APLICACIÃN DE MACHINE LEARNING A UNA BASE DE DATOS DE MARCADORES STR (short tandem repeats) DE INDIVIDUOS DE DIFERENTES REGIONES DE VENEZUELA.

YASSER VEGA

7 de julio de 2016

.

DESCRIPCION DEL TRABAJO

Se estudiÃ³ una base de datos de la Unidad de Estudios GenÃ©ticos y Forenses de usuarios de pruebas de filiaciÃ³n biolÃ³gica a los que se les determinÃ³ los genotipos de 15 STRs.

Los datos corresponden 312 cromosomas de individuos provenientes de 4 grandes regiones del paÃs codificadas como: 1 = Centro. 2 = Occidente. 3 = Llanos. 4 = Oriente.

alt text

alt text

Como objetivo general, se plantea aplicar Redes Neuronales y Support Vector Machine para estimar si los individuos se pueden diferenciar geneticamente de acuerdo a las regiones de procedencia y generar un modelo que permita predecir la procedencia de un individuo con la informaciÃ³n de marcadores STR estudiados.

Los STR (short tandem repeats) son elementos repetitivos dispersos en el ADN que son estudiados ampliamente en la identificaciÃ³n humana y para pruebas de filiaciÃ³n biolÃ³gica por ser altamente polimorficos en las poblaciones.

DESCRIPCION DE LOS DATOS

```
DataYasser.Individuos.78 <- read.csv("~/DataYasser-Individuos-78.csv", sep=";")
```

```
str(DataYasser.Individuos.78)
```

```
## 'data.frame':    312 obs. of  20 variables:
##  $ D8S1179           : int  12 11 11 12 12 10 10 12 10 10 ...
##  $ D21S11            : int  29 33 30 30 28 30 27 29 28 28 ...
##  $ D21S11_2          : int  0 2 0 0 0 0 0 0 0 0 ...
##  $ D7S820            : int  8 11 9 11 10 11 10 11 10 11 ...
##  $ CSF1PO            : int  9 11 10 10 11 11 10 11 12 11 ...
##  $ D3S1358           : int  14 15 14 15 15 15 16 17 16 17 ...
##  $ THO1              : int  6 7 7 6 7 6 7 6 8 7 ...
##  $ THO1_3            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ D13S317           : int  9 13 12 8 12 10 8 8 10 11 ...
##  $ D16S539           : int  11 11 11 9 10 9 9 13 9 11 ...
##  $ D2S1338           : int  17 17 20 17 19 18 17 17 20 19 ...
##  $ D19S433           : int  15 13 12 13 12 12 13 13 14 14 ...
##  $ D19S433_2         : int  2 0 0 0 0 2 0 0 0 0 ...
##  $ VWA               : int  16 17 15 15 15 15 15 16 15 14 ...
##  $ TPOX              : int  9 8 9 8 8 9 8 8 8 8 ...
##  $ D18S51            : int  17 12 13 17 16 12 13 12 15 14 ...
##  $ D5S818            : int  11 11 11 10 11 11 11 11 12 11 ...
##  $ FGA               : int  22 24 20 21 22 19 21 20 23 22 ...
##  $ FGA_2             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ INDIVIDUO.REGION.N: int  1 2 1 1 1 1 1 1 1 1 ...
```

La base de datos inicial estaba sesgada hacia los individuos de la regiÃ³n 1 (CENTRO) por lo tanto se tomo el valor minimo para una region (78) para igualar a todas las regiones.

```
table(DataYasser.Individuos.78$INDIVIDUO.REGION.N)
```

```
##
##  1  2  3  4
## 78 78 78 78
```

NORMALIZACIÃN DE LOS DATOS

```
normalize <- function(x) {
    return((x - min(x)) / (max(x) - min(x)))
}
```

```
Yass_norm <- as.data.frame(lapply(DataYasser.Individuos.78, normalize))
```

```
summary(Yass_norm$D8S1179)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.4000  0.5000  0.4782  0.6000  1.0000
```

