PEC1: Predicción de los splice junctions

Machine Learning

Descripción

Los *splice junctions* son puntos en una secuencia de ADN en los que se elimina el ADN "superfluo" durante el proceso de síntesis de proteínas en organismos superiores. El problema que se plantea en este conjunto de datos es reconocer, dada una secuencia de ADN, los límites entre los exones (las partes de la secuencia de ADN retenidas después del *splicing*) y los intrones (las partes de la secuencia de ADN que se cortan). Este problema consta de dos subtareas: reconocer los límites exón/intrón (denominados sitios EI) y reconocer los límites intrón/exón (sitios IE). En la comunidad biológica, las fronteras de la IE se denominan `acceptors`, mientras que las fronteras de la EI se denominan `donors`. Todos los ejemplos fueron tomados de Genbank 64.1. Las categorías "EI" e "IE" incluyen "genes con splicing" de los primates en Genbank 64.1. Los ejemplos de no *splicing* fueron tomados de secuencias que se sabe que no incluyen un sitio de *splicing*.

Los datos estan disponibles en la PEC en el fichero `splice.txt`. El archivo contiene 3190 filas que corresponden a las distintas secuencias, y 3 columnas separadas por coma. La primera columna correspondiente a la clase de la secuencia (EI, IE o N), la segunda columna con el nombre identificador de la secuencia y la tercera columna con la secuencia propiamente. Tratandose de secuencias de ADN, apareceran los nucleótidos identificados de manera estandar con las letras A, G, T y C. Además, aparecen otros caracteres entre los caracteres estándar, D, N, S y R, que indican ambigüedad según la siguiente tabla:

```
|caracter | significado   |
|---------|---------------|
|D        | A o G o T     |
|N        | A o G o C o T |
|S        | C o G         |
|R        | A o G         |
```

Para más información acerca del caso se puede consultar el link: https://archive.ics.uci.edu/ml/datasets/Molecular+Biology+(Splice-junction+Gene+Sequences), del repositorio de machine learning de la Universidad de California-Irvine (UCI).

La manera elegida para representar los datos es un paso crucial en los algoritmos. En el caso que nos ocupa, análisis basados en secuencias, se usará la codificación `one-hot`.

La codificación `one-hot` representa cada nucleótido por un vector de 8 componentes, con 7 de ellas a 0 y una a 1. Pongamos por ejemplo, el nucleótido A se representa por (1,0,0,0,0,0,0,0), el nucleótido G por (0,1,0,0,0,0,0,0), el T por (0,0,1,0,0,0,0,0) y, finalmente, la C por (0,0,0,1,0,0,0,0) y los caracteres de ambiguedad los representaremos, la D por (0,0,0,0,1,0,0,0), la N por (0,0,0,0,0,1,0,0), la S por (0,0,0,0,0,0,1,0) y la R por (0,0,0,0,0,0,0,1).

Entonces, cada secuencia de 60 nucleótidos se convertirá en un vector de 8*60=480 componentes resultado de concatenar los vectores para cada uno de los 60 nucleótidos. A modo de ejemplo, se muestra la primera secuencia de la base de datos y el resultado de la codificación one-hot. Observar que será necesario eliminar los espacios en blanco previos a las secuencias.

```
[1] "               CCAGCTGCATCACAGGAGGCCAGCGAGCAGGTCTGTTCCAAGGGCCTTCGAGCCAGTCTG"
```

Podemos usar la función `str_trim` del paquete stringr.

```
[1] "CCAGCTGCATCACAGGAGGCCAGCGAGCAGGTCTGTTCCAAGGGCCTTCGAGCCAGTCTG"
```

