

Clasificador: prueba no TCGA (colonomics)

Alberto Joven Álvarez

7 de enero, 2023

Índice

1	Lista de librerías empleadas	1
2	Entrenamiento del modelo red neuronal con 1.000 sondas más variables.	2
2.1	Carga de los datos: obteto summarizedExperiment y desglose por tipo de sonda	2
2.2	El modelo	2
2.3	Entrenamiento del modelo con todas las observaciones	3
3	Predicción con los datos de colonomics	4
3.1	Carga de los datos	4
3.2	Estimación de datos de las sondas que faltan en colonomics	5
3.3	Estimación de los valores faltantes con <code>impute</code>	5
3.4	Carga valores de fenotipos de colonomics	6
3.5	predicción con las betas de colonomics	7
	Bibliografía	7

1 Lista de librerías empleadas

```
library(knitr)
library(dplyr)
library(readr)
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
library(FDb.InfiniumMethylation.hg19)
library(impute)
library(SummarizedExperiment)
library(GEOquery)
library(caret)
library(ggplot2)
library(keras)
```

2 Entrenamiento del modelo red neuronal con 1.000 sondas más variables.

2.1 Carga de los datos: obteto summarizedExperiment y desglose por tipo de sonda

Se han seleccionado las 1.000 sondas más variables, una vez excluidas sondas problemáticas

```
load("C:/Users/usuario/TFM/sondas_dep_1000.Rda")
sondas_dep_1000

## class: RangedSummarizedExperiment
## dim: 1000 9267
## metadata(0):
## assays(1): counts
## rownames(1000): cg10255237 cg12036633 ... cg11594770 cg23950788
## rowData names(11): addressA addressB ... probeTarget sd
## colnames(9267): TCGA-2F-A9K0-01A-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

train <- assay(sondas_dep_1000, "counts") %>% t()
label <- sondas_dep_1000$label %>% factor()
label_c <- to_categorical(as.integer(label))
```

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

2.2 El modelo

```
build_model <- function() {

  model <- keras_model_sequential() %>%
    layer_dense(units=64, activation="relu", input_shape=dim(train)[[2]]) %>%
    layer_dense(units=32, activation="relu") %>%
    layer_dense(units=64, activation = "relu") %>%
    layer_dense(units=35, activation = "softmax")

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

}

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

```
## Model: "sequential"
## -----
## Layer (type)                Output Shape          Param #
## =====
## dense_3 (Dense)              (None, 64)             64064
## dense_2 (Dense)              (None, 32)             2080
## dense_1 (Dense)              (None, 64)             2112
## dense (Dense)                (None, 35)             2275
## =====
## Total params: 70,531
## Trainable params: 70,531
## Non-trainable params: 0
## -----
```

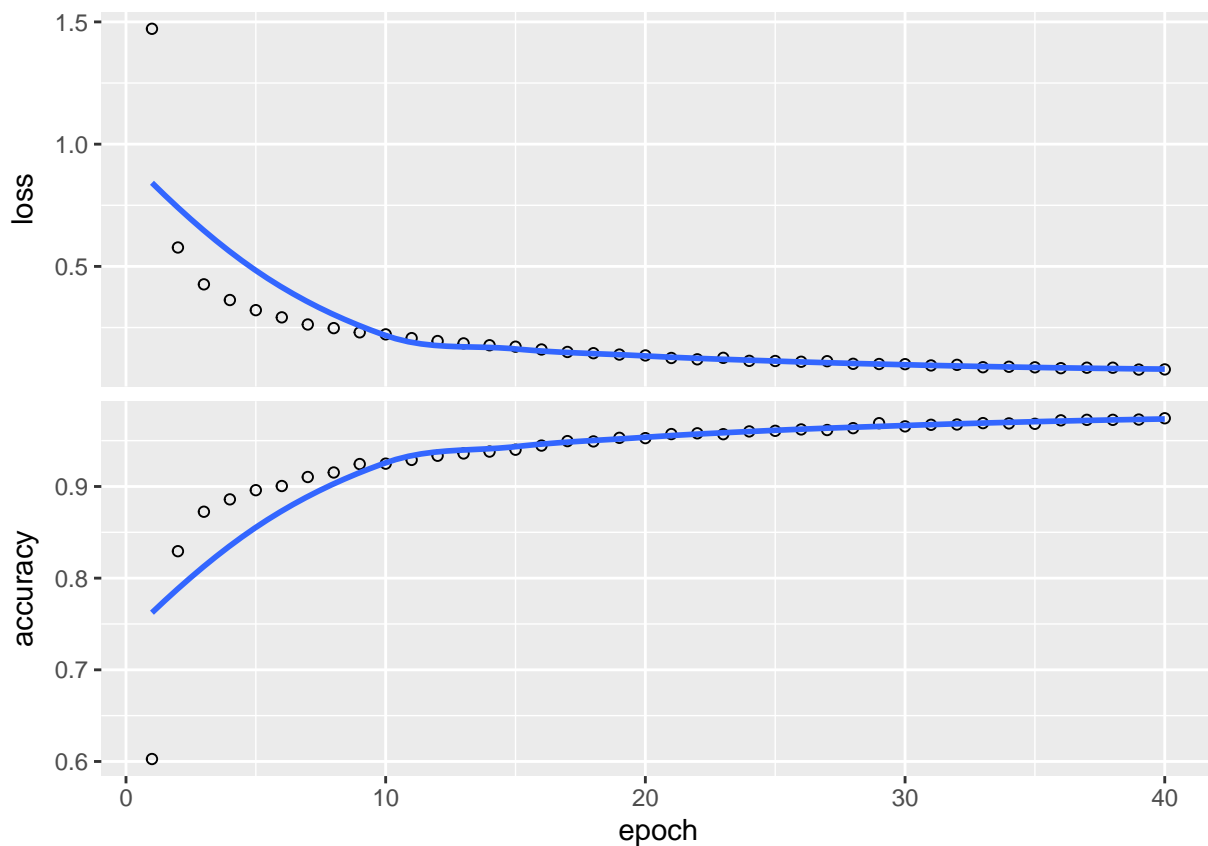
2.3 Entrenamiento del modelo con todas las observaciones

Solo estan las muestras de tumores primarios.