Se divide la data en un grupo de entrenamiento y otro de prueba.

```
Yass_train <- Yass_norm[1:234, ]
Yass_test <- Yass_norm[235:312, ]
```

1- APLICACIÃN DE REDES NEURONALES

Primero se realizÃ³ el Bagging, para determinar cuantas neuronas de la capa oculta (hidden) pueden optimizar la red neuronal.

library(lattice)

library(ggplot2)

library(caret)

set.seed(300)

mybag <- train(INDIVIDUO.REGION.N ~ D8S1179 + D21S11 + D21S11_2 + D7S820 + CSF1PO + D3S1358 + THO1 + THO1_3 + D13S317 + D16S539 + D2S1338 + D19S433 + D19S433_2 + VWA + TPOX + D18S51 + D5S818 + FGA + FGA_2, data = Yass_norm, method = ânnetâ)

mybag

Neural Network

312 samples

19 predictor

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 312, 312, 312, 312, 312, 312, â¦

Resampling results across tuning parameters:

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were size = 1 and decay = 0.1.

```
library(MASS)
library(grid)
library(neuralnet)
```

```
## Warning: package 'neuralnet' was built under R version 3.2.5
```

```
set.seed(12345)
```

```
Yass_model <- neuralnet(INDIVIDUO.REGION.N ~ D8S1179 + D21S11
                          + D21S11_2 + D7S820 + CSF1PO + D3S1358 + THO1 + THO1_3 + D13S317 + D16S539 + D2S1338 + D19S433 + D19S433_2 + VWA + TPOX + D18S51 +  D5S818 +  FGA + FGA_2,
                          data = Yass_train, hidden = 1)
```

```
plot(Yass_model)
```

alt text

alt text

```
model_results <- compute(Yass_model, Yass_test[1:19])
```

```
predicted_region <- model_results$net.result
```

```
cor(predicted_region, Yass_test$INDIVIDUO.REGION.N)
```

```
##              [,1]
## [1,] 0.1089192354
```

Aplico la Red Neuronal a la data completa

```
set.seed(12345)
```

```
Yass_model_T <- neuralnet(INDIVIDUO.REGION.N ~ D8S1179 + D21S11
                          + D21S11_2 + D7S820 + CSF1PO + D3S1358 + THO1 + THO1_3 + D13S317 + D16S539 + D2S1338 + D19S433 + D19S433_2 + VWA + TPOX + D18S51 +  D5S818 +  FGA + FGA_2,
                          data = Yass_norm, hidden = 1)
```

```
plot(Yass_model_T)
```

alt text

alt text

```
model_results <- compute(Yass_model_T, Yass_test[1:19])
```

```
predicted_region2 <- model_results$net.result
```

```
cor(predicted_region2, Yass_test$INDIVIDUO.REGION.N)
```

```
##              [,1]
## [1,] 0.2782029083
```

```
cor(predicted_region, predicted_region2)
```

```
##               [,1]
## [1,] 0.03659059318
```

2- APLICACIÃN DE SUPPORT VECTOR MACHINE

Primero convierto a factor la columna de INDIVIDUO.REGION.N

```
DataYasser.Individuos.78 <- read.csv("~/DataYasser-Individuos-78.csv", sep=";")
```

```
DataYasser.Individuos.78$INDIVIDUO.REGION.N <- factor(DataYasser.Individuos.78$INDIVIDUO.REGION.N, levels = c("1", "2", "3", "4"),
labels = c("CENTRO", "OCCIDENTE", "LLANOS", "ORIENTE"))
```

```
table(DataYasser.Individuos.78$INDIVIDUO.REGION.N)
```

```
##
##    CENTRO OCCIDENTE    LLANOS   ORIENTE
##        78        78        78        78
```

