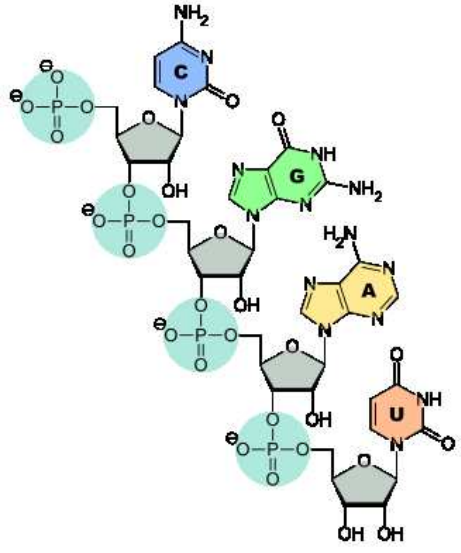
Bioinformatics
(RNA structures)

Marc Hellmuth

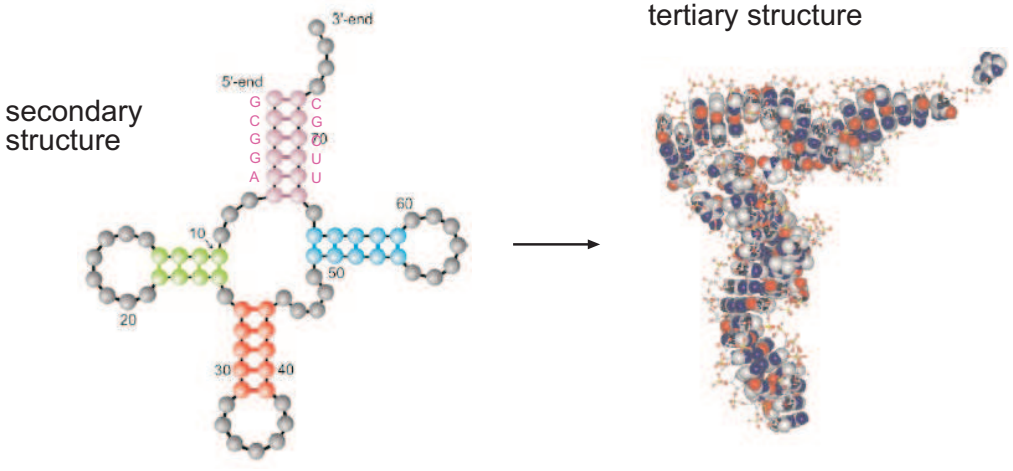
What is RNA (Ribonucleic acid)?

- single-stranded polymer
- polymer made of nucleotides+backbone
- guanine (G), adenine (A), uracil (U), cytosine (C)
- alternating sugar (ribose) and phospat groups (related to phosphoric acid) nucleotides are attached to sugar
- the nucleotides of polymer can bind (A-U, C-G, G-U), i.e., unlike DNA it is more often found in nature as a single-strand folded unto itself, rather than a paired double-strand.

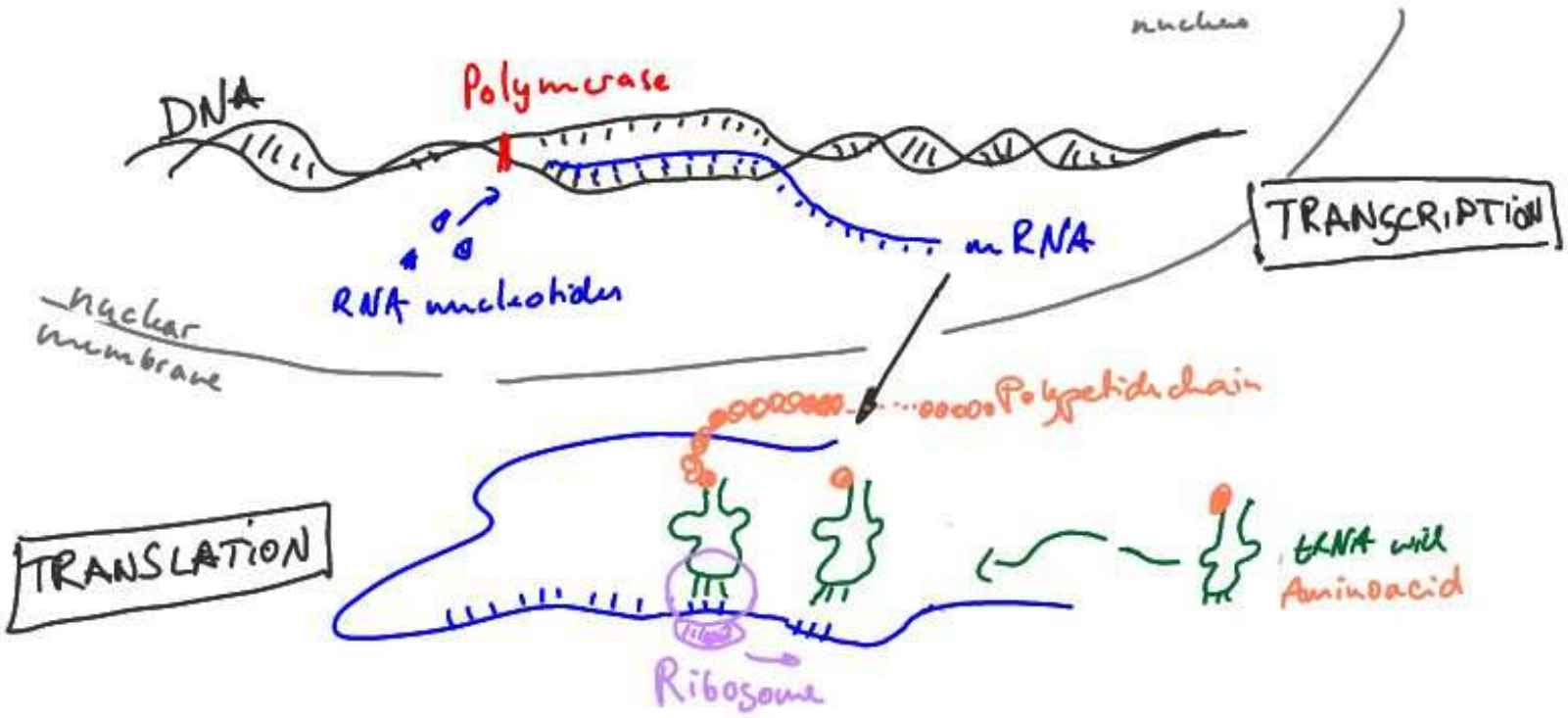


primary structure

5'-end GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA 3'-end



Protein Synthesis - well-known mRNA



RNA function

coding RNA

- mRNA: convey genetic information from DNA to the ribosome

many RNAs do not code for protein
(about 97% are non-protein- coding in eukaryotes)

