Visualización de Bases de datos

Gabriel Illanes

Centro de Matemática
Facultad de Ciencias
Universidad de la República

12 de diciembre de 2018

Los datos

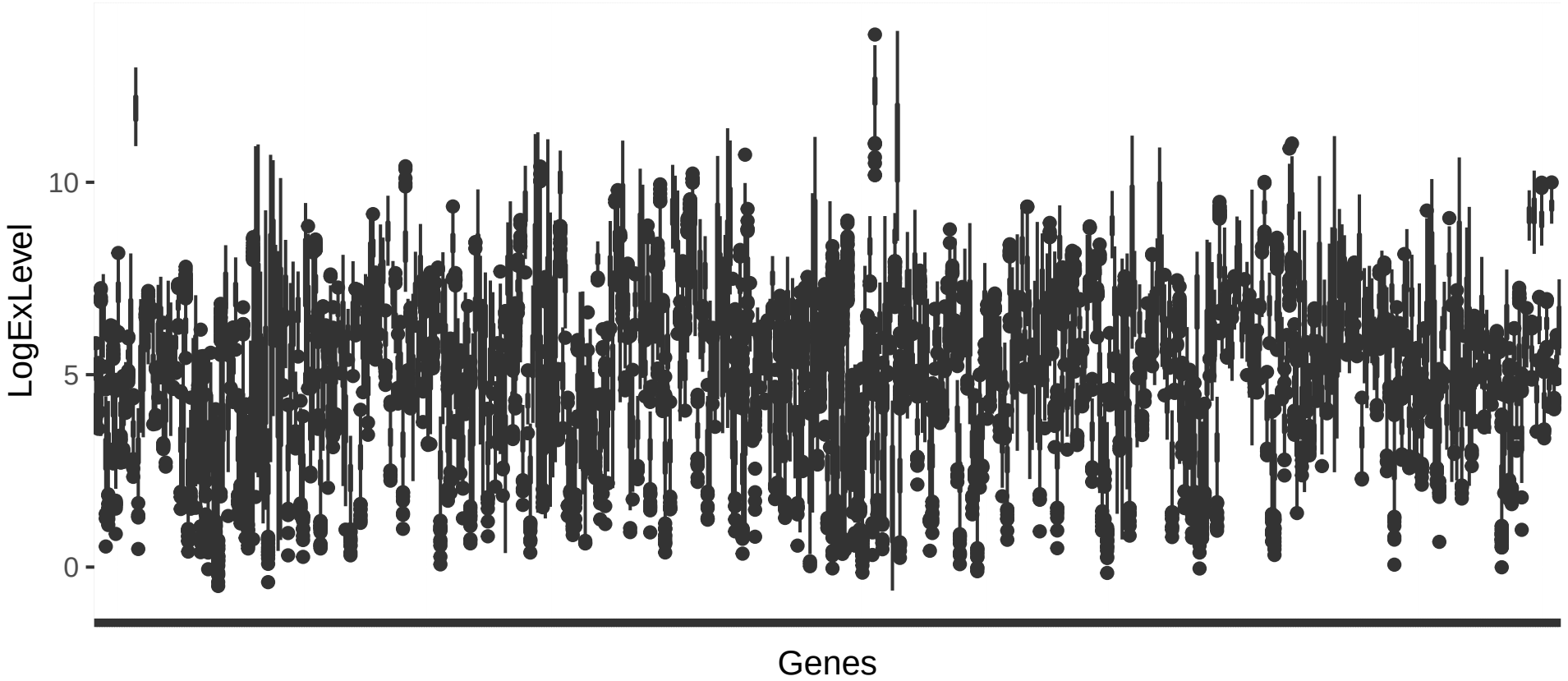
- 25 donantes de sangre "sanos".
- 28 estímulos, incluyendo un estímulo de referencia.
- 587 genes relacionados al sistema inmune.
- La sangre de cada individuo se inserta en tubos con estímulo. Luego de 22hs de incubación, se obtiene la cantidad (normalizada) de ARNm relativas a cada uno de los genes de referencia.

```
## # A tibble: 6 x 11
##   Donor Gender StimulusName ABCB1  ABL1   ADA   AHR AICDA  AIRE   APP  ARG1
##   <int> <fct>  <fct>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    25 F      Null         217.  115.   78.8  400. 12.9  14.1   214.  64.7
## 2    25 F      C12IEDAP     233.  117.   83.0  457. 21.3  19.1   180.  66.2
## 3    25 F      aCD3aCD28     75.9  64.5  45.1  785.  3.99  9.70  258.  30.2
## 4    25 F      CPPD         181.  111.   97.1  444. 18.4  18.4   237.  67.9
## 5    25 F      Gardiqiumod  182.  123.  765.   963. 16.4  27.4   215.  73.3
## 6    25 F      FLA          183.  123.  689.  1097. 10.9  21.9   224.  57.5
```

