Computational Biology

Lecture 20

Saad Mneimneh

RNA secondary structure

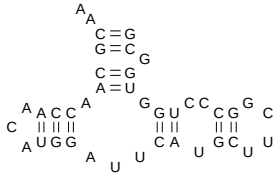
- Recall, unlike DNA, RNA molecule is single stranded chain of nucleotides A, C, G, U.
- A nucleotide in one part of the molecule can base-pair with a complementary nucleotide in another part.
- Therefore, RNA folds. The RNA sequence completely determines the folding (in 3D).
- We would like to predict the *secondary structure* of the RNA: Which bases pair with which?

Saad Mneimneh

Representation

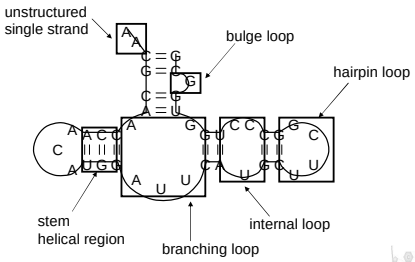
- RNA secondary structure is typically represented by a two-dimensional picture (although folding is 3D).
- Example:

  r = AACGGAACCAACAUGGAUUCACGCUUCGGCCCUGGUCGCG



Saad Mneimneh

Elements of the structure



Saad Mneimneh

Pairs

The secondary structure for an RNA $r = r_1 \ldots r_n$ can be described as a set S of ***disjoint*** pairs (r_i, r_j), where $1 \leq i < j \leq n$

Saad Mneimneh

Knots

- We exclude a configuration called knot.
- A knot exists when r_i is paired with r_j and r_k is paired with r_l such that:

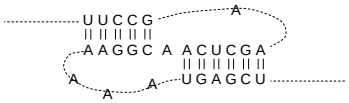
$$i < k < j < l$$

  i.e. overlapping pairs.

- We consider only nested pairs because knots are not frequent.

Saad Mneimneh

Example knot



Knots are not frequent

Saad Mneimneh

Energy

- RNA folds into the minimum free-energy structure.
- Each base pair (r_i, r_j) where r_i and r_j are complementary contributes a negative energy $\alpha(r_i, r_j) < 0$, and $\alpha(r_i, r_j) = 0$ otherwise.
- We must find the minimum free-energy structure.

Saad Mneimneh

Formulation

- Let $E(S)$ be the total free-energy for a set of pairs S:

$$E(S) = \Sigma_{(ri,rj)\in S}\, \alpha(r_i, r_j)$$

- Assume $\alpha(r_i, r_j)$ is independent of all other pairs
- Then we can use solutions for smaller strings to determine the solutions for larger strings.
- Let $S_{i,j}$ be the minimum free-energy structure for $r_i \ldots r_j$.

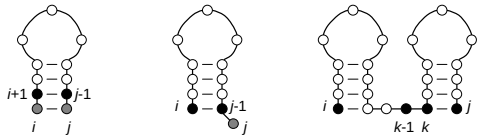
Saad Mneimneh

Dynamic programming

- $S_{i,j}$ is the minimum free-energy structure for $r_i \ldots r_j$.
- What can happen to r_j?
  - r_j unpaired: $E(S_{i,j}) = E(S_{i,j-1})$
  - r_j paired with r_i: $E(S_{i,j}) = \alpha(r_i, r_j) + E(S_{i+1,j-1})$
  - r_j paired with r_k ($k \neq i$): $E(S_{i,j}) = E(S_{i,k-1}) + E(S_{k,j})$ for some $i < k < j$ (we can do this because we assumed no knots)

Saad Mneimneh

Illustrations



(*i*, *j*) pair *j* unpaired branching

Saad Mneimneh

Dynamic programming (Nussinov algorithm)

$$E(S_{i,j}) = \min \begin{cases} E(S_{i+1,j-1}) + \alpha(r_i, r_j) \\ E(S_{i,k-1}) + E(S_{k,j}) \quad i < k \leq j \end{cases}$$

$k = j$ corresponds to the case of unmatched j ($S_{j,j} = 0$)

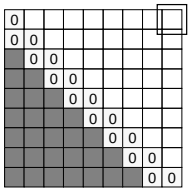
$E(S_{1,n})$ is the lowest free-energy

The actual structure must be obtained by tracing back

Saad Mneimneh

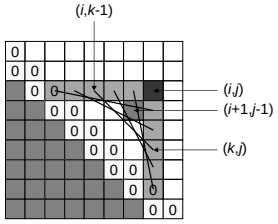
Initialization

- $E(S_{i,i}) = 0$, $i = 1 \ldots n$
- $E(S_{i,i-1}) = 0$, $i = 2 \ldots n$



Saad Mneimneh

Computation



running time: $O(n^2)$ entries
each requires $O(n)$ time => $O(n^3)$

Saad Mneimneh

Example

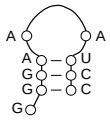
- r = GGGAAAUCC
- $\alpha(C,G) = \alpha(G,C) = -1$
- $\alpha(A,U) = \alpha(U,A) = -1$
- Otherwise $\alpha(x, y) = 0$

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	-1	-2	-3
G	0	0	0	0	0	0	-1	-2	-3
G		0	0	0	0	0	-1	-2	-2
A			0	0	0	0	-1	-1	-1
A				0	0	0	-1	-1	-1
A					0	0	-1	-1	-1
U						0	0	0	0
C							0	0	0
C								0	0

Saad Mneimneh

Tracing back

	G	G	G	A	A	A	U	C	C
G	0	0	0	0	0	0	-1	-2	-3
G	0	0	0	0	0	0	-1	-2	-3
G		0	0	0	0	0	-1	-2	-2
A			0	0	0	0	-1	-1	-1
A				0	0	0	-1	-1	-1
A					0	0	-1	-1	-1
U						0	0	0	0
C							0	0	0
C								0	0



The trace back takes linear time $O(n)$ if we keep back pointers.
The trace above is unbranched. In general we need to keep track of multiple traces.

Saad Mneimneh

Tracing back (no pointers)

trace (1,n)

trace (i,j)
if $i < j$
then **for** $k = i+1$ to j
do **If** $E(S_{i,k-1}) + E(S_{k,j}) = E(S_{i,j})$
then trace (i,k-1)
trace (k,j)
return

// $E(S_{i+1,j-1}) + \alpha(r_i, r_j) = E(S_{i,j})$
// and $\alpha(r_i, r_j) < 0$
pair (r_i, r_j)
trace (i+1, j-1)
return

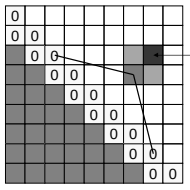
The order is important
If $\alpha(r_i, r_j) = 0$ then we might have many options that result in the same $E(S_{i,j})$, in this case we should not favor a pair

Same preference should be made when keeping pointers

Saad Mneimneh

Illustration

What if $\alpha(r_i, r_j) = 0$ and $E(S_{i+1,j-1}) + \alpha(r_i, r_j)$ is the best option?



If $\alpha(r_i, r_j) = 0$,

then $E(S_{i,j-1})$ is at least as good

Saad Mneimneh