RNA Structures
From Sequence to 2D

Ivo Hofacker

Institute for Theoretical Chemistry, University of Vienna

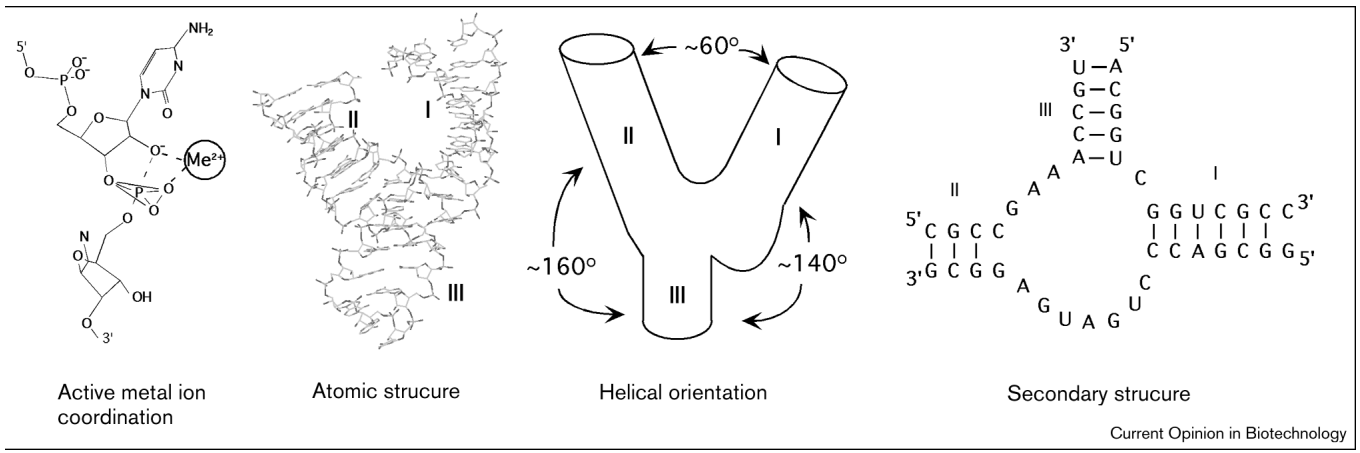
http://www.tbi.univie.ac.at/

AlgoSB 2025
Marseille, December 2025

What is RNA Bioinformatics?

Algorithms on sequences don't care whether it's RNA or protein, so what's special?

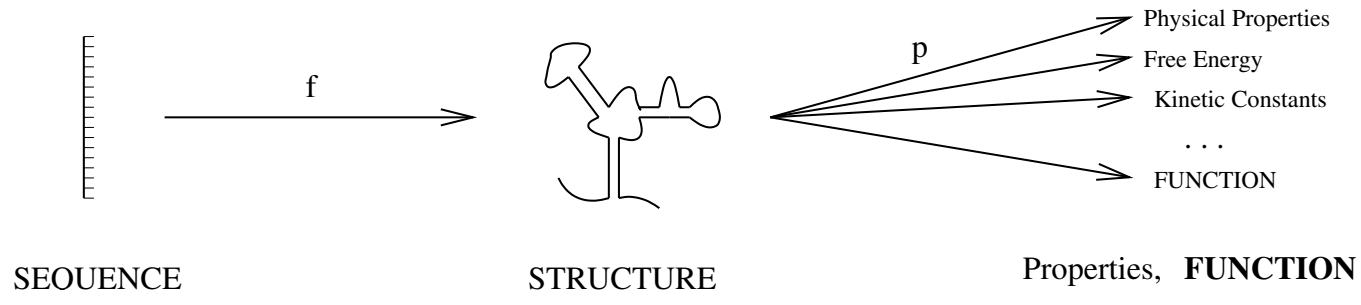
- In contrast to plain sequence analysis we're interested in *structure*
- When taking about *structural bioinformatics* most people think protein structure – RNA is different.
- Structure can be described at many levels
  Very useful for RNA work is usually the *secondary structure*



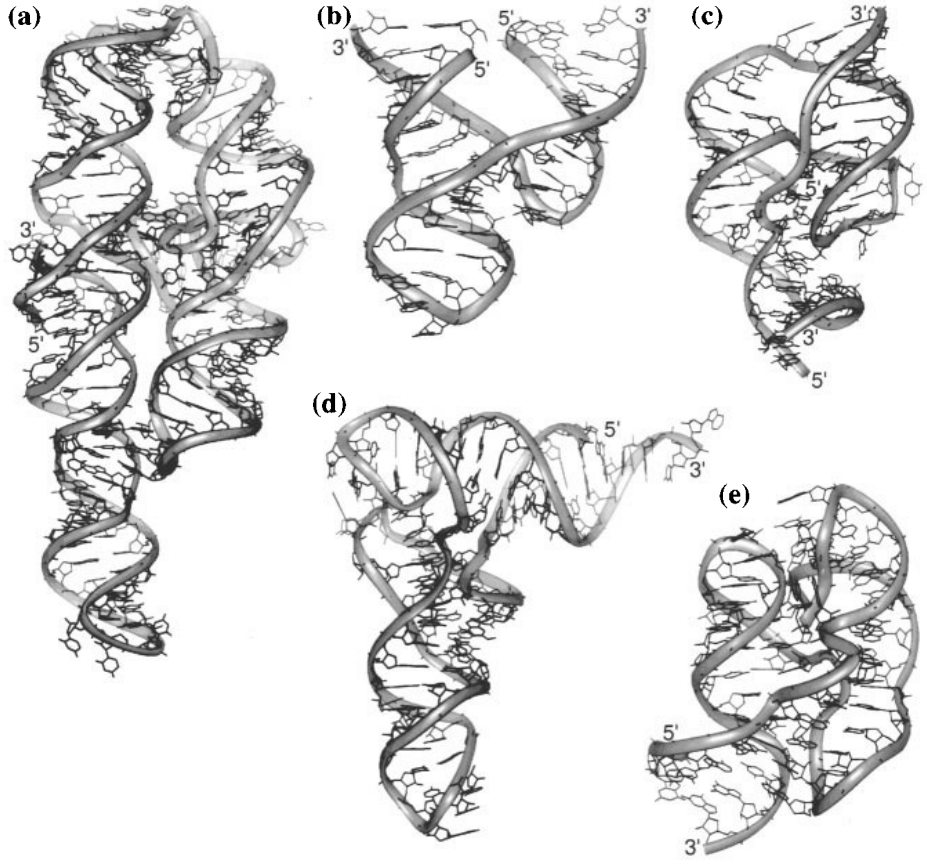
Why look at Structure?

