion    0.99963   0.99972   0.99980   0.99983   0.99986
##                            PC11     PC12     PC13     PC14     PC15     PC16
## Standard deviation     81.04241 73.10545 66.49509 58.84027 55.01264 52.91723
## Proportion of Variance  0.00002  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99987  0.99989  0.99990  0.99991  0.99992  0.99993
##                            PC17     PC18     PC19     PC20  PC21  PC22  PC23
## Standard deviation     51.47128 47.51466 44.66063 43.33484 43.07 40.29 39.04
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00  0.00  0.00
## Cumulative Proportion   0.99993  0.99994  0.99994  0.99995  1.00  1.00  1.00
##                            PC24  PC25  PC26  PC27  PC28  PC29  PC30  PC31 PC32
## Standard deviation     38.53 37.07 35.89 35.27 34.09 33.79 31.77 31.08 30.8
## Proportion of Variance  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.0
## Cumulative Proportion   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.0
##                            PC33  PC34  PC35  PC36
## Standard deviation     30.36 28.67 28.13 27.47
## Proportion of Variance  0.00  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.00  1.00  1.00
```

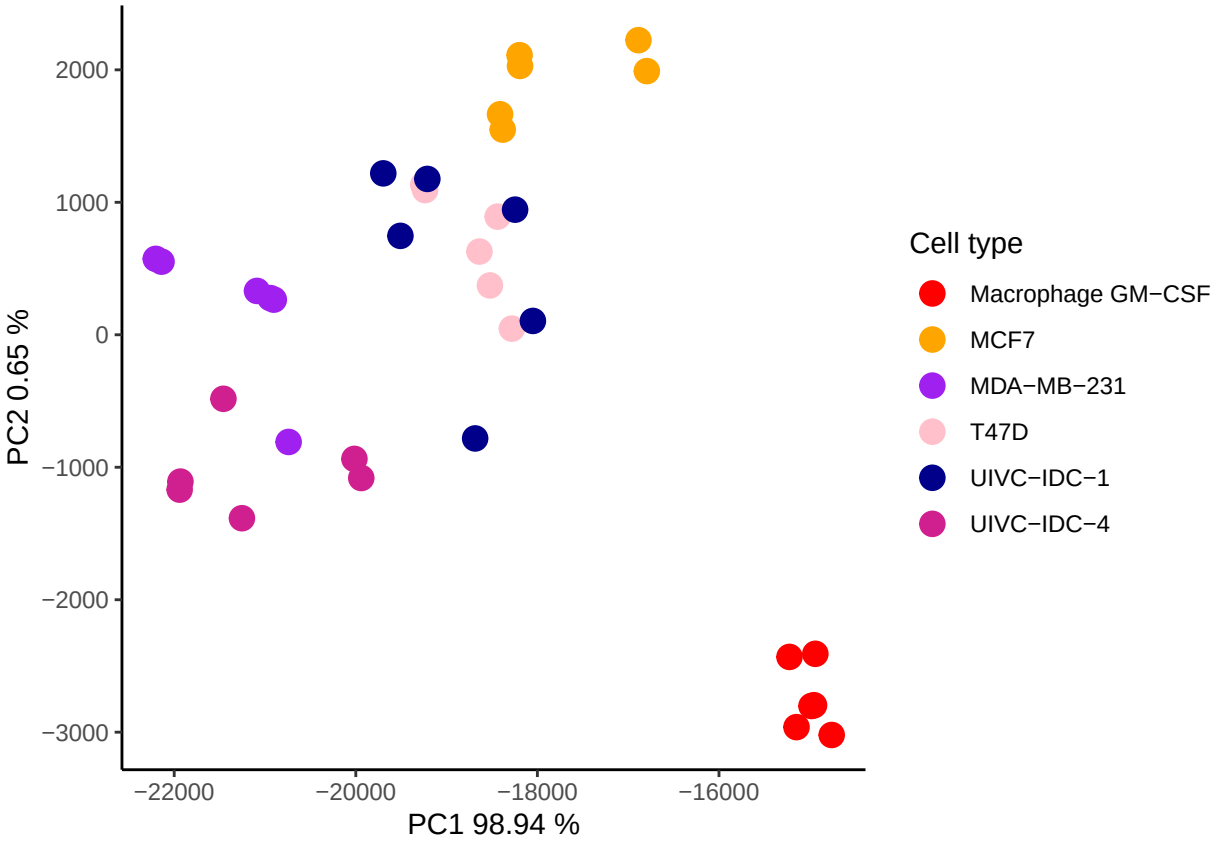
```r
# Realizar el análisis de componentes principales
pca_result <- prcomp(df_tras[, c(1:7640)], center = F, scale. = F)
# Crear un data frame con los resultados del PCA
pca_data <- as.data.frame(pca_result$x)
head(pca_data)
```

```
##               PC1       PC2        PC3       PC4       PC5       PC6       PC7
## gmcsf_1 -14758.46 -3021.151  -893.8309 -88.59559 352.96885  96.47729  80.18060
## gmcsf_2 -14938.70 -2408.405 -1099.5154 344.90653  66.68538 112.20820  75.79576
## gmcsf_3 -15223.36 -2431.690 -1071.1758 360.59410  79.62151 120.48917 -46.75906
## gmcsf_4 -15146.07 -2961.691  -873.2196 -69.45410 329.70800 134.17051 -57.08255
## gmcsf_5 -14952.97 -2796.503 -1121.8777 560.77036  22.65780  96.79155  59.19710
## gmcsf_6 -14980.92 -2802.287 -1128.9078 541.04339  41.40835  83.54041  58.49837
##               PC8       PC9        PC10       PC11       PC12       PC13
## gmcsf_1 -53.20500  34.56155 -102.716540  78.658136 -38.114134   3.278557
## gmcsf_2  70.15855  75.84209  -98.964037  14.962904  -9.815736   6.831146
## gmcsf_3 115.85485  37.05186  108.157435 -64.139384 -98.020214   7.427841
## gmcsf_4 -13.12296  58.91218   45.770171  14.945186 -90.602165  36.863994
## gmcsf_5  79.76506 -13.72278   -6.197554 -42.230933  96.134925 -31.916117
## gmcsf_6  76.59432 -18.02350  -20.621540  -9.077053 101.631322  -9.504401
##                PC14      PC15       PC16       PC17        PC18      PC19
## gmcsf_1  -99.3761717 -26.57158  100.71857  39.581693   -7.740003 -81.13079
## gmcsf_2 -141.3115353 -16.49616  119.75230  23.148943   21.774529  69.40999
## gmcsf_3    7.1856014  46.00304  -48.13474  26.467119 -161.704626  29.71120
## gmcsf_4    0.2549373 -53.25410 -132.28037   8.273259   21.167280 -19.90818
## gmcsf_5  102.3648976  12.37943  -44.74589 -57.965660  110.271230  86.28261
## gmcsf_6  109.2355289  29.81687  -14.86594 -29.427727    7.982789 -80.56637
##               PC20       PC21       PC22        PC23      PC24       PC25
## gmcsf_1 -13.873287 -31.580366  10.045422  11.5775372  70.55197  37.001688
## gmcsf_2   3.990615   3.201626  34.701093 -37.8561367 -50.70348 -16.674768
## gmcsf_3   8.666499  32.897239 -32.121465  51.0627334  55.23448   6.069394
## gmcsf_4  74.651657 -83.632705 -55.445112   2.8903496 -71.38828 -25.950644
## gmcsf_5 -57.295239 -29.870662  36.070661  -0.8800487  34.05176 -16.254011
## gmcsf_6 -19.810863 113.271980   4.432192 -23.9570650 -36.30430  14.551042
##               PC26       PC27       PC28       PC29      PC30       PC31
## gmcsf_1  45.829123   6.106181 -54.594116 -13.467447 -61.19506 -20.271660
## gmcsf_2 -66.054346  25.351340  41.646675   4.673024  59.25344  24.932449
## gmcsf_3 -11.382126 -34.805657 -11.607851 -10.003237  32.25929 -19.072935
## gmcsf_4  10.097811   3.100117  28.288692  44.843744 -16.85637   2.145119
## gmcsf_5  14.263152  30.077532   2.049739 -11.085882 -22.12872 -62.364642
## gmcsf_6   9.278752 -32.977942  -2.644670 -14.959240   8.08511  73.297949
##               PC32       PC33        PC34      PC35       PC36
## gmcsf_1  -4.823764 -32.886825   5.0328291  20.86220   5.494048
## gmcsf_2  15.802981  21.204449   0.4389781 -14.52952  -2.990358
## gmcsf_3   3.138550  17.638668 -21.8246762 -48.93349   2.000886
## gmcsf_4  -7.462147  14.768596  19.3364023  45.04915  10.290922
## gmcsf_5 -10.421872  -2.633153 -16.0512192 -29.34717  22.767499
## gmcsf_6   3.748829 -18.188980  12.9271429  25.42250 -39.184177
```

```r
# Funcion para asignar las etiquetas
assign_label <- function(cond_name) {
  if (grepl("^basal_", cond_name)) {
    return("Monocyte")
```

```
  } else if (grepl("^gmcsf_", cond_name)) {
    return("Macrophage GM-CSF")
  } else if (grepl("^uivc_hs", cond_name)) {
    return("HS578T")
  } else if (grepl("^mcf7_", cond_name)) {
    return("MCF7")
  } else if (grepl("^mda231_", cond_name)) {
    return("MDA-MB-231")
  } else if (grepl("^uivc_p16_", cond_name)) {
    return("MBCDF-16")
  } else if (grepl("^t47d_", cond_name)) {
    return("T47D")
  } else if (grepl("^uivc_160_", cond_name)) {
    return("UIVC-IDC-2")
  } else if (grepl("^uivc_169_", cond_name)) {
    return("UIVC-IDC-3")
  } else if (grepl("^uivc_172_", cond_name)) {
    return("UIVC-IDC-1b")
  } else if (grepl("^uivc_183_", cond_name)) {
    return("UIVC-IDC-9")
  } else if (grepl("^uivc1_", cond_name)) {
    return("UIVC-IDC-1")
  } else if (grepl("^uivc4_", cond_name)) {
    return("UIVC-IDC-4")
  } else {
    return("Other")  # Añadir un caso para cualquier otra condición
  }
}


# Agregar la condición desde los nombres de las columnas
pca_data$Condicion <- sapply(rownames(df_tras), assign_label)



# Agregar la condición desde los nombres de las columnas
pca_data$Condicion <- sapply(rownames(df_tras), function(cond_name) {
  assign_label(cond_name)
})

# Graficar
ggplot(pca_data, aes(x = PC1, y = PC2, color = Condicion)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2,1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2,2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Cell type",
                     values = c("Monocyte" = "blue",
                                "Macrophage GM-CSF" = "red",
                                "HS578T" = "green",
                                "MCF7" = "orange",
                                "MDA-MB-231" = "purple",
                                "MBCDF-16" = "brown",
```

```
                                "T47D" = "pink",
                                "UIVC-IDC-2" = "cyan",
                                "UIVC-IDC-3" = "magenta",
                                "UIVC-IDC-1b" = "yellow",
                                "UIVC-IDC-9" = "salmon",
                                "UIVC-IDC-1" = "darkblue",
                                "UIVC-IDC-4" = "violetred"))
```



Si solo contemplamos los macrófagos el procentaje explicado por el PC1 sigue sinedo muy elevado, lo cual no necesariamente es algo bueno.

En teoría estos datos me dan confizan porque el lote 1 tiene una gran calidad de secueciación y alineamiento. Pero los porcentajes tan altos del PCA me hacen mucho ruído... ¿Qué está pasando?

Ahora vamos integrar el lote 2 a nuestro análisis. En este punto tomé ambos lotes de secuecuaciación y los ensamble y cuantifiqué juntos, para que la normalización fuera homogénea.

