**Modulo 4: Minería de Datos y Modelos Ocultos de Markov**

Objetivos Generales. En este módulo utilizaremos por linea de comando dos programas muy utilizados en bioinformática, MAFFT para hacer alineamientos múltiples y HMMER para trabajar con Modelos Ocultos de Markov (HMM del inglés, Hidden Markov Model). El objetivo general es poder utilizar estos programas para realizar nuestro propio HMM y MSA (del inglés Mutiple Sequence Alignment) de un grupo de proteínas de interés y realizar la búsqueda de todas las secuencias que componen nuestro grupo en una dada base de datos. Además aprenderemos a definir los umbrales, y analizar la capacidad de nuestro modelo en cuanto a sensibilidad y especificidad.

Los objetivos específicos son:

i) Instalación y uso de HMMER y MAFFT por línea de comando.

ii) Parseo de salidas de HMMER.

iii) Construcción de histogramas de puntaje.

**Para empezar, contemos una pequeña historia**... “*Su jefe tiene un set de casi 38 mil secuencias de dominios de proteínas de las 3 grandes familias de globinas, pero accidentalmente borró la única copia del archivo fasta con dichas secuencias. Afortunadamente, su jefe dispone de otro archivo con las secuencias completas de las proteínas (es decir, no solo el dominio globina sino otros dominios que tuviera la secuencia completa también). Sin embargo, su jefe le dice que existe un modo fácil de recuperar los dominios globina de cada familia, y eso es a través de la minería de datos de secuencias utilizando HMMs. Su jefe le pide que primero empiece por separar las proteínas de la familia de las protoglobinas, para lo cual debe descargar un subconjunto de secuencias llamado “seed” o “semilla” de la base de datos Pfam correspondiente a la familia “Protoglobin”. Con estas secuencias puede crear un MSA (alineamiento múltiple de secuencias, o Multiple Sequence Alignment), con el cual generar un HMM y realizar la minería de datos sobre el total de secuencias completas de globinas, y recuperar con eficiencia la gran parte de los dominios globina de la familia protoglobina....”*

Pongámonos manos a la obra:

**Objetivo 1: instalar y correr MAFFT y HMMER**

**1a) Instalar MAFFT y HMMER**

En primer lugar necesitará instalar algunos programas. Éstos son MAFFT, para el MSA, y HMMER para generar HMMs y realizar minería de datos.

sudo apt-get install mafft

sudo apt-get install hmmer

**1b) Obtener conjunto de secuencias “semilla”**

Si usted recuerda cómo se utilizan los HMM en el contexto de Bases de Datos Secundarias, recordará que debemos ser capaces de: i) resolver el problema de scoring, ii) resolver el problema de alineamiento y iii) resolver el problema de entrenamiento. Además recordará que el punto de partida para construir un HMM es tener un conjunto de secuencias “semilla”. En este trabajo utilizaremos como punto de partida el *seed* que se bajar de PFAM (cosa que por obvias razones suele ser siempre una buena idea).

Entonces, Vaya al sitio web de **Pfam (recuerde que ahora esta dentro de interpro), y en la barra de búsqueda que está en la parte superior derecha busque “Protoglobin”**.

Como resultado obtendrá una lista de resultados de búsqueda correspondientes a familias de proteínas. Elija la primera, que tiene el nombre “Protoglobin”. Luego, en el menú a la izquierda elija la opción “alignments”. En La parte llamada “Format an Alignment” seleccione el seed, “format”: fasta, “Gaps”: ‘no gaps (unaligned)’ y finalmente click en “generate”.

