Clasificador diseasecodes TCGA. Red neuronal con sondas de mayor variabilidad

Alberto Joven Álvarez

16 de noviembre, 2022

Índice

1 Lista de librerías empleadas

```
library(knitr)
library(dplyr)
library(readr)
library(curatedTCGAData)
library(TCGAutils)
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
library(FDb.InfiniumMethylation.hg19)
library(impute)
library(SummarizedExperiment)
library(GEOquery)
library(caret)
library(FactoMineR)
library(factoextra)
library(Rtsne)
library(ggplot2)
library(keras)
library(sva)
library(randomForest)
```

2 Planteamiento general del trabajo

Se plantea como pregunta inicial de interés biológico:

¿Es posible entrenar una red neuronal para que a partir de los datos de posiciones metiladas de una muestra sea capaz de identificar a qué categoría de tejido pertenece clasificándola en una de las 34 clases de tumores que establece el proyecto The Cancer Genome Atlas o bien considerarla como muestra no tumoral?

Este trabajo se inspira en el artículo (Capper et al. 2018) que propone un clasificador Machine Learning para la determinación del tipo de tumor del SNC a partir de los datos de metilación (betavalues) de las 20.000 sondas del array Illumina 450k que presentan una mayor variabilidad. También ha servido de guía el planteamiento descrito en (Maros et al. 2020) que expone una aplicación general de esa metodología.

En breve síntesis el trabajo ha consistido en:

1. Descarga de los datos: desde el repositorio del Broad Institute GDAC Firehouse se descargaron todos los análisis del proyecto TCGA con datos de metilación obtenidos con sondas Illumina 450k.
2. Se construye un objeto R *Summarized Experiment* a partir de los valores betas de todos los estudios, una vez suprimidas las sondas con más de un 10% de muestras con valor ausente.
3. Se eliminan las sondas previsiblemente problemáticas de acuerdo con los trabajos (Price et al. 2013) y (Zhou, Laird, y Shen 2017).
4. Se seleccionan las muestras con códigos TCGA: 1 "Primary Solid Tumor" y 03 "Primary Blood Derived Cancer-Peripheral Blood" (TB).
5. Se desglosan las muestras en los grupos train y test (75% y 25%) utilizando la librería *caret* (Kuhn 2017).
6. Se calcula la desviación estándar de las sondas, seleccionándose las 20.000 sondas con mayor variabilidad.
7. Análisis gráfico previo: se muestran los gráficos de componentes principales y la técnica t-SNE (Stochastic Neighbor Embedding), así como los histogramas las sondas con mayor variabilidad, para ello empleo el grupo de sondas de mayor variabilidad y el grupo de muestras *train*.
8. Se diseña el algoritmo Red Neuronal y se evalúa mediante cross-validation entrenando tan solo con las muestras *train*.
9. Se selecciona el algoritmo de mejor resultado y se evalúa con las muestras *test*.
10. Se entrena un algoritmo random Forest y se evalúa su grado de acierto.

2.1 Tabla de códigos de tumor del proyecto TCGA

El detalle de los códigos de tumor del TCGA usados en este cuaderno es:

```
read.delim("C:/TFM UOC/R/codigos_tumores.txt",  sep="\t") %>% kable()
```

CODIGO	DESCRIPCION
ACC	Adrenocortical carcinoma
BLCA	Bladder Urothelial Carcinoma
BRCA	Breast invasive carcinoma
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma
CHOL	Cholangiocarcinoma
CNTL	Controls
COAD	Colon adenocarcinoma
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma
ESCA	Esophageal carcinoma
GBM	Glioblastoma multiforme
HNSC	Head and Neck squamous cell carcinoma
KICH	Kidney Chromophobe
KIRC	Kidney renal clear cell carcinoma
KIRP	Kidney renal papillary cell carcinoma
LAML	Acute Myeloid Leukemia
LGG	Brain Lower Grade Glioma
LIHC	Liver hepatocellular carcinoma
LUAD	Lung adenocarcinoma
LUSC	Lung squamous cell carcinoma
MESO	Mesothelioma
OV	Ovarian serous cystadenocarcinoma
PAAD	Pancreatic adenocarcinoma
PCPG	Pheochromocytoma and Paraganglioma
PRAD	Prostate adenocarcinoma
READ	Rectum adenocarcinoma

CODIGO	DESCRIPCION
SARC	Sarcoma
SKCM	Skin Cutaneous Melanoma
STAD	Stomach adenocarcinoma
TGCT	Testicular Germ Cell Tumors
THCA	Thyroid carcinoma
THYM	Thymoma
UCEC	Uterine Corpus Endometrial Carcinoma
UCS	Uterine Carcinosarcoma
UVM	Uveal Melanoma

2.2 Tabla con los códigos de las muestras

El detalle de los códigos de las muestras TCGA usados en este cuaderno es:

```
read.table("C:/TFM UOC/R/codigos_muestras.txt", head=TRUE, sep="\t") %>% kable()
```

Code	Definition	Short.Letter.Code
1	Primary Solid Tumor	TP
2	Recurrent Solid Tumor	TR
3	Primary Blood Derived Cancer - Peripheral Blood	TB
4	Recurrent Blood Derived Cancer - Bone Marrow	TRBM
5	Additional - New Primary	TAP
6	Metastatic	TM
7	Additional Metastatic	TAM
8	Human Tumor Original Cells	THOC
9	Primary Blood Derived Cancer - Bone Marrow	TBM
10	Blood Derived Normal	NB
11	Solid Tissue Normal	NT
12	Buccal Cell Normal	NBC
13	EBV Immortalized Normal	NEBV
14	Bone Marrow Normal	NBM
15	sample type 15	15SH
16	sample type 16	16SH
20	Control Analyte	CELLC
40	Recurrent Blood Derived Cancer - Peripheral Blood	TRB
50	Cell Lines	CELL
60	Primary Xenograft Tissue	XP
61	Cell Line Derived Xenograft Tissue	XCL
99	sample type 99	99SH

En este trabajo usaré solo las muestras de códigos 01 Primary Solid Tumor (TP), 03 Primary Blood Derived Cancer - Peripheral Blood (TB).

2.3 Detalle de las muestras descargadas para el estudio

```
read_delim("C:/TFM UOC/R/muestras.txt", delim = "\t",
    escape_double = FALSE, col_types = cols(TP = col_integer()),
```

```
    trim_ws = TRUE)  %>%  kable()
```

title assay	1-TP	3-TB	Total
ACC_Methylation-20160128_assays	80	0	80
BLCA_Methylation-20160128_assays	412	0	412
BRCA_Methylation_methyl450-20160128_assays	783	0	783
CESC_Methylation-20160128_assays	307	0	307
CHOL_Methylation-20160128_assays	36	0	36
COAD_Methylation_methyl450-20160128_assays	293	0	293
DLBC_Methylation-20160128_assays	48	0	48
ESCA_Methylation-20160128_assays	185	0	185
GBM_Methylation_methyl450-20160128_assays	140	0	140
HNSC_Methylation-20160128_assays	528	0	528
KICH_Methylation-20160128_assays	66	0	66
KIRC_Methylation_methyl450-20160128_assays	319	0	319
KIRP_Methylation_methyl450-20160128_assays	275	0	275
LAML_Methylation_methyl450-20160128_assays	0	194	194
LGG_Methylation-20160128_assays	516	0	516
LIHC_Methylation-20160128_assays	377	0	377
LUAD_Methylation_methyl450-20160128_assays	458	0	458
LUSC_Methylation_methyl450-20160128_assays	370	0	370
MESO_Methylation-20160128_assays	87	0	87
OV_Methylation_methyl450-20160128_assays	10	0	10
PAAD_Methylation-20160128_assays	184	0	184
PCPG_Methylation-20160128_assays	179	0	179
PRAD_Methylation-20160128_assays	498	0	498
READ_Methylation_methyl450-20160128_assays	98	0	98
SARC_Methylation-20160128_assays	261	0	261
SKCM_Methylation-20160128_assays	105	0	105
STAD_Methylation_methyl450-20160128	395	0	395
TGCT_Methylation-20160128_assays	134	0	134
THCA_Methylation-20160128_assays	503	0	503
THYM_Methylation-20160128_assays	124	0	124
UCEC_Methylation_methyl450-20160128_assays	431	0	431
UCS_Methylation-20160128_assays	57	0	57
UVM_Methylation-20160128_assays	80	0	80
Total	8339	194	8533

3 Descarga de los datos

3.1 Obtención de los beta_values

El código empleado para la descarga de los datos fue:

```
estudios <- curatedTCGAData(
  diseaseCode = c("ACC", "BLCA"), assays = "Methyl*", version = "1.1.38", dry.run = FALSE
)
```

Sucesivamente se descargaron los 33 códigos de tumor que se han detallado anteriormente (en el ejemplo de código solo estan los códigos **ACC** y **BLCA**).