- Basic paradigm of structural biology:
  Sequence $\rightarrow$ Structure $\rightarrow$ Function
- Understanding structure is a first step toward function
- Function is what we're ultimately interested in
  Sequences is what we have in plenty
- Structure should be conserved more strongly than sequence

Structure based methods can succeed where sequence analysis fails



Many non-coding RNAs are Structural RNAs



(a) Group I intron
(b) Hammerhead ribozyme
(c) HDV ribozyme
(d) Yeast $\mathrm{tRNA}^{\mathrm{phe}}$
(e) L1 domain of 23S rRNA

Hermann & Patel, JMB 294, 1999

All the "classical" ncRNAs depend on well-defined and evolutionary conserved **structure**

Known 3D Structures: Protein vs. RNA

Structures & residues in the PDB database over time

	$\leq$ 1980	1981-90	1991-2000	2001-10	2011-22	total	% total
Proteins	78	634	12121	43205	108677	164715	91.57
AA $\leq$ 2Å	5050	45236	1609401	11390238	28513777	41 563 702	99.78
RNA	2	23	306	1392	4488	6211	3.45
nts $\leq$2Å	0	0	270	5974	26921	34 165	0.08

- Much fewer RNA 3D structure than protein
- Even more pronounced for residues and high-res structures

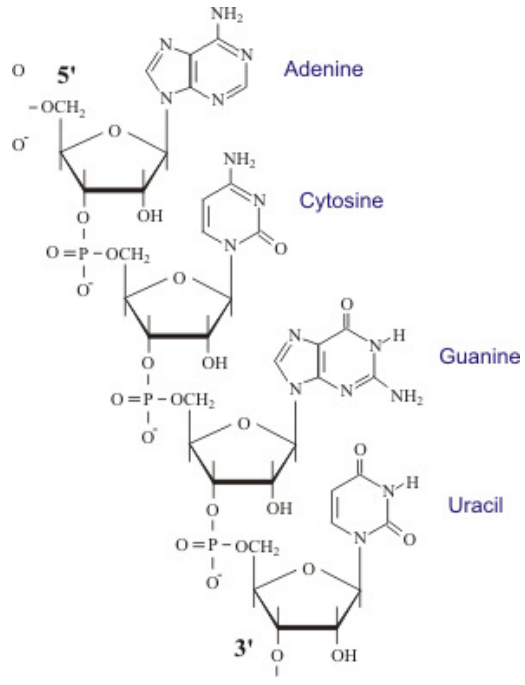
[0]Schneider et al, NAR, 2023

Basic Tasks

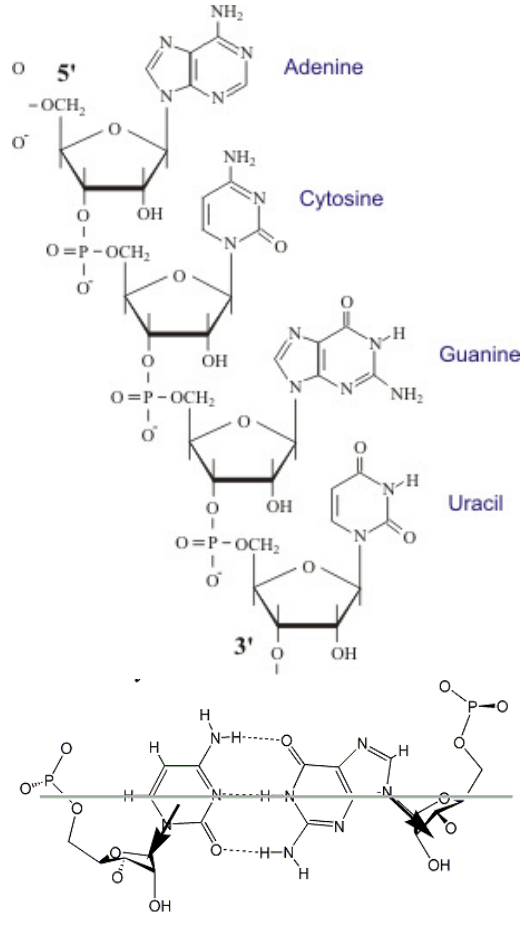
Mostly the same questions are the same as in sequence analysis
Here we want to base them on structure

- Structure prediction (sequence $\rightarrow$ structure)
  (single or multiple sequences, w/o pseudo-knots, tertiary structure)
- Classification:
  - Recognizing new members of a known RNA class (tRNAs, snoRNAs, ...)
  - Homology searches — Given a novel RNA find all the homologs where in Evolution did it first occur?
- *De novo* prediction
  Annotate all functional RNAs in genomes or transcriptomes
- Many specialized problems for particular RNA classes
  e.g. micro RNA target prediction