Separo la data test y de entrenamiento con la columna de region como factor.

```
Yass_train <- DataYasser.Individuos.78[1:234, ]
Yass_test <- DataYasser.Individuos.78[235:312, ]
```

```
library(kernlab)
```

```
clasificador_region <- ksvm(INDIVIDUO.REGION.N ~ ., data = Yass_train,
kernel = "vanilladot")
```

```
##  Setting default kernel parameters
```

```
clasificador_region
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Linear (vanilla) kernel function.
##
## Number of Support Vectors : 216
##
## Objective Function Value : -110.7714 -60.5523 -69.9406 -71.7895 -69.1013 -43.4563
## Training error : 0.465812
```

```
predictor_region <- predict(clasificador_region, Yass_test[, 1:19])
```

```
table(predictor_region, Yass_test$INDIVIDUO.REGION.N)
```

```
##
## predictor_region CENTRO OCCIDENTE LLANOS ORIENTE
##        CENTRO         0         0      5       2
##        OCCIDENTE      1         0     21      22
##        LLANOS         0         0     10       2
##        ORIENTE        0         0      3      12
```

```
agreement <- predictor_region == Yass_test$INDIVIDUO.REGION.N
```

```
table(agreement)
```

```
## agreement
## FALSE  TRUE
##    56    22
```

```
prop.table(table(agreement))
```

```
## agreement
##        FALSE         TRUE
## 0.7179487179 0.2820512821
```

Prueba aumentando el costo y cambiando el Kernel a rbfdot.

```
clasificador_region2 <- ksvm(INDIVIDUO.REGION.N ~ ., data = Yass_train,
kernel = "rbfdot", C = 2)
```

```
predictor_region2 <- predict(clasificador_region, Yass_test[1:19])
```

```
table(predictor_region2, Yass_test$INDIVIDUO.REGION.N)
```

```
##
## predictor_region2 CENTRO OCCIDENTE LLANOS ORIENTE
##         CENTRO         0         0      5       2
##         OCCIDENTE      1         0     21      22
##         LLANOS         0         0     10       2
##         ORIENTE        0         0      3      12
```

```
agreement2 <- predictor_region2 == Yass_test$INDIVIDUO.REGION.N
```

```
table(agreement2)
```

```
## agreement2
## FALSE  TRUE
##    56    22
```

```
prop.table(table(agreement2))
```

```
## agreement2
##        FALSE         TRUE
## 0.7179487179 0.2820512821
```

CONCLUSIONES

1- El modelo de Redes Neuronales, revela que la algunos marcadores STR muestran pesos diferentes para la clasificaciÃ³n de los individuos por regiones, lo cual pudiera ser Ãºtil si se combina la informaciÃ³n de Ã©sos marcadores para dar mas peso a la clasificaciÃ³n.

2- Al comparar los datos de entrenamiento y de prueba, la correlaciÃ³n es muy baja (0,278) por lo tanto estos datos y/o los marcadores STR no son suficientes para establecer una buena correlaciÃ³n entre la data de entrenamiento y la data de prueba y asÃ poder predecir la region de origen de un individuo dados los genotipos de los 15 marcadores STR estudiados.

3- En el modelo de Support Vector Machine, se observÃ³ que solo hubo buena clasificaciÃ³n para 12 individuos de Oriente y 10 de los llanos, lo cual representa solo un 28 % de la prediccIÃ³n.

4- Cambiando el Kernel y el costo para este modelo, se obtuvo el mismo resultado.

5- Se puede decir que los datos analizados y/o los marcadores estudiados no aportan suficiente informaciÃ³n para lograr diferenciar a los individuos en regiones geograficas de Venezuela y asÃ establecer una adecuada prediccIÃ³n de su origen geogrÃ¡fico.

6- Pudiera ser Ãºtil, aumentar la muestra y estudiar otros marcadores de ADN que pudieran ser mÃ¡s informativos, asÃ como establecer combinaciones de los marcadores que aportan mayor o menor peso a la regiÃ³n (del modelo de redes neurales) y asÃ probar Ã©stos y otros modelos de clasificaciÃ³n.

Graciasâ¦