non-coding RNA

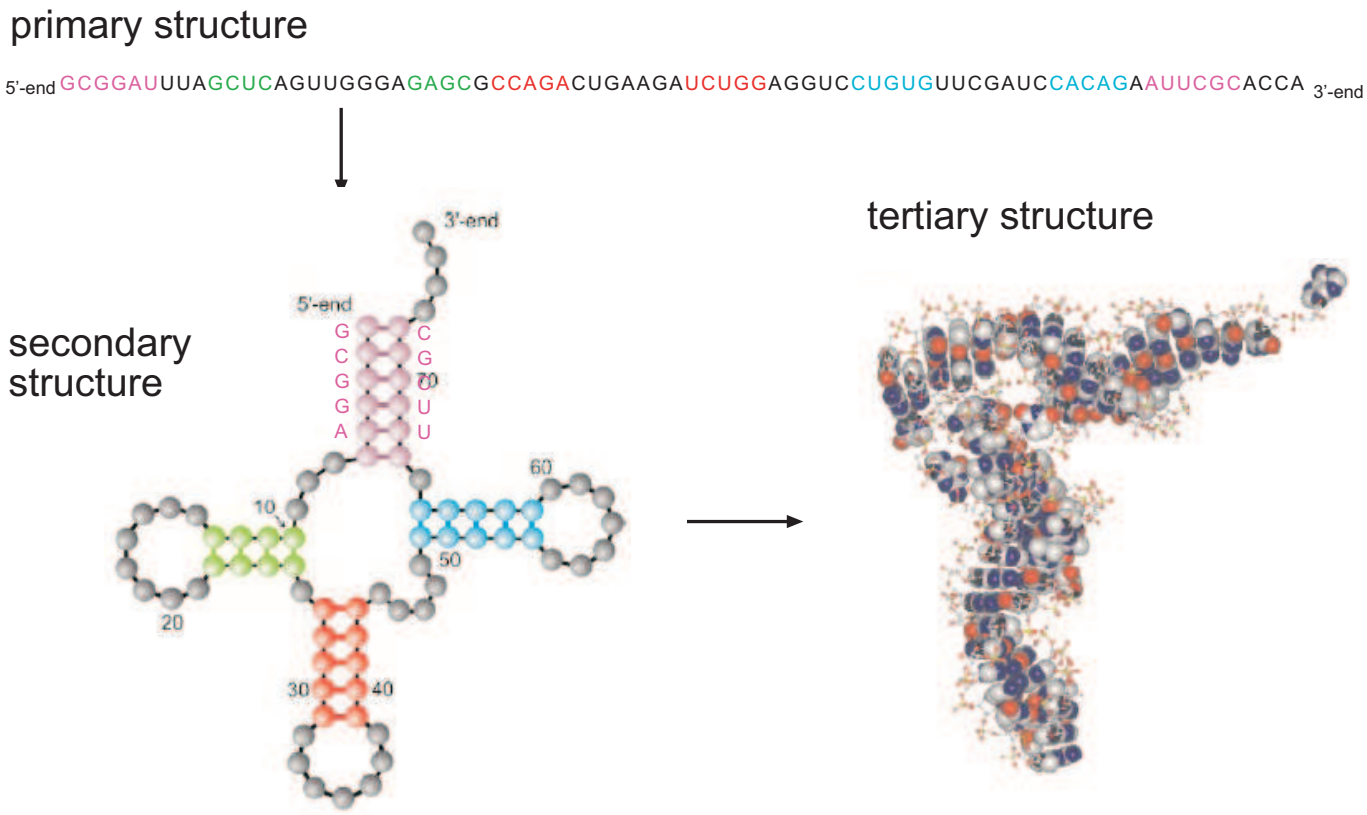
- tRNA: linking codons to aminoacids
- rRNA (ribosomalRNA): part of of the ribosome, essential for protein synthesis
- snRNA (Small nuclear RNA, $\sim$ 150 nt) : splicing and other functions
- miRNA (microRNA, 21-22 nt): regulation of gene expression
- ...

RNA can even act as genome (virus)

RNA - WORLD - HYPOTHESIS: self-replicating RNA molecules are precursors to all current life on Earth.

RNA Structure

Function is determined by structure.



Base Pairing Rules $\mathbb{B}$

- Watson-Crick base pairings
  - Adenin (A) - Uracil (U)
  - Cytosin (C) - Guanin (G)
- wobble base pairings
  - Guanin (G) - Uracil (U)

RNA Structure

In what follows: $\mathbb{A} = \{A, C, G, U\}$

A **primary structure** (of length n) is a sequence $s = s_1 \dots s_n \in \mathbb{A}^n$.

A **secondary structure** *Sec* is a collection of ordered pairs (i, j), where $1 \leq i < j \leq n$, s.t. the following properties hold:

1. If $(i, j), (k, l) \in Sec$, then it is not the case that $i < k < j < l$.
2. If $(i, j), (k, l) \in Sec$ and $i \in (k, l)$ implies that $i = k$ and $j = l$.
3. If $(i, j) \in Sec$, then $j > i + \theta$, where θ is a fixed integer and usually taken to be 3.

A secondary structure *Sec* for a given sequence $s = s_1 \dots s_n \in \mathbb{A}^n$ is a secondary structure fullfilling in addition

4. If $(i, j) \in Sec$, then $s_i s_j \in \mathbb{B} = \{AU, UA, GC, CG, GU, UG\}$.

If item 4. is fullfilled, then we say that the sequence $s \in \mathbb{A}^n$ realizes *Sec*.

A **tertiary structure** is basically the 3D structure, i.e., refers to locations of the atoms in three-dimensional space.

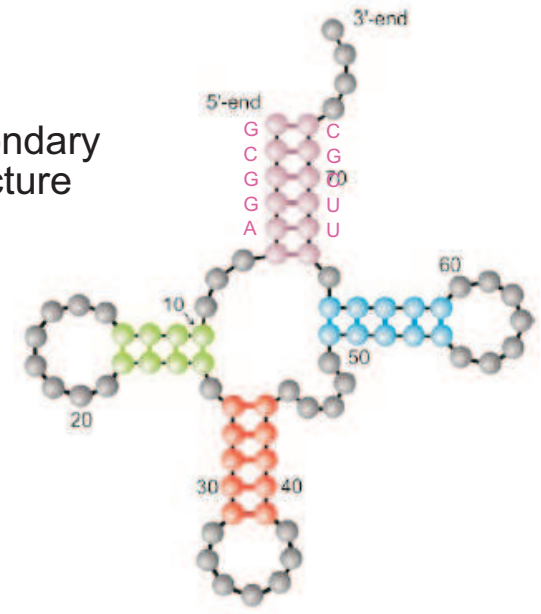
RNA Structure

$$s = s_1 \dots s_n \in C_n = \{A, C, G, U\}^n$$

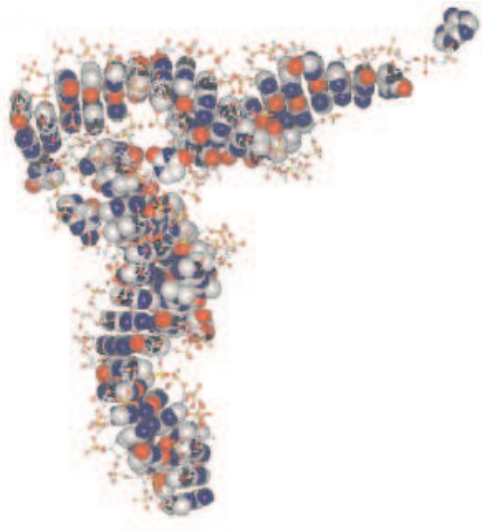
primary structure

5'-end GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAUCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA 3'-end

