4 RNA Secondary Structure

Sources for this lecture:

- R. Durbin, S. Eddy, A. Krogh und G. Mitchison, Biological sequence analysis, Cambridge, 1998
- J. Setubal & J. Meidanis, Introduction to computational molecular biology, 1997.
- D.W. Mount. Bioinformatics: Sequences and Genome analysis, 2001.
- M. Zuker, Algorithms in Computational Molecular Biology. Lectures, 2002, `http://www.rpi.edu/~zukerm/MATH-4961/PostScript/rnafold.ps`.

4.1 RNA

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

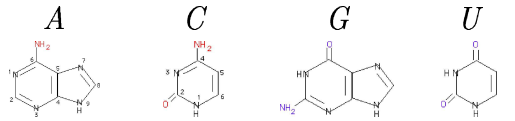
- DNA is used to store and replicate genetic information,
- proteins are the basic building blocks and active players in the cell, and
- RNA plays a number of different important roles in the production of proteins from instructions encoded in the DNA.

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 originally, life was based on RNA and over time RNA delegated the data storage problem to DNA and the problem of providing structure and catalytic functionality to proteins.

4.2 RNA secondary structure

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



adenine cytosine guanine uracil.

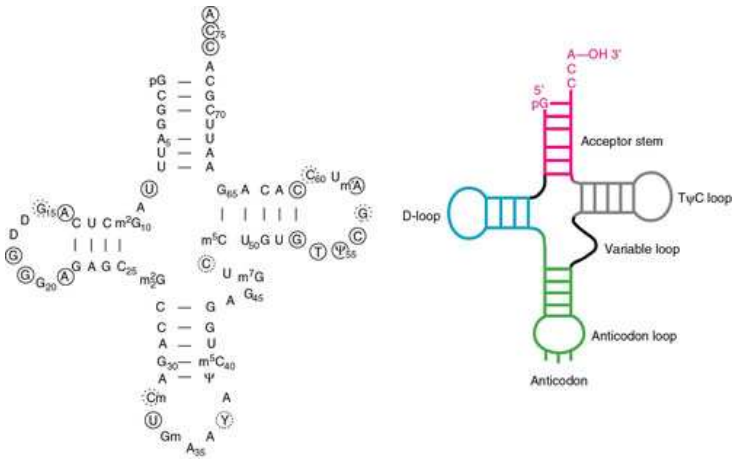
(source: Zuker)

Unlike DNA, RNA is single stranded. However, complementary bases `C` – `G` and `A` – `U` form stable *base pairs* with each other using hydrogen bonds. These are called *Watson-Crick*

pairs. Additionally, one sometimes considers the weaker $\mathtt{U}-\mathtt{G}$ *wobble pairs*. These are all called *canonical base pairs*.

When base pairs are formed between different parts of a RNA molecule, then these pairs are said to define the *secondary structure* of the RNA molecule.

The secondary structure of a tRNA:



This particular tRNA is from yeast and is for the amino acid phenylalanine. (source: http://www.blc.arizona.edu/marty/411/Modules/ribtRNA.html)

4.3 Definition of RNA secondary structure

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

Definition For our purposes, a *RNA molecule* is simply a string

$$x = (x_1, x_2, \ldots, x_L),$$

with $x_i \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$ for all i.

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

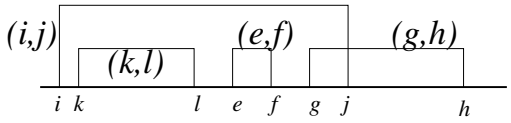
1. $j - i > 3$, i.e. the bases are not too close to each other, (although we will ignore this condition below), and
2. $\{i, j\} \cap \{i', j'\} = \emptyset$, i.e. the base pairs don't conflict.

Definition 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').