Boxplots

```
LabEx_long <- LabEx %>%
  select(c(Donor, StimulusName, ABCB1:TUBB)) %>%
  gather(Genes, ExpressionLevel, ABCB1:TUBB, factor_key = TRUE) %>%
  mutate(LogExLevel = log(ExpressionLevel))
```

```
ggplot(LabEx_long, aes(x = Genes, y = LogExLevel)) +
  geom_boxplot() +
  theme(plot.margin = margin(0, 0, 10, 0, "cm"),
        axis.text.x = element_text(size = 0))
```



Escalamiento Multidimensional

Queremos visualizar las observaciones en un espacio de dimensión baja (2 o 3). Las opciones más comunes son Componentes Principales (PCA, o Kernel PCA) o Escalamiento Multidimensional (MDS).

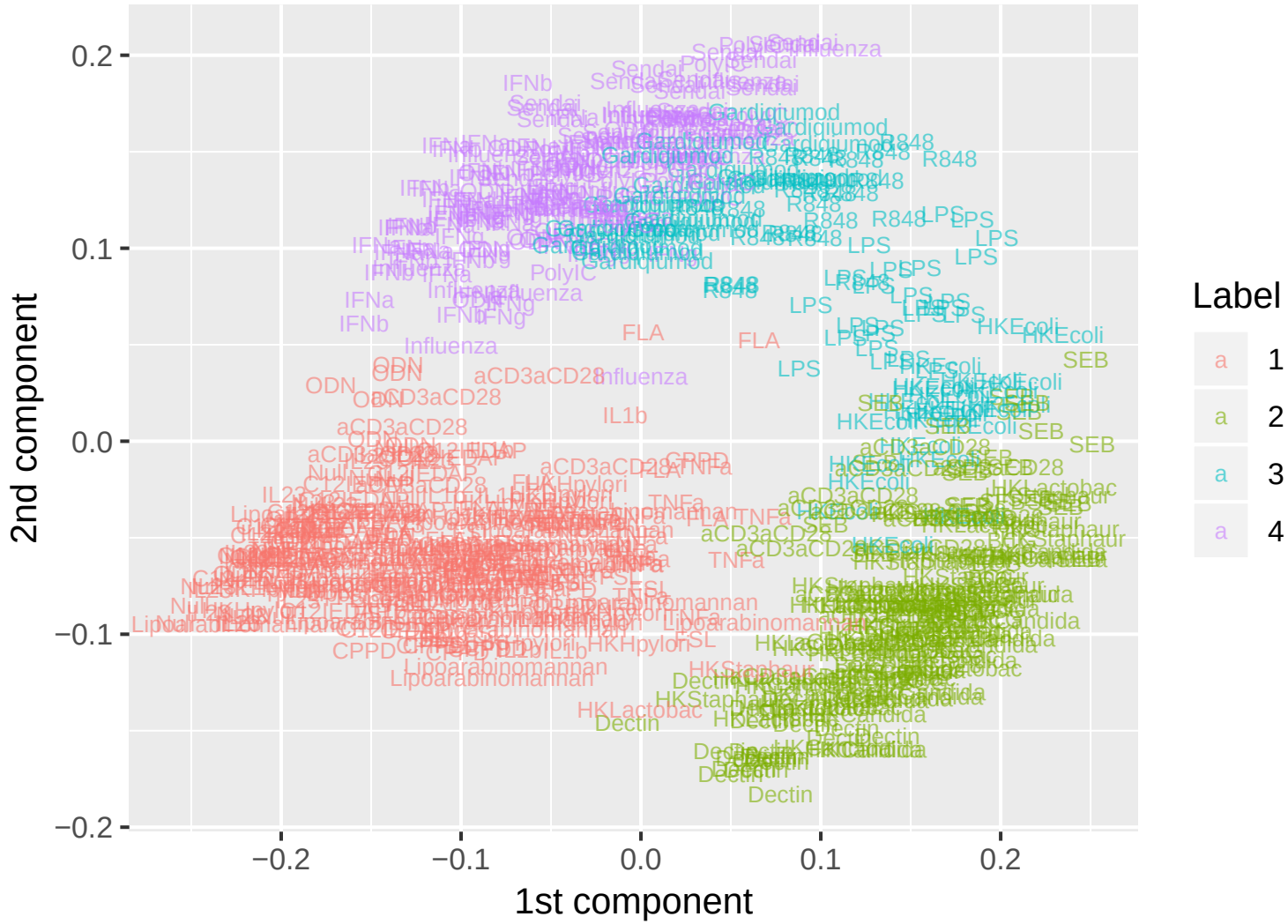
PCA es muy conocido y usado, pero la necesidad de escalar los datos implica problemas que llevarían a un análisis poco robusto.

