Bioinformática

Práctica sobre motivos funcionales

Javier Juan Albarracín, Mabel Mompeán Escartí, Carlos Sáez Silvestre

Departamento de Física Aplicada, Universitat Politècnica de València

2023

SECUENCIA CONSENSO

Uno de los métodos para realizar búsquedas de motivos funcionales en secuencias de proteínas es mediante la *Secuencia Consenso*. Esta es la secuencia de aminoácidos más frecuente para cada posición del alineamiento (ver ejemplo en transparencias).

Para este ejercicio, os proporcionamos un fichero de nombre *sequences.txt* donde se encuentra un conjunto de secuencias de una proteína desconocida. Os pedimos:

1) **Implementa un script en R que obtenga la secuencia consenso del conjunto de secuencias del fichero "Secuencias.txt".**
   Para ello ayúdate de las funciones que están descritas en el ANEXO I.
2) **Realiza una búsqueda en PROSITE de la *Secuencia Consenso* obtenida. Introducir la secuencia en el apartado *Sequence(s) to be scanned* y dejar el resto de parámetros por defecto. (**https://prosite.expasy.org/scanprosite/**)**
3) **¿A qué proteína corresponde la secuencia consenso?**
4) **Accede al link de la proteína en la base de datos PROSITE. Busca el patrón de consenso y comprueba que tu secuencia consenso se ajusta a este patrón.**
5) **Interpreta el perfil de la secuencia. Para ello accede al link *Retrieve the sequence logo from the alignment.***
6) **Realizar una búsqueda en BLAST (**http://blast.ncbi.nlm.nih.gov/Blast.cgi**) utilizando el programa *protein_blast*. ¿Cuál es el mejor alineamiento que da BLAST para esta *Secuencia Consenso* para el *Homo Sapiens*? ¿Cuál es su tasa de acierto? Busca en su ficha (a través del *accesion*) de qué longitud es la proteína y en qué posiciones se encuentra el motivo. Visualiza su estructura 3D y mediante su enlace en *PDB* busca en qué cromosoma se encuentra (en ficha PDB ver *Genome*).**

EXPRESIONES REGULARES.

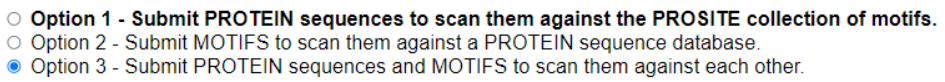
Otro de los métodos para la búsqueda de motivos funcionales es la búsqueda mediante expresiones regulares. Un ejemplo de expresión regular seria: **E-x-[KR]-E-x(2)-E-[KR]-[LF]-[LIVMA]-x(2)-Q-N-x-R-x-G-R**, que indica que las secuencias que se ajustan a la expresión comienzan con una "E", seguido de cualquier carácter, seguido de o una "K" o una "R", etc.

En el fichero BD.txt se encuentra una base de datos de diferentes subsecuencias de proteínas. Estas subsecuencias de proteínas corresponden a: Ferritina: PS00540, Insuline: PS00262, Transaminase: PS00105, Tissue factor: PS00621, con sus identificadores en PROSITE. Os pedimos:

1) **Realizar la búsqueda del identificador de la Ferritina (*signature* 1) en PROSITE (http://expasy.org/prosite/). ¿Qué es la Ferritina?**

2) **Completa el script *findMotifs.R* con la expresión regular de la Ferritina 1. Para ello será necesario traducir la expresión regular de PROSITE a la sintaxis de *grep*. Ejecutar el script y éste realizará una búsqueda de motivos mediante expresión regular en el fichero BD.txt y devolverá las secuencias que se ajustan a ese patrón en un fichero.**

3) **Comprobar que la expresión regular que habéis implementado es correcta comparando el resultado obtenido por el script con PROSITE (https://prosite.expasy.org/scanprosite/). Para ello, marcar la opción 3 y pegar la base de datos y el motivo:**



4) **Realizar el alineamiento múltiple del fichero de resultados anterior mediante la herramienta Clustal Omega. Dejar todos los parámetros por defecto. (http://www.ebi.ac.uk/Tools/msa/clustalo/)**

5) **Analizar los resultados de las secuencias obtenidas. ¿Cuáles son las secuencias más parecidas?, ¿y las menos?, ¿Cuales guardan un ancestro común? ¿Influye este ancestro en la similitud de las secuencias? Para ello observar el *Result Summary* y el *Guide Tree*.**

6) **¿Qué significa la *especificidad* y la *sensibilidad* en expresiones regulares y en concreto en el campo de la BioInformática?**

ANEXO I

Funciones R de ayuda. Para más información teclear en la consola de R ?<comando>.

- **c(nelem):** Crea un vector de “nelem” posiciones (?c).
- **dim(x):** Devuelve la dimensión de una matriz o vector (?dim).
- **max(x):** Calcula el máximo de un conjunto de valores (?max).
- **rep(x, times):** Crea un vector repitiendo el valor de “x” “times” veces. (?rep).
- **matrix(x, nrow, ncol):** Crea una matriz con los valores de la variable “x” y de “nrow” número de filas y “ncol” número de columnas. (?matrix).
- **names(x):** Devuelve el nombre de la fila o columna de la matriz. Ver también **rownames(x), colnames(x)** (?names).
- **ncol(x):** Devuelve el número de columnas de una matriz (?ncol).
- **nrow(x):** Devuelve el número de filas de una matriz (?nrow).
- **print(x):** Imprime por pantalla el contenido de una variable (?print).
- **read.table(file = “nomFich.txt”, stringsAsFactors = TRUE):** Carga un fichero en forma de data.frame, estableciendo a las cadenas de texto la clase de categoría Factor.
- **summary(x):** Realiza el sumatorio del conjunto de valores “x”, que puede ser tanto valores enteros como lógicos (?summary).
- **which.max(x):** Es la función argmax, devuelve el argumento del máximo del conjunto de valores “x” (?which).
- **write.table(var, file=”nomFich.txt”):** Escribe el contenido de la variable “var” en el fichero de nombre “nomFich.txt” (?write).
- **print(x, quote = FALSE):** Imprime por pantalla x quitando las comillas. (?print).
- **paste(vector, collapse = x)**: Une los elementos de texto de vector usando x como nexo. (?paste).