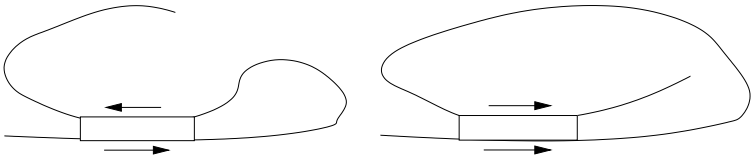
4.4 Nested structures

In the following, we only will consider *nested* secondary structures, as the more complicated non-nested structures are not tractable with the methods we will discuss.



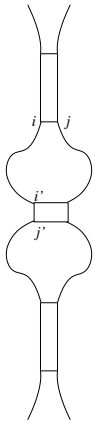
Here, the interactions (i, j) and (g, h) are not nested.

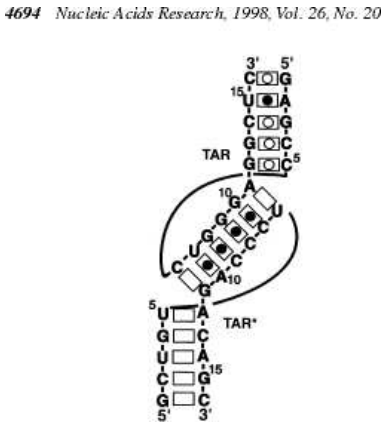
Interactions that are not nested give rise to a *pseudo knot* configuration in which segments of sequence are bonded in the "same direction":



unknoted pseudo knot

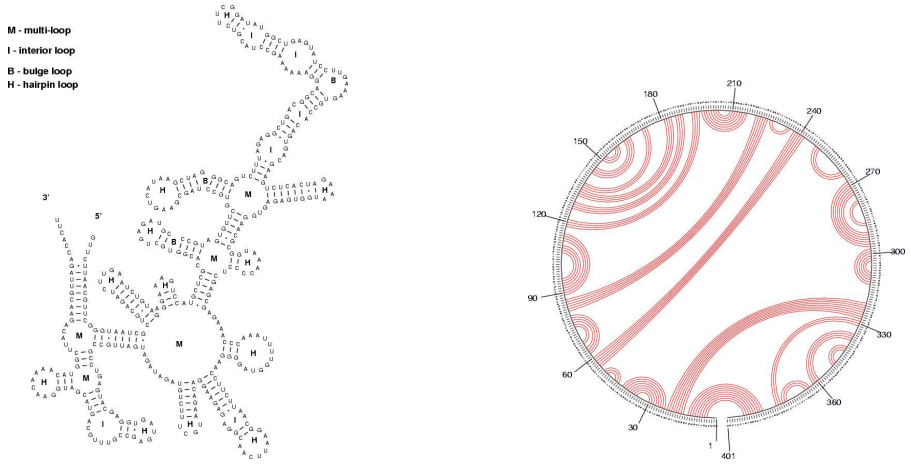
The nested requirement excludes other types of configurations, as well, such as *kissing hairpins*, for example:





4.5 Example of secondary structure

Predicted structure for Bacillus Subtilis RNAase P RNA:



(source: Zuker)

This example shows the different types of single- and double-stranded regions in RNA secondary structures:

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

4.6 Prediction of RNA secondary structure

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.

The goal of aligning two or more biological sequences is to determine whether they are homologous or just similar. In contrast, a secondary structure for an RNA is a simplification of the complex three-dimensional folding of the RNA molecule.

Problem Determine the true secondary structure of an RNA.

Variants: find a secondary structure that:

1. maximizes the number of base pairs,
2. minimizes the "free energy", or
3. is optimal, given a family of related sequences.