**1c) Realizar el MSA con MAFFT.** Una vez se tienen las secuencias *seed* hay que alinearlas. Para ello se usará el programa MAFFT mediante un comando de este estilo:

mafft seed.fasta > seed\_msa.fasta

Como resultado, MAFFT le devolvió sus secuencias alineadas en formato fasta (pero usted puede elegir otro formato de salida también). Si lo desea puede visualizar las secuencias alineadas en MSA (e incluso editar el MSA) con un visualizador de alineamientos como **Jalview** (puede instalarlo con “sudo apt-get install” como hizo con MAFFT y HMMER pero se descarga una versión vieja, si lo baja del sitio web puede obtener la más nueva para cualquier sistema operativo [http://www.jalview.org/getdown/release/#](http://www.jalview.org/getdown/release/))**.**

**1d) Crear el HMM.** Ahora que dispone de un *seed* alineado en un MSA puede crear un HMM con él. Para ello usará una de las herramientas que dispone HMMER: *hmmbuild.* Esta herramienta funciona con un comando de este formato:

hmmbuild seed\_modelo.hmm seed\_msa.fasta

El archivo “secuencias\_modelo.hmm” (o el nombre que le haya puesto) es un archivo de texto que contiene las probabilidades que hacen al HMM en cuestión. Este archivo, junto con las secuencias target, le permitirán realizar la minería de datos.

**Ejercicio 1a-d:** corra todos los comandos anteriores y analice un poco el archivo “secuencias\_modelo.hmm”, puede identificar en él los parámetros que usted conoce debe poseer el modelo (probabilidad de emisión de cada residuo en cada columna (estado match), columnas importantes, probabilidad de inserción, etc).

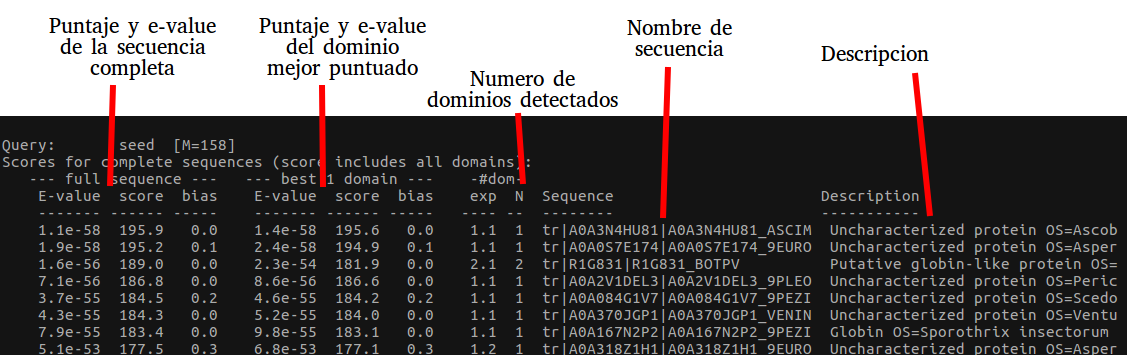
**1e) Realizar Minería de datos usando HMM.**

Para hacer esto recurriremos a otra herramienta de HMMER llamada *hmmsearch*, y utilizaremos como conjunto sobre el cual realizar la búsqueda (o sea la base de datos) un archivo local “secuencias\_target.fasta” (que lo encontrará en el drive de esta unidad de la materia).

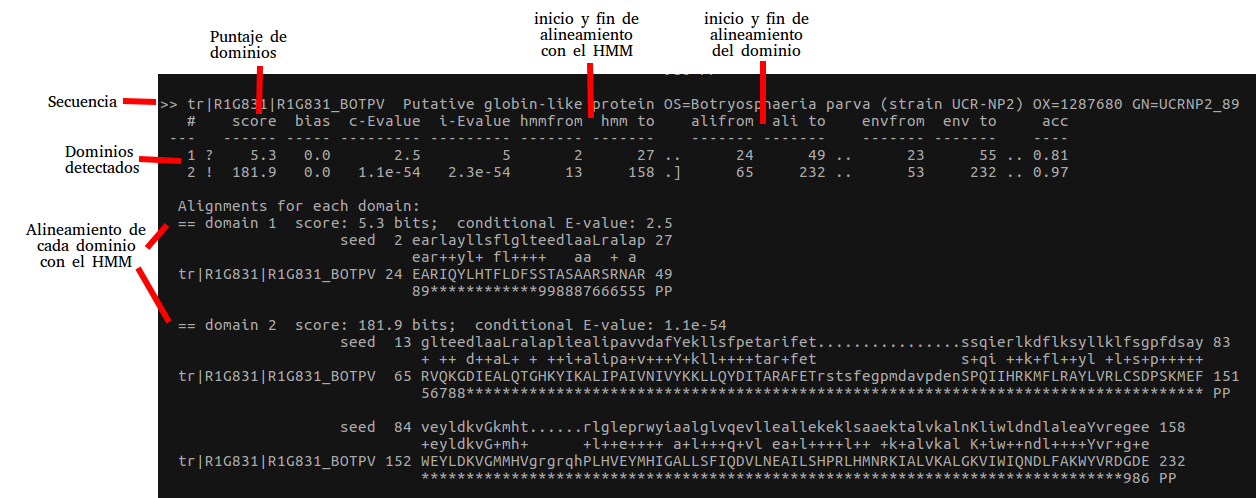
Hagamos entonces el hmmsearch:

