Estabilidad y deformación del dúplex de DNA. Predicción de promotores

Citlali Gil Aguillon.
Jessica Danielly Medina Sánchez.
Anali Migueles Lozano.

1) Completar el código fuente del programa 1.1 para implementar el predictor de Kanhere y Bansal, justificando los valores de cutoff1 y cutoff2 de acuerdo con las figuras de su artículo. Pueden usar el lenguaje de programación que quieran, siempre que haya un compilador disponible.

Modelo Nearest Neighbor para la predicción de promotores a través de un enfoque estructural. Análisis de secuencias de ORFs de Escherichia coli (84 secuencias) con coordenadas -400,+50.

Para escoger los cutoff1 y cutoff2 se utilizó la referencia del artículo Kanhere & Bansal (2005), utilizando los parámetros con más sensibilidad (mayor detección de verdaderos positivos, al no aceptar cualquier valor).

Sensitivity	Cut-off for D	Cut-off for E1 (kcal/mole)	Frequency of false positives	
			FP (1/nt)[a]	FP (1/nt)[b]
0.13	3.4	-15.99	1/16214	1/261000
0.22	3.4	-16.7	1/11350	1/130500
0.32	3.3	-17.1	1/8407	1/65250
0.40	3.3	-17.55	1/6486	1/29000
0.50	2.76	-17.53	1/3914	1/13737
0.60	2.45	-17.64	1/2467	1/7250
0.70	2.35	-18.07	1/1621	1/2747
0.81	1.9	-18.15	1/1086	1/1878
0.90	0.97	-18.37	1/572	1/967

```
#!/usr/bin/perl –w
# prog1.1
# Bruno Contreras-Moreira
#Citlali Gil Aguillon
#Jessica Danielly Medina Sánchez
#Anali Migueles Lozano
# Nearest Neighbor dG calculator
use strict;
sub complement{ $_[0] =~ tr/ATGC/TACG/; return $_[0] }
# global variables
#variable donde guardamos primera mitad de la ventana para poder ver correccion simetrica
my $seqWin='';
#variable en la que se guarda segunda mitad de ventana traducida para ver correccion simetrica
my $seqWinR='';
#vector en el que se guardan las energias libres de todos los dinucleotidos
my @dGn;
my $T          = 37; # temperature(C)
my $windowL    = 15;  # window length, http://www.biomedcentral.com/1471-2105/6/1
my %NNparams   = (
      # SantaLucia J (1998) PNAS 95(4): 1460-1465.
      # [NaCl] 1M, 37C & pH=7
      # H(enthalpy): kcal/mol , S(entropy): cal/k.mol
      # stacking dinucleotides
      'AA/TT' , {'H',-7.9, 'S',-22.2},
      'AT/TA' , {'H',-7.2, 'S',-20.4},
      'TA/AT' , {'H',-7.2, 'S',-21.3},
```

```
        'CA/GT' , {'H',-8.5, 'S',-22.7},
        'GT/CA' , {'H',-8.4, 'S',-22.4},
        'CT/GA' , {'H',-7.8, 'S',-21.0},
        'GA/CT' , {'H',-8.2, 'S',-22.2},
        'CG/GC' , {'H',-10.6,'S',-27.2},
        'GC/CG' , {'H',-9.8, 'S',-24.4},
        'GG/CC' , {'H',-8.0, 'S',-19.9},
        # initiation costs
        'G'     , {'H', 0.1, 'S',-2.8 },
        'A'     , {'H', 2.3, 'S',4.1  },
        # symmetry correction
        'sym'   , {'H',   0, 'S',-1.4 } );
my $infile = $ARGV[0] || die "# usage: $0 <promoters file>\n";
print "# parameters: Temperature=$T \c Window=$windowL\n\n";

open(SEQ, $infile) || die "# cannot open input $infile : $!\n ";
while(<SEQ>){
        if(/^(b\d{4}) \\ ([ATGC]+)/){
                my ($name,$seq) = ($1,$2);
                printf("sequence %s : ",$name);
                                                        #####BLOQUE PRIMERO#####

                ##MODIFICACION CODIGO: SUBRUTINA ACOPLADA CON CODIGO
                # calculate NN free energy of a DNA duplex , dG(t) = (1000*dH - t*dS) / 1000
                # parameters: 1) DNA sequence string; 2) Celsius temperature
                # returns; 1) free energy scalar
                # uses global hash %NNparams

                #variables DNAstep: contiene dinucleotido,  nt: utilizado para correciones,  dG: se utiliza como
intermediario en energia libre
                my ($DNAstep,$nt,$dG) = ("","",0);
                my @sequence = split(//,uc($seq));
                my $tK = 273.15 + $T;
                #vector en el que contendra el valor que corresponde a cada ventana
                my @dGn=();
                #creacion de VEctOR  dG, donde se guardaran para cada 15 sitios de la secuencia
                # add dG for overlapping dinculeotides

               #loop que recorre secuencia para las posibles ventanas, por tanto lo hace 436 veces
                for (my $n=0; $n<(length($seq)-14); $n++){
                        #loop anidado que recorre las 15 posiciones siguiente formando la ventana
                        for(my $i=0; $i<$windowL-1; $i++){
                           $dG=0;
                           $DNAstep='';
                           ####CAMBIO