Por lo tanto, si podemos encontrar una distancia adecuada para estos datos (que evite escalamientos), podemos aplicar MDS. Proponemos la distancia de canberra (escalada)

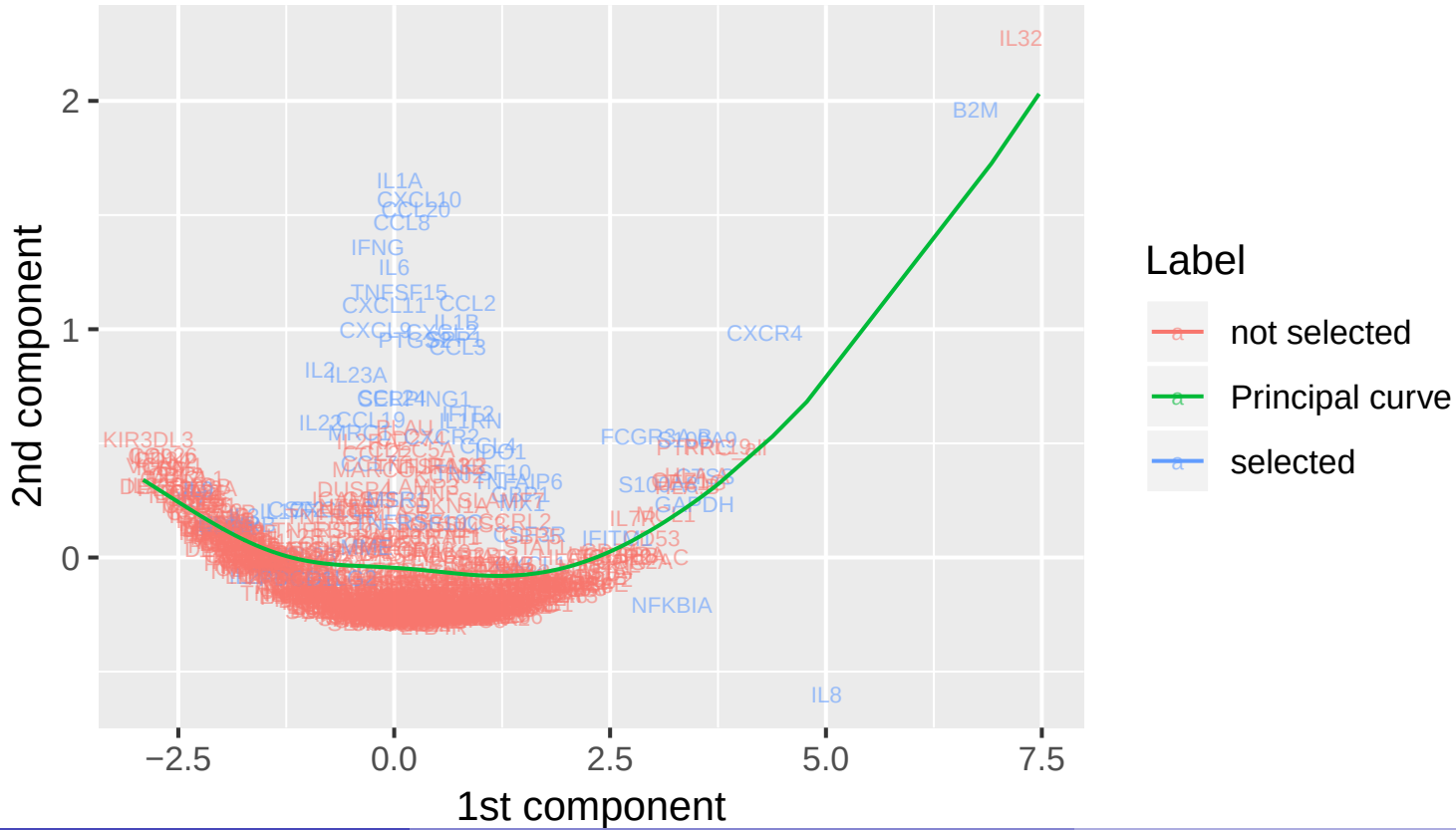
$$d_C(u,v) = \sum_{i=1}^{n} \frac{|u_i - v_i|}{u_i + v_i}$$