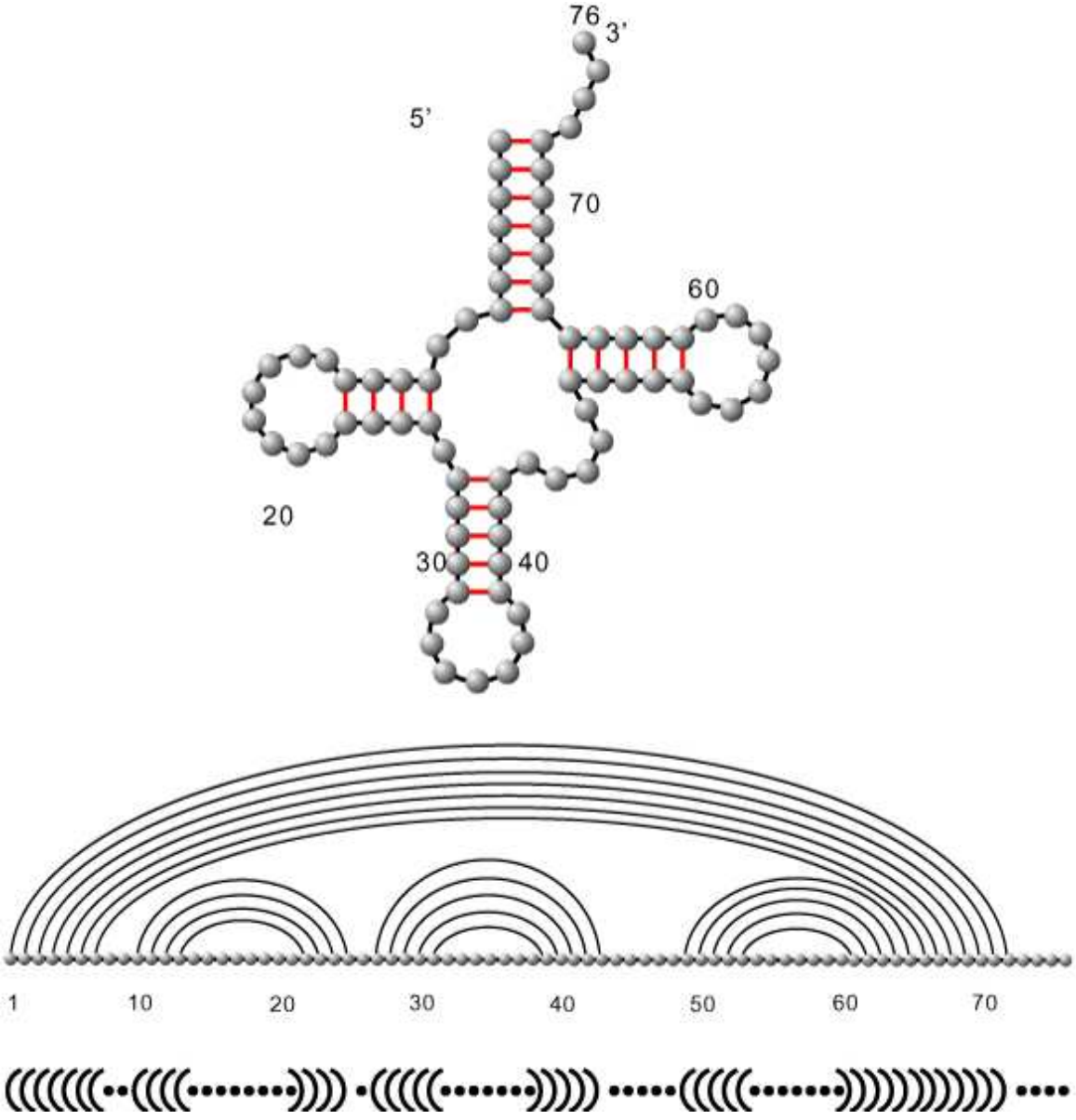
secondary structure



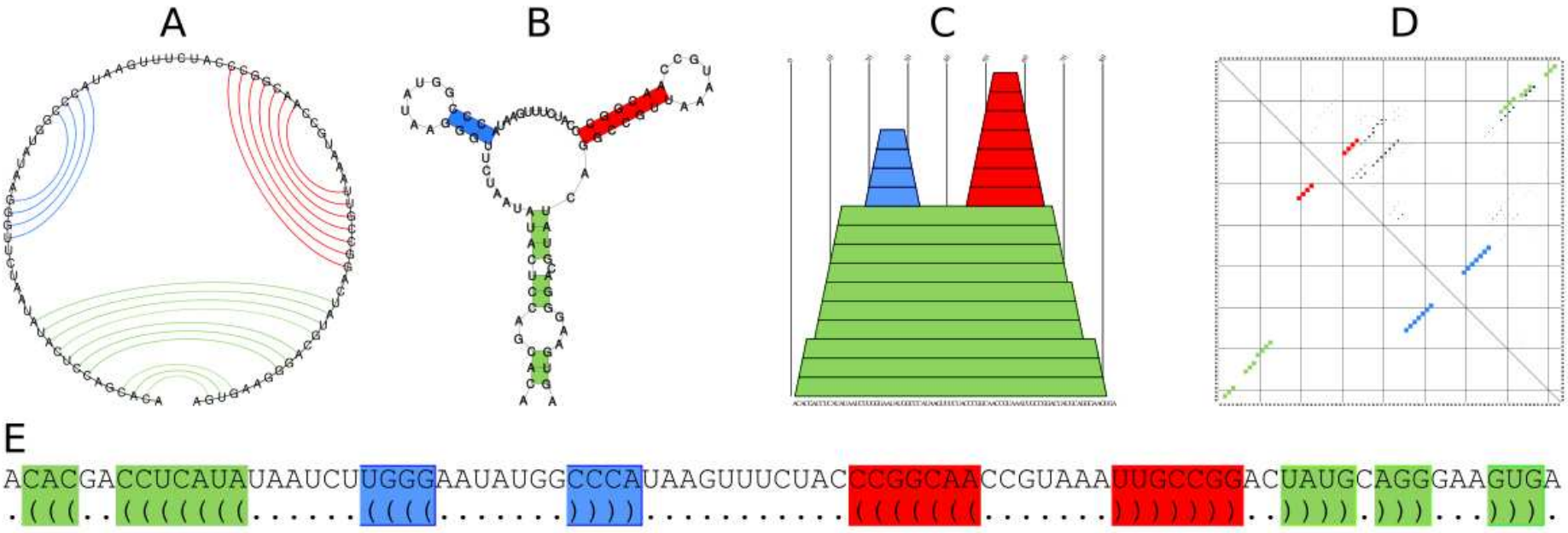
tertiary structure



RNA Secondary Structure Representation



RNA Secondary Structure Representation



A Circular Plot

B "Squiggle" Plot

C Mountain Plot

D Dot Plot

E Bracket Notation

RNA Secondary Structure Prediction

RNA folding: hierarchical process in which secondary structure is broadly considered as sufficient approximation assessing the most relevant characteristics of an RNA molecule

Prediction w.r.t. what? We need some optimization criterion. e.g.

- Number of Basepairs
- Min. Free Energy
- Mutual Information Content

Set of all feasible second.structures $\mathbb{S}(x)$ of an RNA sequence x is called *folding space*. Structure prediction means to select the "most-likely" structure from elements of $\mathbb{S}(x)$.

Counting Secondary Structures

Theorem

Let $S(n)$ denote the number of secondary structures of size n and $\theta = 1$. Then $S(0) = 0$, $S(1) = 1$ and for $n \geq 1$,

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

and

$$S(n) \geq 2^{n-2}.$$

Theorem

Let $S(n,k)$ denote the number of secondary structures of size n that contain exactly k basepairs ($\theta = 1$). Set $S(n,0) = 1$ for all n and $S(n,k) = 0$ for $k \geq \frac{n}{2}$. Then for $n \geq 2$,

$$S(n+1,k+1) = S(n,k+1) + \sum_{j=1}^{n-1} \left[\sum_{i=0}^{k} S(j-1,i)S(n-j,k-i) \right].$$

Corollary

$$S(n) = \textstyle\sum_{k=0}^{\lfloor n/2 \rfloor} S(n,k).$$

Realizability

A sequence $s \in \mathbb{A}^n$ *realizes* or *is compatible with* a secondary structure S of length n if for any $(i,j) \in S$ it holds that $s_i, s_j \in \mathbb{B}$.