```
modelo <- build_model()

num_epoch = 40
history <- modelo %>% fit(train, label_c, epoch= num_epoch)

plot(history)
```



```
# save_model_hdf5(modelo, "G:/TFM UOC/datos/Clasificador_variabilidad/modelo_dep_1000.h5")

modelo <- load_model_hdf5("C:/Users/usuario/TFM/modelo_1000.h5", compile=FALSE)
```

3 Predicción con los datos de colonomics

3.1 Carga de los datos

```
betas_colonomics <- read_csv("G:/TFM UOC/datos/test_colonomics/CLX_methylation_betas_2014Apr25.txt")
```

```
## Rows: 430086 Columns: 248
## -- Column specification -----
## Delimiter: ","
## chr (1): cpg
## dbl (247): E2069_B, K2068_B, P2055_B, P2078_B, S2039_B, W2072_B, Z2038_B, A6...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
betas_colonomics[1:5, 1:5]
```

```
## # A tibble: 5 x 5
##   cpg          E2069_B K2068_B P2055_B P2078_B
##   <chr>          <dbl>   <dbl>   <dbl>   <dbl>
## 1 cg13869341      888     840     868     800
## 2 cg24669183      855     838     856     823
## 3 cg15560884      817     818     808     804
## 4 cg01014490       69      48      51      40
## 5 cg17505339      903     906     891     898
```

```
dim(betas_colonomics)
```

```
## [1] 430086    248
```

```
betas <- as.matrix(betas_colonomics[, 2:dim(betas_colonomics)[2]])
```

```
sondas_c <- betas_colonomics[, 1]
row.names(betas) <- sondas_c$cpg
betas[1:3, 1:5]
```

```
##           E2069_B K2068_B P2055_B P2078_B S2039_B
## cg13869341      888     840     868     800     829
## cg24669183      855     838     856     823     878
## cg15560884      817     818     808     804     832
```

```
dim(betas)
```

```
## [1] 430086    247
```

3.2 Estimación de datos de las sondas que faltan en colonomics

Primero se crea un data.frame con los nombres de las 1.000 sondas más variables, y con la función `merge` se le añaden los datos betas del proyecto colonomics

```
# se crea el data frame con solo el nombre de las filas: las 1.000 sondas más variables
betas_s <- data.frame(row.names=row.names(sondas_dep_1000))