$$d_C^*(u,v) = -\log(1 - d_C(u,v)/n)$$

```r
LabEx_stim <- LabEx %>%
  select(ABCB1:TUBB)
LabEx$Group_c <- factor(clusters) # hice trampa, lo muestro más tarde
mds_c = cmdscale(-log(1- dist(LabEx_stim, method = "canberra")/ncol(LabEx_stim)), k = 2)
LabEx$PC1_s <- mds_c[,1]
LabEx$PC2_s <- mds_c[,2]
ggplot(data = LabEx, aes(x = PC1_s, y = PC2_s, colour = Group_c, label = StimulusName)) +
  labs(x = "1st component", y = "2nd component", colour = "Label") +
  geom_text(aes(label = StimulusName), size = 2.5, alpha = 0.6) +
  theme(plot.margin = margin(0, 2, 8, 2, "cm"))
```



```r
mds_g <- cmdscale(-log(1-dist(t(LabEx_stim), method = "canberra")/ncol(t(LabEx_stim))), k = 296)
pcur <- principal_curve(mds_g, thresh = 1e-5, trace = FALSE, maxit = 100)
dist_pcur <- apply((pcur$s - mds_g)^2, 1, sum)
mds_g <- as.data.frame(mds_g)
names(mds_g) <- c("PC1", "PC2")
mds_g$Pcur1 <- pcur$s[,1]
mds_g$Pcur2 <- pcur$s[,2]
threshold <- 1.1
mds_g$Group <- c("selected", "not selected")[(dist_pcur < threshold) + 1]
mds_g$Genes <- names(LabEx_stim)
mds_g$Dist <- dist_pcur
mds_g$index <- 1:587
ggplot(data = mds_g, aes(x = PC1, y = PC2, colour = Group, label = Genes)) +
  labs(x = "1st component", y = "2nd component", colour = "Label") +
  geom_text(aes(label = Genes), size = 2, alpha = 0.6) +
  geom_line(data = mds_g, aes(x = Pcur1, y = Pcur2, colour = "Principal curve")) +
  theme(plot.margin = margin(0, 2, 10, 2, "cm"))
```



