DNA SECONDARY STRUCTURE PREDICTION WITH WORD COMBINATIONS

DANIELLE DEES

1. Introduction

Zuker's original work on RNA secondary structures provides an algorithm to predict the secondary structure of a single RNA strand which will output a negative number if this strand folds into a secondary structure and a positive number if this strand remains unfolded. This algorithm is quite useful to learn about a single sequence of RNA neucleotides; however, many of strands that are currently used in DNA computing are made up of smaller strands that are called words. The science of choosing these words has become quite complex, with many rules governing the order of neucleotides in order to obtain words which can be combined into a strand that will not fold onto itself. The algorithm that we present here will take in $2n$ words, where a word is defined as a sequence of RNA neucleotides of length l. These words will be labeled as $w(1), \overline{w(1)}, w(2), \overline{w(2)}, \dots w(n), \overline{w(n)}$. The resultant strand will the $\sum_{i=1}^{n} w(i)$ or $\overline{w(i)}$. Our algorithm outputs a negative number if one of the 2^n strands that result from the concatenation of these words will fold into a secondary structure or a positive number if each possible strand will not form a secondary structure. Although the same result could be obtained by running each of the 2^n structures through Dr. Zuker's algorithm, which has a runtime of $O(n^4)$, this new algorithm will solve this problem in polynomial time according to n.

2. Basic Algorithm

A secondary structure formed from a strand $s \in S$ where S is the set of all strands that can be created, will have $1 \leq i \leq |s|$ where S_i is the base at position#(i) in word $w(word\#(i))$ or $\overline{w(word\#(i))}$. The position#(x) and word#(x) functions determine the word number and position within the word based on the word length and parameter x. It is assumed that no psuedo-knots exist in any of the secondary structures and that each base can only bond to one other base. Each psuedo-knot free structure can be viewed as a collection of stacked pairs and loops with exterior base pairs closing each loop. The types of loops are hairpin, internal, and multibranched, with a bulge loop being a particular instance of an internal loop. RNA secondary structure prediction is really just determining the most stable comfiguration of the given neucleotides. The optimal structure will have the lowest free energy. Therefore, we can use recurance relations to determine the free energy over all 2^n structures by taking advantage of dynamic programming and modifying Zuker's recurance relations for single strand folding. The modifications that we have made to the original algorithm is that we have added additional parameters into all the equations that specify whether S_i is in $w(word\#(i))$ or $\overline{w(word\#(i))}$ based on the parameter $b_i \in \{T, B, E\}$ in which case T would represent $w(word\#(i))$, B would represent $\overline{w(word\#(i))}$, and E is the case where it does not matter to which word S_i belongs. The word specification is only T or B if the recurrance calls functions with a parameter that would be in the same word as a second parameter. Specifing the word makes sure that the optimal stucture is a concatenation of whole words and not parts of words.

- The energy of the optimal structure from S_1 to S_j where j is the length of $\forall s \in S$ is:

$$W'_S(j) = W'_S(E,j) \tag{1}$$

$$W'_S(b_j,j) = \min \begin{cases} W'_S(x,j-1) & \begin{array}{r} x = b_j \text{ if } j-1 \bmod l \neq 0 \\ x = E \text{ if } j-1 \text{mod} l = 0 \end{array} \\ \min\limits_{\substack{1\leq i\leq j-1 \\ i-1 \bmod l\neq 0 \\ b_i\in\{B,T\}}} V'_S(b_i,b_j,i,j) + W'_S(b_i,i-1) \\ \min\limits_{\substack{1\leq i\leq j-1 \\ i-1 \bmod l=0}} V'_S(E,b_j,i,j) + W'_S(E,i-l) \end{cases} \tag{2}$$

- The energy of a loop which is closed with the base pair S_i, S_j is:

$$V'_S(b_i,b_j,i,j) = \min \begin{cases} eH'_S(b_i,b_j,i,j) \\ eS'_S(b_i,b_j,i,j) + V'_S(x,y,i+1,j-1) \\ VBI'_S(b_i,b_j,i,j) \\ VM'_S(b_i,b_j,i,j \end{cases} \tag{3}$$

where x which corresponds to b_{i+1} and y which corresponds to $b_{j-1}$are determined by the following psudocode:

```
     x = E;
     y = E;
     if(word#(i) == word#(i+1)) then
x = $b_i$;
    if(word#(j) == word#(j-1)) then
y = $b_j$;
  if(word#(i+1) == word#(j-1)) then
if(x == E and y == E) then
x = b_1;
y = b_1;
else if(x == E) then
x = y;
else
y = x;
```

- The energy of an internal loop which is closed by the pair S_i, S_j is:

$$VBI'_S(b_i,b_j,i,j) = \min \begin{cases} +\infty \text{ for} j < i+4 \\ \min\limits_{i<i'<j'<j} eL(b_i,b_j,x,y,i,j,i',j') + V'_S(x,y,i',j') \end{cases} \tag{4}$$

where x which corresponds to $b_{i'}$ and y which corresponds to $b_{j'}$ are determined by the following psudocode:

```
     x = E;
     y = E;
     if(word#(i) == word#(i')) then
x = $b_i$;
    if(word#(j) == word#(j')) then
y = $b_j$;
  if(word#(i') == word#(j')) then
if(x == E and y == E) then
x = b_1;
y = b_1;
else if(x == E) then
x = y;
else
y = x;
```

- The energy of a multibranched loop closed by the base pair S_i, S_j is:

$$VM'_S(b_i, b_j, i, j) = \min_{i+1<h\leq j-1} WM(w, y, i+1, h-1) + WM(x, z, h, j-1) + a \tag{5}$$

$$WM_S(b_i, b_j, i, j) = \min \begin{cases} V'_S(b_i, b_j, i, j) + b \\ \min\limits_{i<h\leq j} WM'_S(b_i, y, i, h-1) + WM'_S(x, b_j, h, j) \end{cases} \tag{6}$$

where w which corresponds to b_{i+1},x which corresponds to b_h, y which correponds to b_{h-1}, and z which corresponds to $b_{j-1}$are determined by the following psudocode:

```
    w = E;
    x = E;
y = E;
z = E;
    if(word#(i+1) == word#(i)) then
w = $b_i$;
else if(word#(i+1) == word#(h-1)) then
w = $b_1$;
   if(word#(h-1) == word#(i+1)) then
x = w;
else if(word#(h-1) == word#(h)) then
x = $b_2$;
if(word#(h) == word#(h-1)) then
y = x;
else if(word#(h) == word#(j-1)) then
y = $b_3$;
if(word#(j-1) == word#(h)) then
z = y;
if(word#(j-1) == word#(j)) then
if(z == $b_{num}$) then
$b_{num}$ = $b_j$;
else
z = $b_j$;
```

This algorithm is dependent on eH'_S, eS'_S, and eL'_S which are the free energy equations for a hairpin loop, a stacked pair, and an internal loop respectively. The details of these calculations are discussed in greater detail later in this paper.