                              #ya que la subrutina afecta a la secuencia introducida, debemos usar variables
temporales para no afectar la secuencia original
                           my $temporal=$sequence[$n+$i].$sequence[$n+$i+1];
                           $temporal= complement($temporal);
                           $DNAstep = $sequence[$n+$i].$sequence[$n+$i+1].'/'.$temporal;

                           if(!defined($NNparams{$DNAstep})){
                              $DNAstep = reverse($DNAstep);
                           }

                           $dG = ((1000*$NNparams{$DNAstep}{'H'})-($tK*$NNparams{$DNAstep}{'S'}))/
```

```
1000;
                        #rellenar vector
                        $dGn[$n] += $dG;

                         #realizar bloque solo cuando se encuentre en la ultima vuelta
                        if($i==$windowL-2){

                           # add correction for helix initiation
                           $nt = $sequence[$n]; # first pair
                           if(!defined($NNparams{$nt})){
                              $nt = complement($nt)
                           }
                           $dGn[$n] += ((1000*$NNparams{$nt}{'H'})-($tK*$NNparams{$nt}{'S'}))/ 1000;

                           $nt = $sequence[$n+14]; # last pair
                           if(!defined($NNparams{$nt})){ $
                              nt = complement($nt)
                           }
                           $dGn[$n] += ((1000*$NNparams{$nt}{'H'})-($tK*$NNparams{$nt}{'S'}))/ 1000;

                           #please complete for symmetry correction [ AD ]
                           #checar si la secuencia es palindromica  CAMBIOO
                                                    for(my $a=0; $a<7; $a++){
                              $seqWin .= $sequence[$n+$a];
                              $seqWinR .= $sequence[$n+14-$a];
                           }
                           #sacar complementaria de la segunda parte de la ventana
                           $seqWinR= complement($seqWinR);
                           #si se cumple la igualdad debe sacarse correccion para ella
                           if($seqWin eq $seqWinR){
                              #se agrega a la ventana correspondiente
                              $dGn[$n] += ((1000*$NNparams{'sym'}{'H'})-($tK*$NNparams{'sym'}
{'S'}))/1000;
                           }
                        }
                     }
                  }
               }
                                       #######BLOQUE SEGUNDO#########
                                       # en donde se sacaran E1 E2 D#
#valores de cortes tomados del articulo de referencia
my $cutoff_1= 3.4;
my $cutoff_2= -15.99;
   ##CODIGO IMPLEMENTADO
   #calcular E1, E2, D
   #vector donde guardaremos posicion de ventana
         my @dGSeq;
   #valores para E1
         my $E1=0;
   #valores para E2
         my $E2=0;
   #valores para D
         my $D=0;
#recorrer todas las ventanas para calcular D
my $j=0;
#recorre todas las ventanas, con variable 'i'
for( my $i=0; $i<((scalar(@dGn))-199); $i++){
```

```
            #calcula E1
            for (my $n=0; $n<49; $n++){
                        $E1+=$dGn[$n+$i];
            }#promedio
   $E1/=50;
            #calcula E2
            for (my $n=0; $n<99; $n++){
                        $E2+=$dGn[$n+99+$i];
            }#promedio
   $E2/=100;
            #diferencia de E1 y E2
            $D= $E1- $E2;
            #con el vector vamos a evaluar si respeta los puntos de corte
            if( $cutoff_2 > $E1 && $cutoff_1 > $D){
                        #guarda posicion de valor relevante
      ##de acuerdo con los criterios revisamos que se encuentre entre las posiciones -150 y 50
                        if(($i-350)>-150 && ($i-350)<50){
          #se guarda la posicion del posible promotor y se aumenta el contador 'j'
                                    $dGSeq[$j++]=$i-350;
          #se agregan 24 posiciones, porque los colindantes se toman como la misma señal (son 25, pero se
aumenta otro en el 'for')
                                    $i+=24;
                        }
            }
   #reiniciar las variables para no tener problemas
            $E1=0;
            $E2=0;
            $D=0;
}
#imprimimos vector en el que se guardan los inicios de la secuencias predichas donde los valores son
aceptables
print join(", ", @dGSeq);
print "\n";
}
}close(SEQ);
print "\n";
```