Con el código siguiente se extrajo uno a uno la información (objetos) contenida en las descargas anteriores. Este código se ejecuta para cada uno de los 33 assays expuestos en la primera columna de la tabla anterior:

```
matriz1 <- assays(estudios)
matriz1 <- matriz1[["BLCA_Methylation-20160128"]]
matriz1 <- as.matrix(matriz1)

matriz2 <- assays(estudios)
matriz2 <- matriz2[["ACC_Methylation-20160128"]]
matriz2 <- as.matrix(matriz2)
```

El paso siguiente fue unir sucesivamente una a una (en otro caso daba error) las matrices con los valores betas con *cbind*:

```
matriz <- cbind(matriz1, matriz2)
rm(matriz1, matriz2)

matriz <- cbind(matriz, matriz3)
rm(matriz3)
```

3.2 Tratamiento de los datos faltantes

Tratamiento de los valores faltantes **NAs**: se eliminaron las sondas con más del 10% de las observaciones NAs, quedan 395319:

```
matriz2 <- matriz[nas < 9707 * 0.1, ]
```

El resto de valors faltantes se imputaron con la función *impute* de la librería del mismo nombre, que utiliza el nearest neighbor averaging para su cálculo. Se divide la matriz de datos en 9 tramos para posibilitar el cálculo y, una vez calculada la estimación de lo valores faltantes, se vuelven a unir. El código utilizado fue:

```
library(impute)

matriz3 <- matriz2[ , 1:1000]
matriz4 <- matriz2[ , 1001:2000]
matriz5 <- matriz2[ , 2001:3000]
matriz6 <- matriz2[ , 3001:4000]
matriz7 <- matriz2[ , 4001:5000]
matriz8 <- matriz2[ , 5001:6000]
matriz9 <- matriz2[ , 6001:7000]
matriz10 <- matriz2[ , 7001:8000]
matriz11 <- matriz2[ , 8001:9000]
matriz12 <- matriz2[ , 9001:9707]

rm(matriz2)

matriz33 <- impute.knn(matriz3)
matriz44 <- impute.knn(matriz4)
matriz55 <- impute.knn(matriz5)
matriz66 <- impute.knn(matriz6)
matriz77 <- impute.knn(matriz7)
matriz88 <- impute.knn(matriz8)
```

```
matriz99 <- impute.knn(matriz9)
matriz1010 <- impute.knn(matriz10)
matriz1111 <- impute.knn(matriz11)
matriz1212 <- impute.knn(matriz12)


matriz_sna <- cbind(matriz33$data, matriz44$data, matriz55$data, matriz66$data,
                    matriz77$data, matriz88$data,
                    matriz99$data, matriz1010$data, matriz1111$data, matriz1212$data)
```

3.3 Construcción de Data Frame con los fenotipos de las muestras

El Data.Frame con los fenotipos de las muestras lo construyo manualmente. Los campos a incorporar son:

1. Nombre de las muestras: el bar-code que identifica cada muestra en la matriz con los beta-values.
2. Bar-code: El bar-code de identificación de las muestras.
3. El identificador de individuo (los 12 primeros caracteres del bar-code).
4. Categoría: es el tipo de muestra: 01 Tumor primario, 11 Tejido normal etc...
5. type: es la descripción del tumor extraída de los datos de fenotipos de cada ensayo: Histological Type
6. disease_code: tipo de tumor extraído de lkos datos de fenotipos de cada ensayo.
7. assay: identifica el ensayo.
8. label: si la muestra es tumoral, figura el identificador del ensayo. Si la muestra no es tumoral, código de muestra "11" figura la categoría "Control".

```
# Se carga la matriz con todos los datos betas de todos los ensayos descargados:
# sus columnas están rotuladas con los  bar-codes.

load("G:/TFM UOC/datos/matriz.Rda")

# Se cargan los objetos MultiassayExperiment descargados anteriormente

load(file="G:/TFM UOC/estudios/estudios_1")
load(file="G:/TFM UOC/estudios/estudios_2")
load(file="G:/TFM UOC/estudios/estudios_3")
load(file="G:/TFM UOC/estudios/estudios_4")
load(file="G:/TFM UOC/estudios/estudios_5")
load(file="G:/TFM UOC/estudios/estudios_6")
load(file="G:/TFM UOC/estudios/estudios_7")
load(file="G:/TFM UOC/estudios/estudios_8")
load(file="G:/TFM UOC/estudios/estudios_9")
load(file="G:/TFM UOC/estudios/estudios_10")

# Los bar-codes y los 12 primeros caracteres de los mismos
# forman los dos primeros campos.

etiqueta <- data.frame(bar_code = colnames(matriz),
                       sujeto=substring(colnames(matriz), 1, 12))
etiqueta$categoria <- substring(colnames(matriz), 14,15) %>% as.factor()

# para añadir el histological_type y el disease_code

fenotipos1 <- colData(estudios)
```

```
fenotipos2 <- colData(estudios_2)
fenotipos3 <- colData(estudios_3)
fenotipos4 <- colData(estudios_4)
fenotipos5 <- colData(estudios_5)
fenotipos6 <- colData(estudios_6)
fenotipos7 <- colData(estudios_7)
fenotipos8 <- colData(estudios_8)
fenotipos9 <- colData(estudios_9)
fenotipos10 <- colData(estudios_10)
tipos1 <- data.frame(sujeto = rownames(fenotipos1),
                     type = fenotipos1$histological_type,
                     disease_code=fenotipos1$admin.disease_code)
tipos2 <- data.frame(sujeto = rownames(fenotipos2),
                     type = fenotipos2$histological_type,
                     disease_code=fenotipos2$admin.disease_code)
tipos3 <- data.frame(sujeto = rownames(fenotipos3),
                     type = fenotipos3$histological_type,
                     disease_code=fenotipos3$admin.disease_code)
tipos4 <- data.frame(sujeto = rownames(fenotipos4),
                     type = fenotipos4$histological_type,
                     disease_code=fenotipos4$admin.disease_code)
tipos5 <- data.frame(sujeto = rownames(fenotipos5),
                     type = fenotipos5$histological_type,
                     disease_code=fenotipos5$admin.disease_code)
tipos6 <- data.frame(sujeto = rownames(fenotipos6),
                     type = fenotipos6$histological_type,
                     disease_code=fenotipos6$admin.disease_code)
tipos7 <- data.frame(sujeto = rownames(fenotipos7),
                     type = fenotipos7$histological_type,
                     disease_code=fenotipos7$admin.disease_code)
tipos8 <- data.frame(sujeto = rownames(fenotipos8),
                     type = fenotipos8$histological_type,
                     disease_code=fenotipos8$admin.disease_code)
tipos9 <- data.frame(sujeto = rownames(fenotipos9),
                     type = fenotipos9$histological_type,
                     disease_code=fenotipos9$admin.disease_code)
tipos10 <- data.frame(sujeto = rownames(fenotipos10),
                      type = fenotipos10$histological_type,
                      disease_code=fenotipos10$admin.disease_code)

tipos <- rbind(tipos1, tipos2, tipos3, tipos4, tipos5, tipos6,
               tipos7, tipos8, tipos9, tipos10)

etiqueta <- merge(etiqueta, tipos, by="sujeto", all.x=TRUE)

# Para añadir el estudio assay

maps1 <- as.data.frame(sampleMap(estudios))
maps2 <- as.data.frame(sampleMap(estudios_2))
maps3 <- as.data.frame(sampleMap(estudios_3))
maps4 <- as.data.frame(sampleMap(estudios_4))
maps5 <- as.data.frame(sampleMap(estudios_5))
maps6 <- as.data.frame(sampleMap(estudios_6))
```

```
maps7 <- as.data.frame(sampleMap(estudios_7))
maps8 <- as.data.frame(sampleMap(estudios_8))
maps9 <- as.data.frame(sampleMap(estudios_9))
maps10 <- as.data.frame(sampleMap(estudios_10))

tipos_2 <- rbind(maps1, maps2, maps3, maps4, maps5, maps6,
                 maps7, maps8, maps9, maps10)
tipos_2$assay <- as.character(tipos_2$assay)
ensayos <- strsplit(tipos_2$assay, split="_")
ensayos <- lapply(ensayos, "[", 1) %>% unlist()
tipos_2$assay <- ensayos

# elimino duplicados en los indicadores de muestra
tipos_2 <- tipos_2[!duplicated(tipos_2$colname), c(1,3) ]

etiqueta <- merge(etiqueta, tipos_2, by.x="bar_code", by.y= "colname", all.x=TRUE)
rownames(etiqueta) <- etiqueta$bar_code

etiqueta$label <- etiqueta$assay
etiqueta$label[etiqueta$categoria == "11"] <- "Control"

# muestras puestas en el mismo orden que la matriz de expresión
etiqueta <- etiqueta[colnames(matriz), ]

table(etiqueta$assay, useNA="always")
table(etiqueta$label, useNA="always")

sum(colnames(matriz) != rownames(etiqueta))

sujetos <- unique(etiqueta$sujeto)
table(table(etiqueta$sujeto))
```

El data.frame etiqueta, al final tiene el siguiente aspecto (se traspone para mejor visualización:

```
etiqueta <- read.table("C:/TFM UOC/R/etiqueta.txt")

kable(t(etiqueta[1:2, ] ))
```

	TCGA-2F-A9KO-01A-11D-A38H-05	TCGA-2F-A9KP-01A-11D-A38H-05
bar_code	TCGA-2F-A9KO-01A-11D-A38H-05	TCGA-2F-A9KP-01A-11D-A38H-05
sujeto	TCGA-2F-A9KO	TCGA-2F-A9KP
categoria	1	1
type	muscle invasive urothelial carcinoma (pt2 or above)	muscle invasive urothelial carcinoma (pt2 or above)
disease_code	blca	blca
assay	BLCA	BLCA
label	BLCA	BLCA