The RNA Molecule

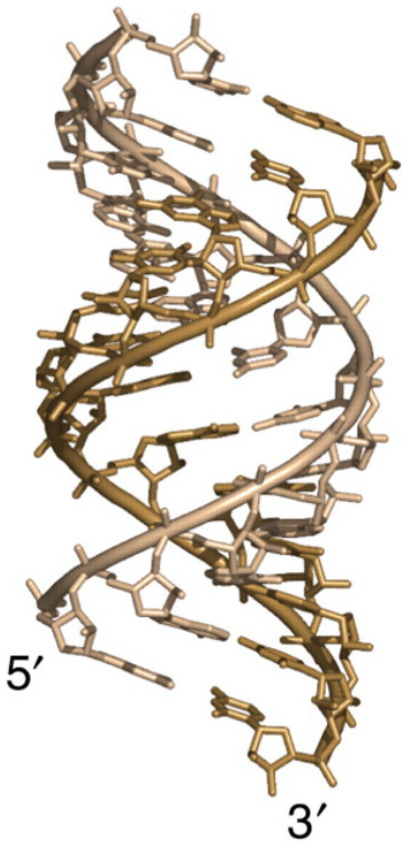


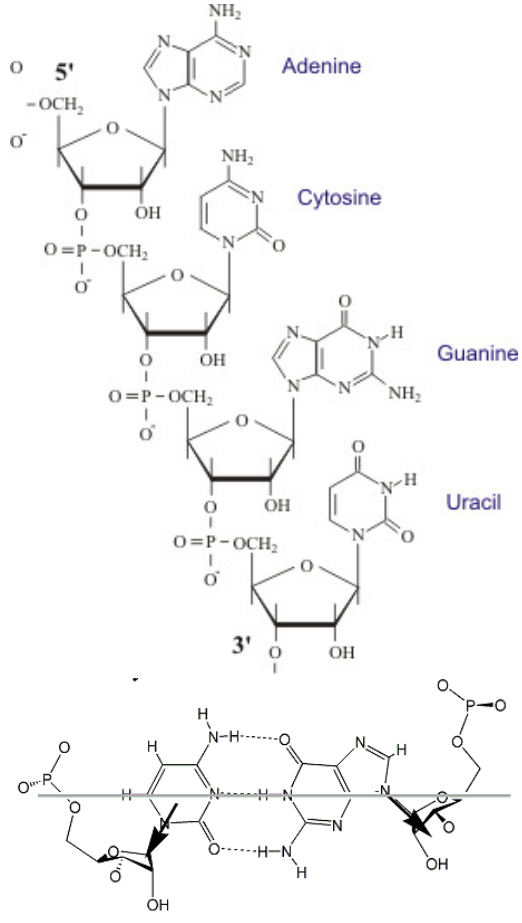
The RNA Molecule



Double helices formed by Watson-Crick AU, UA, GC, CG, GU, UG pairs.

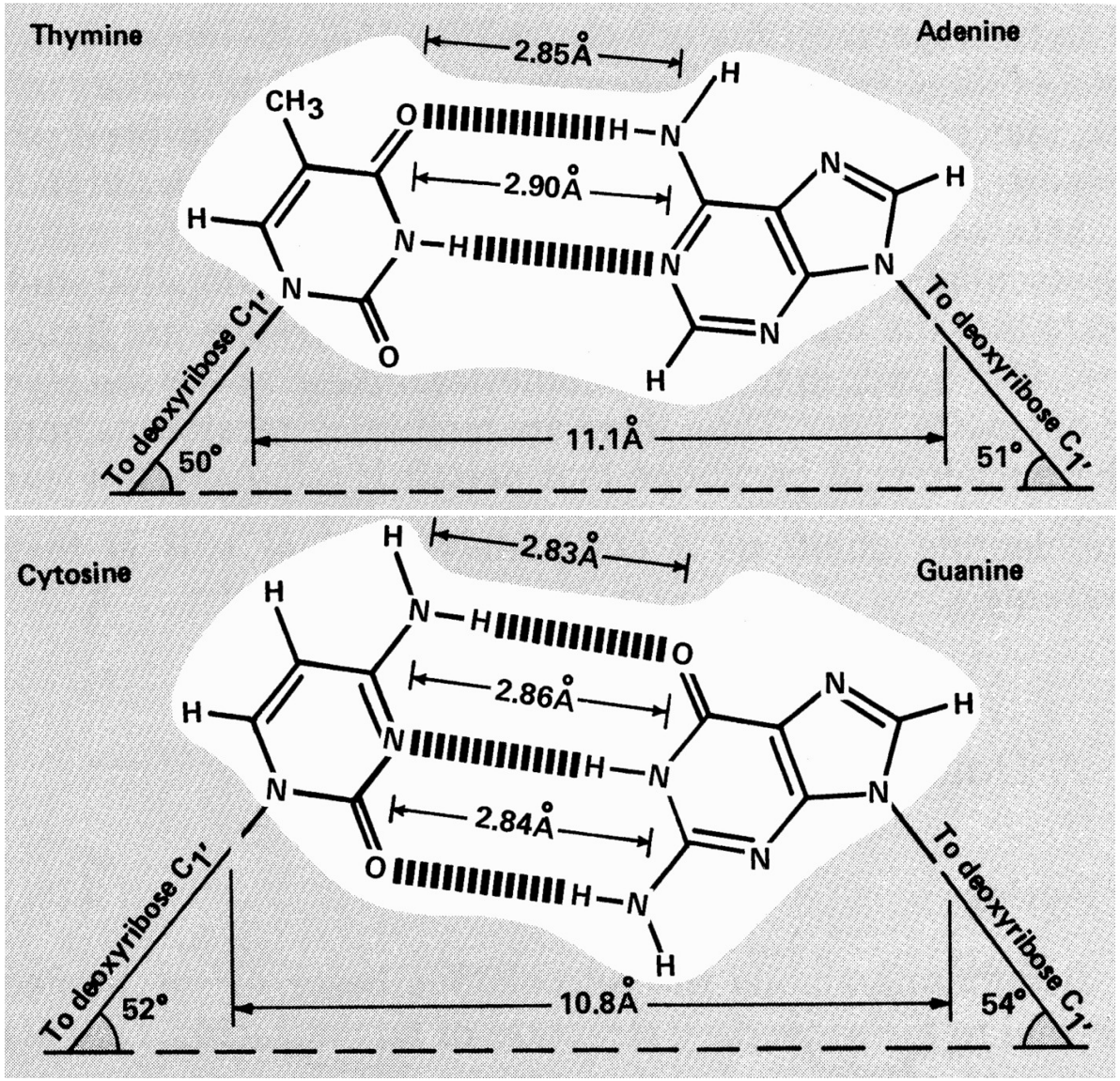
The RNA Molecule



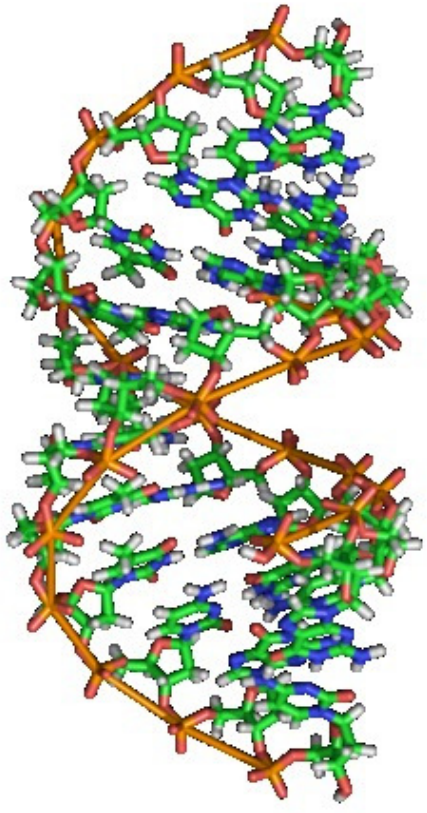


Double helices formed by Watson-Crick AU, UA, GC, CG, GU, UG pairs.

