Part II: On structures

10 RNA secondary structure

10.1 Overview

- Introduction to RNAs, types, function
- Structure of RNA, definition and visualization
- Secondary structure
- Another dynamic programming algorithm: Nussinov algorithm
- Prediction using covariation

10.2 RNA: Introduction

RNA, *DNA* and *proteins* are the basic molecules of life on Earth.

Recall that:

- DNA is used to store and replicate genetic information.
- Proteins are the basic building blocks and active players in the cell.
- RNA plays a number of different important roles in the production of proteins from instructions encoded in the DNA.

In eukaryotes, DNA is transcribed into pre-mRNA, from which introns are spliced to produce mature mRNA, which is then translated by ribosomes to produce proteins with the help of tRNAs. A substantial amount of a ribosome consists of RNA.

The *RNA-world* hypothesis suggests that life was originally based on RNA (Gilbert, 1986) Then, over time the "data storage problem" was delegated to DNA and the problem of providing structure and catalytic functionality was delegated to proteins.

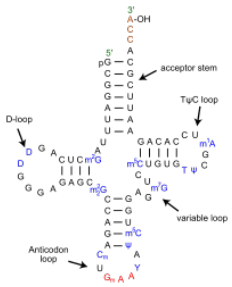
There are a number of different types of RNA molecules. We distinguish: RNA genes and regulatory RNAs.

And within the class of RNA genes we distinguish *protein-coding* and *non-coding* RNAs.

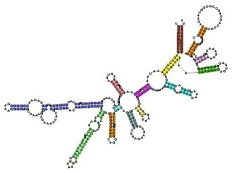
Proteins are built from the messenger RNA (mRNA), which carries the DNA to the ribosomes.

Among the many different non-coding RNAs the most prominent are:

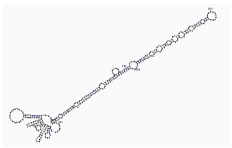
- transfer RNA (tRNA)



- ribosomal RNA (rRNA)



- micro RNA (miRNA)



- small nucleolar RNA (snoRNA)

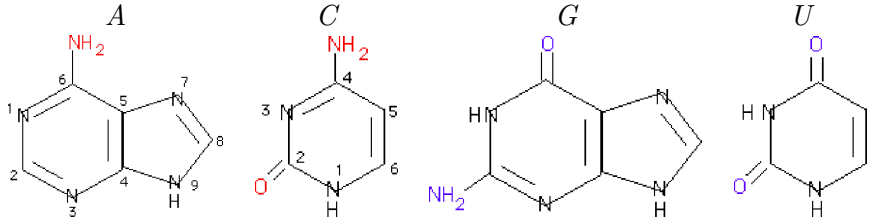
- small interfering RNA (siRNA)

They play many different roles:

- according to central dogma: transmitter of genetic information (mRNA)
- adaptor molecule (tRNA)
- make up ribosome (rRNA)
- carrier of genetic information (RNA virus)
- regulator of gene expression (siRNA)
- catalyst (ribozyme)
- many more

10.2.1 The structure of RNA

An RNA molecule is a polymer composed of four types of (ribo)nucleotides, each specified by one of four bases:



(Image source: Zuker)

RNA is single-stranded, in contrast to DNA, which is double-stranded.

However, complementary bases $\mathtt{C}-\mathtt{G}$ and $\mathtt{A}-\mathtt{U}$ form stable *base pairs* with each other using hydrogen bonds. These are called *Watson-Crick* pairs or *canonical base pairs*. Additionally, one sometimes considers the weaker $\mathtt{G}-\mathtt{U}$ *wobble pairs*.

10.3 RNA secondary structure

Some simple definitions:

Definition 10.3.1 (RNA molecule) *For our purposes, an* RNA molecule *is a sequence* $A = a_1 a_2 \dots a_L$ *of nucleotides* $a_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$.