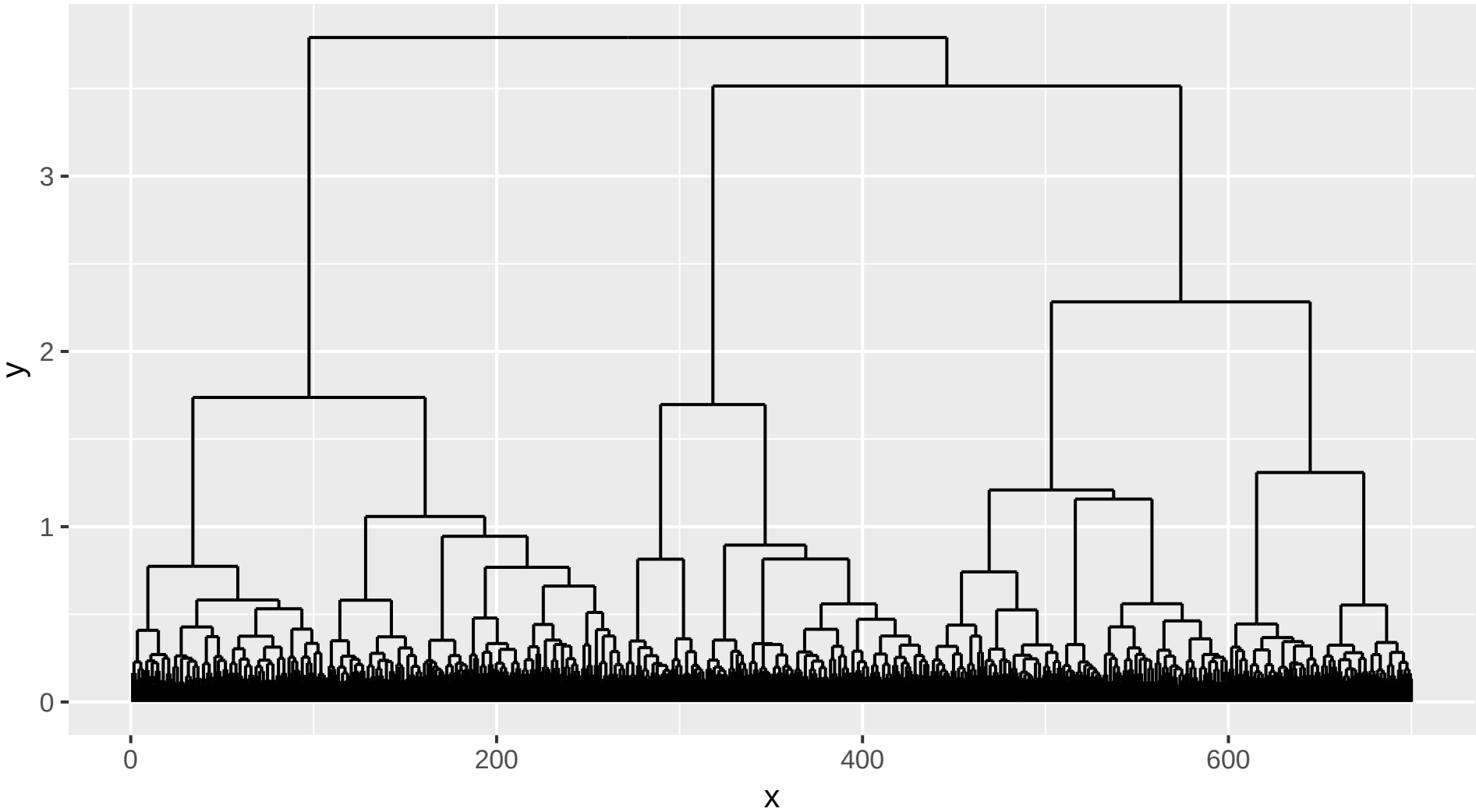
Agrupamiento jerárquico

Una vez que tenemos una idea sobre como se agrupan los distintos estímulos y genes, podemos elegir el método más adecuando para aplicar agrupamiento jerárquico.