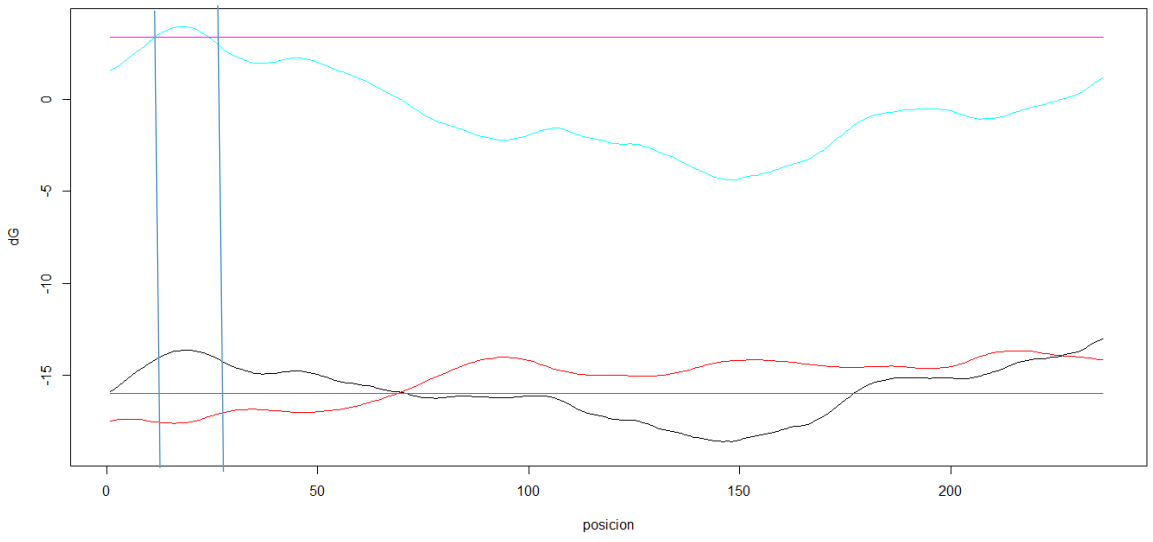
2) Diseñar una figura donde se muestra gráficamente D, E1 y E2 para una posición n.

Script de R para sacar grafica por secuencia, utilizando variación del código entregado anteriormente.

```
#leer archivo con datos de salida del código para predecir secuencias
tabla2 <- read.table(file="/users/BEAST/Desktop/b1182")
#puntos de corte para  D y E1 respectivamente
        cutoff <- rep(3.4, 236)
        cutoff2 <- rep(-15.99, 236)
plot(x=0:236, ylim=c(-19,4), type="n", xlab="posicion", ylab="dG")
#grafica de 'E1' color negro
lines(tabla2$V1, col=1)
#grafica de 'E2' color rojo
lines(tabla2$V2, col=42)
#grafica de 'D' color azul
lines(tabla2$V3, col=5)
#grafica de cutoff para D color rosa
lines(cutoff, col=22)
#grafica de cutoff2 para E1 color rosa
lines(cutoff2, col=22)
```

La gráfica que presenta todos los Deltas Gs para todas las posibles secuencias promotoras. Que son 236. En el eje de las 'y' tenemos los dG y en 'x' las posiciones. Las líneas de color rosas horizontales representan los cortes utilizados para aceptar los valores. **La línea negra representa E1, la línea roja a E2 y la línea azul es D.** Aquellas partes en las que supera los cortes se tomaran en cuenta como posibles promotores por su inestabilidad. Tomamos secuencia B1189 y graficamos