- *dependency graph or shape graph* $G(S_1, \ldots, S_k)$

QUESTION:

On what conditions is it possible to find *one* sequence, which is realizing *all* sec.str. $S_1, \ldots, S_k$ of the same size?

equivalent to:

What properties have to be fulfilled in $G(S_1, \ldots, S_k)$, s.t. there exists a single sequence, which is realizing all $S_1, \ldots, S_k$?

Intersection Theorem

$C(S)$ denotes the set of all sequences that realize Secondary Structure S.

Intersection Theorem [Reidys et al. 1995]
For any two secondary structures S_1 and S_2 of same size holds: $C(S_1) \cap C(S_2) \neq \emptyset$.

Generalized Intersection Theorem [Flamm et al. 2001]
$\bigcap_{i=1}^{k} C(S_i) \neq \emptyset \;\Leftrightarrow\; G(S_1, \ldots, S_k)$ is bipartite.

RNA folding

Aim: For given sequence $s \in \mathbb{A}^n$ find "most-likely" or "optimal" structure.
"optimal" := e.g. max. number of basepairs, min free energy,
mutual information content (for more than one sequence) ...

Inferring Structure by Comparative Sequence Analysis

RNA sequence evolution is constrained by structure ($\rightarrow$ function)

"homologous" RNA's = RNA's that share a "common" structure

There are exampled of different RNA's that have a common structure, but almost no sequence similarity (conserved structure)
(e.g. tRNA structure conserved across species)

Drastic changes in sequences can often be tolerated as long compensatory mutations maintain base pairings.

Inferring Structure by Comparative Sequence Analysis

The key idea is to identify the interactions (i.e., the Watson-Crick correlated positions that result from compensatory mutations) in a multiple alignment, e.g.:

seq1	G	C	C	U	U	C	G	G	G	C
seq2	G	A	C	U	U	C	G	G	U	C
seq3	G	G	C	U	U	C	G	G	C	C

The amount of correlation of two columns of a multiple alignment 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* ?"

Inferring Structure by Comparative Sequence Analysis

A method used to locate covariant positions in a multiple sequence alignment is the mutual information content of two columns (=measure of the variables' mutual dependence).

$f_i(X)$ is rel. frequency of base $X \in \{A, C, G, U\}$ in i-th column of $\mathscr{A}$.

$f_{ij}(XY)$ is rel. frequency of simultaneously finding $X \in \{A, C, G, U\}$ in i-th column of $\mathscr{A}$ and $Y \in \{A, C, G, U\}$ in j-th column of $\mathscr{A}$

If the base frequencies of any two columns i and j are independent of each other then,

$$\frac{f_{ij}(XY)}{f_i(X) f_j(Y)} \approx 1$$

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

Inferring Structure by Comparative Sequence Analysis

Given alignment $\mathscr{A}$ of RNA sequences.
Mutual Information Score[1] is defined as

$$M_{i,j} = \sum_{XY} f_{ij}(XY) \log_2 \frac{f_{ij}(XY)}{f_i(X) f_j(Y)},$$

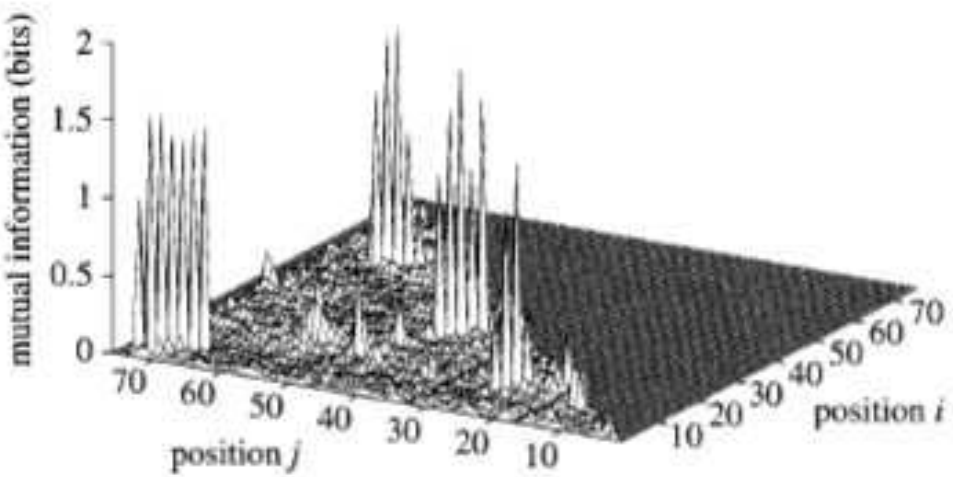
with: $f_i(X)$ is rel. frequency of base $X \in \{A, C, G, U\}$ in i-th column of $\mathscr{A}$
and $f_{ij}(XY)$ is rel. frequency of simultaneously finding $X \in \{A, C, G, U\}$ in i-th column of $\mathscr{A}$ and $Y \in \{A, C, G, U\}$ in j-th column of $\mathscr{A}$

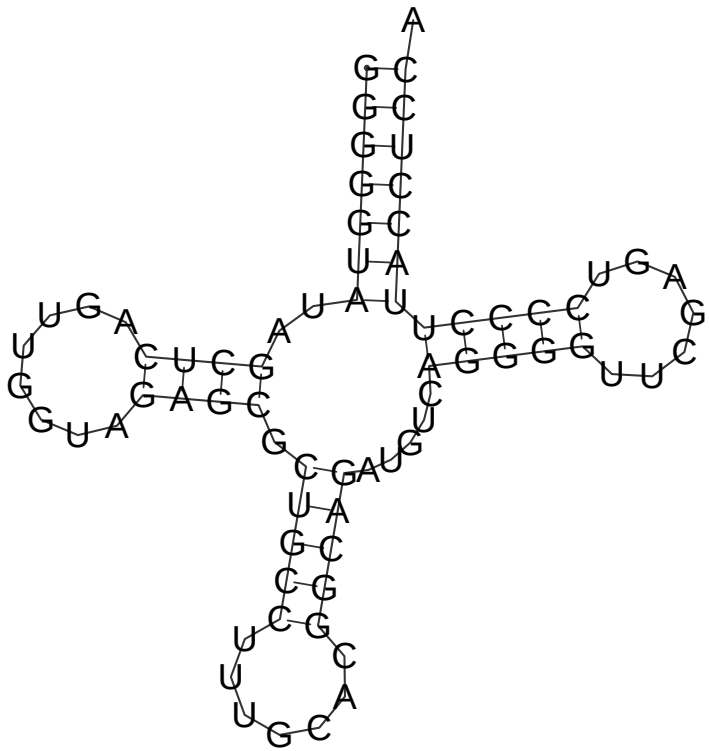
$$M_{i,j} \in [0, 2]$$

Intuitively, $M_{A,B}$ measures the information that A and B share: For example, if variables A and B are independent ($f(AB) = f(A)f(B) \Rightarrow \log(1) = 0$), then knowing A does not give any information about B and vice versa, so $M_{A,B} = 0$. At the other extreme, if A is a deterministic function of B (and vice versa) then all information conveyed by A is shared with B: knowing A determines the value of B and vice versa.