```
# Cargar datos
setwd("D:/marval_windows/JR_MARVAL/himfg/maestria/rnaseq_macrophage/DEA_ballgown_5_all_samples/batch/ba
list.files()
```

```
##  [1] "batch_pyjn.ipynb"
##  [2] "batch_pyjn_function.ipynb"
##  [3] "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv"
##  [4] "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv"
```

```
##  [5] "fpkm_all_samples_with_genes_wiso_mean_L1.csv"
##  [6] "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv"
##  [7] "fpkm_macs_with_genes_wiso_mean_L1.csv"
##  [8] "fpkm_macs_with_genes_wiso_mean_L1_median.csv"
##  [9] "fpkm_without_gmcsf_with_genes_wiso_mean_L1.csv"
## [10] "heatmap_all_data.R"
## [11] "rna_pca_batc.html"
## [12] "rna_pca_batc.Rmd"
## [13] "rna_pca_batc_files"
## [14] "work_flow_transcriptome.png"
```

```
data <- read.table(file = "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv", sep = ",", head=T, row.names
head(data)
```

```
##            basal_1   basal_2   basal_3   basal_4  basal_5   basal_6  gmcsf_1
## 5_8S_rRNA 0.000000 0.3295147 0.2612993 0.4521663 0.380779 0.3609917 3.805476
## A1BG      2.018176 2.5895070 2.5014310 1.5421420 2.580991 2.9403150 1.613248
## AAAS      4.643210 6.4536940 3.5578310 6.1616320 3.880429 4.5584150 6.617348
## AACS      2.542018 2.4306390 2.5398120 2.6464310 2.602396 2.6253910 7.555202
## AAGAB     5.171589 5.2833220 4.8610035 4.7664350 4.677665 4.7426860 5.858877
## AAK1      0.953462 0.0000000 1.8896470 1.4826380 0.250284 1.6539590 0.554020
##            gmcsf_2   gmcsf_3  gmcsf_4  gmcsf_5   gmcsf_6 uivc_hs_1 uivc_hs_2
## 5_8S_rRNA 3.228030 0.2115733 0.000000 0.445400 0.2647877 0.4613917 0.8666923
## A1BG      2.731089 1.3409300 2.502300 1.062667 1.5177880 0.6426820 1.8573130
## AAAS      4.571088 3.6608490 6.751473 4.075548 4.2847330 5.5377660 3.7269190
## AACS      7.603384 7.7584920 7.206987 7.314979 6.9703910 5.9976080 6.4691310
## AAGAB     6.047381 6.0325320 6.193422 6.186050 6.4635775 6.5061110 6.1066425
## AAK1      3.754945 1.4567430 0.000000 1.625898 0.2817040 0.0000000 0.0000000
##           uivc_hs_3 uivc_hs_4 uivc_hs_5 uivc_hs_6   mcf7_1   mcf7_2   mcf7_3
## 5_8S_rRNA  2.009060  1.849251  2.097440  2.780988 1.724103 1.247547 0.499947
## A1BG       0.682260  0.407238  2.104228  2.405066 1.169993 1.979829 1.951125
## AAAS       4.420382  4.229813  3.139040  2.778528 5.066211 7.230433 7.182122
## AACS       4.068125  3.753875  4.883587  4.794360 4.435555 4.530611 4.002409
## AAGAB      7.714627  8.628127  6.050537  5.273953 5.044572 4.882823 5.131884
## AAK1       0.000000  0.000000  0.524761  0.000000 0.333017 2.237235 0.319808
##             mcf7_5   mcf7_6 mda231_1 mda231_2 mda231_3 mda231_4  mda231_5
## 5_8S_rRNA 8.970628 9.162631 0.396269 0.206297 0.628268 0.506981 0.8267517
## A1BG      5.287287 4.749459 3.374202 4.151723 5.765807 2.337247 1.4690920
## AAAS      5.367146 4.765695 3.152331 6.385787 5.165659 2.166771 5.3261880
## AACS      4.558067 4.345533 3.442099 3.599744 3.745793 3.808038 3.9467970
## AAGAB     4.835360 5.112352 4.799007 4.249025 4.221697 4.359663 4.4332305
## AAK1      0.000000 0.701440 0.262317 0.246300 0.000000 3.648735 0.5253880
##           mda231_6 uivc_p16_1 uivc_p16_2 uivc_p16_3 uivc_p16_4 uivc_p16_5
## 5_8S_rRNA 1.018049   0.000000  0.3994747   1.451787   1.343182   2.332635
## A1BG      3.889778   0.000000  0.5858740   2.611511   0.376547   0.907030
## AAAS      3.026159   3.780964  3.0524190   2.855243   3.308149   2.774287
## AACS      3.426885   3.393279  3.8824370   3.959445   4.128744   3.302765
## AAGAB     4.565231   8.118073  8.5784085   4.287048   5.301029   4.510503
## AAK1      0.000000   0.000000  2.3389420   0.000000   1.346121   2.102538
##           uivc_p16_6    t47d_1    t47d_3   t47d_4    t47d_5    t47d_6
## 5_8S_rRNA   2.730117 0.5060667 0.4452117 0.501758 0.3960237 0.3390333
## A1BG        0.485591 1.3169720 1.0546900 0.970205 3.4099290 1.9477190
## AAAS        4.269573 9.9056840 9.1596600 6.082649 4.9021130 5.6833790
## AACS        3.392378 5.3864430 4.9369420 4.928674 4.3663650 4.1626140
```

```
## AAGAB       4.359627 4.9514560 5.5515515 5.366778 4.9517205 5.3096805
## AAK1        0.000000 0.2205440 3.3694980 0.527517 0.6181070 0.4347660
##           uivc_160_1 uivc_160_2 uivc_160_3 uivc_160_4 uivc_160_5 uivc_160_6
## 5_8S_rRNA   1.209545   1.463899   0.557046  0.2828523   1.080663   0.482817
## A1BG        1.287749   2.512699   0.656408  1.9604710   1.710555   1.629011
## AAAS        5.135188   6.084133   0.000000  4.3911430   7.825346   6.083337
## AACS        6.933368   6.842249   3.700268  3.6698310   4.843826   4.308089
## AAGAB       7.209487   6.046748   6.641164  6.5847525   5.815954   5.710891
## AAK1        0.000000   0.000000   0.000000  0.0000000   0.000000   0.000000
##           uivc_169_1 uivc_169_2 uivc_169_3 uivc_169_4 uivc_169_5 uivc_169_6
## 5_8S_rRNA  0.7516103  0.5277087  0.4965837  0.4387597   0.884378  0.6635793
## A1BG       0.4613590  0.0000000  1.4810040  0.4289050   0.666354  2.2177050
## AAAS       3.1357990  1.9210310  5.9061070  3.1960360   6.620033  8.0661570
## AACS       5.5429020  5.6314300  4.5946000  4.1186150   6.699352  5.7913990
## AAGAB      7.8844570  7.2700640  6.0200335  6.6298270   5.671249  4.4393215
## AAK1       3.0860320  0.0000000  0.4795190  0.0000000   0.000000  0.0000000
##           uivc_172_1 uivc_172_2 uivc_172_3 uivc_172_4 uivc_172_5 uivc_172_6
## 5_8S_rRNA   0.000000  0.5495993   0.000000   0.000000  0.4416363  0.4247763
## A1BG        3.071792  3.2386290   0.808219   1.189324  1.0490840  1.6396100
## AAAS        5.375921  5.1097580   7.290517   6.731126  3.3670090  6.2064380
## AACS        4.958736  5.2478570   4.139521   5.118015  6.0855590  5.7446100
## AAGAB       7.580497  6.4023270   6.116886   4.956798  5.3205030  5.1400265
## AAK1        0.000000  2.7127730   0.363941   0.000000  0.0000000  0.0000000
##           uivc_183_1 uivc_183_2 uivc_183_3 uivc_183_4 uivc_183_5 uivc_183_6
## 5_8S_rRNA   0.591303  0.5543893   0.974838   1.648021  0.6304287  0.4012353
## A1BG        2.580105  2.5440310   0.000000   0.394871  2.1127600  0.4415450
## AAAS        2.001447  5.0403040   2.896499   2.977566  3.1725350  3.9077620
## AACS        5.926133  5.9216650   5.883516   5.127281  5.6641600  5.0629470
## AAGAB       7.043604  7.1200440   7.929241   7.941213  6.2284595  7.0248815
## AAK1        0.000000  2.3351550   1.237737   0.000000  1.3109140  0.0000000
##            uivc1_2  uivc1_3  uivc1_4  uivc1_5   uivc1_6   uivc4_1   uivc4_2
## 5_8S_rRNA 0.588970 1.781583 1.989139 0.417952 0.4780183 0.2264417 0.5721347
## A1BG      1.235478 3.833647 2.701246 0.749198 2.0432050 1.3276110 1.8027380
## AAAS      5.893254 4.626503 4.535137 5.443699 5.6635610 3.7506040 4.4148840
## AACS      5.250632 5.346076 5.293288 4.961098 4.9900630 4.1932530 4.1395330
## AAGAB     5.162096 5.000882 5.610289 5.662820 5.8837240 4.6519525 4.2673200
## AAK1      0.000000 2.861372 3.062424 0.000000 0.3659970 0.0000000 1.6978040
##            uivc4_3  uivc4_4   uivc4_5   uivc4_6
## 5_8S_rRNA 0.296019 0.374932 0.3028623 0.3500447
## A1BG      2.359144 3.153597 3.8049200 2.6434820
## AAAS      4.184424 5.217705 4.8383790 3.6288590
## AACS      3.535354 3.319669 4.0914650 3.9395960
## AAGAB     4.644629 4.701441 4.6289125 4.5463450
## AAK1      0.443703 0.000000 3.3420390 0.2843740
```

