8 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. Waterman. Introduction to computational biology, 1995.
- NC Jones & PA Pevzner. An Introduction to Bioinformatics Algorithms, 2004.

8.1 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.

8.1.1 RNAs and their functions

Types of RNA:

- transfer RNA (tRNA)
- messenger RNA (mRNA)
- ribosomal RNA (rRNA)
- small interfering RNA (siRNA)
- micro RNA (miRNA)
- small nuclear RNA (snoRNA)
- ...

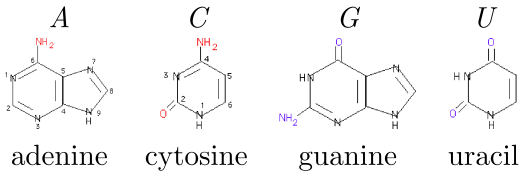
Functions:

- adaptor molecule (tRNA)
- transmitter of genetic information (mRNA)

- carrier of genetic information (virus)
- regulator of gene expression (siRNA, miRNA)
- catalyst (ribosome)
- many more

8.1.2 Definition of RNA structure

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



(source: Zuker)

Unlike DNA, RNA is single-stranded. The single-stranded RNA is viewed as a linear sequence $a = a_1 a_2 \dots a_n$ of ribonucleotides. The sequence a is called the *primary structure*.

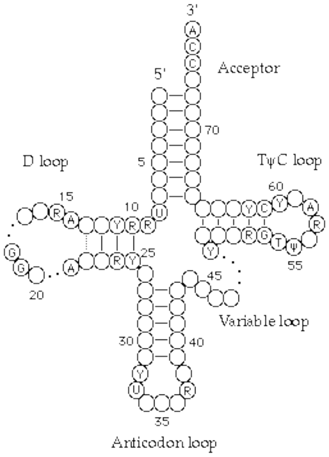
However, complementary bases $\mathtt{C} - \mathtt{G}$ and $\mathtt{A} - \mathtt{U}$ form stable *base pairs* with each other using hydrogen bonds. These are called *Watson-Crick* pairs. Additionally, one sometimes considers the weaker $\mathtt{G} - \mathtt{U}$ *wobble pairs*. These are all called *canonical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

GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAUCUGGAGGUCCUGUG

UUCGAUCCACAGAAUUCGCACCA



Definition 8.1.1 *For our purposes, a* RNA molecule *is simply a string*

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

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

Definition 8.1.2 *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. any base participates in at most one base pair.

Definition 8.1.3 *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*

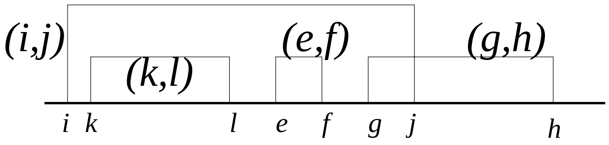
1. $i < j < i' < j'$, i.e. (i, j) precedes (i', j'), or

2. $i < i' < j' < j$, i.e. (i, j) includes (i', j').

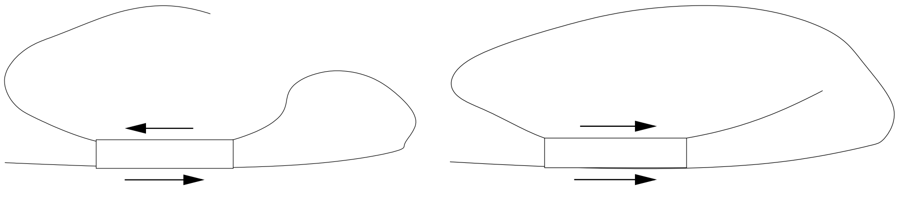
8.1.3 Nested structures

In the following, we only will consider *nested* secondary structures, as the non-nested structures are not tractable using the discussed methods.