hmmsearch secuencias\_modelo.hmm secuencias\_target.fasta > busqueda.out

Por default, *hmmsearch* le devuelve un archivo de texto con una larga tabla de resultados. Cada fila corresponde a una de las secuencias evaluadas con el HMM, y se indica el puntaje y *e-value* globales de la secuencia, el puntaje y *e-value* correspondientes al dominio mejor puntuado, la cantidad de “dominios” detectados en esa secuencia (es decir, cuántas veces el HMM “alineó” contra la secuencia), nombre y descripción de la secuencia (si la hubiera).



Además, luego de esa tabla, se profundiza en detalles de cada secuencia analizada (tiene que “scrollear” el archivo hasta que la tabla termina).



En esta sección, para cada secuencia se muestra una lista de todos los dominios detectados con sus respectivos puntajes en que alinearon el HMM y la secuencia, seguido de los alineamientos correspondientes.

**Nota:** en realidad uno buscaría sobre toda una base de datos (como Uniprot, Pfam o PDB) o el proteoma de algún organismo, pero como eso tarda mucho (y arriesgan quemar sus compus) vamos a usar un conjunto más pequeño ya disponible. De todos modos, si usted lo desea puede descargar todo Uniprot de la página web en formato fasta y correr HMMsearch contra esta (pero cuidado, Uniprot pesa más de 60Gb). Otra alternativa es correr HMMER en el server: <https://www.ebi.ac.uk/Tools/hmmer/search/hmmsearch>

Ejercicio 1e: analice un poco el archivo de salida de HMMsearch, trate de identificar dónde están los resultados obtenidos para cada secuencia query, y de ellos donde esta el E-value, la cobertura, etc.

**Objetivo 2: Recuperar los dominios con Python**

Si a esta altura usted está aburrido con esta unidad y tiene ganas de programar, no se preocupe, ahora viene lo divertido.

**2a) Histograma de puntajes de dominio**. El primer objetivo busca Implementar un script en python que grafique un histograma de puntajes de secuencia POR DOMINIO (recuerde, usted le interesa recuperar los dominios globina, no la secuencia entera de cada proteína), a partir de la salida de *hmmsearch*. Necesitará una librería para graficar cómo matplotlib y la librería de biopython SearchIO, para poder parsear el archivo salida de *hmmsearch*.

Les dejamos una ayudita con “pseudo-còdigo” para empezar:

from Bio import SearchIO

import matplotlib.pyplot as plt

resultados=list(SearchIO.read(‘archivo\_salida\_hmmer.out’, 'hmmer3-text'))

Ahora ustede debe:

Iterar secuencias (análogo a los hits de Blast)

Iterar dominios detectados (análogo a los HSP de Blast)

Añadir puntaje de dominio a lista

Para hacer el histograma y grabar/mostrar la figura puede utilizar los siguientes comandos:

histogram=plt.hist(puntajes\_lista, bins=numero\_de\_bins)

plt.savefig(nombreFigura)

plt.show()

**2b) Determinación de puntaje de corte y Extracción del archivo de los dominios de interés**. Con el gráfico que hizo mediante su script, establezca (a ojo) un puntaje de corte para recuperar los dominios de secuencia que estén por encima de dicho corte. Recuerde que el puntaje de corte es aquel valor tal que aquellas secuencias que cuando sean contrastadas contra el HMM den un valor superior al valor de corte elegido, serán consideradas como miembros de la familia/dominio/grupo que usted está analizando.