Definition 10.3.2 (Primary structure of RNA) *The sequence* $A = a_1 a_2 \dots a_L$ *of ribonucleotides is called the* primary structure *of an RNA molecule.*

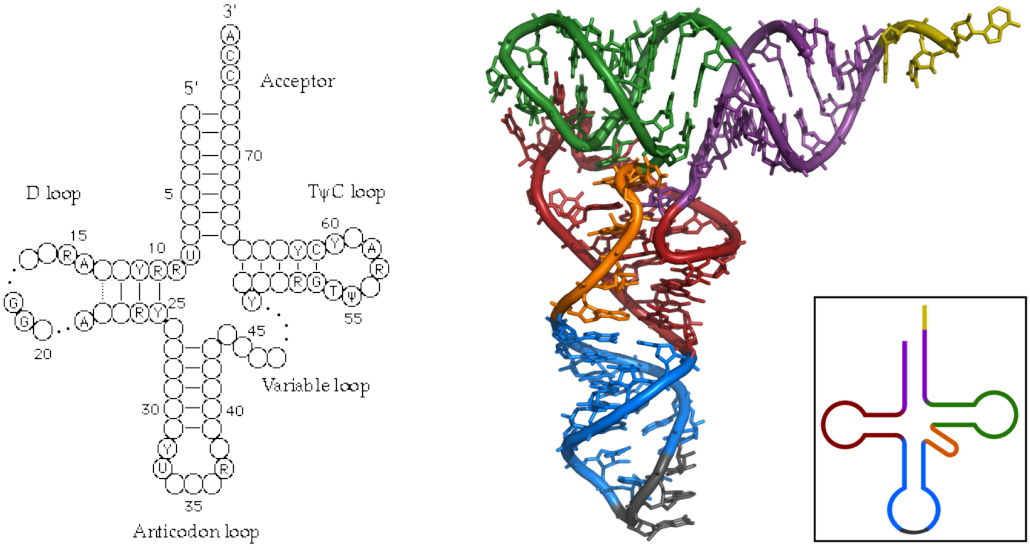
Definition 10.3.3 (Secondary structure of RNA) *A* secondary structure *of* A *is a set* P *of ordered* base pairs (i, j), *with* $1 \leq i < j \leq L$, *satisfying:*

- *Any two different base pairs are disjoint, that is:* $(i, j) \neq (i', j')$ *implies* $\{i, j\} \cap \{i', j'\} = \emptyset$.

The *true secondary structure* of a real RNA molecule is the set of base pairs that occur in its actual three-dimensional structure *in vivo*.

Example of the primary, secondary and tertiary structure of a tRNA:

GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAUCUGGAGGUCCUGUG
UUCGAUCCACAGAAUUCGCACCA



(Source: Wikipedia)

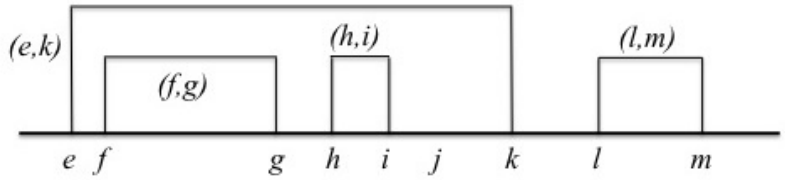
10.3.1 Nested structures

We will only consider so-called *nested* secondary structures.

Definition 10.3.4 (Nested secondary structure) *A secondary structure is called* nested*, if for any two base pairs* (i, j) *and* (i', j')*, w.l.o.g.* $i < i'$*, we have either*

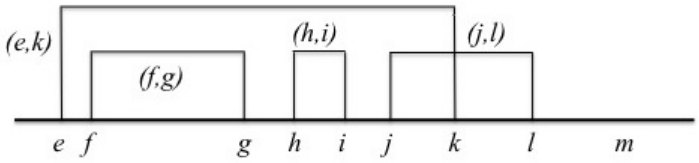
1. $i < j < i' < j'$*, i.e.* (i, j) *precedes* (i', j')*, or*
2. $i < i' < j' < j$*, i.e.* (i, j) *includes* (i', j')*.*

Example: in the following figure



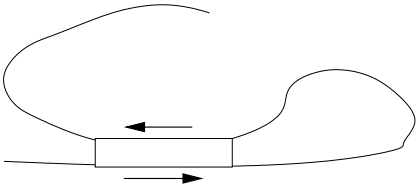
(e, k) includes (f, g) and (h, i), and these all precede (l, m).

In the following figure two base pairs are not nested:

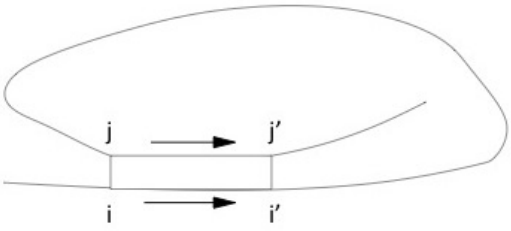


These are the interactions (e, k) and (j, l).