3.4 Granges con las anotaciones de las sondas

Se obtienen de las anotaciones de Illumina.

```
sondas <- get450k()
sondas <- sondas[rownames(matriz_sna)]
```

3.5 Construcción del objeto Summarized Experiment

```
data <- SummarizedExperiment::SummarizedExperiment(
  assays=S4Vectors::SimpleList(counts=matriz_sna),
  rowRanges = sondas,
  colData = etiqueta
)
```

4 Selección de las 20000 sondas con mayor variabilidad

4.1 Exclusión de sondas problemáticas de acuerdo a (Price et al. 2013) y (Zhou, Laird, y Shen 2017)

4.1.1 Descarga de las anotaciones adicionales de las sondas

Se descargan las anotaciones adicionales de las sondas desde la documentación adicional de (Price et al. 2013)

```
elist <- getGEO("GSE42409")
GSE42409 <- elist[[1]] %>% featureData()
```

4.1.2 Supresión de sondas problemáticas

Se excluirán las sondas identificadas con:

1. Aquellas cuya denominación comienza con rs o ch: identificadas como SNP por las anotaciones de Illumina o que no están mapeadas al genoma.
2. Las que tienen en el fichero de anotación adicional de (Price et al. 2013) el campo `Target CpG SNP` no vacío.
3. Las que tiene más de una localización in silico de acuerdo al fichero de (Price et al. 2013).
4. Las que apuntan a ADN repetitivo según el fichero de (Price et al. 2013).

```
GSE42409_df <- GSE42409@data
GSE42409_df <- select(GSE42409_df, c(1,6,12,13,14))

nombres_sondas <- data.frame(sondas = row.names(data))
nombres_sondas <- filter(nombres_sondas, substring(sondas, 1, 2) == "cg")

sondas <- merge(nombres_sondas, GSE42409_df, by.x="sondas", by.y="ID", all.x=TRUE)

sondas_dep <-filter(sondas, (`Target CpG SNP`=="" &
```

```
                      AlleleA_Hits == 1 &
                      AlleleB_Hits == 0 &
                      n_bp_repetitive == 0) | is.na(`Target CpG SNP`))

data_dep <- data[row.names(data) %in% sondas_dep$sondas , ]
```

4.1.3 Selección de las muestras tipos 01 TP y 03 TB tumores primarios

```
codigos <- c("01", "03")
data_sondas_dep <- data_sondas_dep[ , data_sondas_dep$categoria %in% codigos]
```

4.2 Selección de las 20000 sondas de mayor desviación estándar

4.2.1 Cálculo desviaciones estandar

```
betas <- assay(data_sondas_dep, "counts")
sds <- apply(betas, 1, sd)

rowRanges(data_sondas_dep)$sd <- sds

save(data_sondas_dep, file="G:/TFM UOC/datos/Clasificador_variabilidad/data_sondas_dep_sd.Rda")
```

4.2.2 Selección de las 20000 sondas de mayor desviación estándar

```
selecciona_sondas <- function(data, n=20000){
  sds <- rowRanges(data)$sd
  orden <- order(sds, decreasing=TRUE)
  sds_o <- sds[orden][1:n]
  d <- data[names(sds_o)]
}

n = 20000

sondas_20000 <- selecciona_sondas(data_sondas_dep, n)
save(sondas_20000, file="G:/TFM UOC/datos/Clasificador_variabilidad/sondas_20000.Rda")
```

4.2.3 Distribución de las desviaciones tipicas

```
load(file="G:/TFM UOC/datos/Clasificador_variabilidad/sondas_20000.Rda")
sondas_20000
```

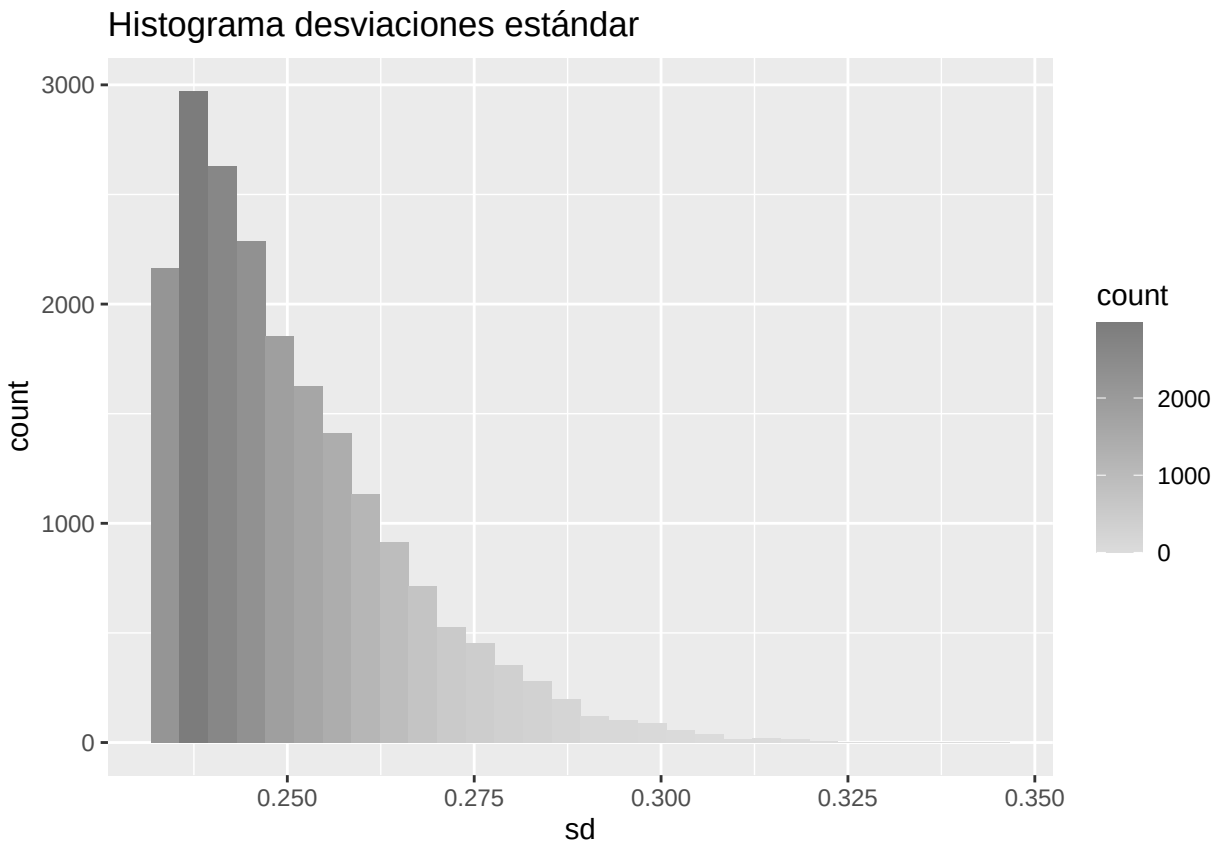
```
## class: RangedSummarizedExperiment
## dim: 20000 8533
## metadata(0):
## assays(1): counts
```

```
## rownames(20000): cg20148575 cg04039555 ... cg12908916 cg19561076
## rowData names(11): addressA addressB ... probeTarget sd
## colnames(8533): TCGA-2F-A9KO-01A-11D-A38H-05
##   TCGA-2F-A9KP-01A-11D-A38H-05 ... TCGA-ZA-A8F6-01A-23D-A365-05
##   TCGA-ZQ-A9CR-01A-11D-A398-05
## colData names(7): bar_code sujeto ... assay label
```

```
df <- rowRanges(sondas_20000)$sd
df <- data.frame(sd = df)

summary(df$sd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2331  0.2392  0.2470  0.2509  0.2587  0.3441
```

```
ggplot(data = df , aes(x=sd)) +
  geom_histogram(aes( fill = ..count..), bins=30) +
  scale_fill_gradient(low="#DCDCDC", high = "#7C7C7C") +
  ggtitle("Histograma desviaciones estándar")
```



4.3 Desglose de las muestras en los grupos train y test

Se desglosa el objeto summarized experiment que contiene todas las muestras, en dos, un grupo train con el 75 % de las muestras y un grupo test con el resto. Se usa la función `createDataPartition` de la librería `caret`.

```
set.seed(123)
etiqueta <- sondas_20000$label

in_train <- createDataPartition(etiqueta, p=0.75, list=FALSE) %>% as.vector()

train <- sondas_20000[ , in_train]
test <- sondas_20000[ , -in_train]
train
```

```
## class: RangedSummarizedExperiment
## dim: 20000 6412
## metadata(0):
## assays(1): counts
## rownames(20000): cg20148575 cg04039555 ... cg12908916 cg19561076
## rowData names(11): addressA addressB ... probeTarget sd
## colnames(6412): TCGA-2F-A9KP-01A-11D-A38H-05
##   TCGA-2F-A9KQ-01A-11D-A38H-05 ... TCGA-VQ-AA6K-01A-11D-A411-05
##   TCGA-ZQ-A9CR-01A-11D-A398-05
## colData names(7): bar_code sujeto ... assay label
```

```
test
```

```
## class: RangedSummarizedExperiment
## dim: 20000 2121
## metadata(0):
## assays(1): counts
## rownames(20000): cg20148575 cg04039555 ... cg12908916 cg19561076
## rowData names(11): addressA addressB ... probeTarget sd
## colnames(2121): TCGA-2F-A9KO-01A-11D-A38H-05
##   TCGA-2F-A9KW-01A-11D-A38H-05 ... TCGA-VQ-AA6F-01A-31D-A411-05
##   TCGA-ZA-A8F6-01A-23D-A365-05
## colData names(7): bar_code sujeto ... assay label
```

```
comprueba <- function(data){
  f <- colnames(data) != data$bar_code
  sum(f)
}

comprueba(train)
```

```
## [1] 0
```

```
comprueba(test)
```

```
## [1] 0
```

