Tarea #2

1.

a) Investigue y conteste. ¿Cuántos megabytes requiere la secuencia del genoma de un ser humano (admitiendo agrupar varias bases en un mismo byte, pero sin comprimir más allá de eso)? Si sólo nos interesan las secuencias que codifican proteínas ("CDS"), ¿cuántos megas necesitaríamos?

b) Tenemos un cierto gen (humano) y queremos escoger un trocito de él (un segmento de N bases contiguas) que sea único dentro del genoma completo. Suponiendo que las bases del genoma fuesen equiprobables e independientes, ¿cuál es el menor valor de N que nos garantizaría una probabilidad menor a 0.01 de encontrar la misma secuencia en otro punto del genoma? Explique su cálculo y detalle cualquier supuesto adicional que haga.

c) Pequeñas secuencias como la descrita en la parte anterior se usan en al menos dos tecnologías importantes, la PCR (reacción en cadena de polimerasa) y los microarrays. Averigüe qué son esas tecnologías y comente sobre la necesidad (o no) de unicidad de la secuencia en cada una.

2.

Programe en su lenguaje favorito. Necesitará (al menos) funciones que hagan lo siguiente:

- Generar una secuencia aleatoria de 200 bases (A,C,G,T) equiprobables e independientes.
- Una función que aplique una mutación a una secuencia; la mutación se escoge entre inserción, borrado y reemplazo de manera equiprobable, y su lugar de aplicación se elige al azar a lo largo de la secuencia. El borrado borra una letra, la inserción inserta una letra (equiprobable), y el reemplazo reemplaza una letra por cualquiera de las otras 3 (de manera equiprobable).
- Una función que calcule la distancia de Levenshtein entre dos secuencias (implementando Needleman-Wunsch).

Con esas funciones, hará lo siguiente:

a) Generar una secuencia, y aplicar M mutaciones; para M entre 0 y 300, grafique la relación entre M y D, donde d es la distancia de Levenshtein entre la secuencia final y la secuencia inicial.

b) Genere una secuencia, clónela, y a cada copia aplíquele M mutaciones (de modo que tendrá dos secuencias crecientemente distintas). Grafique la relación entre M y D', donde D' es la distancia entre las dos secuencias que están mutando.

c) Genere 10.000 pares de secuencias (largo 200 c/u) y evalúe su distancia de Levenshtein; haga un histograma de la distribución de estos valores, y calcule media y σ.

d) Considerando (b) y (c), ¿por sobre qué valor de M diría usted que el parentesco entre las secuencias es indetectable?

3.

a) Siga programando en su lenguaje favorito. Esta vez, haga un programa que reciba una secuencia de DNA y encuentre en ella los 10 palíndromes más largos (un palíndrome es una palabra que se lee igual en orden inverso).

b) Aplíquelo al genoma de *Methanococcus jannaschii*; lo puede encontrar en http://cmr.jcvi.org/cgi-bin/CMR/shared/BatchDownload.cgi?molecule=1811&select_type=whole_org_mol

c) Sugiera una estrategia, basada en Smith-Waterman, para realizar esta tarea si nos interesaran palíndromes *aproximados*, donde se permitan reemplazos o inserciones.

4. Retome las 6 proteínas encontradas en la tarea 1 (los primeros 6 matches que les dio BLAST).

a) Alinee sus secuencias usando Clustal (www.clustal.org). Muestre el alineamiento. ¿Quedan alineados entre sí los segmentos que se alineaban con su nombre?

b) Corte, de cada secuencia, el segmento que se alinea con su nombre. Con las 7 secuencias, haga un alineamiento en Clustal.

c) Usando ese alineamiento, construya y dibuje ("a mano", es decir, sin usar software de HMM) un HMM.

d) Determine la secuencia de estados internos más probable en ese HMM para emitir su nombre.