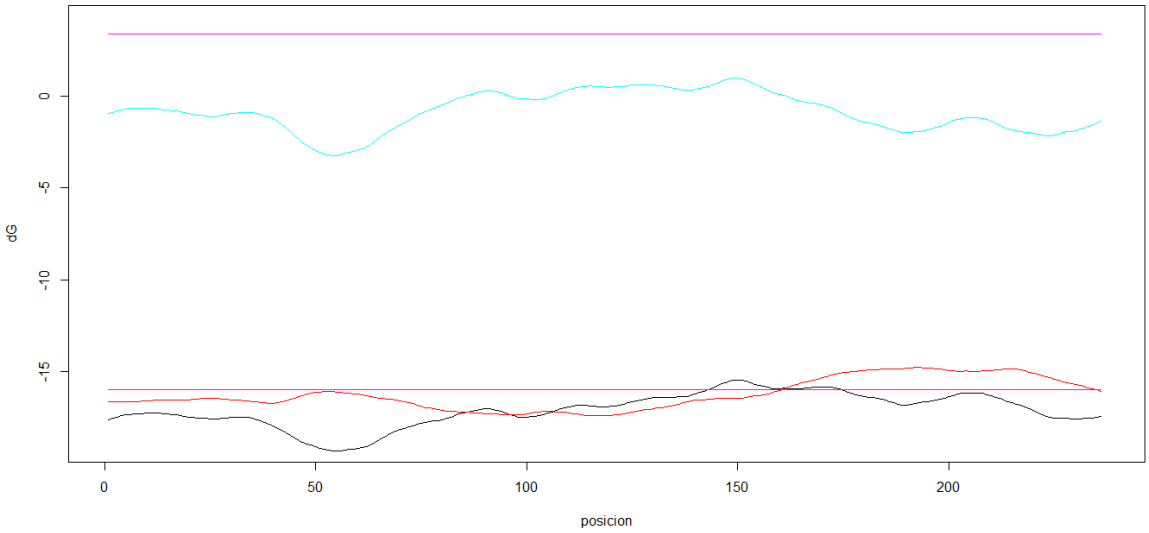
Secuencia B1189:



En la gráfica que se encuentra arriba remarcamos (líneas verticales azules) aquellos valores en los que entre las secuencias 0-50 supera ambos umbrales, indicándonos significancia para predecir promotores, las posiciones donde podría encontrarse dicho promotor aparecen abajo.

Inicio	Final	Secuencia (50nt)
-149	-99	
-124	-174	

Secuencia B1182:



En la gráfica superior podemos observar que en ninguna posición la gráfica 'D' pasa el umbral, lo que nos indica que ninguno de los sitios fue resaltado como posible promotor. En dicha secuencia (B1182) no encontramos ningún posible promotor.

3) Predecir promotores en todas las secuencias del fichero K12_400_50_sites.

Sitios de inicio de presuntos promotores

sequence b0034 :
sequence b0040 : -145
sequence b0041 :
sequence b0063 : -149 -124
sequence b0064 : -149 -124
sequence b0077 :
sequence b0112 : -149 -124
sequence b0113 :
sequence b0116 : -149, -124
sequence b0118 :
sequence b0124 : -149, -124
sequence b0216 : -149, -124
sequence b0241 : -149
sequence b0244 : -149, -124
sequence b0273 : -149, -124
sequence b0313 : -149, -124
sequence b0314 : -149, -124
sequence b0336 : -149, -124
sequence b0338 : -149, -124

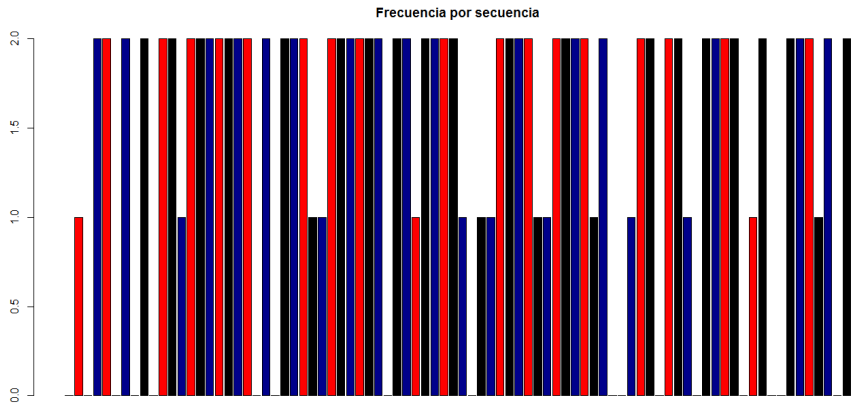
sequence b0339 : -149, -124
sequence b0365 :
sequence b0383 : -149, -124
sequence b0388 :
sequence b0396 : -149, -124
sequence b0399 : -149, -124
sequence b0403 : -149, -124
sequence b0411 : -149
sequence b0432 : -149
sequence b0435 : -149, -124
sequence b0440 : -149, -124
sequence b0450 : -149, -124
sequence b0463 : -149, -124
sequence b0523 : -149, -124
sequence b0536 : -149, -124
sequence b0553 :
sequence b0572 : -147, -122
sequence b0584 : -144, -119
sequence b0585 : -149
sequence b0590 : -149, -124
sequence b0591 : -149, -124
sequence b0592 : -149, -124
sequence b0605 : -149, -124
sequence b0673 : -127
sequence b0678 :
sequence b0679 : -137
sequence b0683 : -149
sequence b0684 : -149, -124
sequence b0698 : -149, -124
sequence b0708 : -149, -124
sequence b0730 : -149, -124
sequence b0754 : -134
sequence b0759 : -149
sequence b0763 : -149, -124
sequence b0774 : -149, -124
sequence b0775 : -149, -124
sequence b0779 : -149, -124
sequence b0781 : -114
sequence b0811 : -149, -124
sequence b0812 :
sequence b0827 :
sequence b0850 : -149
sequence b0887 : -149, -124
sequence b0889 : -149, -124
sequence b0894 :
sequence b0950 : -149, -124
sequence b0954 : -149, -124
sequence b0957 : -137
sequence b0958 :
sequence b0971 : -149, -124
sequence b0972 : -142, -117
sequence b0995 : -149, -124
sequence b0996 : -149, -124
sequence b1015 :
sequence b1020 : -149
sequence b1032 : -149, -124