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



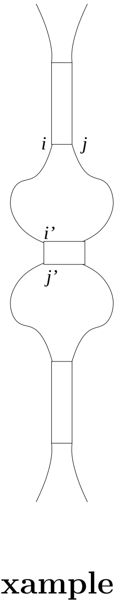
Interactions that are not nested give rise to a *pseudo knot* configuration in which segments of sequence are bonded in the "same direction":

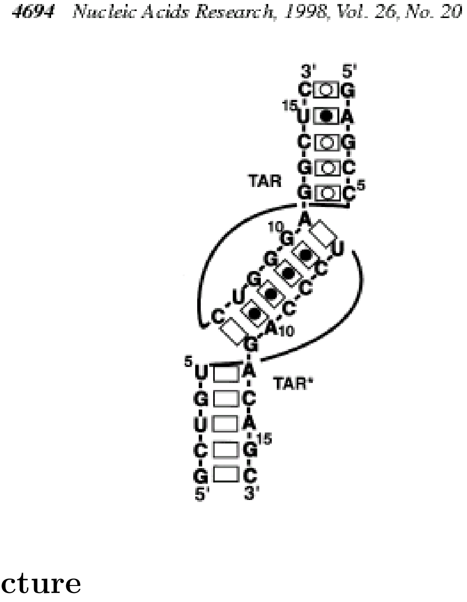


unknoted

pseudo knot

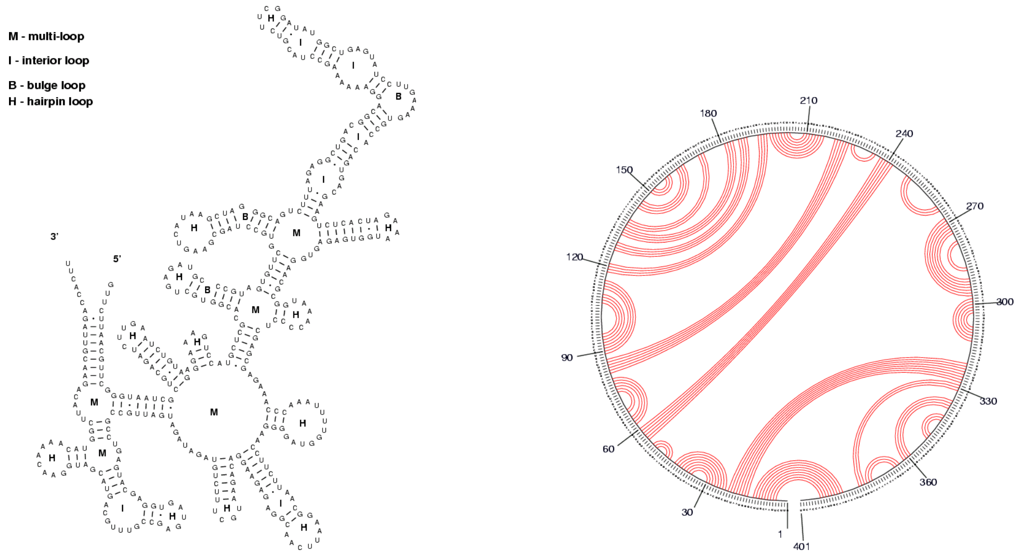
The nested requirement excludes other types of configurations, as well, such as *kissing hairpins*, for example:





8.1.4 Example of secondary structure

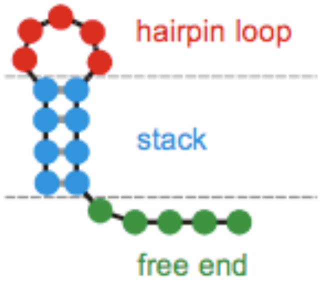
Predicted structure for RNAase P RNA of *Bacillus subtilis*:



(source: Zuker)

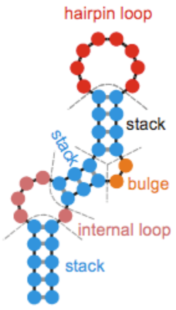
This example shows the different types of single- and double-stranded regions in RNA secondary structures: .3cm