4.3.1 Tabla de distribución de muestras entre train y test

Se detalla la distribución entre los grupos train y test de las muestras.

```
t1 <- train$label %>% table() %>% as.matrix()
t2 <- test$label %>% table() %>% as.matrix()

df <- data.frame(Train= table(train$label),
                 Test = table(test$label),
                 Total = t1+t2)

df[, c(2,4,5) ] %>% kable()
```

	Train.Freq	Test.Freq	Total
ACC	60	20	80
BLCA	309	103	412
BRCA	588	195	783
CESC	231	76	307
CHOL	27	9	36
COAD	220	73	293
DLBC	36	12	48
ESCA	139	46	185
GBM	105	35	140
HNSC	396	132	528
KICH	50	16	66
KIRC	240	79	319
KIRP	207	68	275
LAML	146	48	194
LGG	387	129	516
LIHC	283	94	377
LUAD	344	114	458
LUSC	278	92	370
MESO	66	21	87
OV	8	2	10
PAAD	138	46	184
PCPG	135	44	179
PRAD	374	124	498
READ	74	24	98
SARC	196	65	261
SKCM	79	26	105
STAD	297	98	395
TGCT	101	33	134
THCA	378	125	503
THYM	93	31	124
UCEC	324	107	431
UCS	43	14	57
UVM	60	20	80

4.3.2 Tabla de ubicaciones de las sondas seleccionadas

Se detalla los cromosomas a los que se mapean las 20.000 sondas seleccionadas.

```
rowRanges(train) %>% seqnames() %>% table()
```

```
## .
```

```
##  chr1  chr2  chr3  chr4  chr5  chr6  chr7  chr8  chr9 chr10 chr11 chr12 chr13
##  1795  1629  1188   803  1244  1498  1480  1132   244  1045   989  1047   642
## chr14 chr15 chr16 chr17 chr18 chr19 chr20 chr21 chr22  chrX  chrY
##   564   526   554   843   244   491   332   181   166  1357     6
```

4.3.3 Tabla con la tipología de las sondas seleccionadas

El desglose de las sondas seleccionadas entre Tipo I (Green y Red) y Tipo II (Both)

```
rowRanges(train)$channel %>% table() %>% kable()
```

.	Freq
Both	18748
Grn	245
Red	1007

4.3.4 Tabla con tipos de targets HIL en las sondas seleccionadas

```
elist <- getGEO("GSE42409")
GSE42409 <- elist[[1]] %>% featureData()

GSE42409$HIL_CpG_class[GSE42409$ID %in% names(train)]  %>%  table() %>%  kable()
```

.	Freq
HC	2394
IC	3562
ICshore	1296
LC	10334

5 Análisis gráfico previo de los beta-values

Se utilizan para construir los gráficos tan solo las muestras del grupo train.

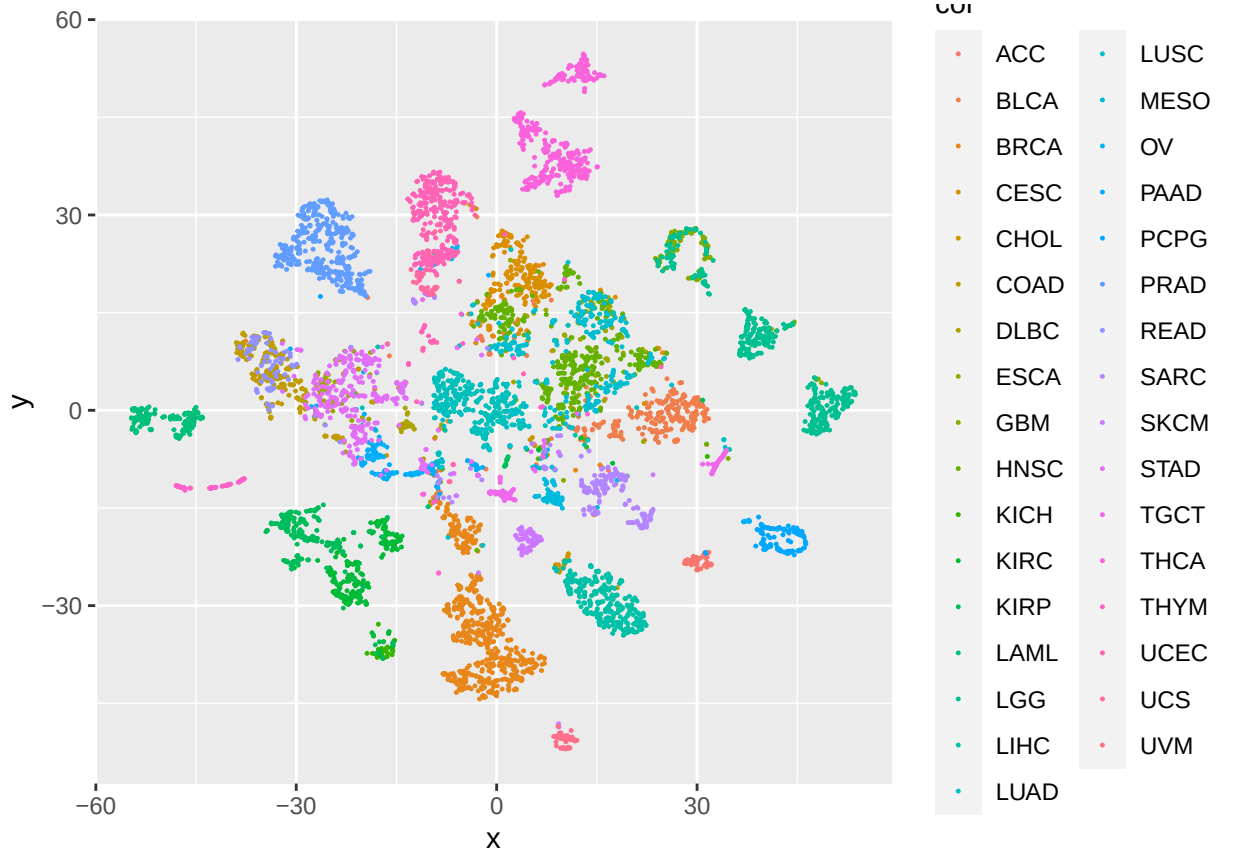
```
train_data <- assay(train, "counts") %>%  t()
label <- train$label %>% factor()
```

5.1 Gráfico previo Rtsne

Tan solo utilizaremos los datos train:

```
sed.seed=123
tsne <- Rtsne(train_data, partial_pca=TRUE, dims=2, perplexity=30, verbose =FALSE, max_iter=1000 )

# Gráfico por patologías
```

```
tsne_plot <- data.frame(x = tsne$Y[,1], y = tsne$Y[,2], col = label)
ggplot(tsne_plot) + geom_point(aes(x=x, y=y, color=col), size=0.2)
```



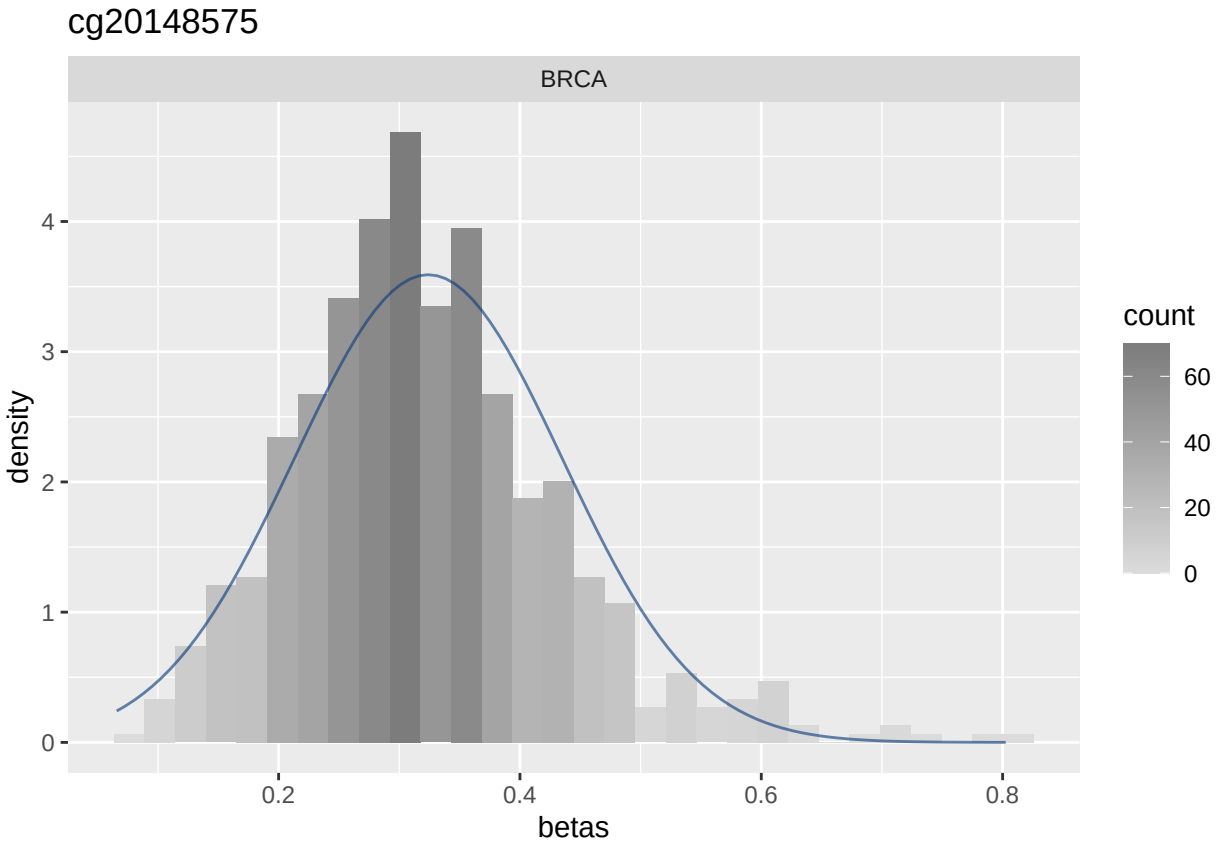
Muchos tumores aparecen claramente diferenciados en el gráfico, sin embargo, otros se presume que va a ser difícil de discriminar con la información contenida en los beta-values de las 20.000 sondas seleccionadas.