An important example of a non-nested secondary structure is a *pseudo knot*, a secondary structure in which segments of sequence are bonded in the "same direction":

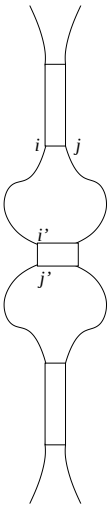


unknoted

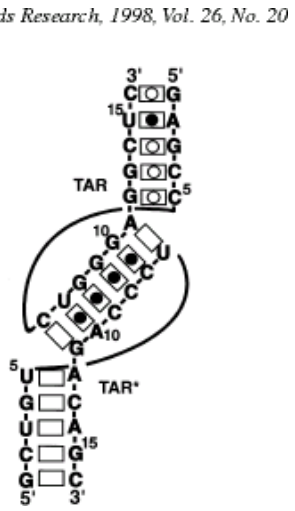


pseudo knot:
(i, j) does not precede nor include (i', j')

The nested requirement excludes other types of configurations, as well, such as *kissing hairpins*, for example:



⇒ biological example



In general, non-nested structures are more complicated and despite being biologically relevant they are not tractable with the methods that we will discuss in this chpater.

10.3.2 Dot-bracket notation

For computer tractability, secondary structures can be stored compactly in strings consisting of dots and matching brackets:

For each position of the sequence A of length L we compute a sequence $RNA(A) = (x_1, \ldots, x_L), x_i \in \{.,(,)\}$. For any pair between positions i and j ($i < j$) we place an open bracket "(" at position i and a closed bracket ")" at j, while unpaired positions in the molecule are represented by a dot (".").

Since base pairs may not cross, the representation is unambiguous.

Example: for the sequence `ACACGACUGAAGUGA` a secondary structure has been computed. Its dot-bracket notation looks as follows:

```
ACACGACUGAAGUGA
.(((.......))).
```

10.3.3 Visualization of a secondary structure

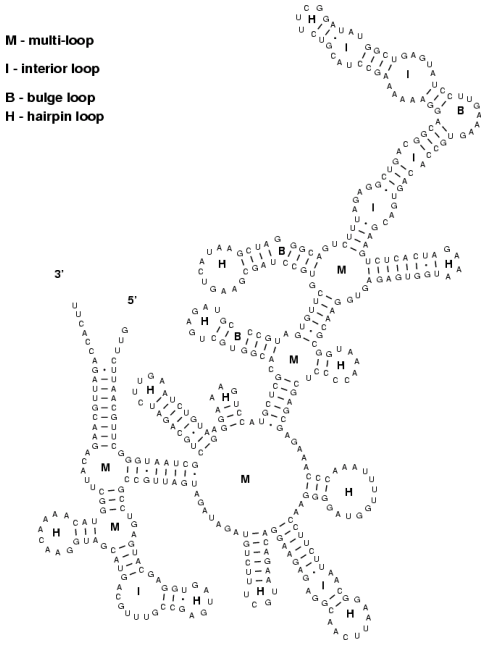
Secondary structures of primary RNA-sequences can be represented by "secondary structure graphs".

In this representation one creates a graph whose each nucleotide of the sequence is a node.

There are two kinds of edges: one representing the adjacency of nucleotides along the RNA sequence, the other representing base pairings.

If there are no pseudoknots, the graph is planar.

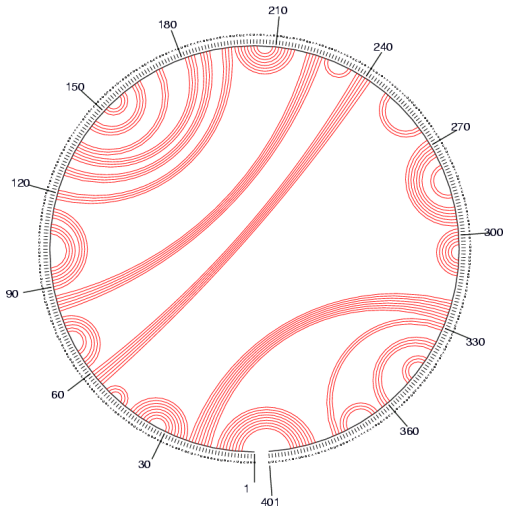
Here is an example of the secondary structure graph of the predicted structure for *Bacillus subtilis* RNAase P RNA:



(source: Zuker)

When there are no pseudoknots, we can also draw the secondary structure by placing the backbone on a circle and drawing a chord for every base pair such that no two chords intersect.