Ejercicio 2b: Desarrolle en python un código que utilice el puntaje de corte recién establecido (esto debe ser un parámetro) para extraer los dominios búscados en las secuencias correspondientes.

El mismo debe devolver un archivo fasta con dichas secuencias recortadas.

El código debe tener además las dos siguientes funciones:

i) Debe ser capaz de obtener los Uniprot IDs y las posiciones de inicio y fin de los dominios y entregar los resultados como lista de listas (o una matriz de numpy o pandas),

ii) Tomar los valores de la lista anterior (IDs y posiciones), y usarlos para cortar las secuencias del registro fasta original.

**Nota**: Para aquellos que lo deseen, a continuación les damos un poco de ayuda con el código.

Los nombres de las secuencias en el archivo de salida debe ser de este formato: P04569/3-154 (que es el formato que tienen las secuencias en la base de datos Pfam). Lo primero (“P04569”) es el identificador de la secuencia en la base de datos Uniprot, lo segundo (“3-154”) son las posiciones de inicio y fin del dominio en la secuencia completa.

Uniprot IDs y posiciones (ejemplo):

A0A0U1M3Y2 39 219

A0A1Y1ZMR4 20 200

F7VKQ0 27 207

A0A2L2TID2 17 197

A0A1Q8S944 18 198

A0A096PC56 17 197

Del formato de Uniprot al formato de Pfam:

|  |  |
| --- | --- |
| tr|F7VKQ0|F7VKQ0\_SORMK | F7VKQ0/3-158 |
| sp|P02144|MYG\_HUMAN | P02144/6-245 |
| tr|A0A2L2TID2|A0A2L2TID2\_9HYPO | A0A2L2TID2/1-144 |

Aclaración: “tr” es por “TrEMBL (secuencias no curadas de Uniprot) y “sp” por Swissprot (secuencias curadas).

Un poco de pseudocódigo de ayuda:

from Bio import SeqIO

Def extraerPosiciones(puntaje\_corte):

lista de límites

searchFile=list(SearchIO.read("resultado\_hmmsearch", 'hmmer3-text'))

Iterar secuencias

Iterar dominios en secuencia

Si dominio.puntaje > puntaje\_corte

añadir id y limites a lista de limites

Devolver lista de limites

Def guardarDominios(lista\_limites):

fasta=list(SeqIO.parse("secuencias\_completas.fasta", 'fasta'))

Iterar Fasta

Iterar lista\_limites

Si hay coincidencia modificar el ID de la secuencia y la secuencia misma

Guardar el registro modificado en una nueva lista

SeqIO.write(registros\_modificados, “salida.fasta”, “fasta”)

Sugerencias: itere lista\_limites con un ciclo “while” utilizando un contador para indexar los elementos visitados. Cuando “matchee” la secuencia en el archivo fasta con el elemento de la lista de límites, elimine el elemento de la lista de límites (ya no lo necesitará más) y resetee el contador. De este modo la lista de límites se hará cada vez más chica con cada coincidencia que logre, recortando así el tiempo de cómputo.

Iterar fasta

contador=0

Mientras no haya coincidencia y contador<len(lista\_limites):

Si coincidencia:

Modificar el registro

Si len(lista\_limites)>1:

lista\_limites.pop(contador)

Si no: contador+=1

Ejercicio 2c. Responda:

¿Cuántos dominios de protoglobina logró recuperar de la minería de datos?

¿Puede afirmar que recuperó todos y que no dejó alguno afuera?

¿O que solo recuperó dominios protoglobina y que no incorporó dominios globina de otro tipo? ¿Por qué?