```
# Eliminar filas cuya suma sea 0
data <- data[!(rowSums(data[,]) == 0), ]
# Transponer dataframe
df_tras = data.frame(t(data[,]))
```

Ahora construimos el PCA considerando ambos lotes de secuenciación. Primero lo haremos con los **datos centrados y escalados:**

```r
# PCA
df <- prcomp(df_tras[,c(1:8663)], center = T, scale. = T)
summary(df)
```

```
## Importance of components:
##                             PC1     PC2      PC3      PC4      PC5      PC6
## Standard deviation     44.8243 34.4767 22.85913 21.50055 18.22867 16.78315
## Proportion of Variance  0.2319  0.1372  0.06032  0.05336  0.03836  0.03251
## Cumulative Proportion   0.2319  0.3691  0.42946  0.48282  0.52118  0.55369
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     15.36393 14.68881 13.22066 12.15135 12.00359 10.16604
## Proportion of Variance  0.02725  0.02491  0.02018  0.01704  0.01663  0.01193
## Cumulative Proportion   0.58094  0.60585  0.62602  0.64307  0.65970  0.67163
##                            PC13    PC14    PC15    PC16    PC17    PC18   PC19
## Standard deviation     10.09314 9.64050 9.14193 8.72616 8.61662 8.18360 8.0605
## Proportion of Variance  0.01176 0.01073 0.00965 0.00879 0.00857 0.00773 0.0075
## Cumulative Proportion   0.68339 0.69412 0.70376 0.71255 0.72112 0.72885 0.7363
##                            PC20    PC21    PC22    PC23    PC24    PC25    PC26
## Standard deviation     7.98183 7.75430 7.71852 7.65477 7.59524 7.53641 7.47324
## Proportion of Variance 0.00735 0.00694 0.00688 0.00676 0.00666 0.00656 0.00645
## Cumulative Proportion  0.74371 0.75065 0.75753 0.76429 0.77095 0.77751 0.78395
##                            PC27    PC28    PC29    PC30    PC31    PC32    PC33
## Standard deviation     7.43358 7.35990 7.29664 7.24011 7.22187 7.18795 7.13162
## Proportion of Variance 0.00638 0.00625 0.00615 0.00605 0.00602 0.00596 0.00587
## Cumulative Proportion  0.79033 0.79658 0.80273 0.80878 0.81480 0.82077 0.82664
##                            PC34   PC35    PC36    PC37    PC38    PC39    PC40
## Standard deviation     7.04724 7.0288 6.99051 6.97570 6.93455 6.83514 6.78691
## Proportion of Variance 0.00573 0.0057 0.00564 0.00562 0.00555 0.00539 0.00532
## Cumulative Proportion  0.83237 0.8381 0.84371 0.84933 0.85488 0.86027 0.86559
##                           PC41   PC42    PC43    PC44   PC45    PC46    PC47
## Standard deviation     6.7780 6.6452 6.54313 6.52519 6.3799 6.36221 6.25289
## Proportion of Variance 0.0053 0.0051 0.00494 0.00491 0.0047 0.00467 0.00451
## Cumulative Proportion  0.8709 0.8760 0.88093 0.88585 0.8905 0.89522 0.89973
##                            PC48    PC49    PC50    PC51   PC52    PC53    PC54
## Standard deviation     6.23021 6.18503 6.08612 6.01085 5.9627 5.93575 5.91112
## Proportion of Variance 0.00448 0.00442 0.00428 0.00417 0.0041 0.00407 0.00403
## Cumulative Proportion  0.90421 0.90863 0.91291 0.91708 0.9212 0.92525 0.92928
##                           PC55    PC56    PC57    PC58    PC59    PC60    PC61
## Standard deviation     5.8844 5.87148 5.84600 5.82206 5.72963 5.72466 5.67523
## Proportion of Variance 0.0040 0.00398 0.00395 0.00391 0.00379 0.00378 0.00372
## Cumulative Proportion  0.9333 0.93726 0.94120 0.94511 0.94890 0.95269 0.95640
##                            PC62    PC63    PC64    PC65    PC66    PC67    PC68
## Standard deviation     5.64078 5.60493 5.59880 5.52655 5.49712 5.44577 5.41831
## Proportion of Variance 0.00367 0.00363 0.00362 0.00353 0.00349 0.00342 0.00339
## Cumulative Proportion  0.96008 0.96370 0.96732 0.97085 0.97434 0.97776 0.98115
##                            PC69    PC70    PC71    PC72    PC73    PC74
## Standard deviation     5.32730 5.31725 5.29400 5.18701 5.13047 5.03997
## Proportion of Variance 0.00328 0.00326 0.00324 0.00311 0.00304 0.00293
## Cumulative Proportion  0.98442 0.98769 0.99092 0.99403 0.99707 1.00000
##                            PC75
## Standard deviation     3.627e-14
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```r
# Realizar el análisis de componentes principales
pca_result <- prcomp(df_tras[, c(1:8663)], center = T, scale. = T)
# Crear un data frame con los resultados del PCA
pca_data <- as.data.frame(pca_result$x)
head(pca_data)
```