Here is the structure of the *Bacillus subtilis* RNAase P RNA drawn like that:



(source: Zuker)

10.3.4 Secondary structure elements

The following figures show the different types of single- and double-stranded regions in RNA secondary structures:

- single-stranded RNA,
- double-stranded RNA in stacked base pairs,
- stem and loop or hairpin loop,
- bulge loop,
- interior loop, and
- junction or multi-loop.

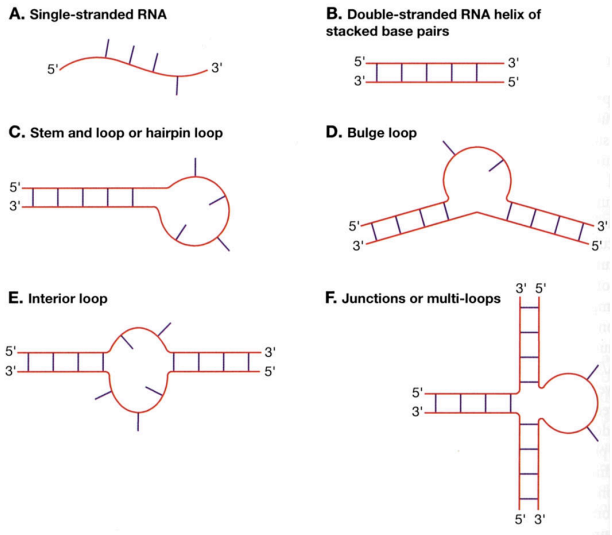


Figure from Mount

10.4 Prediction of RNA secondary structure

RNA secondary structure prediction: For a given RNA primary sequence, the *RNA secondary structure prediction problem* is to predict the true secondary structure of the RNA sequence.

The problem of predicting the secondary structure of RNA has some similarities to DNA alignment, except that the sequence folds back on itself and aligns complementary bases rather than similar ones.

To predict the true secondary structure of RNA, a number of different computational goals have been formulated:

Find a secondary structure that:

1. maximizes the number of base pairs,
2. minimizes the "free energy", or
3. is optimal with respect to the "mutual information content", when considering a comparison of RNA sequences.

10.4.1 Combinatorics of structures

The simplest approach to predicting the secondary structure of RNA molecules is to find the configuration that maximizes the number of paired bases.

The number of possible configurations to be inspected grows exponentially with the length of the sequence:

Theorem 10.4.1 (Number of secondary structures) *Let $S(n)$ be the number of secondary structures for the sequence of length n. Then $S(0) = 0$, $S(1) = 1$, $S(2) = 1$, and for $n \geq 2$,*

$$S(n+1) = S(n) + S(n-1) + \sum_{j=2}^{n-1} S(j-1)S(n-j)$$

Thus, the number of possible configurations to be inspected grows exponentially with the length of the sequence:

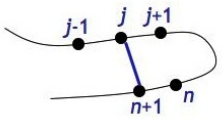
Proof: by induction.
I.S.: Obvious for $n = 1$ and $n = 2$, therefore, S(1) = S(2) = 1.

I.A.: Assume $S(k)$ is known for $1 \leq k \leq n$. Consider the sequence $[1, n+1]$. Then we distinguish two cases: Either $n+1$ is not basepaired or $n+1$ is paired.

Case 1: $n+1$ is not basepaired.

There are $S(n)$ structures, since that is the number of secondary structures $[1, n]$ can form.

Case 2: $n+1$ is basepaired



Two subcases: $S(n+1)$ is paired with $j = 1$, then $[2, n]$ can form $S(n-1)$ substructures.
$S(n+1)$ is paired with $j \geq 2$, then $[1, j-1]$ and $[j+1, n]$ can form $S(j-1)$ and $S(n-j)$ secondary structures independently.

Therefore, altogether we get $S(n+1) = S(n) + S(n-1) + \sum_{j=2}^{n-1} S(j-1)S(n-j)$.
□

10.5 The Nussinov folding algorithm

Given: RNA primary sequence

Task: Compute the secondary structure that maximizes the number of paired bases.

Dynamic programming can again be used to solve the problem efficiently (Nussinov, 1978).

Similar to pairwise alignment, the key idea of the recursive calculation is to consider in this case four ways of getting the best structure for $a_i \dots a_j$ from the best structures of smaller subsequences.