5.2 Gráfico revisión normalidad de las sondas

Se muestra el histograma de los valores betas de una sonda elegida aleatoriamente para las muestras del estudio BRCA

```
label_g <- as.character(label)
l <- label_g[label == "BRCA" ]

df <- data.frame(betas = train_data[ label =="BRCA" , sample(1:dim(train_data)[2], 1)], label = l )

ggplot(data = df , aes(x=betas)) +
  geom_histogram(aes(y=..density.., fill = ..count..)) +
  scale_fill_gradient(low="#DCDCDC", high = "#7C7C7C") +
  stat_function(fun=dnorm, colour="#0C3D7D9F", args=list(mean=mean(df$betas), sd = sd(df$betas))) +
  ggtitle(colnames(train_data)[1]) + facet_grid( . ~ df$label)
```

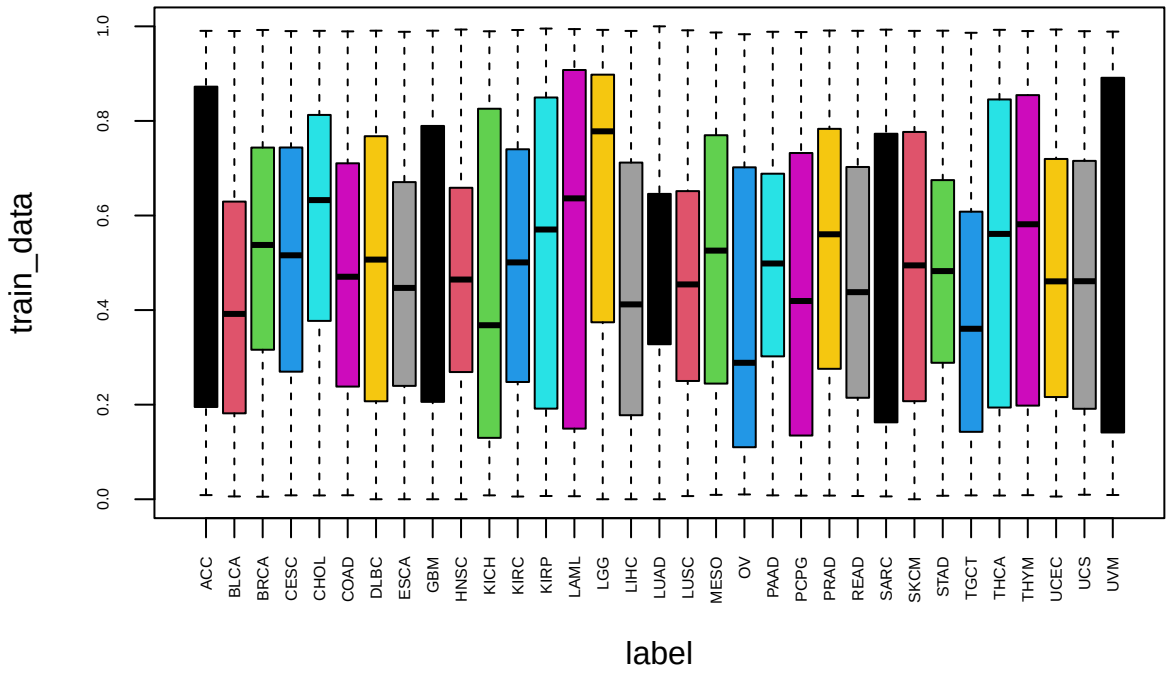
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



No podemos presumir que los valores betas se distribuyan normalmente ni en el conjunto de todas las muestras ni dentro de cada tumor en particular. Esto sería un impedimento en una clasificador que utilizara la regresión logística, sin embargo no aparece como restricción ni en los algoritmos red neuronal ni random forest que son los que aplicaremos seguidamente.

5.3 Histograma de los valores beta

```
boxplot(train_data ~ label, cex.axis=0.5, las=3, col=palette("Polychrome 36"))
```



El histograma de valores betas por tumor (solo consideradas las 20.000 sondas seleccionadas) muestra diferencias significativas entre los diferentes tumores, sin embrago el grupo de control representado en el séptimo lugar no se identifica o discrimina claramente con respecto al resto.

6 El clasificador RED neuronal

6.1 Desglose de las sondas de acuerdo al tipo de sonda

Se desglosan las sondas en los grupos "Green" y "Red", sondas tipo I y "Both" sondas tipo II de acuerdo a las anotaciones de Illumina 450k. También se preparan las etiquetas de las muestras (el tumor al que pertenecen) para poder incorporarse a la red neuronal: se convierte un factor (label) en un array con el procedimiento One-hot encoding.

```
train_data <- assay(train, "counts") %>%  t()
label <- train$label %>% factor()
label_c <- to_categorical(as.integer(label))
```

```
## Loaded Tensorflow version 2.9.2
```

```
tipos_sondas <- rowData(train)$channel %>%  as.factor()
train_green_data <- train_data[ , tipos_sondas=="Grn"]
train_red_data <- train_data[ , tipos_sondas == "Red"]
train_both_data <- train_data[ , tipos_sondas =="Both"]
```

6.2 Formulación del modelo

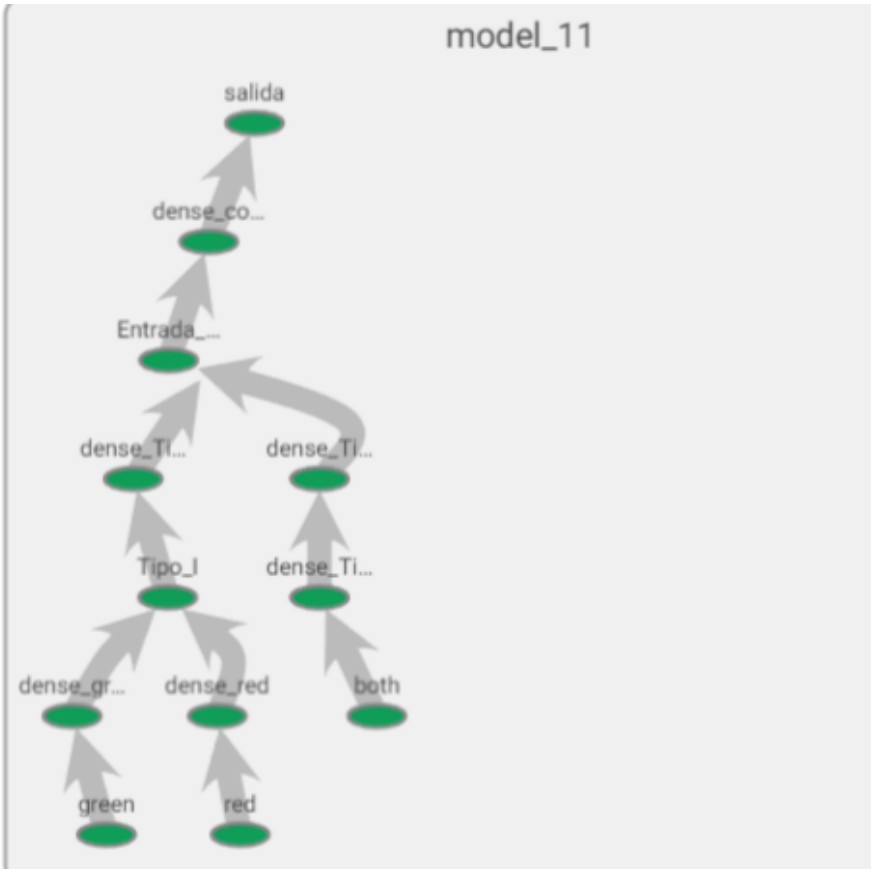
Código adaptado de (Chollet y Allaire 2018)

```
build_model <- function() {
  green <- layer_input(shape= dim(train_green_data)[[2]], name="green")
  red <- layer_input(shape= dim(train_red_data)[[2]], name="red")
  both <- layer_input(shape= dim(train_both_data)[[2]], name="both")

  salida_green <- green %>%
    layer_dense(units=64, activation="relu", name ="dense_green")

  salida_red <- red %>%
    layer_dense(units=64, activation="relu", name="dense_red")

  salida_tipo_I <- layer_concatenate(list(salida_green, salida_red),
                                     name = "Tipo_I")

  tipo_I <- salida_tipo_I %>%
    layer_dense(units = 32, activation = "relu", name="dense_Tipo_I")

  tipo_II <- both %>%
    layer_dense(units=64, activation="relu", name="dense_Tipo_II_1") %>%
    layer_dense(units=32, activation="relu", name = "dense_Tipo_II_2")

  concatenated <- layer_concatenate(list(tipo_I, tipo_II), name="Entrada_Tipo_I_TipoII")

  salida <- concatenated %>%
    layer_dense(units=64, activation = "relu", name="dense_conjunta") %>%
    layer_dense(units=34, activation = "softmax", name="salida")


  model <- keras_model(list(green, red, both), salida)

  model %>%  compile(
    optimizer = "rmsprop",
    loss = "categorical_crossentropy",
    metrics = c("accuracy")
  )
}

build_model() %>% summary()
```

```
## Model: "model"
## ____________________________________________________________________________
##  Layer (type)              Output Shape      Param #  Connected to
## ============================================================================
##  green (InputLayer)        [(None, 245)]     0        []
##  red (InputLayer)          [(None, 1007)]    0        []
##  dense_green (Dense)       (None, 64)        15744    ['green[0][0]']
##  dense_red (Dense)         (None, 64)        64512    ['red[0][0]']
##  both (InputLayer)         [(None, 18748)]   0        []
##  Tipo_I (Concatenate)      (None, 128)       0        ['dense_green[0][0]',
##                                                       'dense_red[0][0]']
```

```
##  dense_Tipo_II_1 (Dense)  (None, 64)        1199936  ['both[0][0]']
##  dense_Tipo_I (Dense)     (None, 32)        4128     ['Tipo_I[0][0]']
##  dense_Tipo_II_2 (Dense)  (None, 32)        2080     ['dense_Tipo_II_1[0][0]']
##  Entrada_Tipo_I_TipoII (C  (None, 64)       0        ['dense_Tipo_I[0][0]',
##  oncatenate)                                          'dense_Tipo_II_2[0][0]']
##  dense_conjunta (Dense)   (None, 64)        4160     ['Entrada_Tipo_I_TipoII[0][[
##                                                      0]']
##  salida (Dense)           (None, 34)        2210     ['dense_conjunta[0][0]']
## ==================================================================================
## Total params: 1,292,770
## Trainable params: 1,292,770
## Non-trainable params: 0
## ----------------------------------------------------------------------------------
```