[1] goes back to Shannon information theory and entropy

Inferring Structure by Comparative Sequence Analysis[2]





[2]Book: Biological Sequence Analysis, Durbin et al.

Inferring Structure by Comparative Sequence Analysis

Exmpl.

	1	2	3	4	5	6
seq1	C	G	C	G	A	U
seq2	C	G	G	C	C	G
seq3	C	G	C	G	G	C
seq4	C	G	G	C	U	A

$M_{3,4} = 1$
$M_{5,6} = 2$
$M_{1,2} = 0$ although in all cases there can be bp

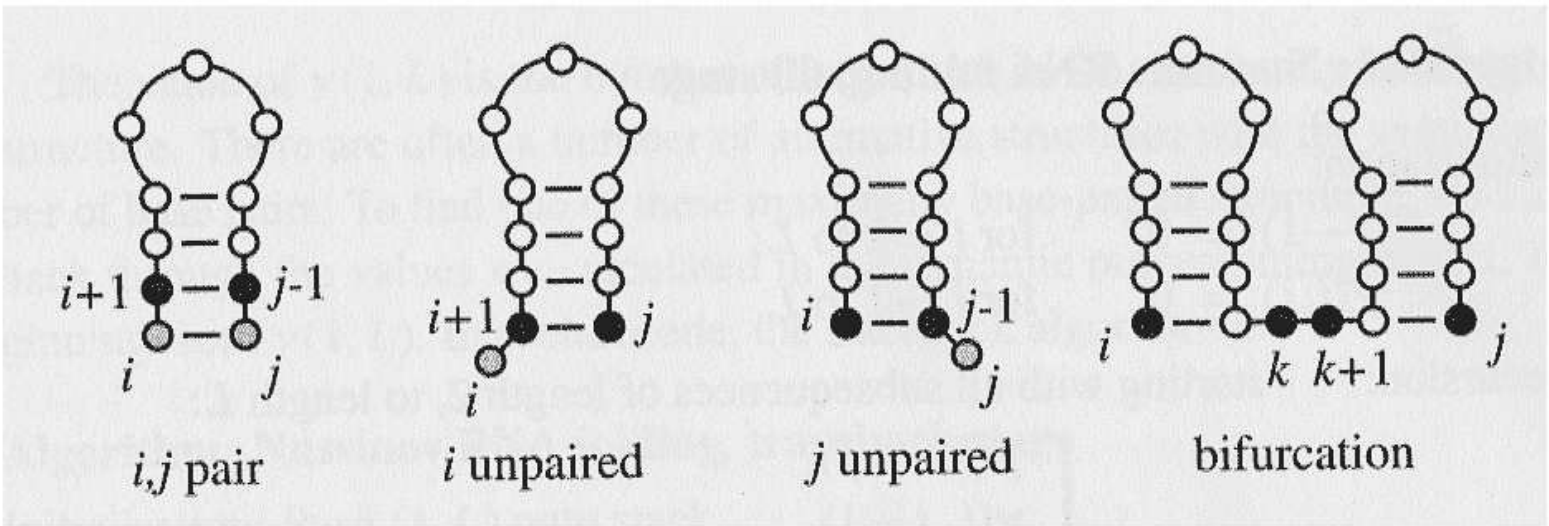
Problems:
One need large datasets of highly diverse and homologous (how do I know this?) sequences.

RNA folding - Maximize Number of Basepairs - Nussinov Algorithms

RNA folding: hierarchical process in which secondary structure is broadly considered as sufficient approximation assessing the most relevant characteristics of an RNA molecule.

Directly compute structure for a given sequence.

Observation: Best substructure $[i, j]$ can be obtained from substructures of $[i, j]$ - 4 cases:



Nussinov Algorithm - Basepair Maximization

1: **INPUT:** $s = s_1 \dots s_n \in \{A, C, G, U\}^n$
2: **for** $i, j = 1, \dots, n$ **do**
3: $E_{i,i-1} = E_{i,i} = 0$, $\delta_{ij} = \begin{cases} 1 & \text{, if } s_i s_j \in \mathbb{B} \\ 0 & \text{, else} \end{cases}$
4: **end for**
5: **for** $L = 1, \dots, n-1$ **do**
6: **for** $i = 1, \dots, n-L$ **do**
7: $j = i + L$
8: $E_{i,j} = \max \begin{cases} E_{i+1,j} \\ E_{i,j-1} \\ E_{i+1,j-1} + \delta_{ij} \\ \max_{i<k<j} \{E_{i,k} + E_{k+1,j}\} \end{cases}$
9: **end for**
10: **end for**
11: print "Max Nr Basepairs:" $E_{1,n}$

Nussinov Algorithm - Basepair Maximization

Given: $s = GGGAAAUCC$
Init: $E_{i,i-1} = 0$ and $E_{i,i} = 0$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**								
2	**G**	**0**	**0**							
3	**G**		**0**	**0**						
4	**A**			**0**	**0**					
5	**A**				**0**	**0**				
6	**A**					**0**	**0**			
7	**U**						**0**	**0**		
8	**C**							**0**	**0**	
9	**C**								**0**	**0**

Nussinov Algorithm - Basepair Maximization

$$E_{i,j} = \max \begin{cases} E_{i+1,j} \\ E_{i,j-1} \\ E_{i+1,j-1} + \delta_{ij} \\ \max_{i<k<j}\{E_{i,k} + E_{k+1,j}\} \end{cases}$$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**							
2	**G**	**0**	**0**	**0**						
3	**G**		**0**	**0**	**0**					
4	**A**			**0**	**0**	**0**				
5	**A**				**0**	**0**	**0**			
6	**A**					**0**	**0**	**1**		
7	**U**						**0**	**0**	**0**	
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Nussinov Algorithm - Basepair Maximization

$$E_{i,j} = \max \begin{cases} E_{i+1,j} \\ E_{i,j-1} \\ E_{i+1,j-1} + \delta_{ij} \\ \max_{i<k<j}\{E_{i,k} + E_{k+1,j}\} \end{cases}$$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**						
2	**G**	**0**	**0**	**0**	**0**					
3	**G**		**0**	**0**	**0**	**0**				
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**		
6	**A**					**0**	**0**	**1**	**1**	
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Nussinov Algorithm - Basepair Maximization

$$E_{4,7} = \max \begin{cases} E_{5,7} \\ E_{4,6} \\ E_{5,6} + \delta_{4,7} \\ \max_{4<k<7}\{E_{4,k} + E_{k+1,7}\} \end{cases}$$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**			
5	**A**				**0**	**0**	**0**	**1**		
6	**A**					**0**	**0**	**1**	**1**	
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Nussinov Algorithm - Basepair Maximization

$$E_{4,7} = \max \begin{cases} E_{5,7} = 1 \\ E_{4,6} = 0 \\ E_{5,6} + \delta_{4,7} = 1 \\ \max_{4<k<7}\{E_{4,k} + E_{k+1,7}\} = 1 \end{cases}$$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**					
2	**G**	**0**	**0**	**0**	**0**	**0**				
3	**G**		**0**	**0**	**0**	**0**	**0**			
4	**A**			**0**	**0**	**0**	**0**	**1**		
5	**A**				**0**	**0**	**0**	**1**		
6	**A**					**0**	**0**	**1**	**1**	
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Nussinov Algorithm - Basepair Maximization

$$E_{i,j} = \max \begin{cases} E_{i+1,j} \\ E_{i,j-1} \\ E_{i+1,j-1} + \delta_{ij} \\ \max_{i<k<j-1}\{E_{i,k} + E_{k+1,j}\} \end{cases}$$