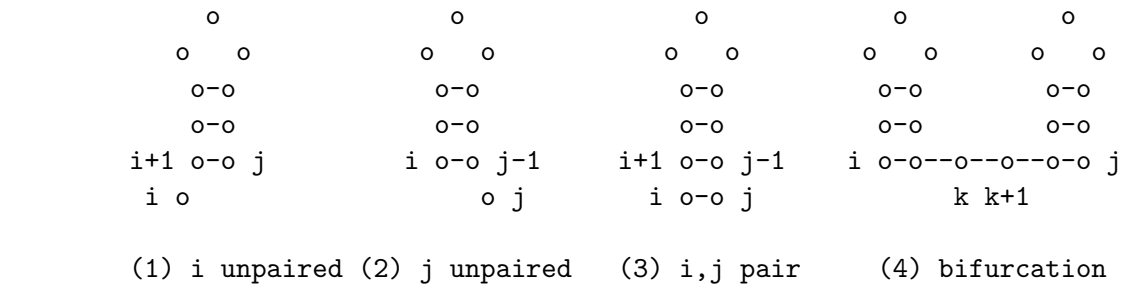
This works only because the structures are assumed to be nested.

Similar to pairwise alignment using dynamic programming, we now want to derive how the secondary structure of a sequence can be computed from the smaller ones.

Consider the string $A_{i,j} = a_i a_{i+1} \dots a_j$.

Here, we note that the smaller structures involve positions either from $i+1, \dots, j$, from $i, \dots, j-1$ or from $i+1, \dots, j-1$.

Observation: There are four ways to get an optimal structure for $A_{i,j}$ from smaller ones:



1. Add an unpaired base i to the best structure for the subsequence $i+1, j$,
2. add an unpaired base j to the best structure for the subsequence $i, j-1$,
3. add paired bases $i-j$ to the best structure for the subsequence $i+1, j-1$, or
4. combine two optimal substructures i, k and $k+1, j$.

For the pairing case $i-j$, we need a scoring function (similar to alignment):
For a given sequence $A = a_1 \dots a_L$ of length L, set

$$\delta(i,j) = \begin{cases} 1, & \text{if } a_i - a_j \text{ is a canonical base pair} \\ 0, & \text{else.} \end{cases}$$

The recursion is then given by:

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j), \\ \gamma(i,j-1), \\ \gamma(i+1,j-1) + \delta(i,j), \\ \max_{i<k<j}[\gamma(i,k) + \gamma(k+1,j)], \end{cases}$$

with initialization: $\gamma(i, i-1) = 0$ for $i = 2$ to L, and $\gamma(i,i) = 0$ for $i = 1$ to L.

The dynamic programming algorithm as usual has two stages:

In the *fill stage*, we will recursively calculate scores $\gamma(i,j)$ which are the maximal number of base pairs that can be formed for subsequences $a_i a_{i+1} \dots a_j$.

In the *traceback* stage, we traceback through the calculated matrix to obtain one of the maximally base paired structures.

Algorithm 10.5.1 (Nussinov, fill stage)

Input: Sequence $A = a_1 a_2 \dots a_L$
Output: Maximal number $\gamma(i,j)$ *of base pairs for* $(a_i, \dots, a_j)$.

Initialization:

$$\begin{aligned} \gamma(i, i-1) &= 0 \qquad \textit{for } i = 2 \textit{ to } L, \\ \gamma(i,i) &= 0 \qquad \textit{for } i = 1 \textit{ to } L; \end{aligned}$$

for $n = 2$ **to** L **do** *// longer and longer subsequences*

for $j = n$ **to** L **do**
Set $i = j - n + 1$

$$\textit{Set } \gamma(i,j) = \max \begin{cases} \gamma(i+1,j), \\ \gamma(i,j-1), \\ \gamma(i+1,j-1) + \delta(i,j), \\ \max_{i<k<j}[\gamma(i,k) + \gamma(k+1,j)]. \end{cases}$$

Return $\gamma(1, L)$

What is the time and space complexity of the Nussinov algorithm? Time: $O(n^3)$, Space: $O(n^2)$

Example: Consider the sequence $A =$ `GGGAAAUCC`. Here is the matrix γ after initialization ($i :\downarrow$, $j :\rightarrow$):

	G	G	G	A	A	A	U	C	C
G	0								
G	0	0							
G		0	0						
A			0	0					
A				0	0				
A					0	0			
U						0	0		
C							0	0	
C								0	0