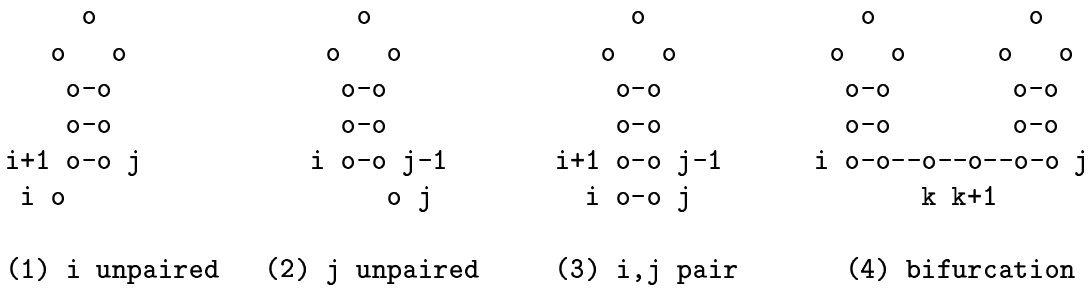
4.7 The Nussinov folding algorithm

The simplest approach to predicting the secondary structure of RNA molecules is to find the configuration with the greatest number of paired bases. The number of possible configurations to be inspected grows exponentially with the length of the sequence.

Fortunately, we can employ dynamic programming to obtain an efficient solution. In 1978 Ruth Nussinov et al. published a method to do just that.

The algorithm is recursive. It calculates the best structure for small subsequences, and works its way outward to larger and larger subsequences. The key idea of the recursive calculation is that there are only four possible ways of the getting the best structure for i, j from the best structures of the smaller subsequences.

Idea: There are four ways to obtain an optimal structure for a sequence i, j from smaller substructures:



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 $k + 1, j$.

Given a sequence $x = (x_1, \ldots, x_L)$ of length L. We set $\delta(i, j) = 1$, if $x_i - x_j$ is a canonical base pair and 0, else.

The dynamic programing algorithm 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 $(x_i, \ldots, x_j)$.

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

4.8 The fill stage

Algorithm (Nussinov RNA folding, fill stage)

Input: Sequence $x = (x_1, x_2, \ldots, x_L)$
Output: Maximal number $\gamma(i, j)$ of base pairs for $(x_i, \ldots, x_j)$.

Initialization:

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

for $n = 2$ **to** L **do** // longer and longer subsequences
  for $j = n$ **to** L **do**
    Set $i = j - n + 1$

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

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

Nussinov Matrix

	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

Here is the matrix γ after executing the recursion ($i :\downarrow$, $j :\rightarrow$):

Nussinov Matrix

	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

b

Values obtained using $\delta(a, b) = \begin{cases} 1 & \text{if } \{a, b\} = \{\mathtt{A}, \mathtt{U}\} \text{ or } \{\mathtt{C}, \mathtt{G}\}, \\ 0 & \text{else.} \end{cases}$

4.9 The traceback stage

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 $\gamma(i,j) = \gamma(i+1,j)$ **then** // case (1)
    traceback$(i+1, j)$
  else if $\gamma(i,j) = \gamma(i,j-1)$ **then** // case (2)
    traceback$(i, j-1)$
  else if $\gamma(i,j) = \gamma(i+1,j-1) + \delta(i,j)$ **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 $\gamma(i,j) = \gamma(i,k) + \gamma(k+1,j)$ **then**
      traceback(i, k)
      traceback$(k+1, j)$
      break
end

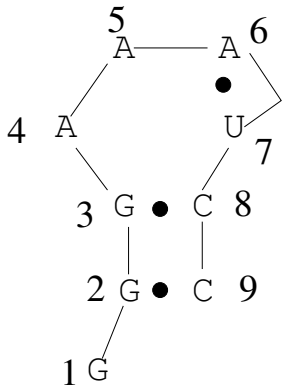
Here is the traceback through γ (i :↓, j :→):

Nussinov Matrix

		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

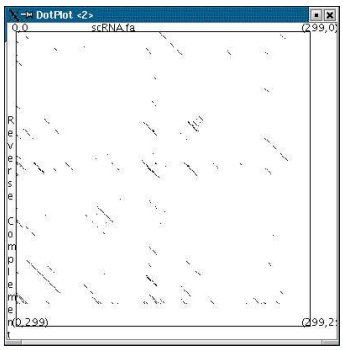
(There is a slight error in the traceback shown in Durbin et al. page 271)