```
##               PC1      PC2      PC3       PC4      PC5         PC6      PC7
## basal_1 -128.8805 45.32952 19.72094 -18.60895 4.213886  -7.3998815 4.816884
## basal_2 -127.8795 44.91732 18.68495 -18.05356 3.258322  -7.6783658 4.119234
## basal_3 -124.3565 44.13016 17.39355 -15.36812 2.394687  -9.6014421 3.239642
## basal_4 -123.7933 44.68590 18.43221 -14.93713 2.146150 -10.2247395 3.443637
## basal_5 -131.9115 39.27130 18.11705 -19.25763 5.596550   1.0349183 6.024673
## basal_6 -131.9925 39.81254 18.23402 -18.59909 5.605838   0.2705734 5.165871
##                PC8        PC9       PC10       PC11      PC12        PC13
## basal_1 -1.8541706  1.0672077 -1.4083661  0.8862151 1.4505704 -0.06320126
## basal_2 -2.4879470  0.2152907 -1.0621576  0.1563702 1.3881524 -2.16921153
## basal_3 -0.5134427 -0.9465825 -2.1865628 -2.6828467 2.0716788  1.26473627
## basal_4  0.4262813 -0.4991724 -2.3638678 -1.7610197 0.9698356  0.51925364
## basal_5 -2.7301758 -2.8518509  0.3644478  1.8727821 1.0374187 -1.40874806
## basal_6 -3.7397782 -3.9809318 -0.3528599 -0.2144588 0.5363021 -0.01539838
##               PC14      PC15       PC16        PC17       PC18       PC19
## basal_1  2.3433478 -2.399089  0.5016418 -1.14833165  3.5513843 -2.5753462
## basal_2  1.1307149 -2.500525 -1.0157976 -1.39860560  1.2740923  0.5808960
## basal_3  0.7665412  1.704940  0.6851174  0.36861992  0.8949366 -2.3225189
## basal_4 -0.3534957  2.083575  1.4610358 -0.66038617  2.5276107 -0.1644683
## basal_5 -1.6725008  1.532105 -2.5485196 -0.01692272 -1.8540821  3.1700632
## basal_6 -2.5312234  2.002560 -1.7975482  1.52511337 -4.3362278  0.9026657
##              PC20       PC21        PC22       PC23       PC24       PC25
## basal_1  3.767725 -1.1470591  1.10804675  0.3228664  2.4410207 -2.1544689
## basal_2  2.251593 -1.0083817 -0.05586621  1.6001599  2.3623863  2.2826399
## basal_3 -1.022329 -1.9514785 -0.92594442 -0.5093113 -0.4662402 -1.0509725
## basal_4 -1.223795 -0.8423473  2.08238011 -1.4499782  1.1076158 -0.9986118
## basal_5 -2.220266  3.2885278 -3.80765657  0.4349535 -1.4497760  0.4072587
## basal_6 -1.390178  1.1899000  1.14044256 -0.3577122 -3.6166179  0.5876790
##               PC26       PC27       PC28       PC29       PC30        PC31
## basal_1 -0.1886707 -1.4754085 -5.6366267  0.120238  3.9149945  0.23121533
## basal_2  3.9345755 -2.5346500 -1.4472308  3.800962  3.8407519 -1.19821984
## basal_3 -1.6735049 -0.2486705  3.8956363  2.925547 -1.1052106  2.87561328
## basal_4  3.1446403 -1.2624544  5.8064401  1.764797 -0.4521156  0.61732141
## basal_5 -3.1829123  3.5678120 -1.1363235 -5.364006 -3.4474960 -0.06849702
## basal_6 -2.3051873  1.8162696 -0.7379414 -3.866867 -3.5614123 -1.95313970
##              PC32      PC33       PC34      PC35      PC36      PC37       PC38
## basal_1  3.907858 -5.044842 -1.4752580 -4.683693 -2.882119  3.636618  2.7624383
## basal_2 -1.832232 -2.636297 -1.9500042 -1.205659  1.398970  3.702558 -1.5013794
## basal_3 -5.646735  2.166398  2.9466339 11.019723  6.466140 -2.696728  0.7558338
## basal_4 -2.776501  6.940330  3.1461715  7.352420  2.359823  2.116695 -2.9257814
## basal_5  2.067011 -2.630326 -1.9773916 -7.454444 -3.914992 -2.483865  1.9926577
## basal_6  4.412928  1.530132 -0.7216772 -6.037978 -3.861558 -4.298559 -1.8622633
##                PC39        PC40       PC41       PC42       PC43       PC44
## basal_1  5.39553632  4.58560687 -11.237386 -5.3291180 -10.524554  29.013408
## basal_2 -0.06089792  1.25678431  -8.909468 -6.3731622  -4.059907  22.160858
## basal_3 -4.05789723 -0.36957707   4.638282  4.6657880   3.730811 -12.226036
## basal_4 -5.41005364  0.44629834   3.210629  7.0296989   7.739373  -9.487291
```

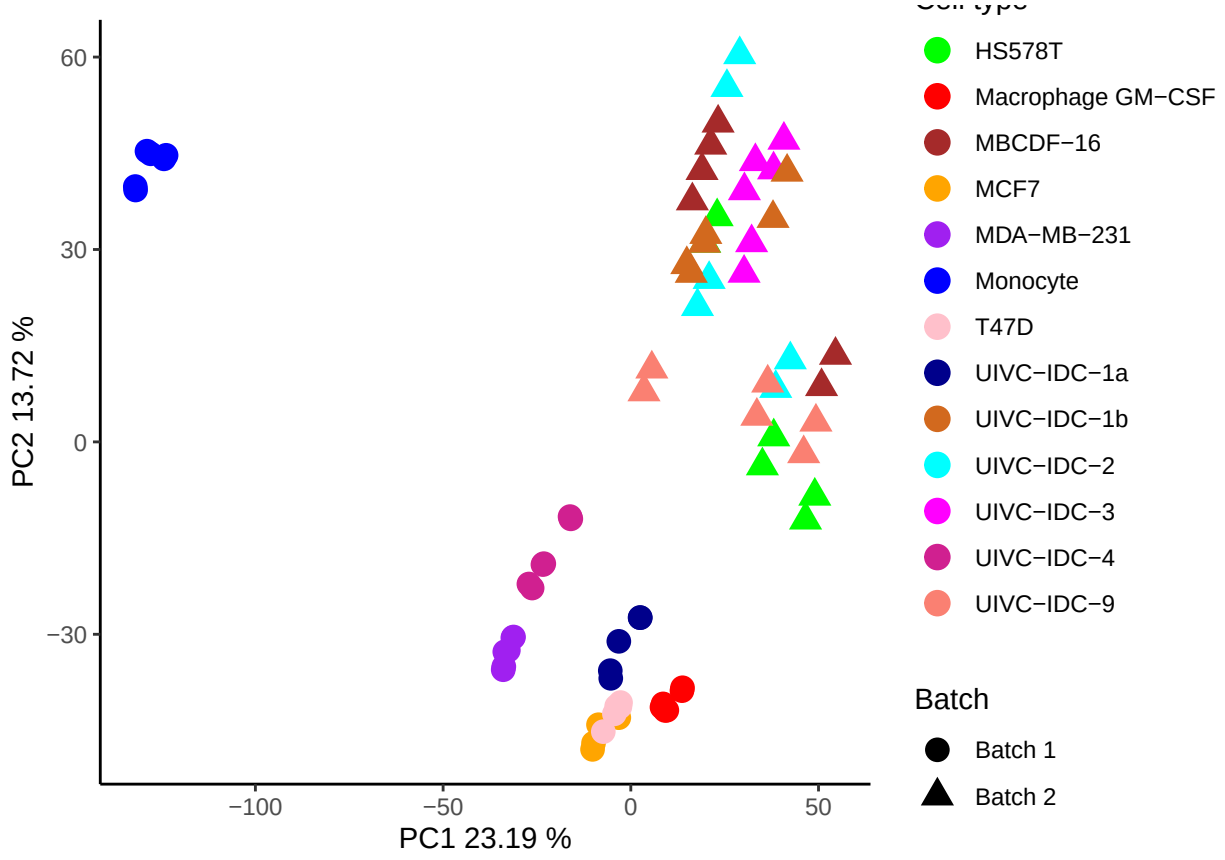
```
## basal_5  2.43850016 -6.37457666   4.294578  1.2435710  -1.419846 -16.225247
## basal_6  1.33126074 -0.08284114   6.881458 -0.9146592   5.087194 -14.758128
##               PC45        PC46       PC47       PC48        PC49       PC50
## basal_1   5.267377  15.8255692 -13.859208  -1.406589   0.2802933  13.163297
## basal_2  -9.223442 -10.9037961  16.481927   7.564073  -4.2600396 -13.372596
## basal_3  -5.018384  -0.9172547  -9.262680   5.019906  14.2873790 -10.947910
## basal_4 -11.164376 -12.0716417  -1.845216  13.221038  14.7070582  -1.122229
## basal_5  24.140122  10.3268466   2.792786 -15.878698  -6.3412425  -3.571184
## basal_6  -4.654586  -3.1065500   5.453965  -8.048735 -16.9929719  15.197510
##                PC51        PC52       PC53       PC54       PC55       PC56
## basal_1 -19.4539012  -1.9915778 -0.1892363  -6.778732  -1.086364  5.699796
## basal_2  26.7148113  -2.3373605 -2.4319671   6.256662   3.137600 -1.752194
## basal_3  -4.9182362  11.6558481  3.1371301  12.966912  -5.335533 10.603973
## basal_4 -12.1991625 -12.7394712 -3.6752169 -10.419564   2.265081 -5.603730
## basal_5   8.8534780   0.3677783 -1.0019856  12.017035 -10.188203 -2.208986
## basal_6   0.9091505   5.3470700  3.8074408 -14.308903  11.443064 -6.409685
##                PC57        PC58        PC59       PC60       PC61       PC62
## basal_1   3.6681693   0.5975798   0.5982069  0.5530086   2.148485 -1.0640126
## basal_2  -4.7230932  -0.3257174   2.9669046  3.8423328   1.291332  0.4517548
## basal_3   0.9865927   9.6292299  12.2158590  1.0253968  12.375661 -9.7537781
## basal_4   6.7819651 -10.3438073 -13.8192997 -4.2570959 -10.075392  6.3172323
## basal_5  13.5647020  -1.7735141  -9.0496905 -5.7948717   0.793373  4.8234148
## basal_6 -19.8821594   2.0918573   6.7944523  4.7669913  -6.373119 -0.6123236
##               PC63       PC64       PC65        PC66       PC67       PC68
## basal_1  -1.490165 -1.493537 -0.6487107 -0.92492908  0.958889  2.543374
## basal_2   5.119539  2.033419 -4.3840499 -0.08580534 -1.966773 -3.035378
## basal_3 -12.092052 -5.511382  1.0596580  5.68174226  3.419920 -4.618131
## basal_4   6.959962  4.238024 -0.7883917 -3.19165364 -6.159957  2.677289
## basal_5   7.808562 -0.709081 -1.7525780 -4.59446762 -5.446176  3.152328
## basal_6  -6.671334  1.100269  6.4224334  3.52627080  9.196472 -1.091980
##               PC69       PC70       PC71       PC72       PC73       PC74
## basal_1 -0.6000691 -1.018557 -0.1085751  1.2351565 -0.6761088  0.9768287
## basal_2 -2.8920681  1.003148 -3.2008690 -1.3788962 -1.0403687 -0.5618638
## basal_3 -0.7092917 -6.406073 -4.2940499  0.8413821  4.2010918 -1.7519851
## basal_4  5.8215612  7.028867  1.7269204 -0.4717626  0.9402712 -0.1611455
## basal_5 -2.4018384  1.105740  2.7749358 -2.4695701 -3.2767705 -0.4080173
## basal_6  0.9851169 -1.978522  2.7755745  2.0832776 -0.0361352  1.7644508
##                 PC75
## basal_1 -2.803140e-14
## basal_2 -1.428981e-13
## basal_3  6.975757e-15
## basal_4 -1.307158e-14
## basal_5 -4.916033e-14
## basal_6 -1.323958e-13
```

Ahora vamos a representar gráficamente nuestros resultado. Para ello nombraremos nuestras muestras por condición, excluyendo su número de replica para que sea más fácil la unificación. Además, pondremos una etiqueta para identificar el lote al que pertenece cada muestra.

```r
# Función para asignar las etiquetas
assign_label <- function(cond_name) {
  if (grepl("^basal_", cond_name)) {
    return("Monocyte")
  } else if (grepl("^gmcsf_", cond_name)) {
```

```
    return("Macrophage GM-CSF")
  } else if (grepl("^uivc_hs", cond_name)) {
    return("HS578T")
  } else if (grepl("^mcf7_", cond_name)) {
    return("MCF7")
  } else if (grepl("^mda231_", cond_name)) {
    return("MDA-MB-231")
  } else if (grepl("^uivc_p16_", cond_name)) {
    return("MBCDF-16")
  } else if (grepl("^t47d_", cond_name)) {
    return("T47D")
  } else if (grepl("^uivc_160_", cond_name)) {
    return("UIVC-IDC-2")
  } else if (grepl("^uivc_169_", cond_name)) {
    return("UIVC-IDC-3")
  } else if (grepl("^uivc_172_", cond_name)) {
    return("UIVC-IDC-1b")
  } else if (grepl("^uivc_183_", cond_name)) {
    return("UIVC-IDC-9")
  } else if (grepl("^uivc1_", cond_name)) {
    return("UIVC-IDC-1a")
  } else if (grepl("^uivc4_", cond_name)) {
    return("UIVC-IDC-4")
  } else {
    return("Other")
  }
}

# Función para asignar el lote
assign_batch <- function(cond_name) {
  lote1 <- c("Monocyte", "Macrophage GM-CSF", "MCF7", "MDA-MB-231", "T47D", "UIVC-IDC-1a", "UIVC-IDC-4")
  if (assign_label(cond_name) %in% lote1) {
    return("Batch 1")
  } else {
    return("Batch 2")
  }
}

# Agregar las columnas 'Condicion' y 'Lote' al marco de datos PCA
pca_data$Condicion <- sapply(rownames(df_tras), assign_label)
pca_data$Lote <- sapply(rownames(df_tras), assign_batch)

# Graficar el PCA con colores para las condiciones y formas para los lotes
ggplot(pca_data, aes(x = PC1, y = PC2, color = Condicion, shape = Lote)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2,1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2,2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Cell type",
                     values = c("Monocyte" = "blue",
                                "Macrophage GM-CSF" = "red",
                                "HS578T" = "green",
                                "MCF7" = "orange",
```

```
                                "MDA-MB-231" = "purple",
                                "MBCDF-16" = "brown",
                                "T47D" = "pink",
                                "UIVC-IDC-2" = "cyan",
                                "UIVC-IDC-3" = "magenta",
                                "UIVC-IDC-1b" = "chocolate",
                                "UIVC-IDC-9" = "salmon",
                                "UIVC-IDC-1a" = "darkblue",
                                "UIVC-IDC-4" = "violetred")) +
  scale_shape_manual(name = "Batch",
                     values = c("Batch 1" = 16,  # Cuadrado
                                "Batch 2" = 17))  # Triángulo
```



En este gráfico podemos observar un posible efecto de lote. Para solucionarlo, el primer abordaje consiste en no centrar ni escalar los datos, dado que todos están medidos en la misma unidad de transcripción FPKM y por ende normalizados. El análisis de los datos sin centrar ni escalar viene a continuación.

Análisis de Componentes Principales y Efecto de Lote

Los datos necesarios para construir este análsis son los mismos que usamos anteriormente, solo cambian los argumentos del algoritmo a *False* para no centrar ni escalar los datos.