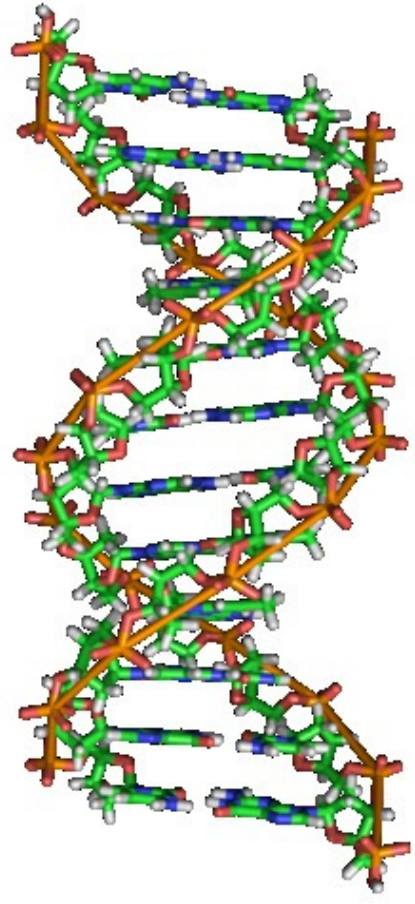
Isostericity of Watson-Crick Pairs



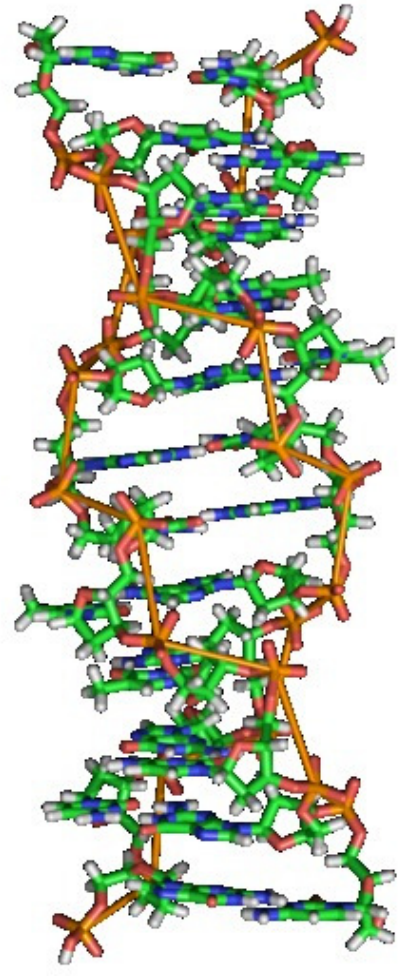
Double Helices



A-form



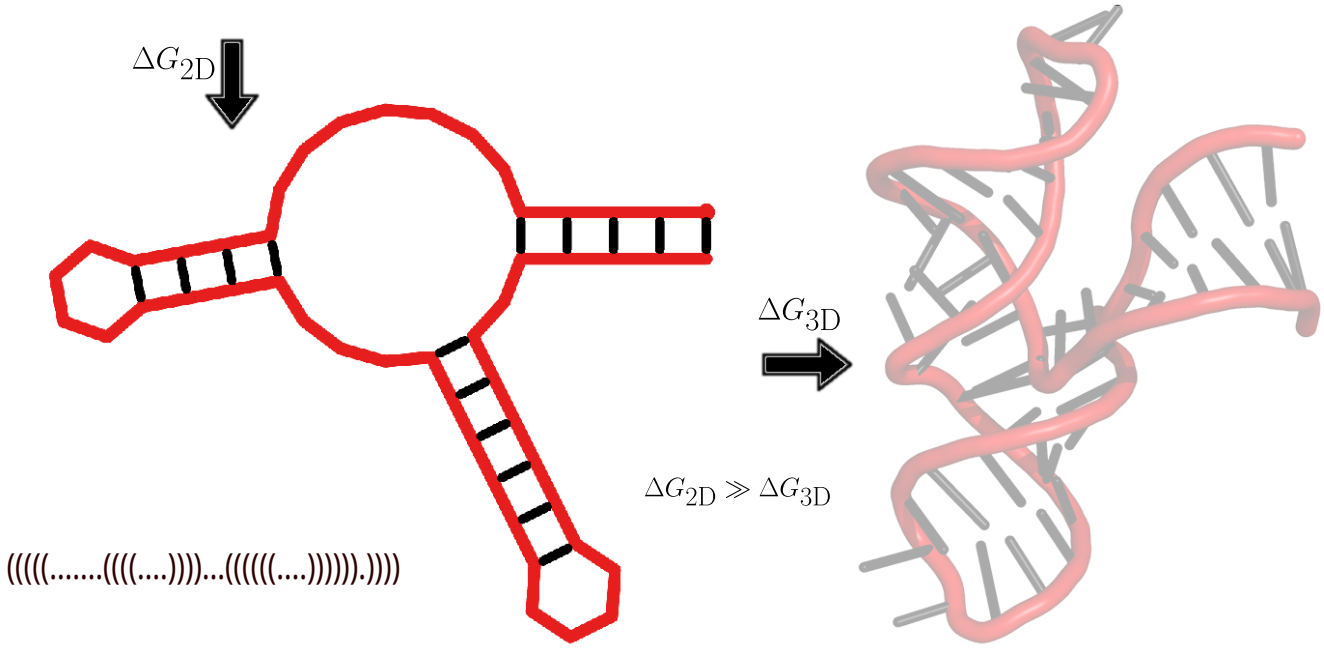
B-form



Z-form

The RNA Folding Problem

GCGCUCUGAUGAGGCCGCAAGGCCGAAACUGCCGCAAGGCAGUCAGCGC

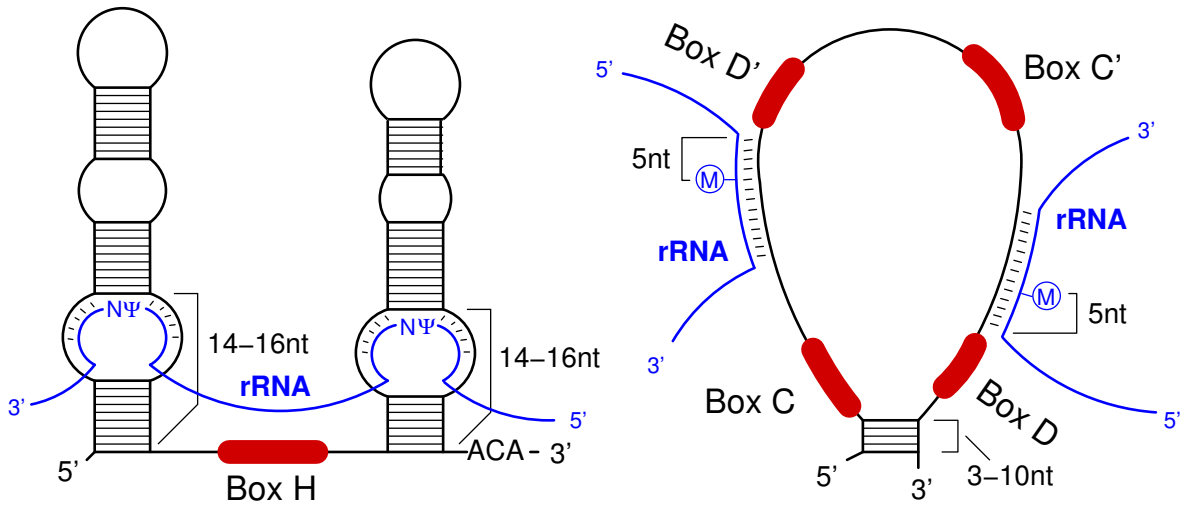


- Hierarchical folding: Secondary structure forms first then helices arrange to form tertiary structure
- Secondary structures cover most most of the folding energy