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
2	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
3	**G**		**0**	**0**	**0**	**0**	**0**	**1**	**2**	**2**
4	**A**			**0**	**0**	**0**	**0**	**1**	**1**	**1**
5	**A**				**0**	**0**	**0**	**1**	**1**	**1**
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Traceback Nussinov Algorithm - Basepair Maximization

```
Traceback(matrix E, i = 1, j = n)
if i < j then
    if E_{i,j} = E_{i+1,j} then                          // i unpaired
        Traceback(E, i+1, j)
    else if E_{i,j} = E_{i,j-1} then                     // j unpaired
        Traceback(E, i, j-1)
    else if E_{i,j} = E_{i+1,j-1} + δ_{i,j} then         // i,j pair
        print "basepair (i,j)"
        Traceback(E, i+1, j-1)
    else for k = i+1,...,j-1 do
        if E_{i,j} = E_{i,k} + E_{k+1,j} then            // combined structures
            Traceback(E, i, k)
            Traceback(E, k+1, j)
            break
```

Nussinov Algorithm - Basepair Maximization

Traceback (similar as in Needleman-Wunsch Algorithm):

		1	2	3	4	5	6	7	8	9
		G	**G**	**G**	**A**	**A**	**A**	**U**	**C**	**C**
1	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
2	**G**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**3**
3	**G**		**0**	**0**	**0**	**0**	**0**	**1**	**2**	**2**
4	**A**			**0**	**0**	**0**	**0**	**1**	**1**	**1**
5	**A**				**0**	**0**	**0**	**1**	**1**	**1**
6	**A**					**0**	**0**	**1**	**1**	**1**
7	**U**						**0**	**0**	**0**	**0**
8	**C**							**0**	**0**	**0**
9	**C**								**0**	**0**

Secondary Structure

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

Nussinov drawbacks

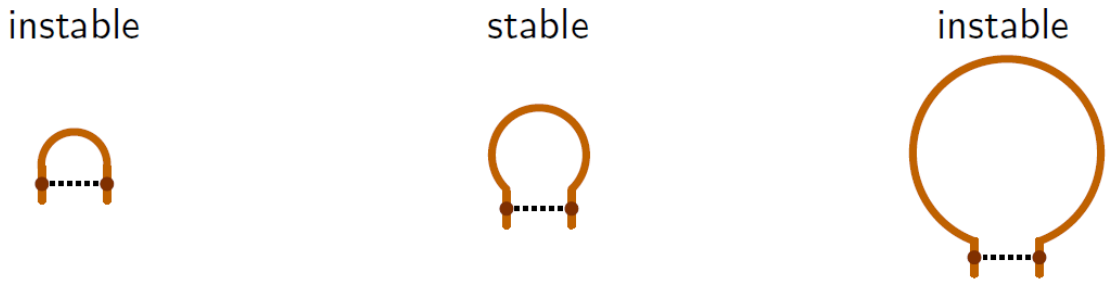
Maximizing the nr of bp does not lead to biological meaningful structures:

Stacking of bp not considered:

(((.))) stable

( ) ( ) . ( ) instable

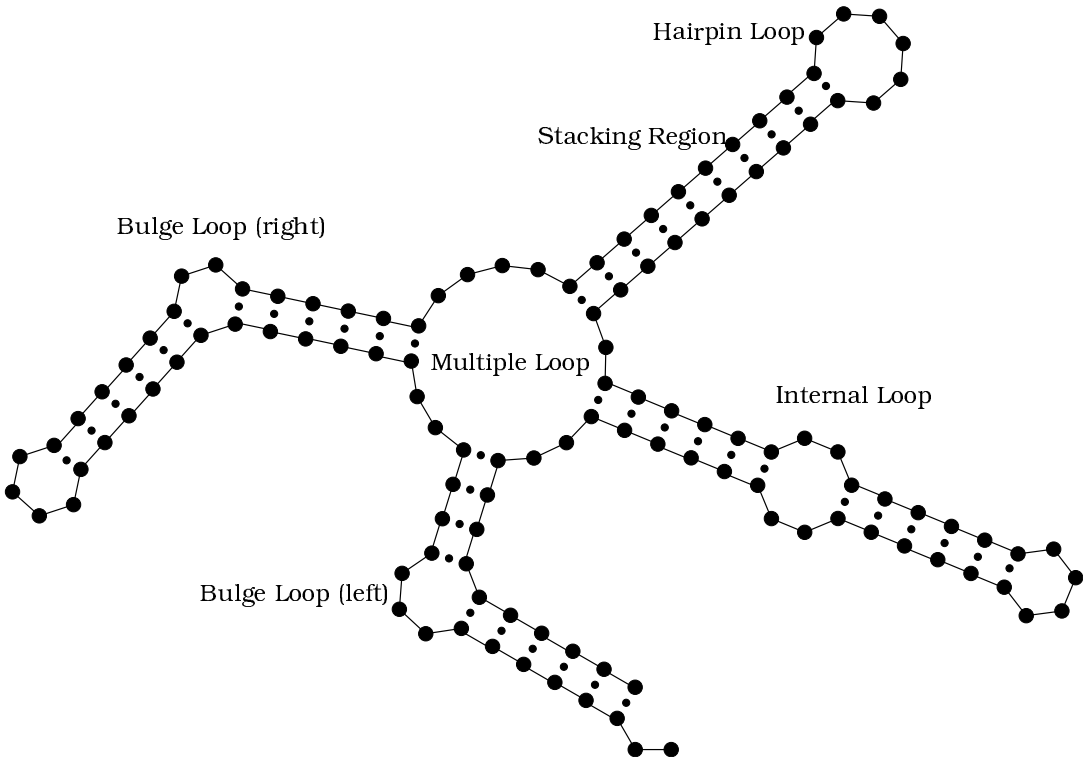
Size of intern loop not considered:



Nevertheless, although Nussinov-Alg is too simple to be accurate, it is stepping-stone for later algorithms

RNA MinFreeEnergy (MFE) Folding

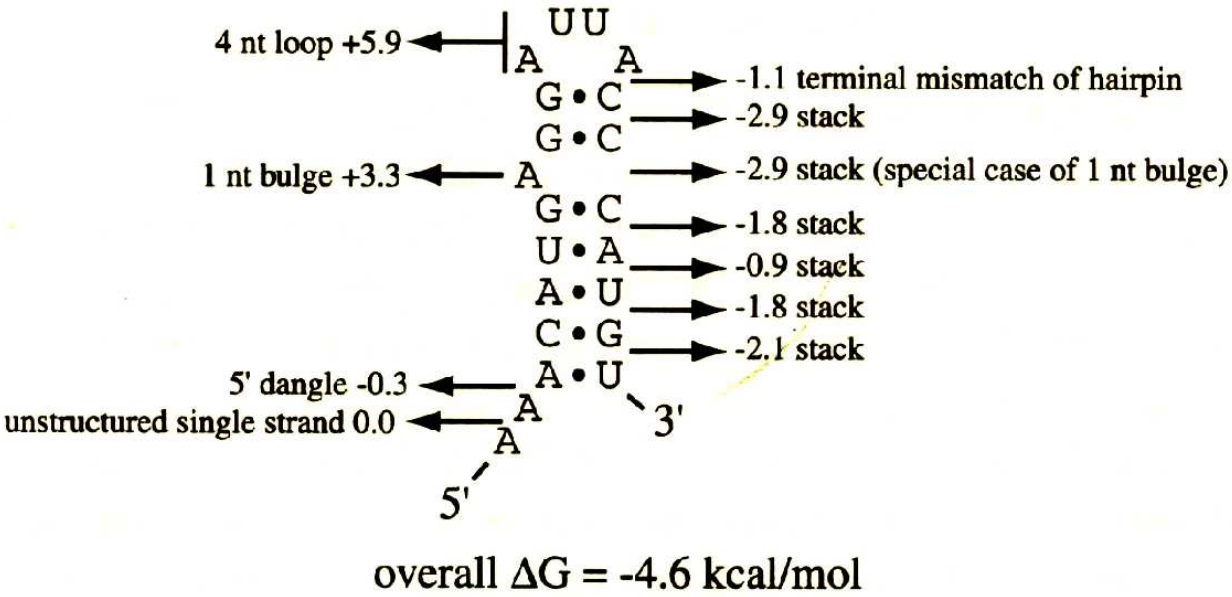
Define energy model for RNA that takes into account local energy contributions from loop and stacking regions.



