

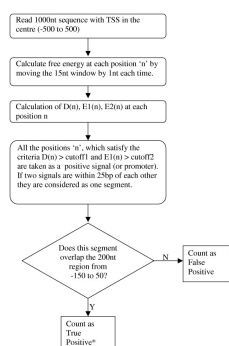
# Predicción de promotores mediante el algoritmo Nearest Neighbor $\Delta G$ calculator

*Ciro Ramírez Suástegui, José Damián Martínez Reyes*

*05/03/2016*

El modelo Nearest Neighbor (NN) aproxima la estabilidad molecular de un ácido nucleico en disolución a partir de su secuencia, que se analiza como una secuencia de dinucleótidos solapantes de interacciones aditivas, cuyas energías se han determinado experimentalmente (SantaLucia, 1998; Breslauer et al., 1986).<sup>1</sup>

El siguiente diagrama muestra el flujo del programa, solo que tendrá algunas modificaciones:



Algoritmo de Kanhere & Bansal (2005), que se basa en calcular la diferencia de  $\Delta G$  entre dos ventanas en torno a una posición  $n$ .

## El código

El archivo del programa estará [aquí](#).

**Agregamos las librerías y las opciones del programa:**

```
use strict; # To restrict unsafe constructs
use Getopt::Long; # Let us take data from the line
my %opts = (); # Hash declaration
# Add keys with their values
GetOptions (\%opts, 'h'); # be sure the field exist
if(($opts{'h'})){
    &PrintHelp();
}
```

Para llamar al programa `program_name [-h] file`  
### Ahora definimos las variables y parámetros globales:

```
# global variables
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/kmol
# 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 %seqs; # Sequences
my @names; #Sequence name
my $g_deltaG;

```

**Leemos el archivo de la secuencia:**

```

my $infile = $ARGV[0] || die "# usage: $0 <promoters file>\n";

print "# parameters: Temperature=$T gC Window=$windowL\n\n";

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

        # Call the subroutine and save the parameters needed
        $g_deltaG = duplex_deltaG($seq, $T);
        $seqs{$name}{'seq'} = $seq; # Save string in a global hash %seqs
        $seqs{$name}{'dG'} = $g_deltaG;
        printf("%s \n", $seqs{$name});
    }
}
close(SEQ);

```

**Subrutinas:**

```

# add dG for overlapping dinucleotides
for($n=0;$n<#$sequence;$n++)
{
    $DNASTep = $sequence[$n].$sequence[$n+1].'/'.
        complement($sequence[$n].$sequence[$n+1]);

    if(!defined($NNparams{$DNASTep})) # Ask if there's the string
    {

```

```

        $DNAstep = reverse($DNAstep); # If no, it reverses
    }

    $dG = ((1000*$NNparams{$DNAstep}{'H'})-
            ($tK*$NNparams{$DNAstep}{'S'}))
            / 1000 ; # Calculating the dG value

    $total_dG += $dG;
}

```

## Resultados