Indrajeet Javeri, 811767637, iyj83163@uga.edu

CSCI 4470/6470 Project

Fall 2019

Structure Prediction

The objective is to compute the maximum number of permissible letter pairs from the input sequence of 4 letters (A, C, G, U). For these letters, legal pairs are A-U, U-A, G-C, C-G while others (such a G-A) are illegal. In addition, a pairing between two letters is exclusive, e.g., if a letter U on the sequence that is paired with a letter A, that letter U cannot be paired with another letter A.

Solution:

(1) Objective Function

Let B(i,j) be the objective function that represents the maximum number of nueclotide base pairs for a given RNA sequence starting from index i and ending at j. Thus, B(i,j) can be defined recursively as follows,

$$B(i,j) = max \ \{ \quad B(i+1,j-1) + 1 \quad \text{(if } i \ \& \ j \text{ form a valid base pair)}$$

$$B(i+1,j-1) \quad \text{(if } i \ \& \ j \text{ do not form a valid base pair)}$$

$$max \ \{ \ B(i,k) \ + \ B(k+1,j) \ \}$$

$$i \leq k < j$$

$$\}$$

Base cases :

$B(i,j) = 0$ if $i = j$

$B(i,j) = 0$ if $j = (i - 1)$

Ultimately, $B(0,n-1)$ would give us the maximum base pair count for the entire RNA sequence assuming the index starts with 0 and n is the total length of the given RNA sequence.

Indrajeet Javeri, 811767637, iyj83163@uga.edu

(2)(a) Pseudocode for computing the objective function

{Assume M and T are two matrices initialized to all zeroes}

MaxBasePair(s, i, j)

p,q,r,m = 0;

if $M[i][j]$!= 0 then return $M[i][j]$

if $i = j \; or \; j = (i-1)$ then return 0

if (i & j *in* s *is a valid pair*) then

    p = MaxBasePair($s, i+1, j-1$) + 1

else

    q = MaxBasePair($s, i+1, j-1$)

for $k = i$ to $j-1$ do

    m = MaxBasePair(s, i, k) + MaxBasePair($s, k+1, j$)

    if $m \geq r$ then

        $r = m$;

        $kVal = k$

$M[i][j] = max(p, q, r)$

if ($M[i][j]$ = p AND p $\neq$ 0) then

    T[i][j] = -1

else if ($M[i][j]$ = q AND q $\neq$ 0) then

    T[i][j] = -2

else

    T[i][j] = kVal

return $M[i][j]$

Indrajeet Javeri, 811767637, iyj83163@uga.edu

(2)(b) Pseudocode for traceback

{Assume t is a list of length n initialized to all "."}

PrintPairs(T, i, j)

if $i = j$ *or* $j = (i-1)$ then return

if T[i][j] = -1 then

    set(t, i , "{" , j , "}")

    PrintPairs($T, i+1, j-1$)

else if T[i][j] = -2 then

    PrintPairs($T, i+1, j-1$)

else

    PrintPairs(T, i, T[i][j])

    PrintPairs(T, T[i][j] + 1, j)

Assume ***set***$(t, i, s1, j, s2)$ *is a function that takes in a list* t *along with two index values* i & j *and two strings* $s1$ *and* $s2$ *such that it* ***assigns*** $s1$ ***at index*** i *and* $s2$ ***at index*** j *in list* t

For traceback we need to call PrintPairs(T,0,$n-1$) and after the recursion ends we have our parenthesizations in the order of the input sequence in list t with denotational symbols "{" , "}" and "."