- Convenient and biologically useful description
- Computationally easy to handle
- Tertiary structure prediction needs knowledge of secondary structure

Example 1: snoRNAs

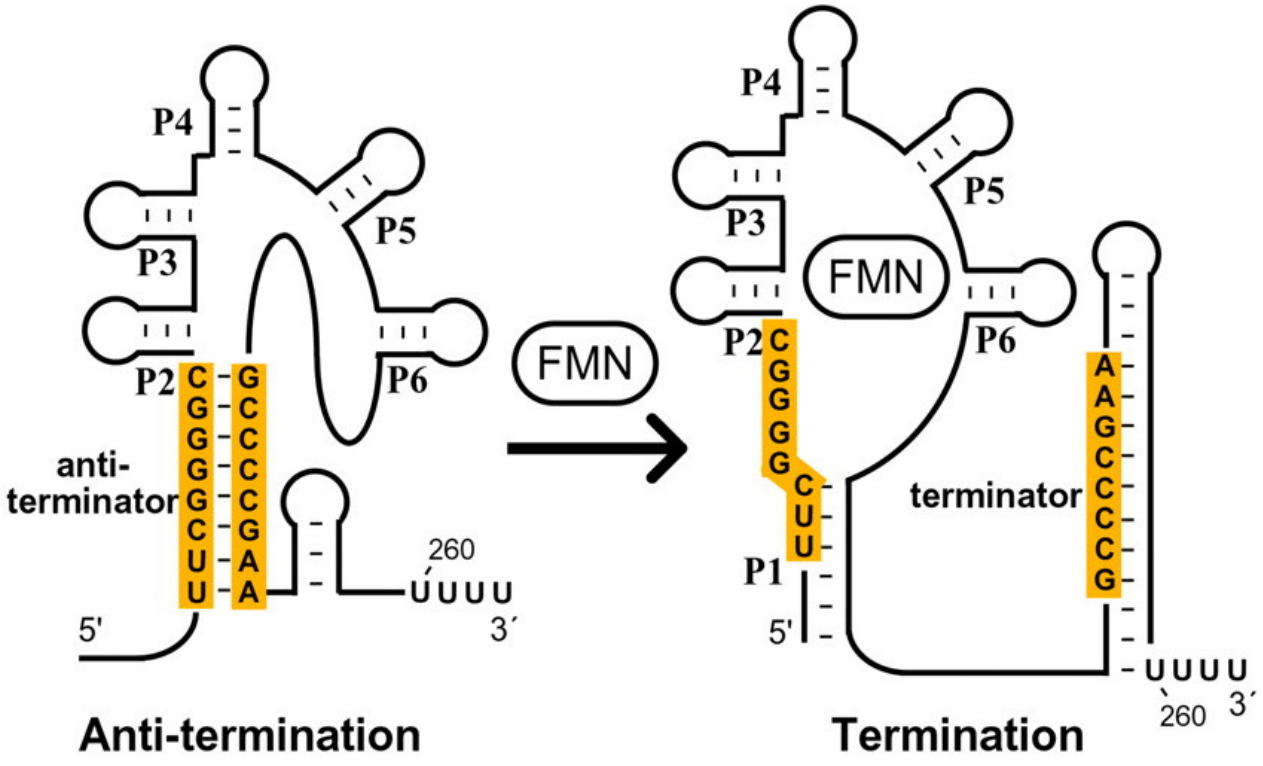
Small **n**ucle**o**lar RNAs are *trans*-acting ncRNAs

- snoRNAs guide chemical modifications of rRNAs, snRNAs, and some mRNAs.
- Characteristic secondary structure and sequence motifs
- Two tasks: How to recognize a snoRNA from its sequence?
  How to predict the targets of a given snoRNA?



