Assigment2 - RNA & Assembly

Weize Xu

May 25, 2018

1 Can you give the pseudocode for the Nussinov folding algorithm? Also, please describe the algorithm to recover the secondary structure.

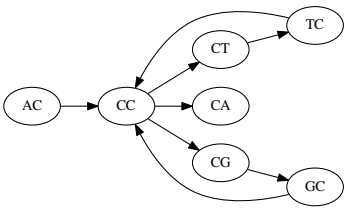
Answer:

Nussinov folding algorithm is a kind of Dynamic Programming algorithm, which goal is to maximize the folding pairs. Pseudocode see Algorithm 1.

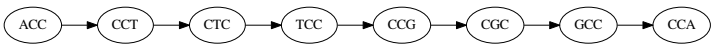
2 Consider the set of reads {ACCTCC, TCCGCC, CCGCCA}. For k=2 or 3, can you build a de Brujin graph H_k ? Can you get the Eulerian path from H_k ? Is the Eulerian path unique?

Answer:

I wrote a little script for building and drawing the de Brujin graph. see https://github.com/Nanguage/Course-Algorithms-in-Bioinformatics/blob/master/L04/debrujin.py. Figure 1 show the de Burjin graph when $k = 2$ and $k = 3$.



(a) $k = 2$



(b) $k = 3$

Figure 1: de Burjin graph H_k when $k = 2$ and $k = 3$

Obvirously, when $k = 3$ there are just one unique euler path. But when $k = 2$, there are two euler paths:

$$AC \to CC \to CT \to TC \to CC \to CG \to GC \to CC \to CA$$
$$AC \to CC \to CG \to GC \to CC \to CT \to TC \to CC \to CA$$

3 Suppose T = {ACT, CAC, CTG, CTT, TCA, TTC} and R = CTGCACT. Can you compute the minimum edit distance between R and a T-string? Please illustrate the steps.

Answer:

I implemented the dynamic programming algorithm for solving the Spectral alignment problem(SAP). Source code see
https://github.com/Nanguage/Course-Algorithms-in-Bioinformatics/blob/master/L04/sap.py
According to my program's computation, the minimum edit distance is 1.

```
   input  : An RNA sequence S[1..n]
   output: A set of pairs P
 1 V ← Matrix(n, n)
 2 InitializeMatrix(V, n)
 3 for m ← 1 to (n − 1) do
 4 |   for i ← 1 to (n − m) do
 5 |   |   j ← i + m
 6 |   |   v_0 ← V[i, j] + δ(S[i], S[j])
 7 |   |   v_1 ← 0
 8 |   |   for k ← i to j do
 9 |   |   |   v ← V[i, k] + V[k + 1, j]
10 |   |   |   if v > v_1 then
11 |   |   |   |   v_1 ← v
12 |   |   |   end
13 |   |   end
14 |   |   V[i, j] ← Max(v_0, v_1)
15 |   end
16 end
17 P ← {}
18 Traceback(P, V, 1, n)
19 return P
20 procedure InitializeMatrix(Mat, n)
21 |   for i ← 1 to n do
22 |   |   Mat[i, i] ← 0
23 |   |   if i ≠ n then
24 |   |   |   Mat[i + 1, i] ← 0
25 |   |   end
26 |   end
27 end
28 procedure Traceback(Pairs, V, i, j)
29 |   if j ≤ i then
30 |   |   return
31 |   else if V[i, j] = V[i, j − 1] then
32 |   |   Traceback(Pairs, V, i, j − 1)
33 |   |   return
34 |   else
35 |   |   for k ← i to j do
36 |   |   |   if δ(S[k], S[j]) ≠ 0 then
37 |   |   |   |   Pairs ∪ (k, j)
38 |   |   |   |   Traceback(Pairs, V, i, k − 1)
39 |   |   |   |   Traceback(Pairs, V, k + 1, j − 1)
40 |   |   |   |   return
41 |   |   |   end
42 |   |   end
43 |   end
44 end
```

Algorithm 1: Nussinov folding