Filling the matrix step-by-step, ending in:
$n = 9, j = 9, \ldots 9, i = 9 - 9 + 1 = 1 \Rightarrow (1, 9)$ // the final cell

$$\gamma(1,9) = \max \left\{ \begin{array}{l} \gamma(2,9) = 3, \\ \gamma(1,8) = 2, \\ \gamma(2,8) + \delta(1,9) = 2 + 1 \text{ (G-C)} \ = 3, \\ \max_{1<k<9}[\gamma(1,k) + \gamma(k+1,9)] = 2. \end{array} \right\} = 3$$

	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

$$\gamma(1, 9) = 3.$$

Two cells give rise to the max, so we have at least two possible tracebacks.

Algorithm 10.5.2 (Nussinov algorithm, traceback)

Algorithm traceback(i, j)
Input: Matrix γ *and positions* i, j.
Output: Secondary structure maximizing the number of base pairs.
Initial call: traceback$(1, L)$.

```
if i < j then
    if γ(i,j) = γ(i+1,j) then                          // case (1)
        traceback(i+1,j)
    else if γ(i,j) = γ(i,j-1) then                     // case (2)
        traceback(i,j-1)
    else if γ(i,j) = γ(i+1,j-1) + δ(i,j) then          // case (3)
        print base pair (i,j)
        traceback(i+1,j-1)
    else for k = i+1 to j-1 do                         // case (4)
        if γ(i,j) = γ(i,k) + γ(k+1,j) then
            traceback(i,k)
            traceback(k+1,j)
            break
end
```

Example: Here is the traceback through γ ($i :\downarrow$, $j :\rightarrow$):

	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

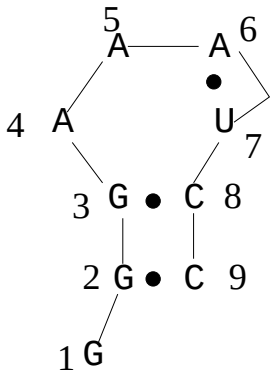
Solution 1 (blue):

```
G G G A A A U C C
. ( ( . . ( ) ) )
```

Solution 2 (red):

```
G G G A A A U C C
( ( . ( . . ) ) )
```

The resulting secondary structure of the first solution is:



Question: how does the secondary structure of solution 2 look like?

10.6 Energy minimization

Unfortunately, this approach does not produce good structure predictions. Better structure predictions are obtained by taking an energy model into account. In addition, when computing a prediction of the secondary structure of an RNA, it should be taken into account that:

- helical stacks of base pairs have a stabilizing effect, whereas
- loops have a destabilizing effect on the structure.

In *Bioinformatics I* we will look at the Zuker algorithm (Zuker & Stiegler 1981, Zuker 1989).

10.7 RNA folding via comparative analysis

Alternative secondary structure prediction algorithms to energy minimization techniques use *comparative* approaches.

Guiding principle in molecular biology:

structure is much more conserved than sequence.

Thus, a change in sequence coincides to maintain structure through base pairs. An impressive example is the tRNA which is not only structurally conserved with respect to the anti-codon family but also across species.

When one base of a pair changes, we usually find that its partner also changes so as to conserve that base pair. This phenomenon is called a *compensatory* base change.

How can we detect them? The key idea is to identify compensatory (Watson-Crick) correlated positions (so-called 'covarying' positions) in a multiple alignment, e.g.:

seq1 G**C**CUUCGG**G**C
seq2 G**A**CUUCGG**U**C
seq3 G**G**CUUCGG**C**C

The two bold columns are *covarying* to maintain Watson-Crick complementarity.

Note, that in order to work well comparative methods require many diverse sequences and highly accurate multiple alignments.

The amount of correlation of two positions can be computed as the *mutual information* content measure:

if you tell me the identity of position i, how much do I learn about the identity of position j?

10.7.1 Mutual information content

A method used to locate *covariant* positions in a multiple sequence alignment is based on the **mutual information content** of two columns:

First, for a column i of the alignment, the relative frequency $f_i(x)$ of a base $x \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ is calculated.

Second, the relative joint frequencies $f_{ij}(x, y)$ of two nucleotides, x in column i and y in column j, are calculated.

Then for a pair of columns i, j we compute the ratio $\frac{f_{ij}(x,y)}{f_i(x) \cdot f_j(y)}$.