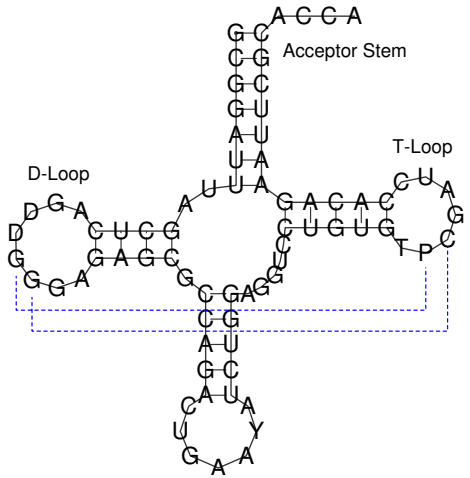
Example 2: The Flavin Mononucleotide Riboswitch

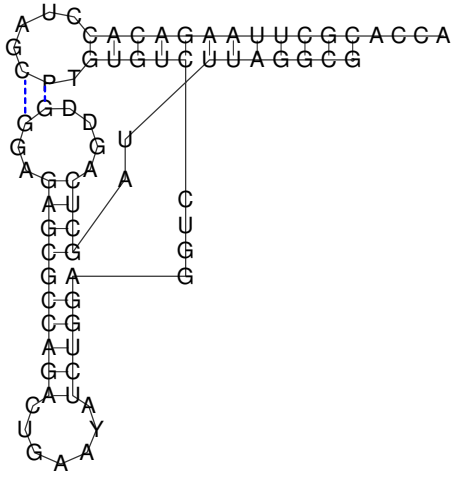
Example for a *cis-acting* RNA element
The FMN-binding riboswitch in the free and bound state

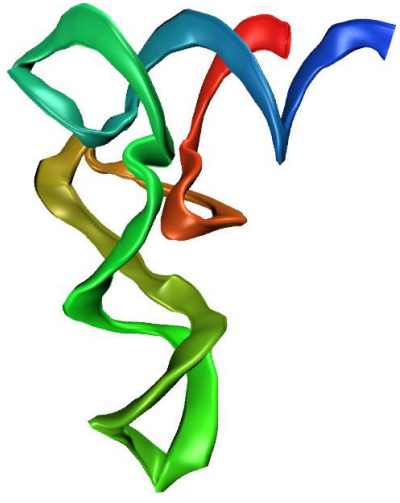


- How can we find novel riboswitches?
- Can we predict how a novel riboswitch works?
  Up-, or down-regulation; transcriptional or translational control?

Secondary Structure Definition







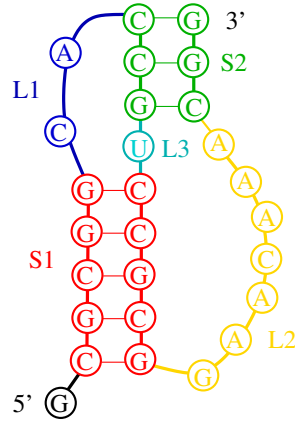
A *secondary structure* is a set of base pairs (i, j) on a sequence x, with

- Any nucleotide (sequence position) can form at most one pair
- No pseudo-knots: No pairs (i, j) and (k, l) with $i < k < j < l$
- If (i, j) is a pair then $x_i x_j \in \{\mathrm{GC}, \mathrm{CG}, \mathrm{AU}, \mathrm{UA}, \mathrm{GU}, \mathrm{UG}\}$
- If (i, j) is a base pair, then $j - i > 3$

Pseudo Knots

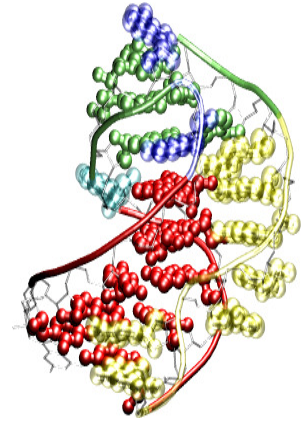
Excluding pseudo knots makes life easy, because

- It greatly simplifies the mathematical model
  $\Longrightarrow$ Simpler algorithms without pseudo knots
- Many pk-structures are sterically not feasible
- Energetics unknown, except for a few data on H-type pseudo-knots