El grafo de modelo extraído de tensorboard:



6.3 Validación cruzada para determinar la capacidad predictiva del modelo

Se realiza la validación cruzada con 4 particiones, pliegues. El grupo train se divide en 4 bloques, se entrena el modelo con tres de ellos, validándose con el restante. Una vez entrenado y validado el modelo las cuatro veces, el `accuracy` propuesto es la media de los 4 valores obtenidos.

Código adaptado de (Chollet y Allaire 2018)

```
k=4
folds <- createFolds(label, k)

num_epoch = 40
all_scores = c()

for (i in 1:k) {
  cat("procesing fold #", i, "\n")
  partial_green_data <- train_green_data[ -folds[[i]] , ]
  partial_red_data <- train_red_data[ -folds[[i]] , ]
  partial_both_data <- train_both_data[ -folds[[i]] , ]
  partial_train_label <- label_c[-folds[[i]] , ]

  val_green_data <- train_green_data[folds[[i]] , ]
  val_red_data <- train_red_data[folds[[i]] , ]
  val_both_data <- train_both_data[folds[[i]] , ]
  val_label <- label_c[folds[[i]],]

  model <- build_model()

  history <- model %>% fit(list(partial_green_data, partial_red_data, partial_both_data ),
                partial_train_label,
                epoch = num_epoch
              )

  results <- model %>%  evaluate(list(val_green_data, val_red_data, val_both_data), val_label)
  all_scores <- c(all_scores, results[2])
  }
```

```
## procesing fold # 1
## procesing fold # 2
## procesing fold # 3
## procesing fold # 4
```

```
all_scores
```

```
##  accuracy  accuracy  accuracy  accuracy
## 0.9284826 0.9383178 0.9150000 0.9218261
```

```
mean(all_scores)
```

```
## [1] 0.9259066
```

6.4 Predicción con lo valores test utilizando el modelo con epoch 40 seleccionado

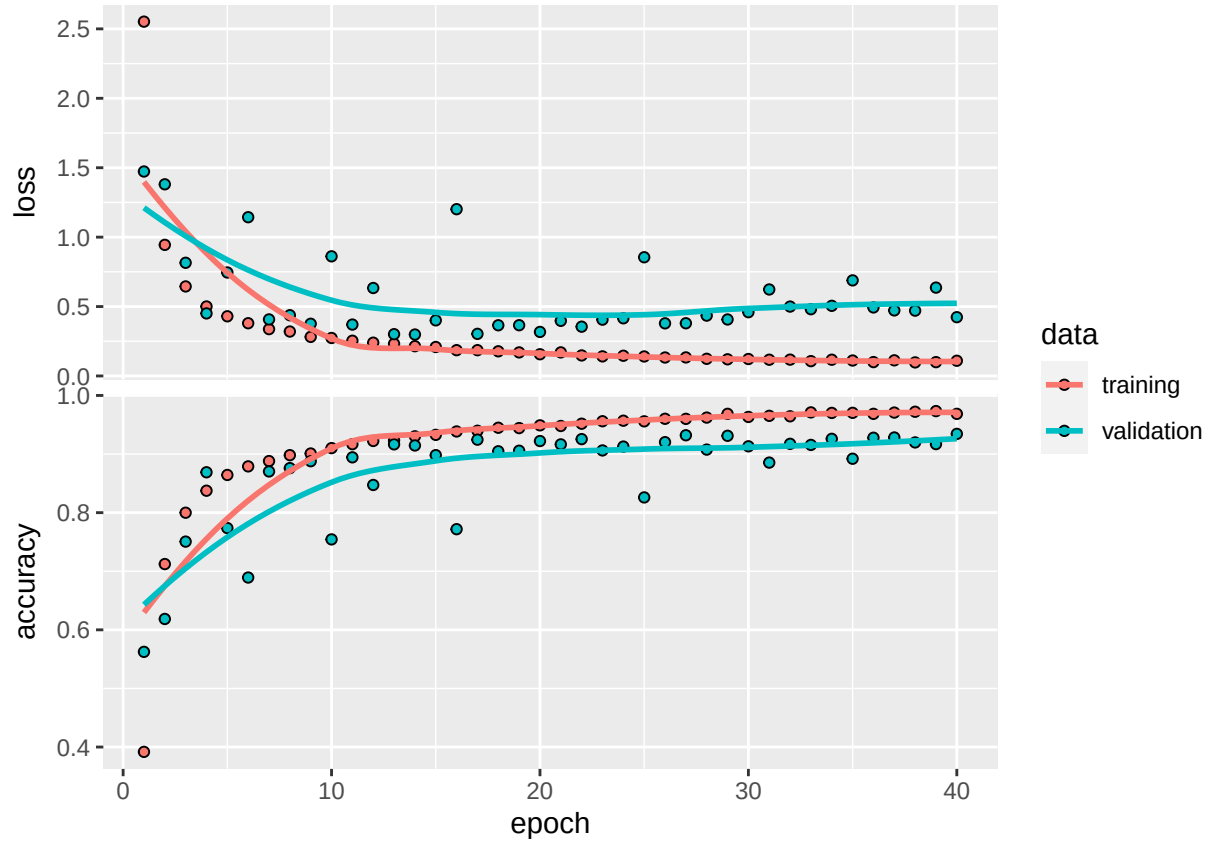
Una vez ajustado el parámetro `epoch` al repetir la validación cruzada con varios valores, el modelo elegido final, que se entrena con todos los valores del subset train, se evalúa con los datos de test.