```
# PCA
df <- prcomp(df_tras[,c(1:8663)], center = F, scale. = F)
summary(df)
```

```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     1.827e+04 2.503e+03 2.080e+03 1.395e+03 1.014e+03
## Proportion of Variance 9.546e-01 1.792e-02 1.238e-02 5.560e-03 2.940e-03
## Cumulative Proportion  9.546e-01 9.726e-01 9.849e-01 9.905e-01 9.934e-01
##                             PC6       PC7      PC8       PC9      PC10
## Standard deviation     851.17734 624.66071 562.4041 426.73287 397.75193
## Proportion of Variance   0.00207   0.00112   0.0009   0.00052   0.00045
## Cumulative Proportion    0.99550   0.99662   0.9975   0.99804   0.99849
##                            PC11     PC12      PC13      PC14     PC15
## Standard deviation     346.21670 322.7583 225.48556 216.72508 185.0256
## Proportion of Variance   0.00034   0.0003   0.00015   0.00013   0.0001
## Cumulative Proportion    0.99884   0.9991   0.99928   0.99941   0.9995
##                            PC16      PC17      PC18      PC19     PC20
## Standard deviation     172.33319 139.18335 129.06498 118.52056 99.49181
## Proportion of Variance   0.00008   0.00006   0.00005   0.00004  0.00003
## Cumulative Proportion    0.99960   0.99965   0.99970   0.99974  0.99977
##                              PC21     PC22     PC23     PC24     PC25     PC26
## Standard deviation     88.88915 86.50937 81.17757 72.65018 67.17863 62.05654
## Proportion of Variance  0.00002  0.00002  0.00002  0.00002  0.00001  0.00001
## Cumulative Proportion   0.99979  0.99981  0.99983  0.99985  0.99986  0.99987
##                              PC27     PC28     PC29     PC30     PC31     PC32
## Standard deviation     57.89719 53.06099 47.63920 45.56889 44.62190 43.09603
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99988  0.99989  0.99989  0.99990  0.99991  0.99991
##                              PC33    PC34    PC35    PC36    PC37    PC38    PC39
## Standard deviation     40.7638 38.5845 36.4229 35.8924 35.4799 34.3228 32.5984
## Proportion of Variance  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
## Cumulative Proportion   0.9999  0.9999  0.9999  0.9999  0.9999  0.9999  0.9999
##                              PC40    PC41  PC42 PC43  PC44  PC45  PC46  PC47  PC48
## Standard deviation     32.1922 31.4358 30.77   30 29.59 29.12 28.85 28.35 28.11
## Proportion of Variance  0.0000  0.0000  0.00    0  0.00  0.00  0.00  0.00  0.00
## Cumulative Proportion   0.9999  0.9999  1.00    1  1.00  1.00  1.00  1.00  1.00
##                             PC49 PC50  PC51  PC52  PC53 PC54  PC55  PC56  PC57
## Standard deviation     26.87 26.6 26.51 25.96 25.74 25.2 24.86 24.41 24.01
## Proportion of Variance  0.00  0.0  0.00  0.00  0.00  0.0  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.0  1.00  1.00  1.00  1.0  1.00  1.00  1.00
##                             PC58 PC59  PC60 PC61  PC62  PC63  PC64  PC65  PC66
## Standard deviation     23.51 23.1 22.82 22.4 22.35 22.18 21.45 21.33 20.97
## Proportion of Variance  0.00  0.0  0.00  0.0  0.00  0.00  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.0  1.00  1.0  1.00  1.00  1.00  1.00  1.00
##                             PC67  PC68  PC69  PC70  PC71  PC72  PC73  PC74  PC75
## Standard deviation     20.31 20.19 19.74 19.57 18.91 18.54 18.15 17.89 17.16
## Proportion of Variance  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
```

```
# Realizar el análisis de componentes principales
pca_result <- prcomp(df_tras[, c(1:8663)], center = F, scale. = F)
# Crear un data frame con los resultados del PCA
pca_data <- as.data.frame(pca_result$x)
head(pca_data)
```

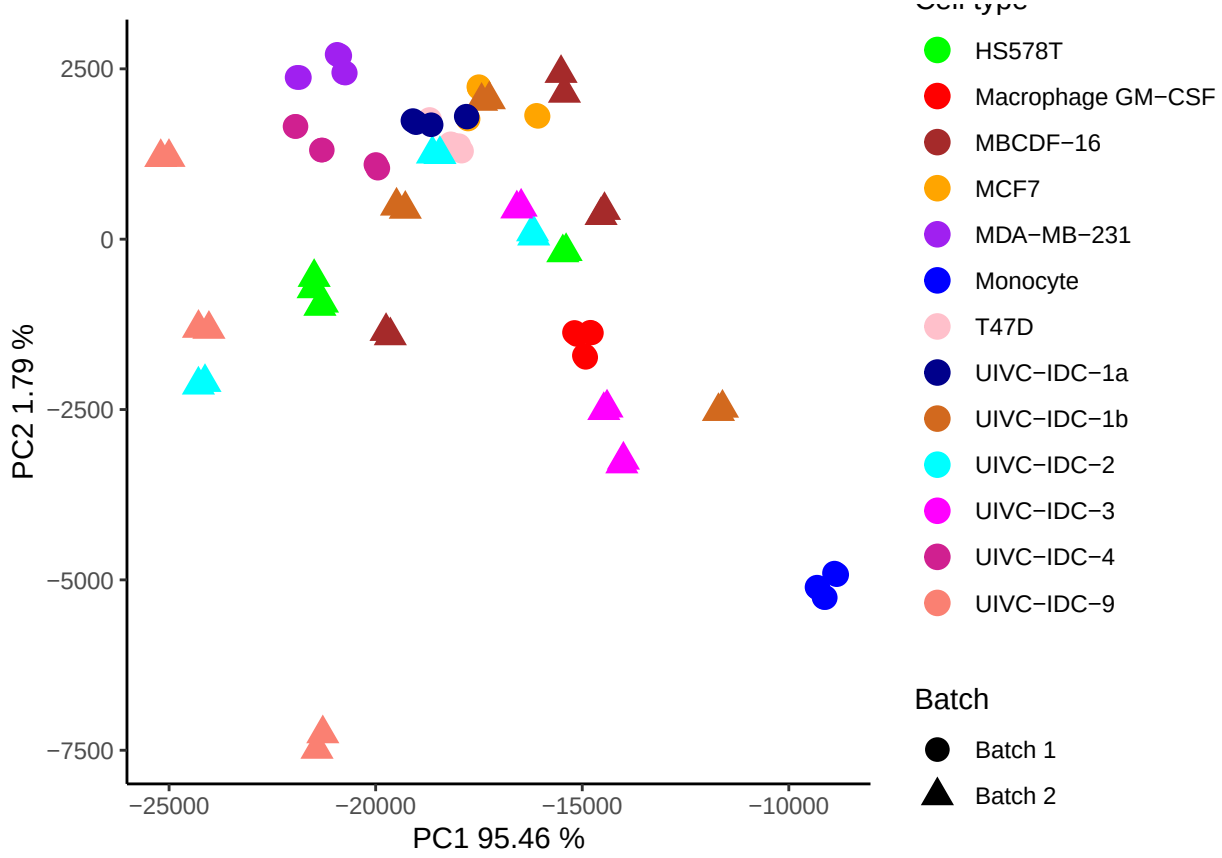
```
##               PC1       PC2       PC3       PC4       PC5      PC6      PC7
## basal_1 -8894.629 -4900.952 -3333.521 -2527.272 -790.5314 883.6319 345.7701
```