En el caso de los estímulos, parece haber "nubes" que contienen a distintos estímulos, podemos suponer que el método *ward* es adecuado.

```r
hc_c <- hclust(-log(1- dist(LabEx_stim, method = "canberra")/ncol(LabEx_stim)),
               method = "ward.D2")
number_clusters = 4
clusters <- cutree(hc_c, k = number_clusters)
LabEx$Group_c <- factor(clusters)
LabEx_table <- LabEx %>%
  select(Donor, StimulusName, Group_c) %>%
  group_by(Group_c, StimulusName) %>%
  dplyr::summarize(counts = n()) %>%
  spread(Group_c, counts, fill= 0, sep = "_")
LabEx_table$Label_c <- max.col(LabEx_table[,(1:number_clusters) + 1])
LabEx_table <- LabEx_table %>%
  arrange(Label_c)
```

```r
dhc <- as.dendrogram(hc_c)
ddata <- dendro_data(dhc, type = "rectangle")
ggplot(segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  theme(plot.margin = margin(0, 0, 8, 0, "cm"))
```



```r
print.data.frame(LabEx_table)
```

```
##          StimulusName Group_c_1 Group_c_2 Group_c_3 Group_c_4 Label_c
## 1           C12IEDAP        25         0         0         0       1
## 2               CPPD        25         0         0         0       1
## 3                FLA        25         0         0         0       1
## 4                FSL        25         0         0         0       1
## 5          HKHpylori        25         0         0         0       1
## 6               IL1b        25         0         0         0       1
## 7               IL23        25         0         0         0       1
## 8  Lipoarabinomannan        25         0         0         0       1
## 9               Null        25         0         0         0       1
## 10              TNFa        25         0         0         0       1
## 11         aCD3aCD28         6        19         0         0       2
## 12               BCG         0        25         0         0       2
## 13            Dectin         0        25         0         0       2
## 14         HKCandida         0        25         0         0       2
## 15        HKLactobac         1        24         0         0       2
## 16        HKStaphaur         1        24         0         0       2
## 17               SEB         0        25         0         0       2
## 18       Gardiqiumod         0         0        25         0       3
## 19           HKEcoli         0         0        25         0       3
## 20               LPS         0         0        25         0       3
## 21              R848         0         0        25         0       3
## 22              IFNa         0         0         0        25       4
## 23              IFNb         0         0         0        25       4
## 24              IFNg         0         0         0        25       4
## 25         Influenza         0         0         0        25       4
## 26               ODN        12         0         0        13       4
## 27            PolyIC         0         0         0        25       4
## 28            Sendai         0         0         0        25       4
```

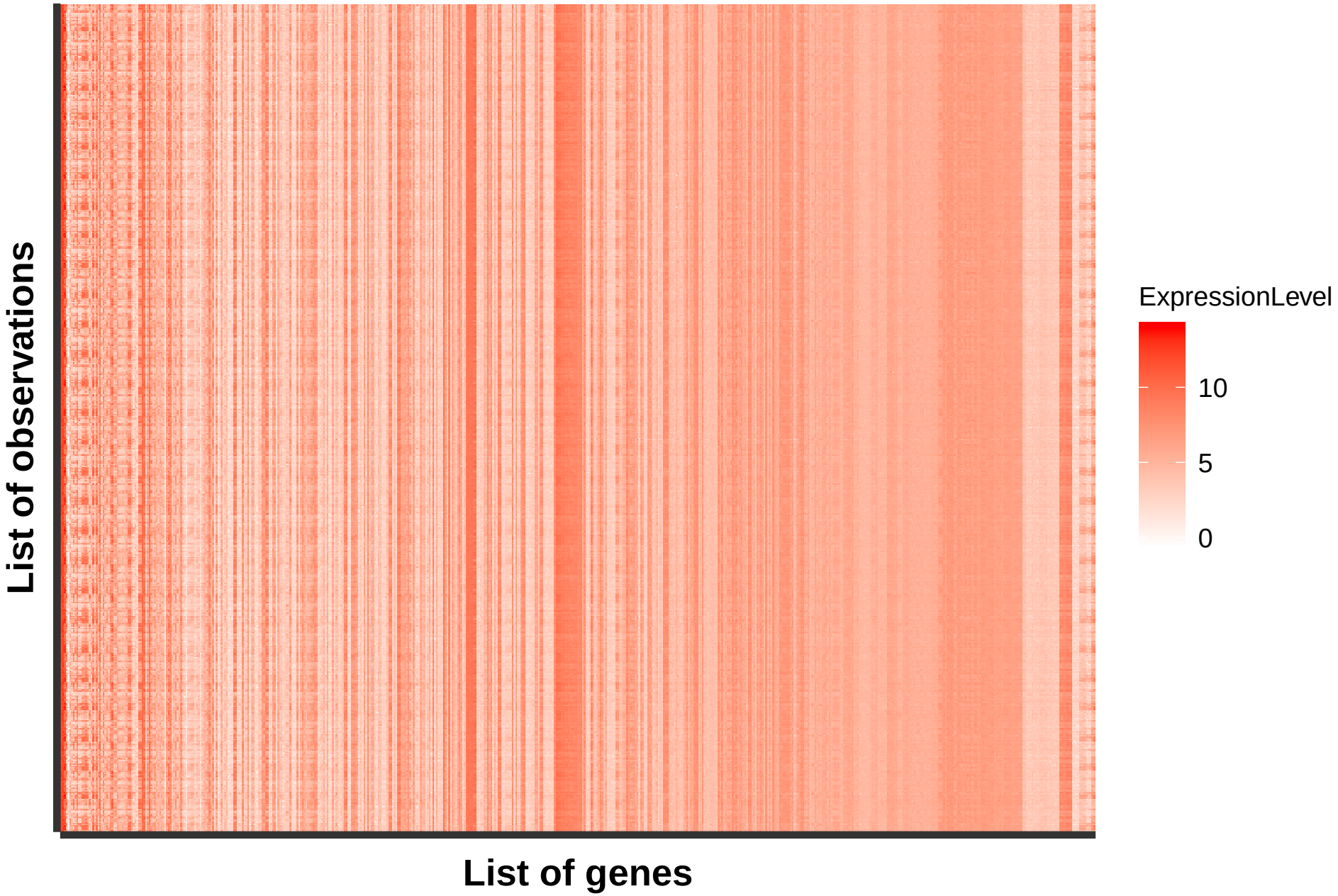
```r
cat("Error de clasificación:", sum((25-apply(LabEx_table[,(1:number_clusters)+1], 1, max))/nrow(LabEx_stim)))
```

```
## Error de clasificación: 0.02857143
```

