Predicción de interacción entre péptido y el complejo mayor de histocompatibilidad tipo I con Artificial Neural Networks (ANN) y Support Vector Machines (SVM)

Introducción

En esta PEC vamos a resolver un análisis relacionado con la modelización de la interacción entre el complejo mayor de histocompatibilidad tipo I (MHCI, en inglés) y péptidos.

En la inmunoterapia contra el cancer las células T deben activarse al exponerse a péptidos tumorales unidos a MHCI (pMHCI). Al analizar la genética del tumor, se pueden identificar péptidos relevantes y, dependiendo del tipo particular de MHCI que tiene el paciente, podemos predecir qué interacción péptido MHCI (pMHCI) es probable que esté presente en el tumor del paciente y, por lo tanto, qué pMHCI se deben usar para activar las células T.

Los ficheros necesarios para realizar la PEC estan en formato csv con separador punto y coma.

En cada registro del fichero peptidos.csv se tienen dos variables: 1) el péptido, 2) la clase de interacción donde NB significa no interacción y SB significa si interacción.

La manera elegida para representar los datos es un paso crucial en los algoritmos de clasificación. En el caso que nos ocupa, análisis basados en secuencias, se usará la transformación denominada **one-hot encoding**.

El one-hot encoding representa cada aminoácido por un vector de 20 componentes, con 19 de ellas a 0 y una a 1 indicando el aminoácido. Pongamos por ejemplo, el aminoácido A se representa por $(1,0,\ldots,0)$ y el aminoácido R por $(0,1,0,\ldots,0)$. Por tanto, para una secuencia de 9 aminoácidos, como en nuestro caso, se obtendrá un vector de 20*9 componentes, resultado de concatenar los vectores para cada uno de los 9 aminoácidos.

Una vez realizada la transformación, one-hot encoding el objetivo es implementar un algoritmo de red neuronal artificial y de support vector machine para predecir si la secuencia peptídica interacciona o no con MHCI.

A modo de ejemplo se muestra el resultado de codificar el primer registro de la base de datos con one-hot encoding

• one-hot encoding: vector de 0s y 1s del primer registro.

Enunciado

- 1. Escribir en el informe dos secciones con los títulos: "Algoritmo Red Neuronal Artificial" y "Algoritmo Support Vector Machine" en el que se haga una breve explicación de su funcionamiento y sus características. Además, se presente una tabla de sus fortaleza y debilidades para cada algoritmo.
- 2. Leer los datos peptidos.csv y hacer una breve descripción de ellos. Incluir en esta descripción el patrón de cada clase de péptido mediante la representación de su secuencia logo (https://en.wikiped

- ia.org/wiki/Sequence_logo). Para realizar esta representación se puede usar el paquete ggseqlogo descargable desde github.
- 3. Desarrollar una función en R que implemente la codificación "one-hot" (one-hot encoding) de las secuencias.
- 4. Transformar las secuencias de aminoácidos en vectores numéricos usando la función de transformación desarrollada en el punto anterior.
- 5. Desarrollar un código en R que implemente un clasificador de red neuronal artificial. El código en R debe:
 - (a) Leer los datos transformados. En caso de no haber podido realizar la función se dispone del fichero peptidos_transf_one_hot.csv.
 - (b) Utilizando la semilla aleatoria 123, separar los datos en dos partes, una parte para training (67%) y una parte para test (33%).
 - (c) Antes de ejecutar cada uno de los modelos de clasificación que se piden a continuación, poner como semilla generadora el valor 1234567.
 - (d) Crear dos modelos de red neuronal artificial de una sola capa oculta con 1 nodo y 3 nodos, respectivamente. Aplicar los datos de training para ajustar los modelos y posteriormente, predecir si la secuencia peptídica interacciona o no con MHCI en los datos del test.
 - (e) Comentar los resultados de la clasificación en función de los valores generales de la clasificación como "accuracy" y otros. Comparar los resultados de clasificación obtenidos para los diferentes valores de nodos usados en la capa oculta.
 - (f) Usar el paquete caret modelo 'mlp' para implementar la arquitectura de 3 nodos en la capa oculta, usando 5-fold crossvalidation. Comentar los resultados.
- 6. Desarrollar un código en R que implemente un clasificador de SVM. El código en R debe:
 - (a) Leer los datos transformados. En caso de no haber podido realizar la función se dispone del fichero peptidos_transf_one_hot.csv.
 - (b) Utilizando la semilla aleatoria 123, separar los datos en dos partes, una parte para training (67%) y una parte para test (33%).
 - (c) Antes de ejecutar cada uno de los modelos de clasificación que se piden a continuación, poner como semilla generadora el valor 1234567.
 - (d) Utilizar la función lineal y la RBF para ajustar un modelo de SVM basado en el training para predecir si la secuencia peptídica interacciona o no con MHCI en los datos del test.
 - (e) Comentar los resultados de la clasificación en función de los valores generales de la clasificación como "accuracy" y otros. Comparar los resultados de clasificación obtenidos para los diferentes funciones kernel usadas.
 - (f) Usar el paquete caret modelo svmRBF para aplicar el algoritmo de SVM con 5-fold crossvalidation. Comentar los resultados.
- 7. Comentar todos los resultados obtenidos y escoger que modelo puede ser el mejor.

Informe de la PEC

Las soluciones se presentarán 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 Red Neuronal Artificial" 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 crea la sección "Algoritmo Support Vector Machine" similar a la anterior sección. En cuarto lugar

se realizan los diferentes apartados de la PEC pero con la estructura de Step1 hasta Step5 para cada tipo de algoritmo. Al final se crea una sección "Discusión final" para comentar todos los resultados obtenidos y escoger el mejor modelo.

Un 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 subiran al registro de entregas un **zip** con los siguientes ficheros:

1. Fichero ejecutable (.Rmd) que incluya un texto explicativo que detalle los pasos implementados en el script y el código de los análisis. No olvidar de incluir todos los ficheros complementarios que hagan falta para la correcta ejecución: ficheros de datos, fichero de bibliografia, imagenes, ...

NOTA: Para facilitar la ejecución, no usar un ruta fija para la lectura del fichero, asociarlo al area de trabajo donde este el fichero .Rmd.

2. Informe (pdf) resultado de la ejecución del fichero Rmd anterior.

Antes de enviar el zip, se recomienda **verificar la reproducibilidad del fichero .Rmd** para obtener el informe en formato pdf sin ninguna dificultad.

Puntuacions de los apartados

Apartado 1 y Apartado 2 (5%), Apartado 3 y Apartado 4 (25%), Apartado 5 (30%), Apartado 6 (30%), Apartado 7 (5%), Calidad del informe dinámico (5%).