```
## basal_2 -8841.032 -4923.452 -3360.101 -2529.989 -782.5950 881.8794 350.7441
## basal_3 -9147.577 -5256.370 -3444.644 -2421.826 -659.7978 884.1301 307.5293
## basal_4 -9119.026 -5259.643 -3456.463 -2430.428 -667.7225 910.6486 309.9510
## basal_5 -9317.014 -5107.218 -3348.153 -2438.260 -767.0721 922.1012 217.5334
## basal_6 -9292.598 -5118.463 -3348.115 -2412.213 -756.5690 914.6107 217.1974
##               PC8      PC9       PC10      PC11      PC12       PC13
## basal_1 -19.518902 231.5271 -47.799681  68.38270 -71.75207  90.041738
## basal_2 -25.031890 220.3149 -48.052992  56.93726 -50.79619  75.548891
## basal_3  -8.651722 142.4930  -7.122334  -9.82084 -17.98230  -3.321365
## basal_4   8.334079 133.7315   1.105693 -12.94360 -12.89105 -20.050494
## basal_5  37.956454 178.6279 -46.045279  90.86784 -47.54689 -28.039385
## basal_6  23.322481 187.5008 -43.416783  96.46271 -40.64276 -37.153446
##               PC14      PC15       PC16      PC17      PC18      PC19
## basal_1   -2.230481 -54.53067  125.95183 -89.97240 117.14748 -12.43670
## basal_2   -2.915590 -38.19332  135.84021 -77.07183  94.82637  16.49636
## basal_3    2.787015 -33.42054  -26.21772 -41.10677 -65.81412 -64.99357
## basal_4    7.531037 -43.00392  -21.13016 -49.73921 -62.09635 -66.97068
## basal_5 -105.628627  15.26874 -152.65114 135.17472 -21.22837  44.10879
## basal_6 -109.277820  26.12039 -143.62587 124.23128 -30.00746  33.23879
##               PC20      PC21       PC22         PC23      PC24       PC25
## basal_1  40.681840 -58.13045  23.0802694   -0.6330434  86.06025  -3.358574
## basal_2  34.262672 -45.78718   4.4773944   14.3644469  64.24776 -10.892392
## basal_3  22.913665  71.10167 -11.1610134   79.0049150  22.38876  34.388628
## basal_4   9.333617  80.27500  20.5558329   73.3924800 -37.01958  55.936198
## basal_5  -6.178867 -34.58153  -4.7993250 -115.2755822 -53.60916 -37.932996
## basal_6 -13.059399 -26.89795  -0.4050102  -97.3923839 -92.79602 -50.725112
##               PC26       PC27       PC28      PC29      PC30      PC31
## basal_1  59.426642 -128.49634  24.365742 -74.10756  80.04119 -23.35944
## basal_2  38.821483  -97.14596  55.500701 -24.40102  90.09870 -20.64959
## basal_3 -34.239734   63.75860 -21.124858 -56.26911 -42.56961 -63.24021
## basal_4 -35.207060   57.59305 -39.191441 -13.74869 -22.14917 -40.56958
## basal_5 -18.103558   35.41341 -23.405431  58.76600 -54.85668  60.11581
## basal_6  -7.087798   50.88091  -5.624969 112.91153 -49.27928  81.61716
##              PC32      PC33      PC34       PC35       PC36        PC37
## basal_1 -48.14504  91.49051  46.01550  86.161510 -20.896073 -39.1291961
## basal_2  12.34276   7.02715 -54.91575 -44.455665  30.903604  13.9335423
## basal_3 -14.37895 -48.00275  60.14334   8.013717  -7.170119  22.3381442
## basal_4 -27.41221 -71.94586  49.93069   2.209579 -78.383804   0.2629732
## basal_5  34.00613  37.45763 -22.07995   9.508358  28.119522 -13.0050336
## basal_6  50.37625 -21.02418 -74.13103 -61.877314  48.371256  17.9307377
##              PC38       PC39      PC40       PC41       PC42       PC43
## basal_1 -44.85299 -11.678023  13.64641  22.139610  55.914958   6.986974
## basal_2 -38.60059  -2.501905  15.99322 -41.574458 -35.961773 -25.677730
## basal_3  76.90921  -6.070726  14.00404 -30.735214  58.227168  40.933135
## basal_4  79.06737   3.869116  44.02893  -5.047738 -57.203048  -9.399805
## basal_5 -25.89953  14.182401 -52.54481  49.243445   4.711438 -12.144269
## basal_6 -37.19642   5.006015 -35.49896   4.497358 -25.746888   1.565101
##               PC44      PC45       PC46       PC47       PC48       PC49
## basal_1   37.552286 -69.08300  -8.9982845  -3.059073  -5.121732 -10.642060
## basal_2   14.617441  24.45183  15.1858527  78.697852   4.615154   3.526714
## basal_3   38.658084  40.80282  27.3605222 -26.723473  -1.733781  25.367711
## basal_4 -105.331180   8.16302   8.5948747   1.420906 -30.234570 -18.636513
## basal_5   12.054198 -21.95329 -41.2183296 -26.186085  25.094767  21.333368
## basal_6    2.425551  17.81942   0.4616691 -23.357369   7.745854 -19.414460
```

```
##                PC50       PC51      PC52      PC53      PC54       PC55
## basal_1  -0.0993786  -1.444176 -12.81607 -16.55047 -4.311431  -6.633540
## basal_2 -52.9028261 -29.397844  33.88480  42.48553 -5.108236  36.566380
## basal_3  57.7728844 -55.606784  34.25206  43.37752 -5.951143 -20.736956
## basal_4 -17.1199449  43.073425 -62.48732 -35.28060 -4.364031  11.650994
## basal_5  30.4717717  54.682499 -30.56420 -79.92671 28.747421 -17.915869
## basal_6 -17.8957286 -12.365893  36.87725  47.02510 -9.624834  -1.759578
##                PC56       PC57        PC58       PC59       PC60       PC61
## basal_1 -15.535157 -19.824064 -13.6022287  12.990703  16.512224  30.624953
## basal_2  37.363435  36.685131   9.6213411 -17.487920  -8.127676   1.473481
## basal_3 -55.835979 -10.633519   0.1001226   8.162397   1.036717 -30.553104
## basal_4  49.610475   9.501894  12.2634422  -6.350804 -15.574180   7.152461
## basal_5  -6.263729   2.265244   3.8592524  -1.313657   8.635274  -2.919499
## basal_6 -10.487528 -17.759676 -10.9692551   4.757501  -3.253049  -5.934502
##                PC62        PC63       PC64      PC65      PC66       PC67
## basal_1   1.962196  15.7608385  29.829831  33.41539 -33.80920 -7.7148540
## basal_2  33.830348 -13.2362580 -27.891820 -32.02213  27.94155 -1.3686493
## basal_3 -38.852582   6.9274132   1.618816 -24.66073  24.16873 -1.3596752
## basal_4  18.461435 -10.5836637  -6.356424  11.95945 -13.86516  2.2471737
## basal_5   8.389411   1.2134063  12.965893 -55.28875  32.47196 -0.1315679
## basal_6 -22.979864  -0.5026463  -9.167315  67.16394 -37.00013  8.4314122
##                PC68       PC69       PC70       PC71       PC72       PC73
## basal_1 -1.4422410  -4.659645  -2.800457 -28.885751  -0.818056  25.45244
## basal_2 -8.0492204  -5.259978  22.529279  27.844101  -3.900473 -20.25707
## basal_3 -0.5056475   7.843428  -7.524104   9.201506  18.370876 -21.29997
## basal_4 -3.0548109   7.377782  -5.427812 -14.208801 -14.571332  11.15371
## basal_5  3.4210522 -24.525884  17.518609  39.215812   6.565222 -20.07574
## basal_6  9.5533802  19.573602 -24.205876 -32.507886  -5.915766  25.44775
##                PC74       PC75
## basal_1  12.887853   3.668745
## basal_2 -11.727715  -4.673471
## basal_3  -2.678400  -4.093036
## basal_4   4.872075   6.932330
## basal_5 -11.556844 -14.013573
## basal_6   8.754728  11.661912
```

```r
# Función para asignar las etiquetas
assign_label <- function(cond_name) {
  if (grepl("^basal_", cond_name)) {
    return("Monocyte")
  } else if (grepl("^gmcsf_", cond_name)) {
    return("Macrophage GM-CSF")
  } else if (grepl("^uivc_hs", cond_name)) {
    return("HS578T")
  } else if (grepl("^mcf7_", cond_name)) {
    return("MCF7")
  } else if (grepl("^mda231_", cond_name)) {
    return("MDA-MB-231")
  } else if (grepl("^uivc_p16_", cond_name)) {
    return("MBCDF-16")
  } else if (grepl("^t47d_", cond_name)) {
    return("T47D")
  } else if (grepl("^uivc_160_", cond_name)) {
    return("UIVC-IDC-2")
```

```
  } else if (grepl("^uivc_169_", cond_name)) {
    return("UIVC-IDC-3")
  } else if (grepl("^uivc_172_", cond_name)) {
    return("UIVC-IDC-1b")
  } else if (grepl("^uivc_183_", cond_name)) {
    return("UIVC-IDC-9")
  } else if (grepl("^uivc1_", cond_name)) {
    return("UIVC-IDC-1a")
  } else if (grepl("^uivc4_", cond_name)) {
    return("UIVC-IDC-4")
  } else {
    return("Other")
  }
}

# Función para asignar el lote
assign_batch <- function(cond_name) {
  lote1 <- c("Monocyte", "Macrophage GM-CSF", "MCF7", "MDA-MB-231", "T47D", "UIVC-IDC-1a", "UIVC-IDC-4")
  if (assign_label(cond_name) %in% lote1) {
    return("Batch 1")
  } else {
    return("Batch 2")
  }
}

# Agregar las columnas 'Condicion' y 'Lote' al marco de datos PCA
pca_data$Condicion <- sapply(rownames(df_tras), assign_label)
pca_data$Lote <- sapply(rownames(df_tras), assign_batch)