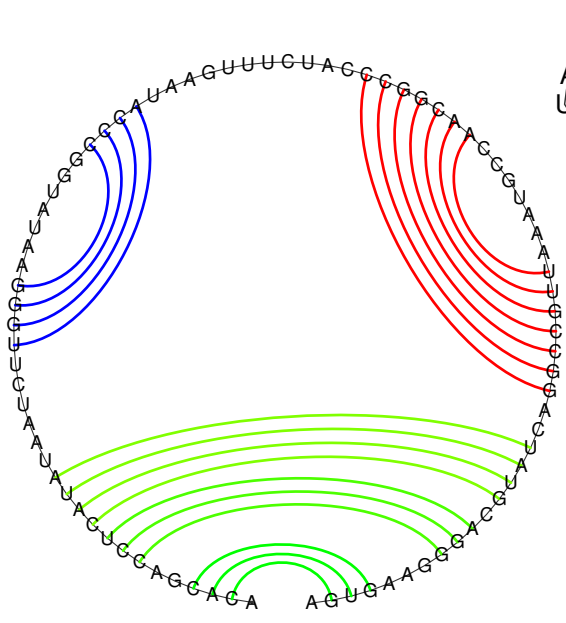


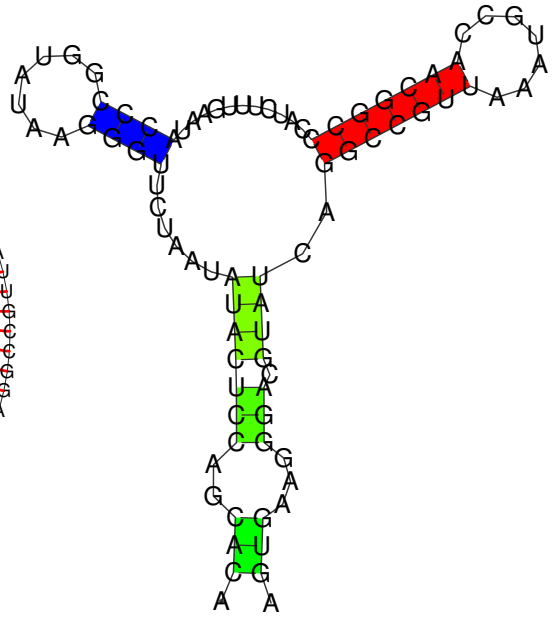
On the other hand

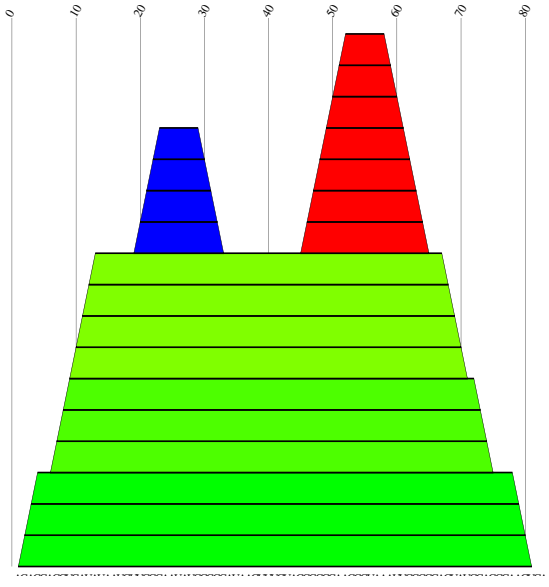
- Pseudo knots can have important function
- First step toward tertiary structure
- H-type knots are tractable with extra computational effort



Representation of Secondary Structures

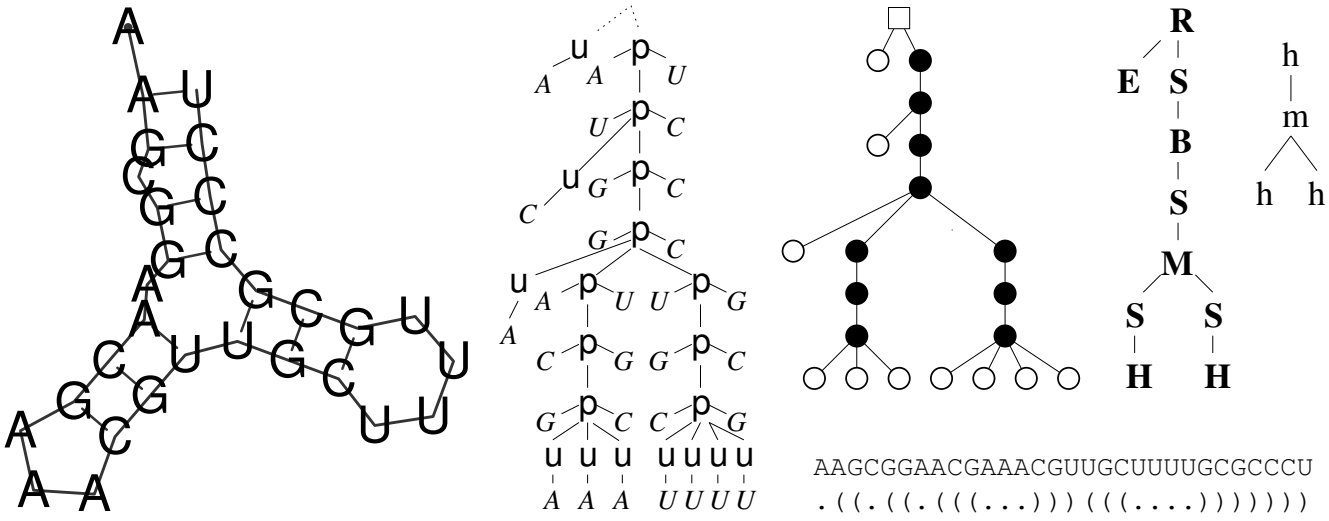






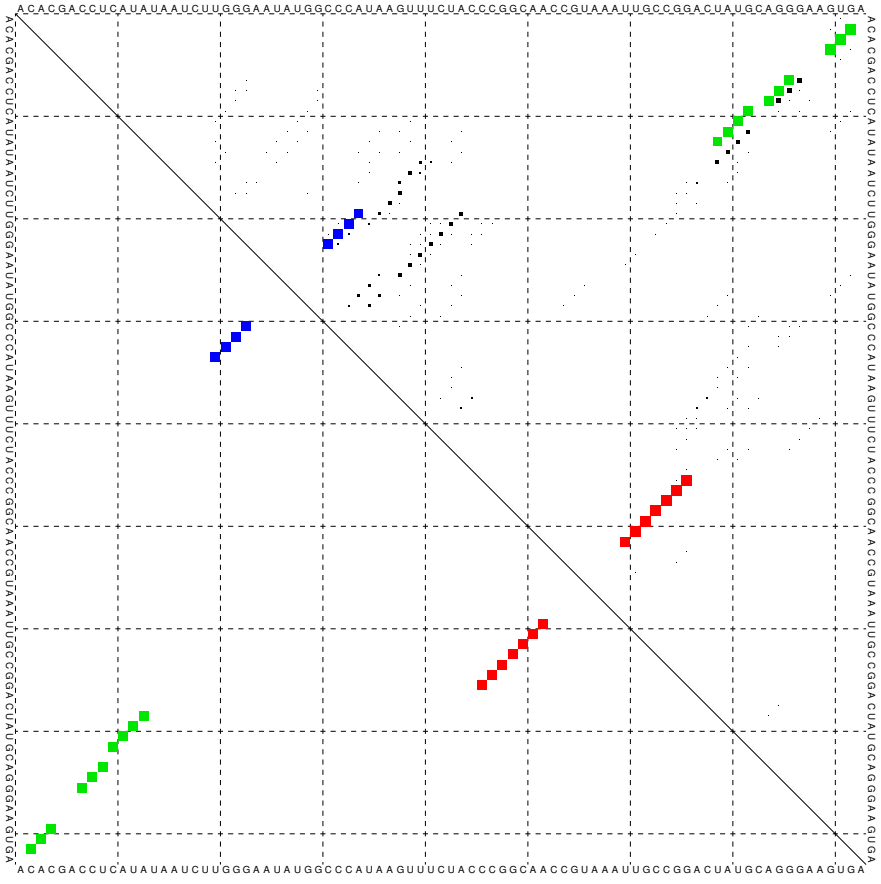
```
ACACGACCUCAUAUAAUCUUGGGAAUAUGGCCCAUAAGUUUCUACCCGGCAACCGUAAAUUGCCGGACUAUGCAGGGAAGUGA
.(((..(((((((.....(((().....)))).............(((((((.....)))))))..)))))..)))...)))).
```