sequence b1040 :
sequence b1041 :
sequence b1062 : -149, -124
sequence b1101 : -148, -123
sequence b1102 : -149, -124
sequence b1109 : -149
sequence b1130 : -149, -124
sequence b1182 :
sequence b1189 : -149, -124

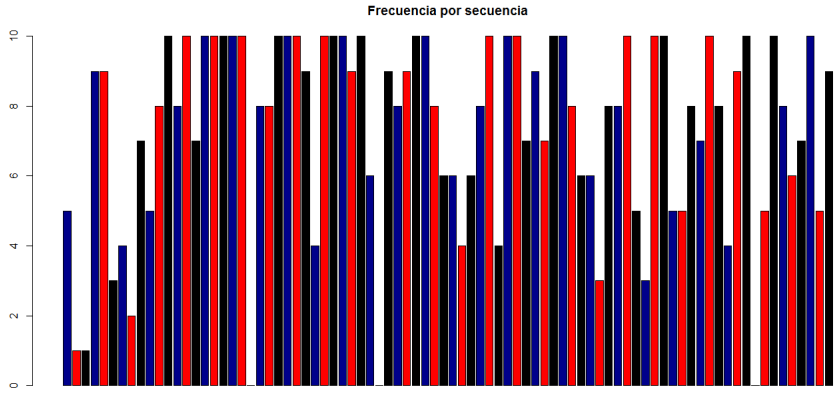
4) Graficar con que frecuencia se predicen promotores en el intervalo -400,50. Con un breve comentario de los resultados es suficiente. Se les ocurre una manera de validar sus resultados, y calcular la tasa de FPs, usando RSAT:: matrix-scan?

Utilizando los datos obtenidos en el punto anterior

Frecuencia de promotores para las 84 secuencias con delimitación de -150 a 50. Máximos encontrados por secuencia son 2 promotores



Frecuencia de promotores para las 84 secuencias sin delimitación. Como punto máximo de posibles promotores por secuencia son 10.



Los resultados observados en las gráficas nos reflejan el tipo de restricciones que hemos impuesto, ya que como está en la referencia Kanhere & Bansal (2005), debemos tomar valores dentro del rango de -150 a 50, además de que si dos resultados obtenidos por nuestro modelo están más cercano de

25 bases deben considerarse como una misma señal, y también el haber puesto valores de cortes tan altos, por lo que nuestros resultados se ven sesgados por los criterios definidos para promotores, lo que explicaría el porqué de que en algunas de las secuencias no se haya detectado promotores.

Matrix scan, el programa ejecuta automáticamente una serie de análisis similares para estimar el intercambio entre la sensibilidad y la tasa de falsos positivos, mediante la combinación de las distribuciones teóricas y empíricas de las puntuaciones calculadas en varias condiciones: el escaneo de un conjunto de secuencia positiva (por ejemplo, sitios anotado); el escaneo de un conjunto negativo (por ejemplo, secuencias aleatorias o secuencias biológicas que contienen ningún sitio) ; el escaneo de los mismos conjuntos de datos con matrices permutados . Yo usaría la lógica de matrix scan, permutar las posiciones de las secuencias de manera aleatoria, para poder tener secuencias al azar con el mismo contenido de GC, y observar el comportamiento con graficas como las del punto ‘2’, donde esperaríamos un comportamiento aleatorio de promotores.