# Graficar el PCA con colores para las condiciones y formas para los lotes
ggplot(pca_data, aes(x = PC1, y = PC2, color = Condicion, shape = Lote)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2,1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2,2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Cell type",
                     values = c("Monocyte" = "blue",
                                "Macrophage GM-CSF" = "red",
                                "HS578T" = "green",
                                "MCF7" = "orange",
                                "MDA-MB-231" = "purple",
                                "MBCDF-16" = "brown",
                                "T47D" = "pink",
                                "UIVC-IDC-2" = "cyan",
                                "UIVC-IDC-3" = "magenta",
                                "UIVC-IDC-1b" = "chocolate",
                                "UIVC-IDC-9" = "salmon",
                                "UIVC-IDC-1a" = "darkblue",
                                "UIVC-IDC-4" = "violetred")) +
  scale_shape_manual(name = "Batch",
                     values = c("Batch 1" = 16,  # Cuadrado
                                "Batch 2" = 17))  # Triángulo
```

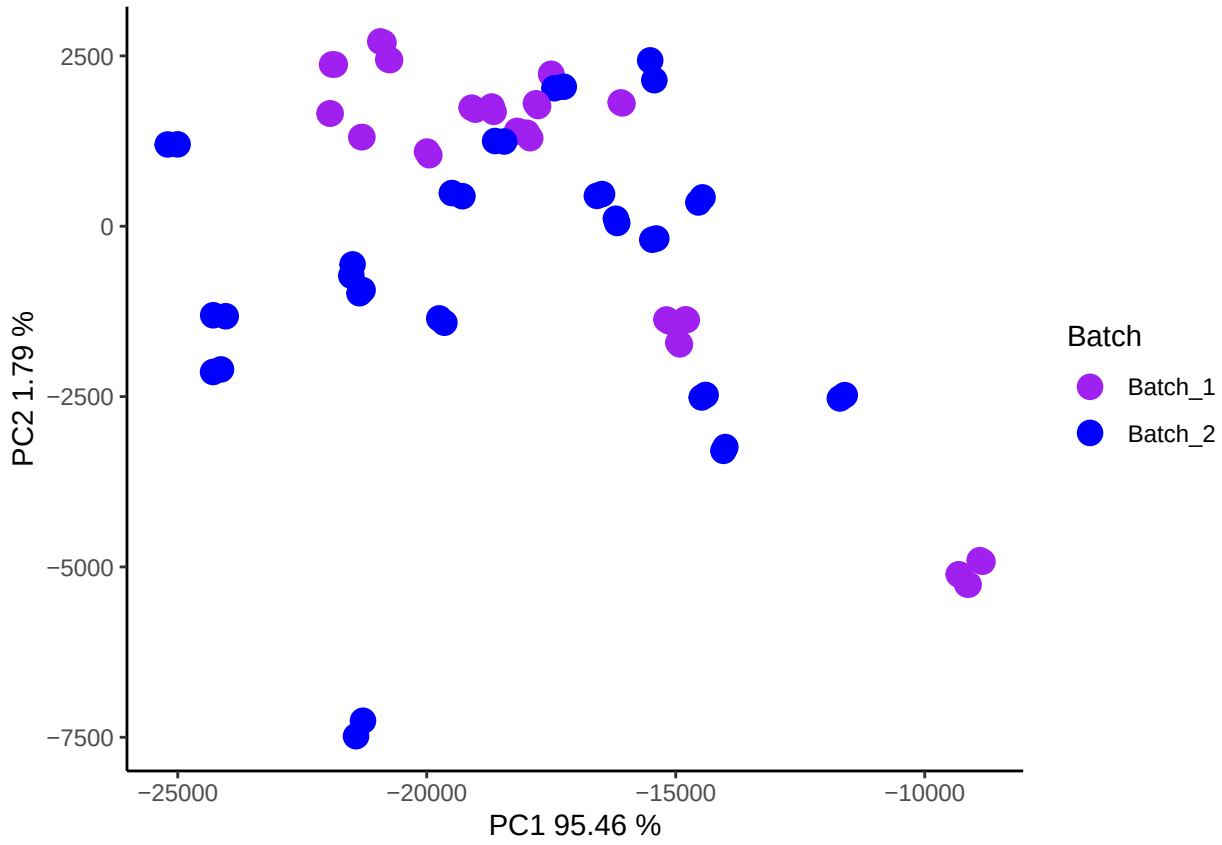


Con este gráfico vemos una mejor armonía entre los macrófagos entrenados por el Microambiente Tumoral (TME), de ambos lotes de secuenciación. **Además, la variación explicada por el PC1 es mucho mayor, tan mayor que preocua que lo que este explicando el PC1 sea el efecto por lote ¿esto es posible?**

Pero aún notamos una mayor dispersión entre las replicas del lote 2, por ello queremos realizar la correción por lote. Lo siguiente es representar un PCA por lotes y ver si en verdad existe este efecto y de ser así solucionarlo. Ahora lo que debemos hacer es añadir la etiqueta de lote a cada una nuestras muestras según corresponda:

```
# Crear un vector de lotes
batch_info <- c("Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                 "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2", "Batch_2",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1",
                "Batch_1", "Batch_1", "Batch_1", "Batch_1", "Batch_1")

# Añadir la información del lote al dataframe transpuesto
df_tras$lote <- batch_info
```

```r
# Verificar efectos de lote con PCA
# Hacer el PCA sin considerar la última columna (lote)
pca_result <- prcomp(df_tras[, -ncol(df_tras)], center = FALSE, scale. = FALSE)
# Convertir el PCA en un dataframe
pca_data <- as.data.frame(pca_result$x)
# Añadir la columna lote al dataframe del PCA
pca_data$lote <- df_tras$lote

# Crear una tabla de relación
tabla_relacion <- data.frame(Muestra = rownames(df_tras), Lote = df_tras$lote)

# Graficar PCA coloreado por lote
ggplot(pca_data, aes(x = PC1, y = PC2, color = lote)) +
  geom_point(size = 4) +
  labs(x = paste("PC1", round(summary(pca_result)$importance[2, 1] * 100, 2), "%"),
       y = paste("PC2", round(summary(pca_result)$importance[2, 2] * 100, 2), "%")) +
  theme_classic() +
  scale_color_manual(name = "Batch",
                     values = c("Batch_1" = "purple", "Batch_2" = "blue"))
```



Ahora lo siguiente es raealizar la corrección por lote con la herramienta ComBat-seq.

Algoritmo de Jerarquización

En esta sección visualizaremos los dato mediante mapas de calor con la intención de observar como se están agrupando los datos. Las librerías a utilizar son:

```
#install.packages("pheatmap")
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.3.3
```

```
#install.packages("ggplot2")
library(ggplot2)
#install.packages("colorspace")
library(colorspace)
```

```
## Warning: package 'colorspace' was built under R version 4.3.3
```

```
#install.packages("grid")
library(grid)
#install.packages("RColorBrewer")
library(RColorBrewer)
```

Primero vamos a visualizar los datos del **lote 1:**

```
# Cargar datos
setwd("D:/marval_windows/JR_MARVAL/himfg/maestria/rnaseq_macrophage/DEA_ballgown_5_all_samples/batch/ba
list.files()
```

```
##  [1] "batch_pyjn.ipynb"
##  [2] "batch_pyjn_function.ipynb"
##  [3] "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv"
##  [4] "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv"
##  [5] "fpkm_all_samples_with_genes_wiso_mean_L1.csv"
##  [6] "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv"
##  [7] "fpkm_macs_with_genes_wiso_mean_L1.csv"
##  [8] "fpkm_macs_with_genes_wiso_mean_L1_median.csv"
##  [9] "fpkm_without_gmcsf_with_genes_wiso_mean_L1.csv"
## [10] "heatmap_all_data.R"
## [11] "rna_pca_batc.html"
## [12] "rna_pca_batc.Rmd"
## [13] "rna_pca_batc_files"
## [14] "work_flow_transcriptome.png"
```

```
data <- read.table(file = "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv", sep = ",", head=T)
head(data)
```

```
##   gene_name  Monocyte Macrophage_GM_CSF      MCF7 MDA_MB_231      T47D
## 1 5_8S_rRNA 0.3467927         0.3558372 1.4928442  0.5701095 0.4752147
## 2      A1BG 2.5519780         1.5718080 2.1351915  3.6117945 1.6382420
## 3      AAAS 4.1000195         3.9749575 4.7281685  3.9296295 5.7812225
## 4      AACS 2.5836655         7.4539210 4.4926840  3.6882900 4.9508285
```

```
## 5     AAGAB 4.8344610         6.1432068 5.1016100  4.4160070 5.3160175
## 6      AAK1 1.1648420         1.1806160 0.3226245  0.1503300 0.5970905
##   UIVC_IDC_1a UIVC_IDC_4
## 1   0.5352285  0.3277103
## 2   1.6642630  2.5093405
## 3   4.2945670  3.9755175
## 4   5.1380185  4.0309815
## 5   5.4053752  4.6515145
## 6   0.1067900  0.2907535
```

Ahora vamos a trabajr un poco los datos. Primero vamos a eliminar los genes que no se expresan en ninguna de las condiciones experimentales. Y después devemmos transdormar nuestro dataframe en una matriz:

```
data <- data[!(rowSums(data[, -1]) == 0), ]
rownames(data) <- data[,1]
samp2 <- data[,-1]
mat_data <- data.matrix(samp2[,1:ncol(samp2)])
colnames(data)
```

```
## [1] "gene_name"         "Monocyte"          "Macrophage_GM_CSF"
## [4] "MCF7"              "MDA_MB_231"        "T47D"
## [7] "UIVC_IDC_1a"       "UIVC_IDC_4"
```

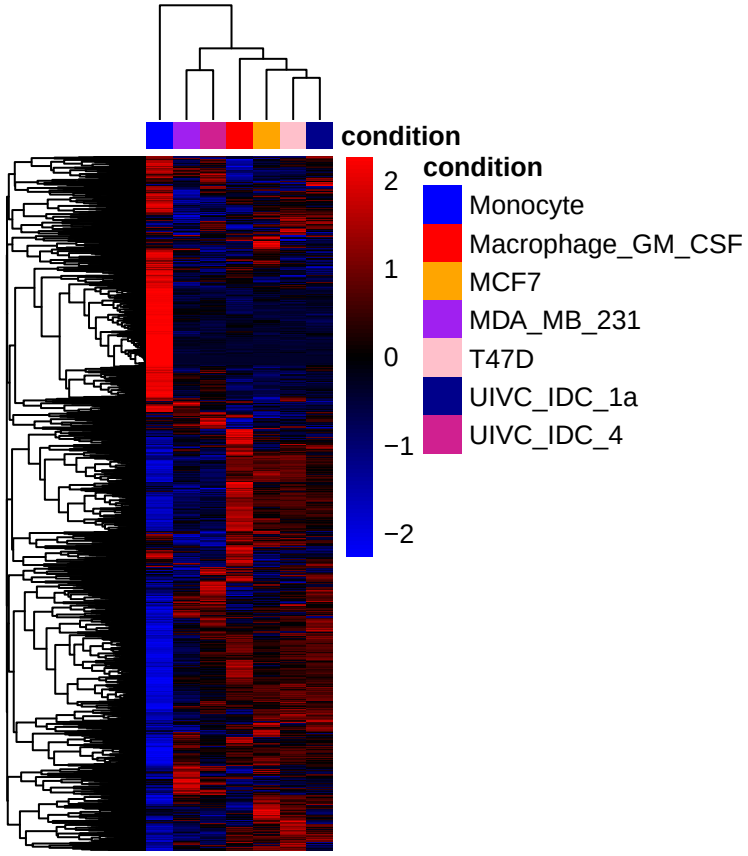
Para que nuestra heatmap muestre el nombre de las condicones experimentales necesitamos agregar las etiquetas necesarias:

```
# Crear el DataFrame de anotaciones de columnas
my_sample_col <- data.frame(
  condition = factor(colnames(mat_data), levels = c("Monocyte", "Macrophage_GM_CSF", "MCF7",
                                                    "MDA_MB_231", "T47D", "UIVC_IDC_1a",
                                                    "UIVC_IDC_4")))

row.names(my_sample_col) <- colnames(mat_data)