# con merge se le añaden los valores betas de colonomics
betas_s2 <- merge(betas_s, betas, by.x=0, by.y=0, all.x=TRUE)

# Vemos que existen NAs en las sondas que faltan en colonomics
betas_s2[1:5, 1:5]
```

```
##      Row.names E2069_B K2068_B P2055_B P2078_B
## 1 cg00008629      341      405      628      593
## 2 cg00025331      916      916      918      925
## 3 cg00087792      892      897      888      892
## 4 cg00151370      362      381      356      404
## 5 cg00223767      857      849      874      820
```

```
# Se nombran las filas con la primera columna
row.names(betas_s2) <- betas_s2$Row.names

# Se elimina la primera columna que tenía el nombre de las sondas
betas_s2 <- betas_s2[, 2:dim(betas_s2)[2]]

# se reordena el data.frame con las betas de colonomics para que
# las sondas estén en el mismo orden que en el modelo red_neuronal
betas_s <- betas_s2[row.names(sondas_dep_1000), ]
dim(betas_s)
```

```
## [1] 1000 247
```

```
betas_s[1:5, 1:5]
```

```
##      E2069_B K2068_B P2055_B P2078_B S2039_B
## cg10255237    119    145     91    116    141
## cg12036633    197    610    596    581    201
## cg08141395    158    173    139    160    170
## cg17415265    160    199    161    153    191
## cg09287864    207    183    178    169    212
```

3.3 Estimación de los valores faltantes con impute

Se estiman los valores faltantes con la función `impute.knn` y se dividen por 1.000 para obtener los valores betas en tanto por uno

```
betas_s_sna <- impute.knn(as.matrix(betas_s))$data / 1000
```

```
## Warning in knnimp(x, k, maxmiss = rowmax, maxp = maxp): 6 rows with more than 50 % entries missing;
## mean imputation used for these rows
```

```
betas_s_sna[1:5, 1:5]
```

```
##           E2069_B K2068_B P2055_B P2078_B S2039_B
## cg10255237  0.119   0.145   0.091   0.116   0.141
## cg12036633  0.197   0.610   0.596   0.581   0.201
## cg08141395  0.158   0.173   0.139   0.160   0.170
## cg17415265  0.160   0.199   0.161   0.153   0.191
## cg09287864  0.207   0.183   0.178   0.169   0.212
```

En el mensaje de aviso vemos que había 6 sondas en el subset de sondas más variables que no estaban en los datos de colonomics

3.4 Carga valores de fenotipos de colonomics

```
fenotipos <- read_csv("G:/TFM UOC/datos/test_colonomics/CLX_ClinicalData.csv")
```

```
## Rows: 250 Columns: 16
## -- Column specification -----
## Delimiter: ","
## chr (10): id_clx, type, id_clx_individual, stage, sex, site, metastasis_site...
## dbl (6): age, event_free, time_free, event_global, time_global, stromal_score
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# selección de los fenotipos incluidos en la matriz de valores betas de colonomics
fenotipos_s <- intersect(fenotipos$id_clx, colnames(betas_s_sna))
fenotipos_r <- fenotipos[fenotipos$id_clx %in% fenotipos_s, ]

# selección de las muestras de colonomics para las que tenemos valor de fenotipo
# y trasposición de la matriz
betas_s_sna <- betas_s_sna[ , fenotipos_s] %>% t()

dim(betas_s_sna)
```

```
## [1] 229 1000
```

```
sum(fenotipos_r$id_clx != rownames(betas_s_sna))
```

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

```
table(fenotipos_r$type)
```

```
##
## Mucosa Normal Tumor
##      37      96      96
```

```
colonomics <- betas_s_sna
```

ya tenemos 229 muestras con su matriz de betas de las 1.000 sondas más variables y sus valores de fenotipos

3.5 predicción con las betas de colonomics

```
prediccion <- modelo %>% predict(colonomics) %>%  
  k_argmax() %>%  
  as.array() %>% as.integer()  
  
l <- as.list(1:34)  
names(l) <- levels(label)  
f <- names(l)[prediccion]  
table(prediccion = f, real_colonomics= fenotipos_r$type)
```

```
##           real_colonomics  
## prediccion Mucosa Normal Tumor  
## COAD      0          5     92  
## Control   37         91      2  
## LAML      0          0      1  
## SARC      0          0      1
```

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.