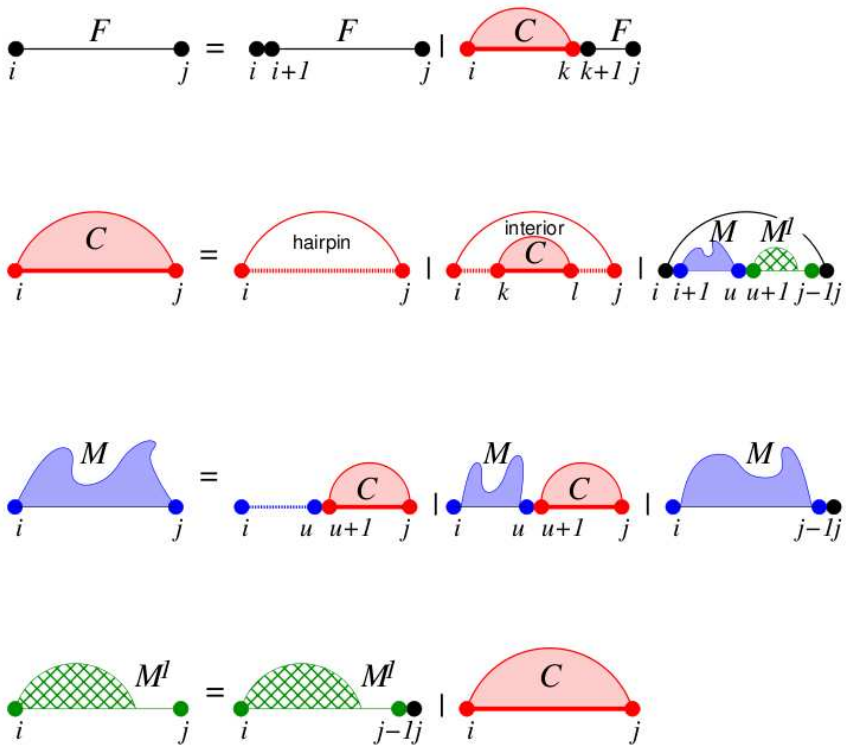
- More realistic: thermodynamics and statistical mechanics.
- Stability of an RNA sec.str. coincides with thermodynamic stability
- Quantified as the amount of free energy released/used by forming bp.

RNA MinFreeEnergy (MFE) Folding

Define energy model for RNA that takes into account local energy contributions contributions from loop and stacking regions.
The Turner rules are a set of experimentally determined parameters which allow us to predict the stability of RNA secondary structures.



MFE-folding via Loop Decomposition



$$F_{i,j} = \min \left\{ \begin{array}{l} F_{i+1,j}, \\ \min\limits_{i<k\leq j} C_{i,k} + F_{k+1,j} \end{array} \right\}$$

$$C_{i,j} = \min \left\{ \begin{array}{l} \mathcal{H}(i,j), \\ \min\limits_{i<k<l<j} C_{k,l} + \mathcal{I}(i,j;k,l), \\ \min\limits_{i<u<j} M_{i+1,u} + M^1_{u+1,j-1} + a \end{array} \right\}$$

$$M_{i,j} = \min \left\{ \begin{array}{l} \min\limits_{i<u<j} (u-i+1)c + C_{u+1,j} + b, \\ \min\limits_{i<u<j} M_{i,u} + C_{u+1,j} + b, \\ M_{i,j-1} + c \end{array} \right\}$$

$$M^1_{i,j} = \min \left\{ \begin{array}{l} M^1_{i,j-1} + c, \\ C_{i,j} + b \end{array} \right\}$$

First proposed by Zuker et al.

init: $F_{i,j} = 0$, $C_{i,j} = M_{i,j} = M^1_{i,j} = \infty$.

- $F_{1,n}$ stores the energy value of the thermodynamically most stable structure, its Minimum Free Energy (MFE).
- traceback structure

MFE-folding via loop decomposition: remarks

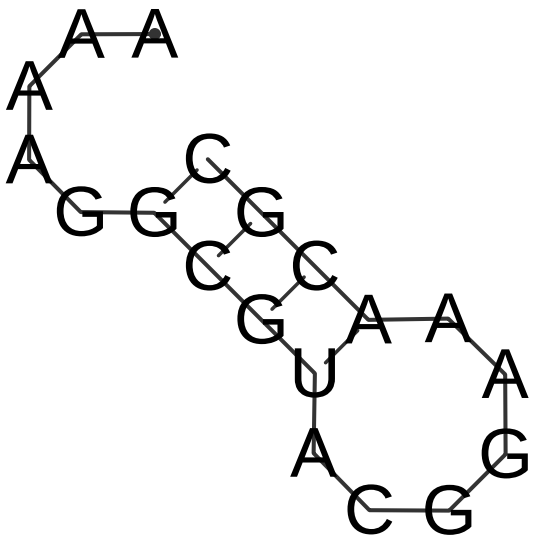
- $F_{i,j}$: free energy of the opt. sub-struct. on the sub-seq. $s_i \dots s_j$.
- $C_{i,j}$: free energy of the opt. sub-struct. on the sub-seq. $s_i \dots s_j$ given that i and j form a base pair.
- $M_{i,j}$: free energy of the opt. sub-struct. on the sub-seq. $s_i \dots s_j$ given that $s_i \dots s_j$ is part of a multi-loop and has at least one "component".
- $M^1_{i,j}$: free energy of the opt. sub-struct. on the sub-seq. $s_i \dots s_j$ given that $s_i \dots s_j$ is part of a multi-loop and has exactly one component which has the closing pair (i, h) for some h satisfying $i < h \leq j$.

RNA Fold

http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi

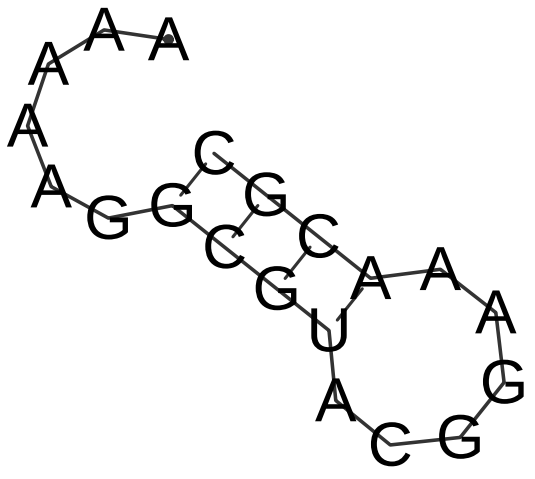
```
  RNAfold < trna.fa
>AF041468
GGGGGUAUAGCUCAGUUGGUAGAGCGCUGCCUUUGCACGGCAGAUGUCAGGGGUUCGAGUCCCCUUACCUCCA
(((((((..((((........)))).(((((.......)))))).....(((((.......)))))))))))).
-31.10 kcal/mol
```

Symmetric Difference



AAAAGGCGUACGGAAACGC

.....((((.....))))