**Objetivo 3: comparar y analizar alineamientos**

**3a) Alinear los dominios**.

Ahora lo que vamos a hacer es utilizar HMMer para alinear y comparar con el alineamiento que haga MAFFT.

El comando para correr *hmmalign* es del tipo*:*

hmmalign -o alin\_seqs.sto secuencias\_modelo.hmm dominios.fasta

El comando para alinear con MAFFT ya lo conoce!

**Ejercicio 3a)** Parsee los MSAs hechos con hmmalign y MAFFT y busque la posición de la histidina F8 (el residuo más conservado y característico de todas las globinas), y calcule la proporción de gaps en el MSA para esa posición. (sugerencia: busque una columna en el MSA donde la histidina representa más del 80% de los aminoácidos presentes en esa columna).

Compare el porcentaje de histidinas en esa columna entre los dos MSAs obtenidos (por MAFFT y *hmmalign)*, y la proporción de gaps obtenida. ¿En qué posición está la histidina F8 en cada MSA? ¿Está en la misma posición en ambos casos? ¿En cuál MSA hay más presencia de gaps?

Para aquellos que lo deseen, les dejamos una ayudita (usando pandas y Bio.AlignIO):

align = AlignIO.read("alineamiento","formato(fasta/stockholm/phylip..")

align\_pd=pd.DataFrame(align) #convierte el alineamiento en un pandas dataframe

align\_pd[k].value\_counts() #devuelve un listado de cantidad de veces que aparecen los elementos de una columna k

align\_pd[k].value\_counts().index #devuelve lista con los elementos que hay en esa columna k

align\_pd[k].value\_counts()["H"] #devuelve la cantidad de veces que aparece el elemento “H” (por histidina) en la columna k (si es que hay un elemento “H”, si no tira error)

align\_pd.shape #devuelve tupla con las dimensiones del dataframe (filas x columnas)

**Ejercicio 3b Minería de datos.**

PFAM posee en su base de datos 3 dominios globina, el ya utilizado protoglobina, y los denominados “Globin” y “Bac\_globin”.

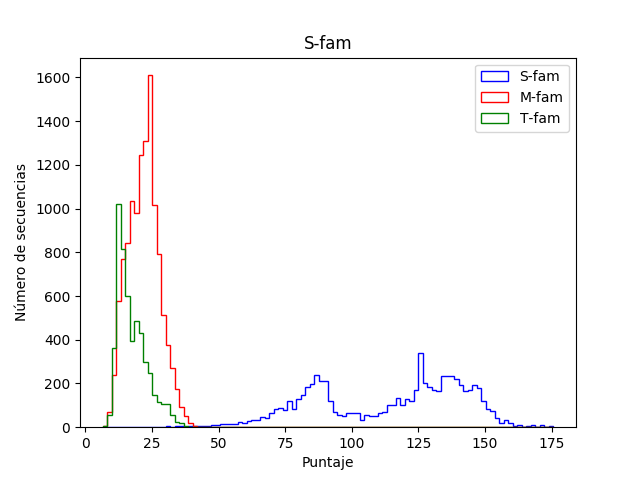
1) Descargue los *seeds* de las éstas dos familias de globinas en Pfam, genere los HMM y corra *hmmsearch* con cada HMM nuevo contra el set de globinas.

2) Genere los histogramas de puntaje de dominio y establezca un umbral de corte para luego extraer los dominios.

3) Compare los 3 sets obtenidos. Hay alguna secuencia presente en más de un set? (use los Uniprot ID de las secuencias para buscar coincidencias entre los sets)

**Ejercicio 3c)** Corra cada HMM contra cada set de dominios obtenidos (si son 3 HMM y 3 sets de dominios recuperados, son un total de 9 corridas de *hmmsearch*).

**Ejercicio 3d)** Usando los resultados de la búsqueda realizada anteriormente genere histogramas de distribución de puntaje para cada HMM (para cada set de secuencias use un color distinto). A modo de ejemplo:



¿Observa algún solapamiento entre las curvas? ¿Cómo lo interpretaría? ¿Fue buena la separación que hizo?