Resultado del proceso `one-hot`

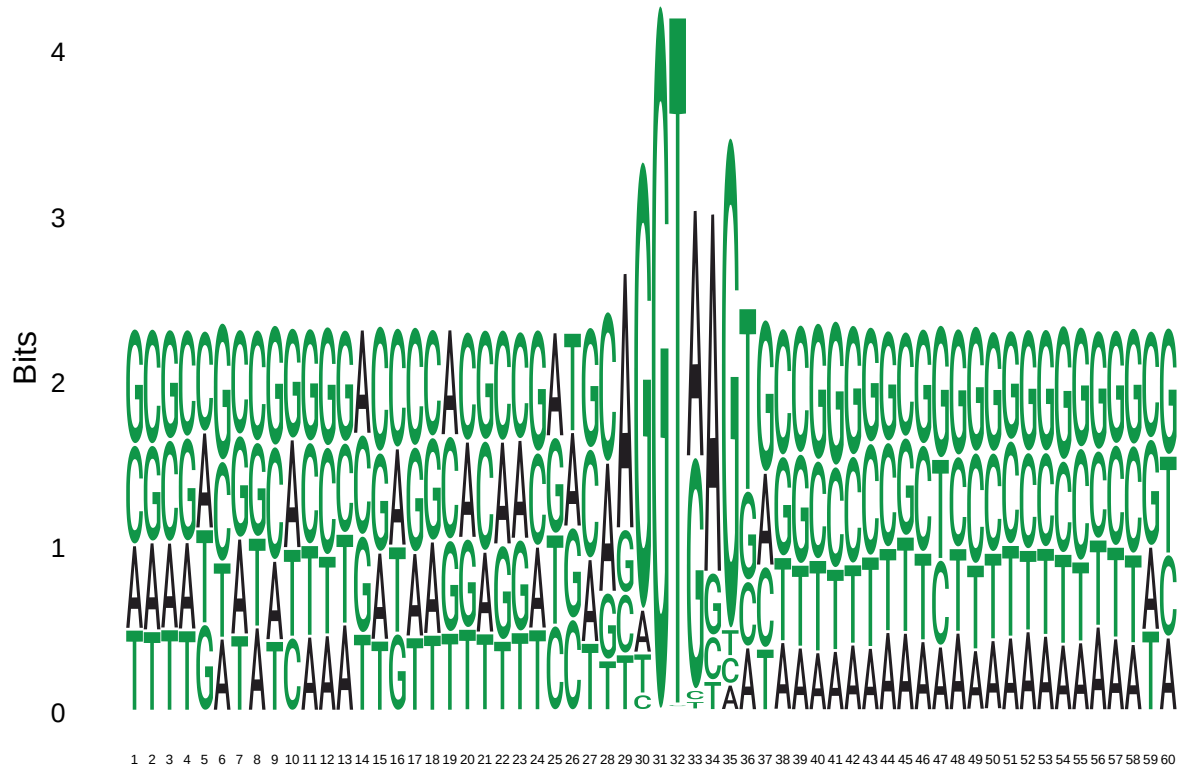
```
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
1  0  0  0  1  0  0  0  0  0   0   0   1   0   0   0   0   1   0   0   0   0
  V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40
1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0   0
  V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57 V58 V59
1   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
  V60 V61 V62 V63 V64 V65 V66 V67 V68 V69 V70 V71 V72 V73 V74 V75 V76 V77 V78
1   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0
  V79 V80 V81 V82 V83 V84 V85 V86 V87 V88 V89 V90 V91 V92 V93 V94 V95 V96 V97
1   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0
  V98 V99 V100 V101 V102 V103 V104 V105 V106 V107 V108 V109 V110 V111 V112 V113
1   0   0    1    0    0    0    0    1    0    0    0    0    0    0    0    0
  V114 V115 V116 V117 V118 V119 V120 V121 V122 V123 V124 V125 V126 V127 V128
1    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0
  V129 V130 V131 V132 V133 V134 V135 V136 V137 V138 V139 V140 V141 V142 V143
1    1    0    0    0    0    0    0    0    0    1    0    0    0    0    0
  V144 V145 V146 V147 V148 V149 V150 V151 V152 V153 V154 V155 V156 V157 V158
1    0    0    1    0    0    0    0    0    0    0    0    0    1    0    0
  V159 V160 V161 V162 V163 V164 V165 V166 V167 V168 V169 V170 V171 V172 V173
1    0    0    0    0    0    1    0    0    0    0    1    0    0    0    0
  V174 V175 V176 V177 V178 V179 V180 V181 V182 V183 V184 V185 V186 V187 V188
1    0    0    0    0    1    0    0    0    0    0    0    0    0    0    1
  V189 V190 V191 V192 V193 V194 V195 V196 V197 V198 V199 V200 V201 V202 V203
1    0    0    0    0    0    1    0    0    0    0    0    0    1    0    0
  V204 V205 V206 V207 V208 V209 V210 V211 V212 V213 V214 V215 V216 V217 V218
1    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
  V219 V220 V221 V222 V223 V224 V225 V226 V227 V228 V229 V230 V231 V232 V233
1    0    1    0    0    0    0    1    0    0    0    0    0    0    0    0
  V234 V235 V236 V237 V238 V239 V240 V241 V242 V243 V244 V245 V246 V247 V248
1    1    0    0    0    0    0    0    0    1    0    0    0    0    0    0
  V249 V250 V251 V252 V253 V254 V255 V256 V257 V258 V259 V260 V261 V262 V263