If the base frequencies of any two columns i and j are *independent* of each other, then the ratio of ≈ 1.

If these frequencies are *correlated*, then this ratio will be significantly greater than 1.

To calculate the *mutual information content* $H(i, j)$ in bits between the two columns i and j, the logarithm of this ratio is calculated and summed over all occurring base-pair combinations:

$$H_{ij} = \sum_{xy} f_{ij}(x, y) \log_2 \frac{f_{ij}(x, y)}{f_i(x) f_j(y)}. \tag{10.1}$$

Example:

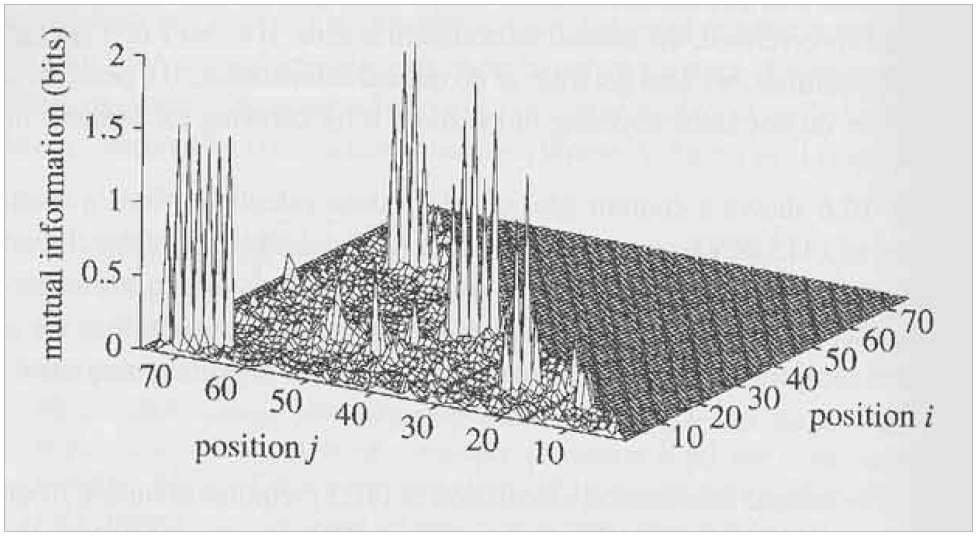
1	2	3	4	5	6
C	G	C	G	A	U
C	G	G	C	C	G
C	G	C	G	G	C
C	G	G	C	U	A

Compute: $H_{12} =$ _____
$H_{34} =$ _____
$H_{56} =$ _____

For RNA sequences, we expect a maximum value of 2 bits when there is perfect correlation and 0 for complete randomness and/or complete conservation.

If either site is conserved, there is less mutual information: for example, if all bases at site i are `A`, then the mutual information is 0, even if site j is always `U`, because there is no covariance.

Example of a mutual information content plot of a tRNA:



(`Source Durbin et al.,1999`).

The main problem with the comparative approach is that we need an accurate multiple alignment to get good structures and we need accurate structures to get a good alignment!

There are methods which address these two problems,folding and alignment, simultaneously: e.g.

Sankoff's DP algorithm

or

Dynalign

10.8 Some related websites

- Vienna RNA Secondary Structure Prediction:
  A web interface to the RNAfold program can be found at:
  `http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi`
- Zuker's mfold Server:
  A web interface to the `mfold` program can be found at:
  `http://mfold.rna.albany.edu`
- Sean Eddy's tRNAScan-SE: Allows one to search for tRNA genes in genomic sequences:
  `http://lowelab.ucsc.edu/tRNAscan-SE/`

Important RNA Databases:

- RFam: The Rfam database of RNA alignments and CMs
  `http://rfam.xfam.org`

Many more links at:

- `http://www.rna.uni-jena.de/en/the-rna-world-website/`

10.9 References and recommended reading

- Clancy S (2008) RNA Functions. Nature Education 1:102.

- Durbin R, Eddy S, Krogh A and Mitchison G (1998) Biological sequence analysis, Cambridge.
- Eddy S (2004) How do RNA folding algorithms work? Nature Biotechnology 22.
- Gilbert W (1986) The RNA world. Nature 319: 618.
- Nussinov R, Pieczenik G, Griggs JG, Kleitman DJ (1978) Algorithms for Loop Matching. SIAM J Appl Math 35, 68-81.
- Zuker M, Stiegler P (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. 9(1):133-48.