- single-stranded RNA,
- double-stranded RNA helix of stacked base pairs,
- stem and loop or hairpin loop



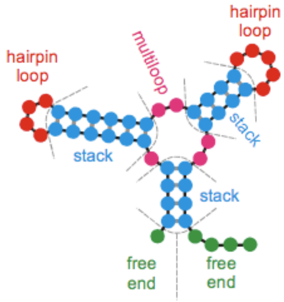
,

- bulge loop,
- interior loop, and



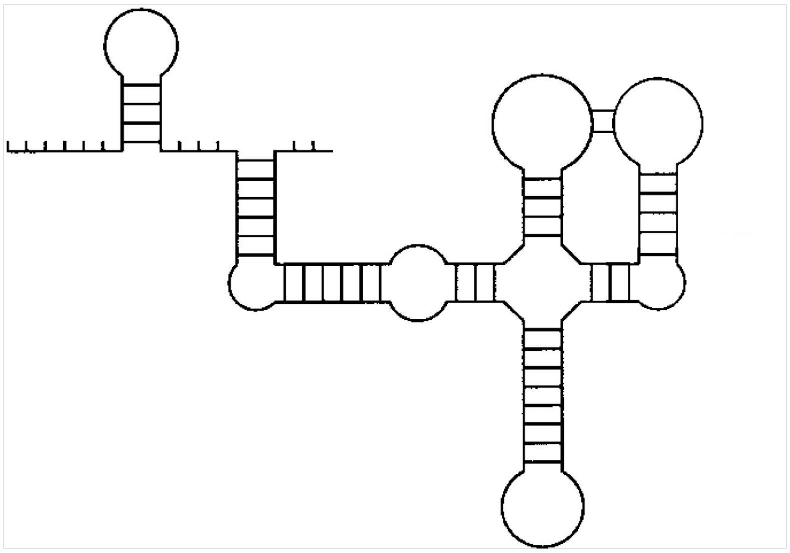
,

- junction or multi-loop.



,

The following example shows all of the above structural elements and a kissing hairpin:



8.1.5 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.

Problem 8.1.4 *Determine the true secondary structure of an RNA.*

By "true secondary structure" we mean the base-pairings that occur in the actual three-dimensional folding of the RNA molecule.

To predict the "true secondary structure" computationally, we solve a related optimization problem, that is, we determine a secondary structure that either:

1. maximizes the number of base pairs,
2. minimizes the "free energy", or
3. is optimal in terms of "mutual information", when an alignment of similar sequences is given.

8.2 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.[1] 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 smaller subsequences.

[1] R Nussinov, G Pieczenik, JG Griggs, DJ Kleitman (1978) Algorithms for Loop Matching. SIAM J Appl Math 35, 68-81

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

```
  o               o                o             o           o
o   o           o   o            o   o         o   o      o   o
o-o             o-o              o-o           o-o        o-o
o-o             o-o              o-o           o-o        o-o
i+1 o-o j         i o-o j-1      i+1 o-o j-1     i o-o--o--o--o-o j
i o                   o j         i o-o j              k k+1

(1) i unpaired  (2) j unpaired    (3) i,j pair      (4) bifurcation
```

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 $x = (x_1, \dots, x_L)$ of length L. Set

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

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 $(x_i, \dots, x_j)$.