# Definir los colores para las anotaciones
my_colour <- list(
  condition = c(Monocyte = "blue",  Macrophage_GM_CSF = "red", MCF7 = "orange",
                MDA_MB_231 = "purple", T47D = "pink", UIVC_IDC_1a = "darkblue",
                UIVC_IDC_4 = "violetred"))
```

Ahora si podemos generar el mapa de calor:

```
pheatmap(mat_data,
         color= colorRampPalette(c("blue", "black", "red"))(100),
         fontsize_col = 8,
         fontsize_row = 8,
         show_rownames = F,
         show_colnames = F,
         cluster_rows = T,
         cluster_cols = T,
         border_color = "grey",
```

```
          scale = "row",
          cellwidth = 10,
          legend = T,
          annotation_legend = T,
          treeheight_col = 40,
          annotation_col = my_sample_col,
          annotation_colors = my_colour,
          annotation_names_col = T
)
```



Ahora veamos en mapa de calor de ambos lotes:

```
# Cargar datos
setwd("D:/marval_windows/JR_MARVAL/himfg/maestria/rnaseq_macrophage/DEA_ballgown_5_all_samples/batch/ba
list.files()
```

```
##  [1] "batch_pyjn.ipynb"
##  [2] "batch_pyjn_function.ipynb"
##  [3] "fpkm_all_samples_with_genes_wiso_mean_L1&2.csv"
##  [4] "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv"
##  [5] "fpkm_all_samples_with_genes_wiso_mean_L1.csv"
##  [6] "fpkm_all_samples_with_genes_wiso_mean_L1_median.csv"
##  [7] "fpkm_macs_with_genes_wiso_mean_L1.csv"
##  [8] "fpkm_macs_with_genes_wiso_mean_L1_median.csv"
##  [9] "fpkm_without_gmcsf_with_genes_wiso_mean_L1.csv"
```

```
## [10] "heatmap_all_data.R"
## [11] "rna_pca_batc.html"
## [12] "rna_pca_batc.Rmd"
## [13] "rna_pca_batc_files"
## [14] "work_flow_transcriptome.png"
```

```
data <- read.table(file = "fpkm_all_samples_with_genes_wiso_mean_L1&2_median.csv", sep = ",", head=T)
head(data)
```

```
##   gene_name  Monocyte Macrophage_GM_CSF   HS578T     MCF7 MDA_MB_231  MBCDF_16
## 1 5_8S_rRNA 0.3452532         0.3550938 1.929156 1.724103  0.5676245 1.3974848
## 2      A1BG 2.5412110         1.5655180 1.269786 1.979829  3.6319900 0.5357325
## 3      AAAS 4.6008125         4.4279105 3.978366 5.367146  4.1589950 3.1802840
## 4      AACS 2.5722070         7.4350905 4.838973 4.435555  3.6727685 3.6378580
## 5     AAGAB 4.8137193         6.1167155 6.306377 5.044572  4.3964470 4.9057655
## 6      AAK1 1.2180500         1.0053815 0.000000 0.333017  0.2543085 0.6730605
##        T47D UIVC_IDC_2 UIVC_IDC_3 UIVC_IDC_1b UIVC_IDC_9 UIVC_IDC_1a UIVC_IDC_4
## 1 0.4452117  0.8188543  0.5956440   0.2123882  0.6108658    0.588970  0.3264535
## 2 1.3169720  1.6697830  0.5638565   1.4144670  1.2771525    2.043205  2.5013130
## 3 6.0826490  5.6092625  4.5510715   5.7911795  3.0750505    5.443699  4.2996540
## 4 4.9286740  4.5759575  5.5871660   5.1829360  5.7738380    5.250632  4.0155305
## 5 5.3096805  6.3157500  6.3249303   5.7186943  7.0818243    5.610289  4.6367710
## 6 0.5275170  0.0000000  0.0000000   0.0000000  0.6188685    0.365997  0.3640385
```

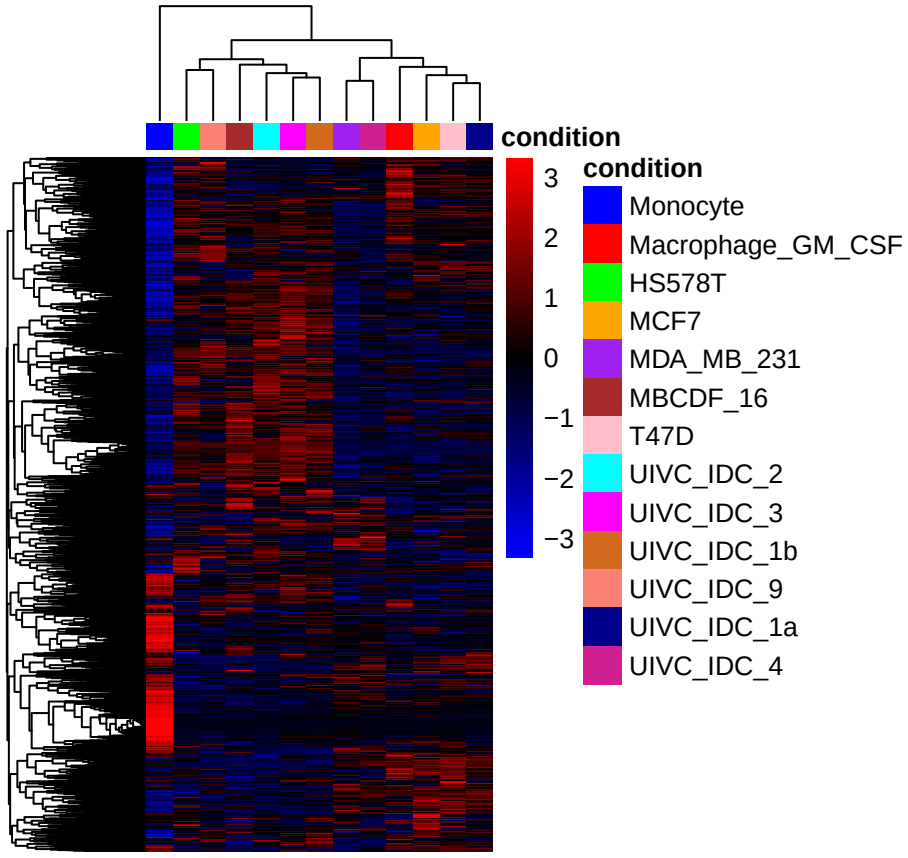
```
data <- data[!(rowSums(data[, -1]) == 0), ]
rownames(data) <- data[,1]
samp2 <- data[,-1]
mat_data <- data.matrix(samp2[,1:ncol(samp2)])
colnames(data)
```

```
##  [1] "gene_name"         "Monocyte"          "Macrophage_GM_CSF"
##  [4] "HS578T"            "MCF7"              "MDA_MB_231"
##  [7] "MBCDF_16"          "T47D"              "UIVC_IDC_2"
## [10] "UIVC_IDC_3"        "UIVC_IDC_1b"       "UIVC_IDC_9"
## [13] "UIVC_IDC_1a"       "UIVC_IDC_4"
```

```
# Crear el DataFrame de anotaciones de columnas
my_sample_col <- data.frame(
  condition = factor(colnames(mat_data), levels = c("Monocyte", "Macrophage_GM_CSF", "MCF7",
                                                    "MDA_MB_231", "T47D", "UIVC_IDC_1a",
                                                    "UIVC_IDC_4", "UIVC_IDC_9",
                                                    "HS578T", "MBCDF_16",  "UIVC_IDC_2",
                                                     "UIVC_IDC_3", "UIVC_IDC_1b" )))

row.names(my_sample_col) <- colnames(mat_data)


# Definir los colores para las anotaciones
my_colour <- list(
  condition = c(Monocyte = "blue",  Macrophage_GM_CSF = "red", HS578T = "green", MCF7 = "orange",
                MDA_MB_231 = "purple", MBCDF_16 = "brown", T47D = "pink",
                UIVC_IDC_2 = "cyan", UIVC_IDC_3 = "magenta", UIVC_IDC_1b = "chocolate",
                UIVC_IDC_9 = "salmon", UIVC_IDC_1a = "darkblue", UIVC_IDC_4 = "violetred"))
```

Ahora si podemos generar el mapa de calor:

```
pheatmap(mat_data,
         color= colorRampPalette(c("blue", "black", "red"))(100),
         fontsize_col = 8,
         fontsize_row = 8,
         show_rownames = F,
         show_colnames = F,
         cluster_rows = T,
         cluster_cols = T,
         border_color = "grey",
         scale = "row",
         cellwidth = 10,
         legend = T,
         annotation_legend = T,
         treeheight_col = 40,
         annotation_col = my_sample_col,
         annotation_colors = my_colour,
         annotation_names_col = T
)
```



El heatmap muestra un claro efecto por lote... ¿Cómo se transforman los valores del PCA para poder usarlos en la construcción del mapa de calor?

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
```

```
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Spanish_Mexico.utf8  LC_CTYPE=Spanish_Mexico.utf8
## [3] LC_MONETARY=Spanish_Mexico.utf8 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Mexico.utf8
##
## time zone: Etc/GMT+6
## tzcode source: internal
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] RColorBrewer_1.1-3 colorspace_2.1-1   pheatmap_1.0.12    ggfortify_0.4.17
## [5] ggplot2_3.5.1
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.5      dplyr_1.1.4       compiler_4.3.1    highr_0.11
##  [5] tidyselect_1.2.1  stringr_1.5.1     gridExtra_2.3     tidyr_1.3.1
##  [9] scales_1.3.0      yaml_2.3.8        fastmap_1.2.0     R6_2.5.1
## [13] labeling_0.4.3    generics_0.1.3    knitr_1.48        tibble_3.2.1
## [17] munsell_0.5.1     pillar_1.9.0      rlang_1.1.3       utf8_1.2.4
## [21] stringi_1.8.4     xfun_0.46         cli_3.6.2         withr_3.0.1
## [25] magrittr_2.0.3    digest_0.6.36     rstudioapi_0.16.0 lifecycle_1.0.4
## [29] vctrs_0.6.5       evaluate_0.24.0   glue_1.7.0        farver_2.1.2
## [33] fansi_1.0.6       rmarkdown_2.27    purrr_1.0.2       tools_4.3.1
## [37] pkgconfig_2.0.3   htmltools_0.5.8.1
```