6.4.1 Preparación de los datos del subset test

```
betas_test <- assay(test, "counts") %>% t()

test_green_data <- betas_test[ , tipos_sondas=="Grn"]
test_red_data <- betas_test[ , tipos_sondas == "Red"]
test_both_data <- betas_test[ , tipos_sondas =="Both"]

fenotipos_test <- colData(test)$label %>% factor(ordered=TRUE)
test_labels <- to_categorical(as.integer(fenotipos_test))
```

6.4.2 Entrenamiento del modelo con todos los datos train

```
modelo <- build_model()

num_epoch = 40
history <- modelo %>% fit(list(train_green_data, train_red_data, train_both_data ),
                          label_c,
                          epoch = num_epoch,
                          validation_data=list(list(test_green_data, test_red_data, test_both_data ),
                                               test_labels)
                          )

plot(history)
```




```
metrics <- modelo %>% evaluate(list(test_green_data, test_red_data, test_both_data), test_labels)
metrics
```

```
##      loss  accuracy
## 0.4233246 0.9344649
```

6.4.3 Matriz de contingencia de los resultados

```
prediccion <- modelo %>% predict(list(test_green_data, test_red_data, test_both_data)) %>%
  k_argmax() %>%
  as.array() %>% as.integer()

l <- as.list(1:34)
names(l) <- levels(fenotipos_test)
f <- names(l)[prediccion]
lev <- levels(fenotipos_test)
prediccion <- factor(f, levels=lev)

c3 <- confusionMatrix(fenotipos_test, prediccion)
c3
```

```
## Confusion Matrix and Statistics
##
```

```
##            Reference
## Prediction ACC BLCA BRCA CESC CHOL COAD DLBC ESCA GBM HNSC KICH KIRC KIRP LAML
##       ACC   19    0    0    0    0    0    0    0   0    0    0    0    0    0
##       BLCA   0  100    0    2    0    0    1    0   0    0    0    0    0    0
##       BRCA   0    0  194    0    0    0    0    0   0    0    0    0    0    0
##       CESC   0    0    0   72    0    0    0    0   0    0    0    0    0    0
##       CHOL   0    0    0    0    5    0    0    0   0    0    0    0    1    0
##       COAD   0    0    0    1    0   59    0    0   0    0    0    0    0    0
##       DLBC   0    2    0    0    0    0   10    0   0    0    0    0    0    0
##       ESCA   0    1    1    0    0    0    0   24   0    6    0    0    0    0
##       GBM    0    0    0    0    0    0    0    0  35    0    0    0    0    0
##       HNSC   0    0    1    1    0    0    0    4   0  122    0    0    0    0
##       KICH   0    0    0    0    0    0    0    0   0    0   14    0    1    0
##       KIRC   0    0    0    0    0    0    0    0   0    0    2   74    2    0
##       KIRP   0    1    0    1    1    0    0    0   0    0    0    4   61    0
##       LAML   0    0    0    0    0    0    0    0   0    0    0    0    0   48
##       LGG    0    0    0    0    0    0    0    0   9    0    0    0    0    0
##       LIHC   0    0    0    0    5    0    0    0   0    0    0    0    0    0
##       LUAD   0    0    1    0    0    0    0    0   0    0    0    0    0    0
##       LUSC   0    1    1    0    0    0    0    0   0    0    0    0    0    0
##       MESO   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       OV     0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       PAAD   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       PCPG   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       PRAD   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       READ   0    0    0    0    0    4    0    0   0    0    0    0    0    0
##       SARC   0    0    1    0    0    0    0    0   0    0    0    1    0    0
##       SKCM   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       STAD   0    0    0    1    0    1    0   13   0    0    0    0    0    0
##       TGCT   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       THCA   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       THYM   0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       UCEC   0    0    0    2    0    0    0    0   0    0    0    0    0    0
##       UCS    0    0    0    0    0    0    0    0   0    0    0    0    0    0
##       UVM    0    0    0    0    0    0    0    0   0    0    0    0    0    0
##            Reference
## Prediction LGG LIHC LUAD LUSC MESO  OV PAAD PCPG PRAD READ SARC SKCM STAD TGCT
##       ACC    0    0    1    0    0   0    0    0    0    0    0    0    0    0
##       BLCA   0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       BRCA   0    0    0    0    0   0    0    0    0    0    1    0    0    0
##       CESC   0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       CHOL   0    2    0    0    0   0    1    0    0    0    0    0    0    0
##       COAD   0    0    0    0    0   0    0    0    0   13    0    0    0    0
##       DLBC   0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       ESCA   0    0    0    0    0   0    1    0    0    1    0    0   12    0
##       GBM    0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       HNSC   0    0    0    4    0   0    0    0    0    0    0    0    0    0
##       KICH   0    0    0    0    0   0    0    0    0    0    1    0    0    0
##       KIRC   0    0    0    0    0   0    0    0    0    0    1    0    0    0
##       KIRP   0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       LAML   0    0    0    0    0   0    0    0    0    0    0    0    0    0
##       LGG  119    0    0    0    0   0    0    0    1    0    0    0    0    0
##       LIHC   0   89    0    0    0   0    0    0    0    0    0    0    0    0
##       LUAD   0    0  109    2    0   0    2    0    0    0    0    0    0    0
```

```
##       LUSC   0    0    4   86    0    0    0    0    0    0    0    0    0    0
##       MESO   0    0    1    0   20    0    0    0    0    0    0    0    0    0
##       OV     0    0    0    0    0    1    0    0    0    0    0    0    0    0
##       PAAD   0    0    0    0    0    0   45    0    0    1    0    0    0    0
##       PCPG   0    0    0    0    0    0    0   44    0    0    0    0    0    0
##       PRAD   0    0    0    0    0    0    0    0  124    0    0    0    0    0
##       READ   0    0    0    0    0    0    0    0    0   20    0    0    0    0
##       SARC   0    0    0    0    0    0    0    0    0    0   63    0    0    0
##       SKCM   0    0    0    0    0    0    0    0    0    0    0   25    0    0
##       STAD   0    0    0    0    0    0    3    0    0    1    0    0   79    0
##       TGCT   0    0    0    0    0    0    0    0    0    0    0    0    0   33
##       THCA   0    0    0    0    0    0    0    0    0    0    0    0    0    0
##       THYM   0    0    0    1    0    0    0    0    0    0    0    0    0    0
##       UCEC   0    0    0    0    0    0    0    0    0    0    0    0    0    0
##       UCS    0    0    0    0    0    0    0    0    0    0    0    0    0    0
##       UVM    0    0    0    0    0    0    0    0    0    0    0    0    0    0
##           Reference
## Prediction THCA THYM UCEC UCS UVM
##       ACC     0    0    0   0   0
##       BLCA    0    0    0   0   0
##       BRCA    0    0    0   0   0
##       CESC    0    0    3   1   0
##       CHOL    0    0    0   0   0
##       COAD    0    0    0   0   0
##       DLBC    0    0    0   0   0
##       ESCA    0    0    0   0   0
##       GBM     0    0    0   0   0
##       HNSC    0    0    0   0   0
##       KICH    0    0    0   0   0
##       KIRC    0    0    0   0   0
##       KIRP    0    0    0   0   0
##       LAML    0    0    0   0   0
##       LGG     0    0    0   0   0
##       LIHC    0    0    0   0   0
##       LUAD    0    0    0   0   0
##       LUSC    0    0    0   0   0
##       MESO    0    0    0   0   0
##       OV      0    0    1   0   0
##       PAAD    0    0    0   0   0
##       PCPG    0    0    0   0   0
##       PRAD    0    0    0   0   0
##       READ    0    0    0   0   0
##       SARC    0    0    0   0   0
##       SKCM    0    0    0   0   1
##       STAD    0    0    0   0   0
##       TGCT    0    0    0   0   0
##       THCA  125    0    0   0   0
##       THYM    0   30    0   0   0
##       UCEC    0    0  102   3   0
##       UCS     0    0    3  11   0
##       UVM     0    0    0   0  20
##
## Overall Statistics
##
```

```
##                Accuracy : 0.9345
##                  95% CI : (0.9231, 0.9446)
##     No Information Rate : 0.0938
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9313
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: ACC Class: BLCA Class: BRCA Class: CESC Class: CHOL
## Sensitivity            1.000000     0.95238     0.97487     0.90000    0.454545
## Specificity            0.999524     0.99851     0.99948     0.99804    0.998104
## Pos Pred Value         0.950000     0.97087     0.99487     0.94737    0.555556
## Neg Pred Value         1.000000     0.99752     0.99740     0.99609    0.997159
## Prevalence             0.008958     0.04950     0.09382     0.03772    0.005186
## Detection Rate         0.008958     0.04715     0.09147     0.03395    0.002357
## Detection Prevalence   0.009430     0.04856     0.09194     0.03583    0.004243
## Balanced Accuracy      0.999762     0.97545     0.98718     0.94902    0.726325
##                      Class: COAD Class: DLBC Class: ESCA Class: GBM Class: HNSC
## Sensitivity              0.92188    0.909091     0.58537    0.79545     0.95312
## Specificity              0.99319    0.999052     0.98942    1.00000     0.99498
## Pos Pred Value           0.80822    0.833333     0.52174    1.00000     0.92424
## Neg Pred Value           0.99756    0.999526     0.99181    0.99569     0.99698
## Prevalence               0.03017    0.005186     0.01933    0.02074     0.06035
## Detection Rate           0.02782    0.004715     0.01132    0.01650     0.05752
## Detection Prevalence     0.03442    0.005658     0.02169    0.01650     0.06223
## Balanced Accuracy        0.95753    0.954072     0.78739    0.89773     0.97405
##                      Class: KICH Class: KIRC Class: KIRP Class: LAML Class: LGG
## Sensitivity             0.875000     0.93671     0.93846     1.00000    1.00000
## Specificity             0.999050     0.99755     0.99660     1.00000    0.99500
## Pos Pred Value          0.875000     0.93671     0.89706     1.00000    0.92248
## Neg Pred Value          0.999050     0.99755     0.99805     1.00000    1.00000
## Prevalence              0.007544     0.03725     0.03065     0.02263    0.05611
## Detection Rate          0.006601     0.03489     0.02876     0.02263    0.05611
## Detection Prevalence    0.007544     0.03725     0.03206     0.02263    0.06082
## Balanced Accuracy       0.937025     0.96713     0.96753     1.00000    0.99750
##                      Class: LIHC Class: LUAD Class: LUSC Class: MESO Class: OV
## Sensitivity              0.97802     0.94783     0.92473    1.000000 1.0000000
## Specificity              0.99754     0.99751     0.99704    0.999524 0.9995283
## Pos Pred Value           0.94681     0.95614     0.93478    0.952381 0.5000000
## Neg Pred Value           0.99901     0.99701     0.99655    1.000000 1.0000000
## Prevalence               0.04290     0.05422     0.04385    0.009430 0.0004715
## Detection Rate           0.04196     0.05139     0.04055    0.009430 0.0004715
## Detection Prevalence     0.04432     0.05375     0.04338    0.009901 0.0009430
## Balanced Accuracy        0.98778     0.97267     0.96089    0.999762 0.9997642
##                      Class: PAAD Class: PCPG Class: PRAD Class: READ
## Sensitivity              0.86538     1.00000     0.99200     0.55556
## Specificity              0.99952     1.00000     1.00000     0.99808
## Pos Pred Value           0.97826     1.00000     1.00000     0.83333
## Neg Pred Value           0.99663     1.00000     0.99950     0.99237
## Prevalence               0.02452     0.02074     0.05893     0.01697
## Detection Rate           0.02122     0.02074     0.05846     0.00943
```

```
## Detection Prevalence     0.02169     0.02074     0.05846     0.01132
## Balanced Accuracy        0.93245     1.00000     0.99600     0.77682
##                       Class: SARC Class: SKCM Class: STAD Class: TGCT
## Sensitivity              0.95455     1.00000     0.86813     1.00000
## Specificity              0.99903     0.99952     0.99064     1.00000
## Pos Pred Value           0.96923     0.96154     0.80612     1.00000
## Neg Pred Value           0.99854     1.00000     0.99407     1.00000
## Prevalence               0.03112     0.01179     0.04290     0.01556
## Detection Rate           0.02970     0.01179     0.03725     0.01556
## Detection Prevalence     0.03065     0.01226     0.04620     0.01556
## Balanced Accuracy        0.97679     0.99976     0.92939     1.00000
##                       Class: THCA Class: THYM Class: UCEC Class: UCS Class: UVM
## Sensitivity              1.00000     1.00000     0.93578   0.733333   0.952381
## Specificity              1.00000     0.99952     0.99751   0.998575   1.000000
## Pos Pred Value           1.00000     0.96774     0.95327   0.785714   1.000000
## Neg Pred Value           1.00000     1.00000     0.99652   0.998102   0.999524
## Prevalence               0.05893     0.01414     0.05139   0.007072   0.009901
## Detection Rate           0.05893     0.01414     0.04809   0.005186   0.009430
## Detection Prevalence     0.05893     0.01462     0.05045   0.006601   0.009430
## Balanced Accuracy        1.00000     0.99976     0.96665   0.865954   0.976190
```