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

8.2.1 The fill stage

Algorithm 8.2.1 (Nussinov, fill stage)

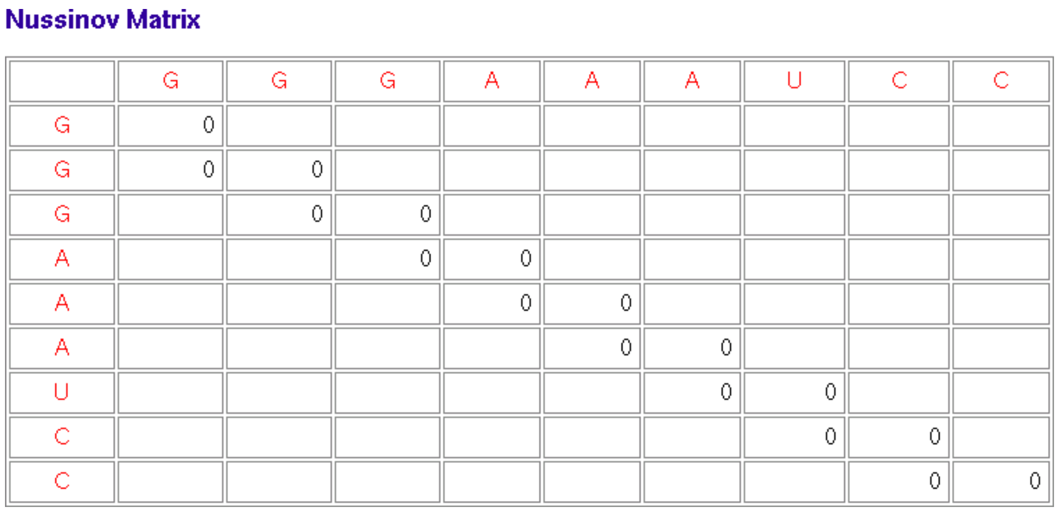
Input: Sequence $x = (x_1, x_2, \dots, x_L)$
Output: Maximal number $\gamma(i, j)$ *of base pairs for* $(x_i, \dots, x_j)$.

Initialization:

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

```
for $n = 2$ to $L$ do        // longer and longer subsequences
        for $j = n$ to $L$ do
            Set $i = j - n + 1$
            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 $\gamma(1, L)$
```

Consider the sequence $x =$ GGGAAAUCC. Here is the matrix γ after initialization (i :↓, j :→):



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

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

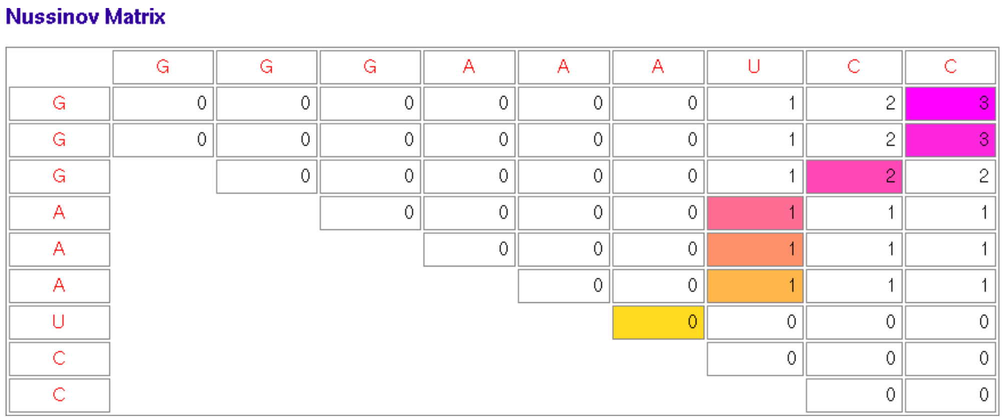
8.2.2 The traceback stage

Algorithm 8.2.2 (Nussinov, traceback)