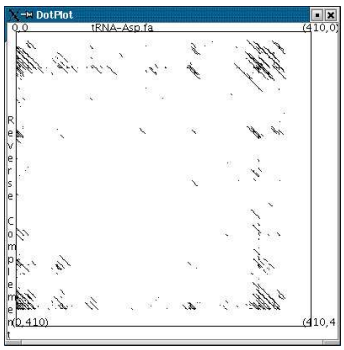
The resulting secondary structure is:



4.10 Application of dot plots

Given an RNA sequence x. Self complementary regions may be found by performing a dot matrix analysis of x with its reverse complement $\bar{x}$. Here are two examples:





4.11 Simple energy minimization

Maximizing the number of base pairs as described above does not lead to good structure predictions. Better predictions can be obtained by minimizing the following *energy function*

$$E((x,P)) = \sum_{(i,j)\in P} e(x_i, x_j),$$

where $e(x_i, x_j)$ is the amount of *free energy* associated with the base pair (x_i, x_j).

Reasonable values for e at 37^oC are -3, -2 and -1 kcal/mole for base pairs $\mathtt{C}-\mathtt{G}$, $\mathtt{A}-\mathtt{U}$ and $\mathtt{G}-\mathtt{U}$, respectively.

Obviously, a few simple changes to the Nussinov algorithm will produce a new algorithm that can solve this energy minimization problem.

Namely, we need to change _____, _____, and _____.

Unfortunately, this approach does not produce good structure predictions because it does not take into account that helical stacks of base pairs have a stabilizing effect whereas loops

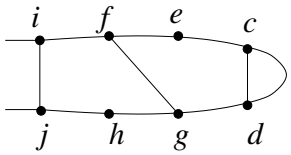
have a destabilizing effect on the structure. A more sophisticated approach is required.

The most sophisticated algorithm for folding single RNAs is the *Zuker* algorithm, an energy minimization algorithm which assumes that the correct structure is the one with the lowest equilibrium free energy ΔG.

The ΔG of an RNA secondary structure is approximated as the sum of individual contributions from loops, base pairs and other secondary structure elements. as we will see, an important difference to the Nussinov calculation is that the energies are computed from loops rather than from base pairs. This provides a better fit to experimentally observed data.

4.12 The k-loop decomposition

If (i, j) is a base pair in P and $i < h < j$, then we say that h is *accessible* from (i, j) if there is no base pair $(i', j') \in P$ such that $i < i' < h < j' < j$. Similarly, we say that (f, g) is *accessible* from (i, j), if both f and g are.



The set s of all $k-1$ base pairs and k' unpaired bases that are accessible from (i, j) is called the *k-loop* closed by (i, j).

The *null* k-loop consists of all *free* base pairs and unpaired bases that are accessible from no base pair.

Fact Given $x = (x_1, x_2, \ldots, x_L)$. Any secondary structure P on x partitions the set $\{1, 2, \ldots, L\}$ into k-loops $s_0, s_1, \ldots, s_m$, where $m > 0$ iff $P \neq \emptyset$.

We can now give a formal definition of names introduced earlier:

1. A 1-loop is called a *hairpin* loop.
2. Assume that there is precisely one base pair (i', j') accessible from (i, j). Then this 2-loop is called
   (a) a *stacked pair*, if $i' - i = 1$ and $j - j' = 1$,
   (b) a *bulge loop*, if $i' - i > 1$ or $j - j' > 1$, but not both, and
   (c) an *interior loop*, if both $i' - i > 1$ and $i - j' > 1$.
3. A k-loop with $k \geq 3$ is called a *multi-loop*.

The following is a consequence of nestedness:

Fact The number of non-null k-loops equals the number of base pairs.

The *size* of a k-loop is the number k' of unpaired bases that it contains.