7 Entrenamiento de la red usando las betas ajustadas con ComBat

En este apartado ajustamos los valores betas por el posible efecto **batch** de la variable `plate_id`. El ajuste solo se realiza a los valores del subset train, dado que si aspiramos a emplear el algoritmo en muestras ajenas al proyecto TCGA, éstas no van a estar ajustadas por esa variable, propia delproyecto.

7.1 Ajuste de los valores betas del subset train con la función Combat

```
edata <- assay(train, "counts")
pdata <- colData(train)

nombres <- assay(train, "counts") %>% colnames()
plate_id <- TCGAbiospec(nombres)
plate_id <- plate_id$plate

train$plate_id <- plate_id
```

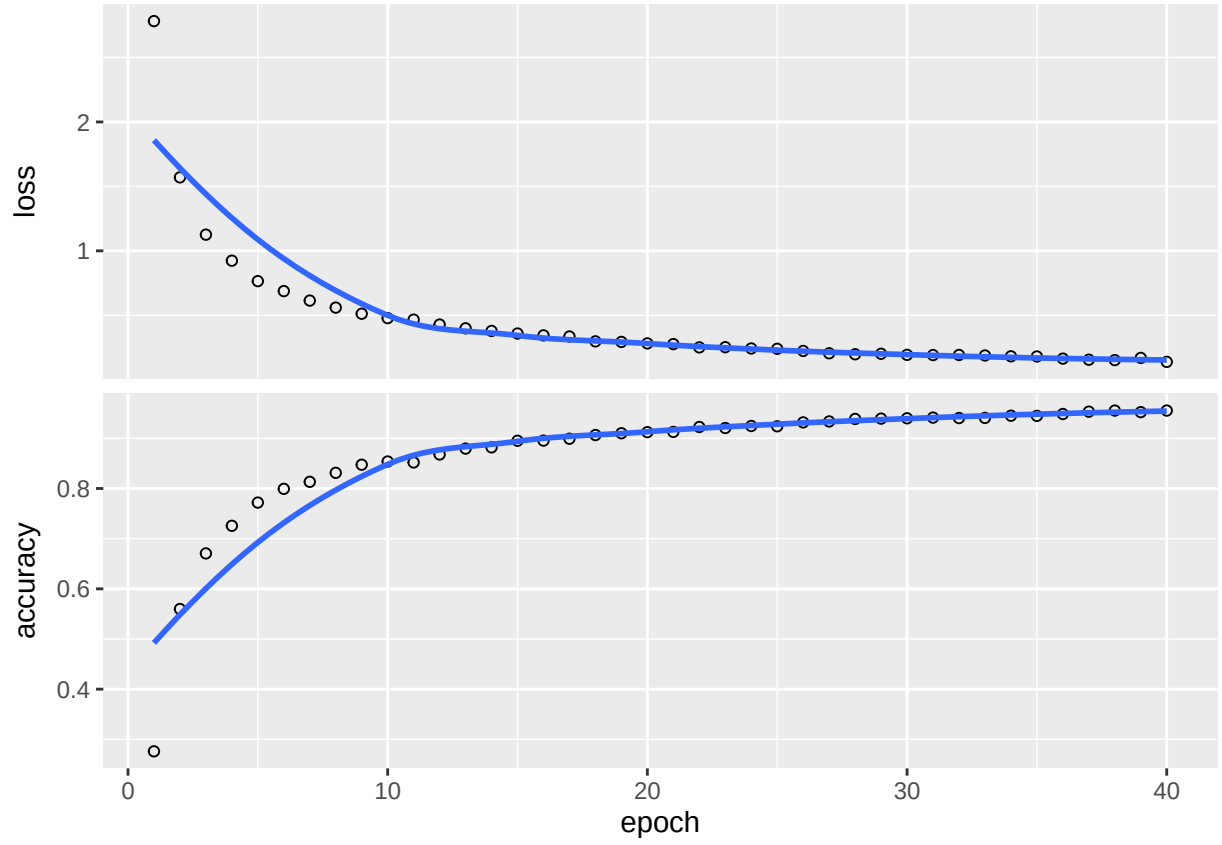
```
mod0 <- model.matrix( ~ 1, data = pdata)
mod <- model.matrix( ~ as.factor(label), data = pdata)

combat_edata <- ComBat(dat = edata, batch = plate_id,
                       mod = mod0, mean.only=TRUE, par.prior=TRUE)

assay(train, "counts") <- combat_edata
```

```
train_data <- assay(train, "counts") %>%  t()
label <- train$label %>% factor()
label_c <- to_categorical(as.integer(label))
```

```
tipos_sondas <- rowData(train)$channel %>%  as.factor()
train_green_data <- train_data[ , tipos_sondas=="Grn"]
train_red_data <- train_data[ , tipos_sondas == "Red"]
train_both_data <- train_data[ , tipos_sondas =="Both"]
```

7.2 Entrenamiento del modelo con los nuevos valores ajustados

```
modelo <- build_model()

history <- modelo %>% fit(list(train_green_data, train_red_data, train_both_data ),
                          label_c,
                          epoch = num_epoch
                          )

plot(history)
```



7.3 Evaluación del modelo

```
metrics <- modelo %>% evaluate(list(test_green_data, test_red_data, test_both_data), test_labels)
metrics
```

```
##      loss  accuracy
## 0.6981411 0.9000471
```

Bibliografía

Capper, David, David TW Jones, Martin Sill, Volker Hovestadt, Daniel Schrimpf, Dominik Sturm, Christian Koelsche, et al. 2018. «DNA methylation-based classification of central nervous system tumours». *Nature* 555 (7697): 469-74.

Chollet, F, y JJ Allaire. 2018. «Deep Learning with R, Manning Publications».

Kuhn, Max. 2017. *A Short Introduction to the caret Package*. https://cran.r-project.org/web/packages/caret/vignettes/caret.pdf.

Lantz, Brett. 2015. *Machine learning with R*. Packt Publishing Ltd. http://www.packtpub.com/books/content/machine-learning-r.

Maros, Máté E, David Capper, David TW Jones, Volker Hovestadt, Andreas von Deimling, Stefan M Pfister, Axel Benner, Manuela Zucknick, y Martin Sill. 2020. «Machine learning workflows to estimate class probabilities for precision cancer diagnostics on DNA methylation microarray data». *Nature protocols* 15 (2): 479-512.

Price, E Magda, Allison M Cotton, Lucia L Lam, Pau Farré, Eldon Emberly, Carolyn J Brown, Wendy P Robinson, y Michael S Kobor. 2013. «Additional annotation enhances potential for biologically-relevant analysis of the Illumina Infinium HumanMethylation450 BeadChip array». *Epigenetics & chromatin* 6 (1): 1-15.

Zhou, Wanding, Peter W Laird, y Hui Shen. 2017. «Comprehensive characterization, annotation and innovative use of Infinium DNA methylation BeadChip probes». *Nucleic acids research* 45 (4): e22-22.