1    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0
  V264 V265 V266 V267 V268 V269 V270 V271 V272 V273 V274 V275 V276 V277 V278
1    0    0    0    1    0    0    0    0    0    0    1    0    0    0    0
  V279 V280 V281 V282 V283 V284 V285 V286 V287 V288 V289 V290 V291 V292 V293
1    0    0    0    0    1    0    0    0    0    0    0    0    1    0    0
  V294 V295 V296 V297 V298 V299 V300 V301 V302 V303 V304 V305 V306 V307 V308
1    0    0    0    0    0    0    1    0    0    0    0    0    0    0    1
  V309 V310 V311 V312 V313 V314 V315 V316 V317 V318 V319 V320 V321 V322 V323
1    0    0    0    0    1    0    0    0    0    0    0    0    1    0    0
  V324 V325 V326 V327 V328 V329 V330 V331 V332 V333 V334 V335 V336 V337 V338
1    0    0    0    0    0    0    1    0    0    0    0    0    0    0    1
  V339 V340 V341 V342 V343 V344 V345 V346 V347 V348 V349 V350 V351 V352 V353
1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0
  V354 V355 V356 V357 V358 V359 V360 V361 V362 V363 V364 V365 V366 V367 V368
1    0    0    1    0    0    0    0    0    0    0    1    0    0    0    0
  V369 V370 V371 V372 V373 V374 V375 V376 V377 V378 V379 V380 V381 V382 V383
1    0    0    1    0    0    0    0    0    0    0    1    0    0    0    0
  V384 V385 V386 V387 V388 V389 V390 V391 V392 V393 V394 V395 V396 V397 V398
1    0    0    0    0    1    0    0    0    0    0    1    0    0    0    0
  V399 V400 V401 V402 V403 V404 V405 V406 V407 V408 V409 V410 V411 V412 V413
1    0    0    1    0    0    0    0    0    0    0    0    1    0    0    0
```

```
  V414 V415 V416 V417 V418 V419 V420 V421 V422 V423 V424 V425 V426 V427 V428
1    0    0    0    0    0    0    1    0    0    0    0    0    0    0    1
  V429 V430 V431 V432 V433 V434 V435 V436 V437 V438 V439 V440 V441 V442 V443
1    0    0    0    0    1    0    0    0    0    0    0    0    0    1    0
  V444 V445 V446 V447 V448 V449 V450 V451 V452 V453 V454 V455 V456 V457 V458
1    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0
  V459 V460 V461 V462 V463 V464 V465 V466 V467 V468 V469 V470 V471 V472 V473
1    0    1    0    0    0    0    0    0    1    0    0    0    0    0    0
  V474 V475 V476 V477 V478 V479 V480
1    1    0    0    0    0    0    0
```