AAAAAGGCGUACGGAAACGC

.....((((.....))))

$S_1 = \{(6,19),(7,18)(8,17)(9,16)\}$
$S_2 = \{(7,20),(8,19)(9,18)(10,17)\}$

$$|S_1 \Delta S_2| = 8$$

String Alignment

$S_1 =$ (((. . (((. . . .))))))

$S_2 =$ (((. . . . . .)))

String Alignment:

(((. . (((. . . .))))))

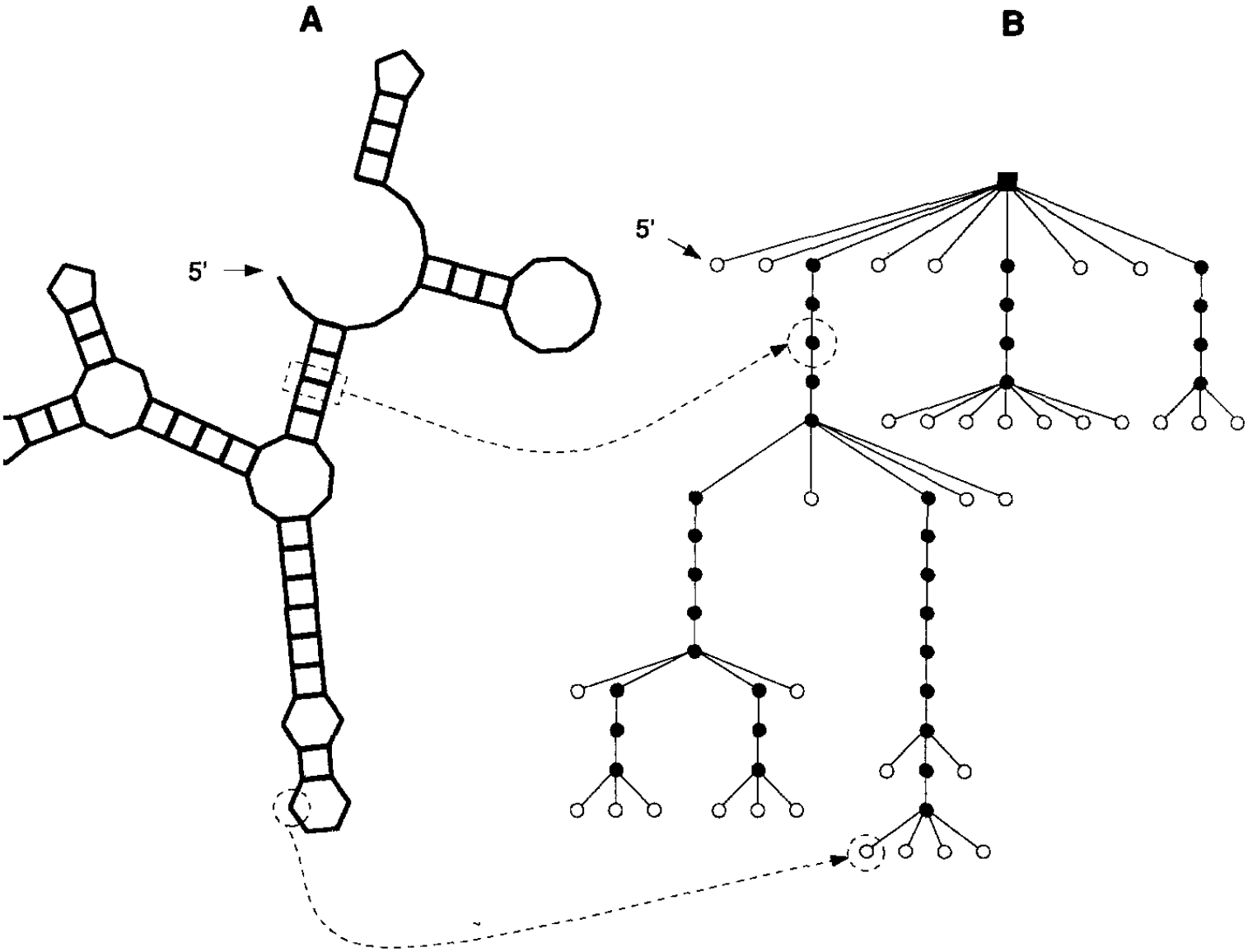
(((. . * * * . . . .))) * * *

"Structure" Alignment:

(((. . (((. . . .))))))

(((. . * * * . . . . * * *)))

Tree Edit Distance



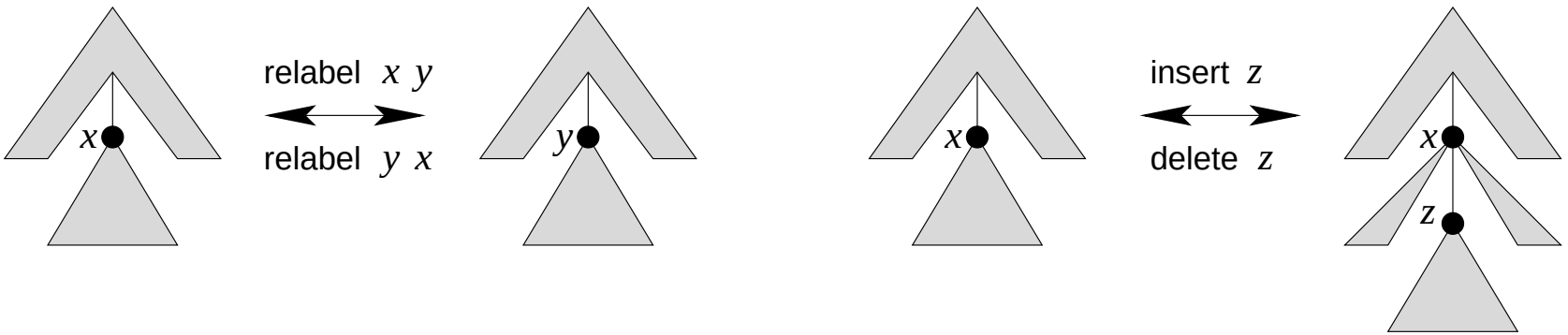
Tree Edit Distance

There are three basic edit operations with associated costs γ:

relabel $x \to y$ with $\gamma(x \to y)$

insert $\emptyset \to z$ with $\gamma(\emptyset \to z)$

delete $z \to \emptyset$ with $\gamma(z \to \emptyset)$



Tree Edit Distance

$\gamma(S)$, (ordered) edit map M , $\gamma(S) = \gamma(M)$

$$D(F_1, F_2) = \min \begin{cases} D(F_1 - v_1, F_2) + \gamma(v_1 \to \text{ø}), \\ D(F_1, F_2 - v_2) + \gamma(\text{ø} \to v_2), \\ D(T(v_1) - v_1, T(v_2) - v_2) + \\ D(F_1 \setminus T(v_1), F_2 \setminus T(v_2)) + \gamma(v_1 \to v_2) \end{cases}$$

where:

$$D(\emptyset, \emptyset) = 0$$

$$D(\emptyset, F_2) = D(\emptyset, F_2 - v_2) + \gamma(\text{ø} \to v_2)$$

$$D(F_1, \emptyset) = D(F_1 - v_1, \emptyset) + \gamma(v_1 \to \text{ø})$$