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

```
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[2] through γ (i :↓, j :→):



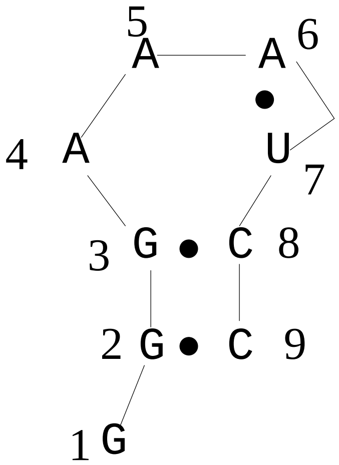
Nussinov Matrix

	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

[2] (Source: http://ludwig-sun2.unil.ch/˜bsondere/nussinov/form.html)

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

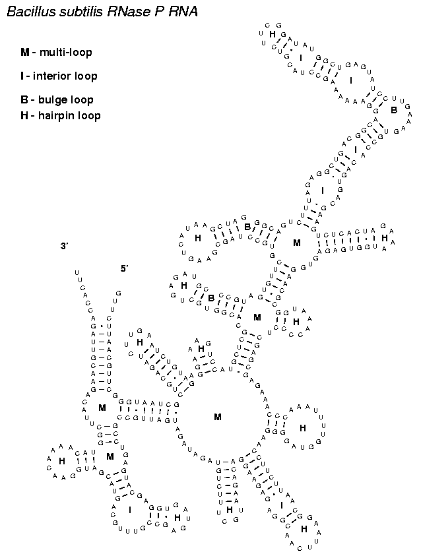
The resulting secondary structure is (assuming minimal hairpin loop length 0):

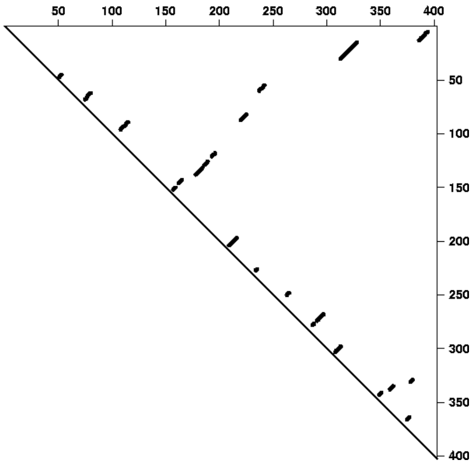


Exercise: draw the other optimal solutions.

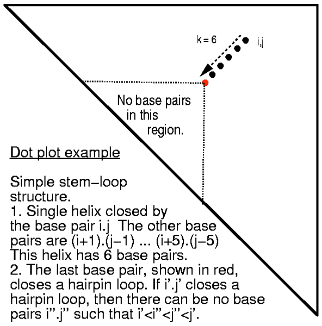
8.3 Application of dot plots

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 is the example for the RNAse P from *Bacillus subtilis*:



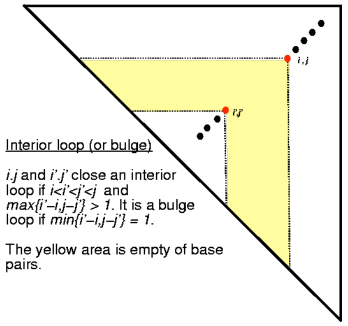


Understanding dot plots (from Zuker):



Dot plot example

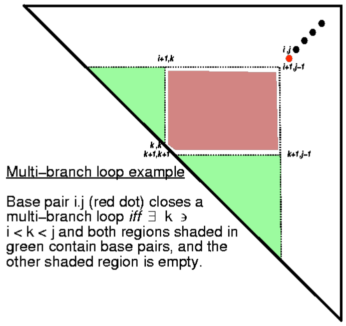
Simple stem–loop structure.
1. Single helix closed by the base pair i.j The other base pairs are (i+1).(j–1) ... (i+5).(j–5) This helix has 6 base pairs.
2. The last base pair, shown in red, closes a hairpin loop. If i'.j' closes a hairpin loop, then there can be no base pairs i''.j'' such that i'<i''<j''<j'.



Interior loop (or bulge)

i.j and *i'.j'* close an interior loop if *i<i'<j'<j* and *max{i'–i,j–j'} > 1*. It is a bulge loop if *min{i'–i,j–j'} = 1*.

The yellow area is empty of base pairs.



Multi–branch loop example

Base pair i.j (red dot) closes a multi–branch loop *iff* ∃ k ∍ i < k < j and both regions shaded in green contain base pairs, and the other shaded region is empty.