Secuencias logo

Una forma de presentar el patrón de la secuencia de un manera gráfica es realizar una secuencia logo (ver https://en.wikipedia.org/wiki/Sequence_logo)). "Para crear logos de secuencias, las secuencias relacionadas de ADN, ARN o proteínas, o bien secuencias de ADN que comparten lugares de unión conservados, son alineadas hasta que las partes más conservadas crean buenos alineamientos. Se puede crear entonces un logo de secuencias a partir del alineamiento múltiple de secuencias conservadas. El logo de secuencias pondrá de manifiesto el grado de conservación de los residuos en cada posición: un menor número de residuos diferentes provocará mayor tamaño en las letras, ya que la conservación es mejor en esa posición. Los residuos diferentes en la misma posición se escalarán de acuerdo a su frecuencia. Los logos de secuencias pueden usarse para representar sitios conservados de unión al ADN, donde quedan unidos los factores de transcripción" (extraido de https://es.wikipedia.org/wiki/Logo_de_secuencias). Para realizar esta representación se usa el paquete ggseqlogo descargable desde CRAN.

A modo de ejemplo, os mostramos la secuencia logo de la clase "EI"



Enunciado

1. Escribir en el informe una sección con el título "Algoritmo k-NN" en el que se haga una breve explicación de su funcionamiento y sus características. Además, se presente una tabla de sus fortaleza y debilidades.
2. Desarrollar una función en R que implemente una codificación "one-hot" (one-hot encoding) de las secuencias. Presentar un ejemplo simple de su uso.
3. Desarrollar un script en R que implemente un clasificador knn. El script realiza los siguientes apartados:
   (a) Leer los datos del fichero `splice.txt` y hacer una breve descripción de ellos.
   (b) Transformar las secuencias de nucleótidos en vectores numéricos usando la función de transformación desarrollada anteriormente. En caso que no se haya implementado la función de codificación one-hot, se puede acceder a los datos ya transformados cargando el fichero `splice_oh.RData`.
   (c) Para el subset formado por las secuencias de las clases "EI" y "N", y para el subset formado por las secuencias de las clases "IE" y "N", realizar la implementación del algoritmo knn, con los siguientes pasos:
      i. Utilizando la semilla aleatoria 123, separar los datos en dos partes, una parte para training (67%) y una parte para test (33%).
      ii. Aplicar el knn ($k = 1, 5, 11, 21, 51, 71$) basado en el training para predecir que secuencias del test son secuencias con puntos de splicing (splice junctions) o no. Además, realizar una curva ROC para cada k y mostrar el valor de AUC.
   (d) Comentar los resultados de la clasificación en función de la curva ROC, valor de AUC y del número de falsos positivos, falsos negativos y error de clasificación obtenidos para los diferentes valores de k. La clase asignada como positiva son las representan secuencias con puntos de splicing.
4. Representar la sequencia logo de las tres clases de secuencias. Comentar los resultados obtenidos.

Informe de la PEC

El informe se presentará mediante un informe dinámico R markdown con la estructura habitual de los ejercicios no evaluables realizados hasta ahora. En primer lugar, el informe tendrá un título (igual que el de la PEC), el autor, la fecha de creación y el índice de apartados de la PEC. En segundo lugar, se crea una sección con el título "Algoritmo k-NN" donde se haga una breve explicación de su funcionamiento y sus características. Además, se presenta la tabla de sus fortaleza y debilidades. En tercer lugar se realizan los diferentes apartados de la PEC. Una característica que se valorará es hasta que punto es el informe "dinámico". En el sentido de adaptarse el informe a cambios en los datos.

Se entregaran dos ficheros:

1. Fichero ejecutable (.Rmd) que incluya un texto explicativo que detalle los pasos implementados en el script y el código de los análisis.
2. Informe (pdf) resultado de la ejecución del fichero Rmd anterior.

Puntuacions de los apartados

Apartado 1 (5%), Apartado 2 (25%), Apartado 3 (50%), Apartado 4 (15%), Calidad del informe dinámico (5%).