Each k-loop is assigned an energy $e(s_i)$ and the energy of a structure P is given by:

$$E(p) = \sum_{i=0}^{m} e(s_i).$$

So now the energy is a function of k-loops instead of a function of base pairs.

4.13 Zuker's algorithm for folding RNA

We will now develop a more involved dynamic program that uses loop-dependent rules. It is due to M. Zuker (Zuker & Stiegler 1981, Zuker 1989). We will use two matrices, W and V.

For $i < j$, let $W(i, j)$ denote the minimum folding energy of all non-empty foldings of the subsequence $x_i, \ldots, x_j$.

Additionally, let $V(i, j)$ denote the minimum folding energy of all non-empty foldings of the subsequence $x_i, \ldots, x_j$, *containing the base pair* (i, j). The following obvious fact is crucial:

$$W(i, j) \leq V(i, j) \text{ for all } i, j.$$

These matrices are initialized as follows:

$$W(i, j) = V(i, j) = \infty \text{ for all } i, j \text{ with } j - 4 < i < j.$$

(Note that we are now going to enforce that two paired bases are at least 3 positions away from each other).

4.14 Loop-dependent energies

We define different energy functions for the different types of loops:

- Let $eh(i, j)$ be the energy of the hairpin loop closed by the base pair (i, j),
- let $es(i, j)$ be the energy of the stacked pair (i, j) and $(i + 1, j - 1)$,
- let $ebi(i, j, i', j')$ be the energy of the bulge or interior loop that is closed by (i, j), with (i', j') accessible from (i, j), and
- let a denote a constant energy term associated with a multi-loop (a more general function for this case will be discussed later).

Predicted free-energy values (kcal/mole at 37^oC) for base pair stacking:

	A/U	C/G	G/C	U/A	G/U	U/G
A/U	-0.9	-1.8	-2.3	-1.1	-1.1	-0.8
C/G	-1.7	-2.9	-3.4	-2.3	-2.1	-1.4
G/C	-2.1	-2.0	-2.9	-1.8	-1.9	-1.2
U/A	-0.9	-1.7	-2.1	-0.9	-1.0	-0.5
G/U	-0.5	-1.2	-1.4	-0.8	-0.4	-0.2
U/G	-1.0	-1.9	-2.1	-1.1	-1.5	-0.4

Predicted free-energy values (kcal/mole at 37^oC) for features of predicted RNA secondary structures, by size of loop:

size	internal loop	bulge	hairpin
1	.	3.9	.
2	4.1	3.1	.
3	5.1	3.5	4.1
4	4.9	4.2	4.9
5	5.3	4.8	4.4
10	6.3	5.5	5.3
15	6.7	6.0	5.8
20	7.0	6.3	6.1
25	7.2	6.5	6.3
30	7.4	6.7	6.5

4.15 The main recursion

For all i, j with $1 \leq i < j \leq L$:

$$W(i,j) = \min \begin{cases} W(i+1,j) \\ W(i,j-1) \\ V(i,j) \\ \min_{i<k<j}\{W(i,k) + W(k+1,j)\}, \end{cases} \tag{4.1}$$

and

$$V(i,j) = \min \begin{cases} eh(i,j) \\ es(i,j) + V(i+1,j-1) \\ VBI(i,j), \\ VM(i,j), \end{cases} \tag{4.2}$$

where

$$VBI(i,j) = \min_{\substack{i < i' < j' < j \\ i' - i + j - j' > 2}} \{ebi(i,j,i',j') + V(i',j')\}, \tag{4.3}$$

and

$$VM(i,j) = \min_{i<k<j-1}\{W(i+1,k) + W(k+1,j-1)\} + a. \tag{4.4}$$

Equation 4.1 considers the four cases in which (a) i is unpaired, (b) j is unpaired, (c) i and j are paired to each other and (d) i and j are paired, but not to each other. In case (c) we reference the auxiliary matrix V.