Heatmaps

La información de los agrupamientos jerárquicos nos ayuda a construir heatmaps más informativos.

```
LabEx_hm <- LabEx
LabEx_hm[,37:623] <- LabEx_stim[hc_c$order, hc_g$order]
LabEx_long <- LabEx_hm %>%
  select(c(Donor, StimulusName, ABCB1:TUBB)) %>%
  gather(Genes, ExpressionLevel, ABCB1:TUBB, factor_key = TRUE) %>%
  mutate(LogExLevel = log(ExpressionLevel), ExpressionLevel = NULL) %>%
  mutate(DonorStim = paste(StimulusName, Donor), StimulusName = NULL, Donor = NULL)
hm = ggplot(LabEx_long) +
  aes(x = Genes, y = DonorStim) +
  geom_tile(aes(fill = LogExLevel)) +
  scale_fill_gradient(low = "white", high = "red") +
  ylab("List of observations") +
  xlab("List of genes") +
  theme(legend.title = element_text(size = 10),
        legend.text = element_text(size = 10),
        plot.margin = margin(1, 0, 5, 0, "cm"),
        axis.title=element_text(size=14,face="bold"),
        axis.text.x = element_text(size = 0),
        axis.text.y = element_text(size = 0)) +
  labs(fill = "ExpressionLevel")
```


List of observations
List of genes
ExpressionLevel
10
5
0

Conclusiones

- Hay muchísimas herramientas de visualización, y todas ayudan en un análisis exploratorio, en donde no hay muchas referencias de las cuales agarrarse, y puede que no sepamos lo que estamos buscando.
- Las distintas herramientas de visualización pueden retroalimentarse, por lo cual hay que conocerlas y dominarlas.
- Es importante tener conocimientos avanzados de estadística para detectar dificultades en el estudio de las bases de datos y poder solucionarlas.
- El uso de *tidyverse* es muy recomendado, ya que aporta mucho control sobre los objetos que creamos. Puede resultar una curva de aprendizaje empinada, pero vale la pena.
- El uso de *R Markdown* y de *knitr* también es recomendado, ya que ayudan a realizar informes de calidad (esta presentación fue hecha usando *knitr*).
- Sé que sé poco, cualquier ayuda o sugerencia es bienvenida :-3

¡Muchas gracias!