Equation 4.2 considers the different situations that arise when bases i and j are paired, closing (a) a hairpin loop, (b) a stacked pair, (c) a bulge or interior loop or (d) a multi-loop. The two latter cases are more complicated and are obtained from equations 4.3 and 4.4.

Equation 4.3 takes into account all possible ways to define a bulge or interior loop that involves a base pair (i', j') and is closed by (i, j). In each situation, we have a contribution from the bulge or interior loop and a contribution from the structure that is on the opposite side of (i', j').

Equation 4.4 considers the different ways to obtain a multi-loop from two smaller structures and adds a constant contribution of a to close the loop.

4.16 Time analysis

The minimum folding energy E_{min} is given by $W(1, L)$.

There are $O(L^2)$ pairs (i, j) satisfying $1 \leq i < j \leq L$.

The computation of

1. W takes $O(L^3)$ steps,
2. V takes $O(L^2)$ steps,
3. VBI takes $O(L^4)$ steps, and
4. VM takes $O(L^3)$ steps,

and so the total run time is $O(L^4)$.

The most practical way to reduce the run time to $O(L^3)$ is to limit the size of a bulge or interior loop to some fixed number d, usually about 30. This is achieved by limiting the search in Equation 4.3 to $2 < i' - i + j - j' - 2 \leq d$.

4.17 Modification of multi-loop energy

In Equation 4.4 we used a constant energy function for multi-loops. More generally, we can use the following function

$$e(\text{multi} - \text{loop}) = a + b \times k' + c \times k,$$

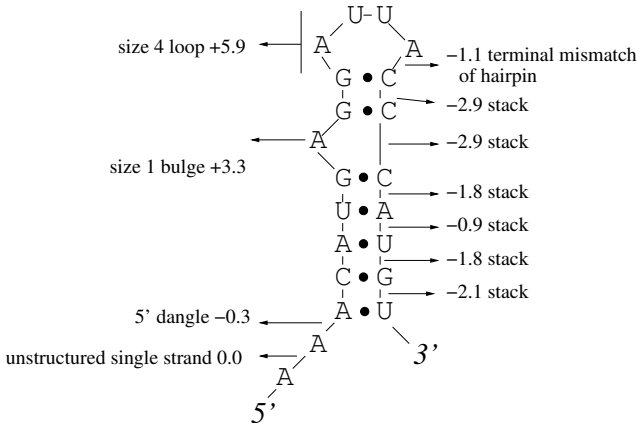
where a, b and c are constants and k' is the number of unpaired bases in the multi-loop.

This is a convenient function to use because, similar to the introduction of affine gap penalties in sequence alignment, a cubic order algorithm remains possible.

A number of additional modifications to the algorithm can be made to handle the stacking of single bases. These modifications lead to better predictions, but are beyond the scope of our lecture.

4.18 Example of energy calculation

Here is any example of the full energy calculation for an RNA stem loop (the wild type $R17$ coat protein binding site):



Overall energy value: -4.6 kcal/mol

4.19 RNA folding via comparative analysis

Although energy minimization techniques are attractive, almost all trusted RNA secondary structures to date where determined using comparative analysis. However, comparative methods require many diverse sequences and highly accurate multiple alignments to work well.

The key idea is to identify Watson-Crick correlated positions in a multiple alignment, e.g.:

```
seq1  GCCUUCGGGC
seq2  GACUUCGGUC
seq3  GGCUUCGGCC
```

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?*

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

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

calculated.

Second, the 16 joint frequencies $f_{ij}(x, y)$ of two nucleotides, x in column i and y in column j, are calculated.

If the base frequencies of any two columns i and j are *independent* of each other, then the ratio of $\frac{f_{ij}(x,y)}{f_i(x) \times f_j(y)} \approx 1$.

If these frequencies are *correlated*, then this ratio will be 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 possible 16 base-pair combinations:

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

This measure is maximum at 2 bits, representing perfect correlation.

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.

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!

4.20 Mutual information content

Examples of how to compute the mutual information content:

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