RNA Secondary Structure as Trees



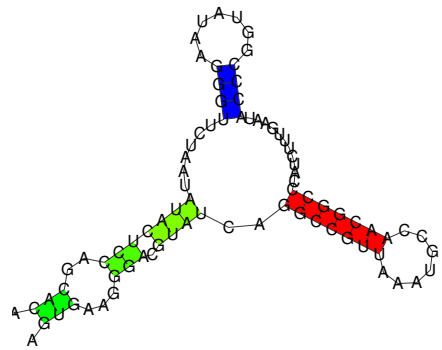
Many RNA algorithms are generalizations of well known *string* algorithms to *trees*

Representing Ensembles of Structures



Ensembles of structures (thermodynamic equilibrium) are best represented by base pair probabilities.

A pair (i, j) with probability p is represented by a square in row i and column j with area p.



Forces stabilizing biomolecular Structures

- Hydrogen bonds
- $\pi - \pi$ stacking
- Coulomb interactions
- Van der Waals Interactions

Forces stabilizing biomolecular Structures

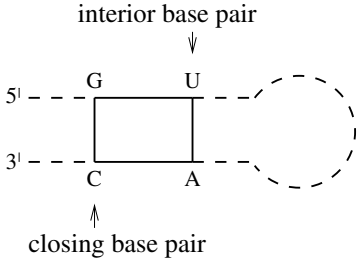
- Hydrogen bonds
- $\pi - \pi$ stacking
- Coulomb interactions
- Van der Waals Interactions
- Hydrophobic effect
- Conformational Entropy

Forces stabilizing biomolecular Structures

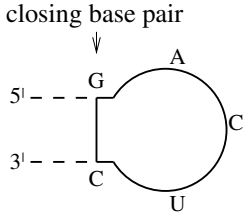
- Hydrogen bonds
- $\pi - \pi$ stacking $\leftarrow$ most important for RNA
- Coulomb interactions
- Van der Waals Interactions
- Hydrophobic effect $\leftarrow$ most important for protein
- Conformational Entropy

Loop Decomposition

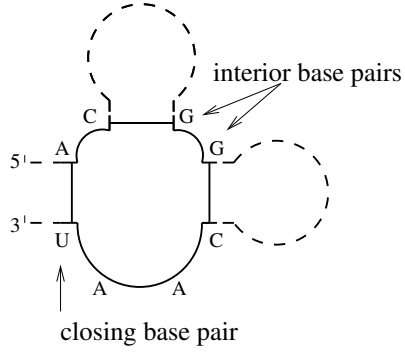
Secondary structures can be uniquely decomposed into loops.
Loops are the *faces* of the secondary structure graph.



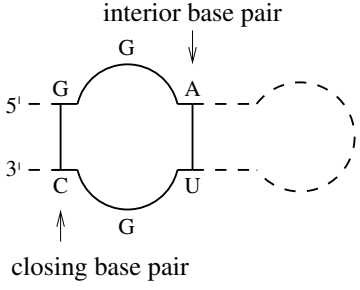
stacking pair



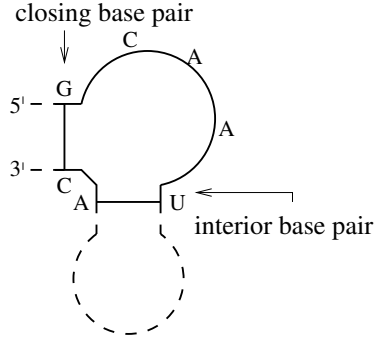
hairpin loop



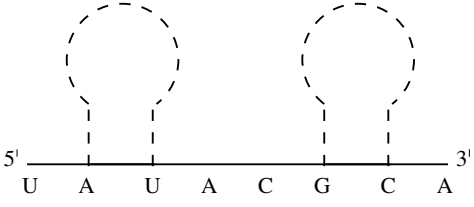
multi loop



interior loop



bulge

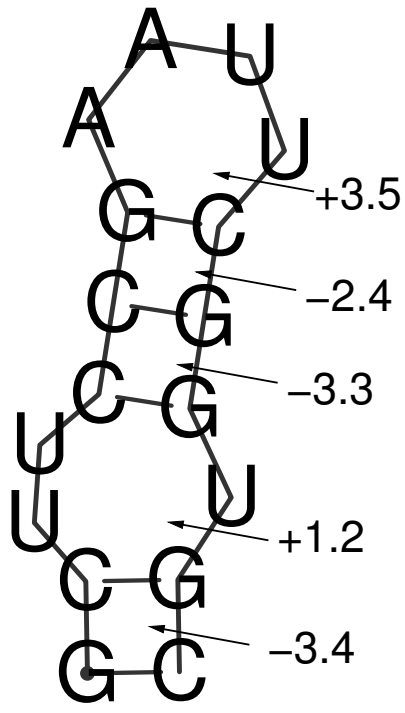


exterior loop

Nearest Neighbor Energies

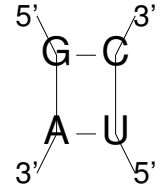
Free energy of a structure approximated as the sum of loop energies

- Good approximation for most oligo-nucleotides
- Loop energies depend on loop type and size, with some sequence dependence
- Most relevant parameters measured experimentally, some still guesswork
- Free energies are dependent on temperature and ionic conditions
- Training parameters is becoming an alternative to experiment



Stacked Pairs

- Major source of stabilizing energy
- all 21 combinations measured, accuracy ≈ 0.1 kcal/mol
- include the hydrogen bonding energy of pair formation
- energies of tandem G·U pairs depend on context, i.e. violate the nearest-neighbor model



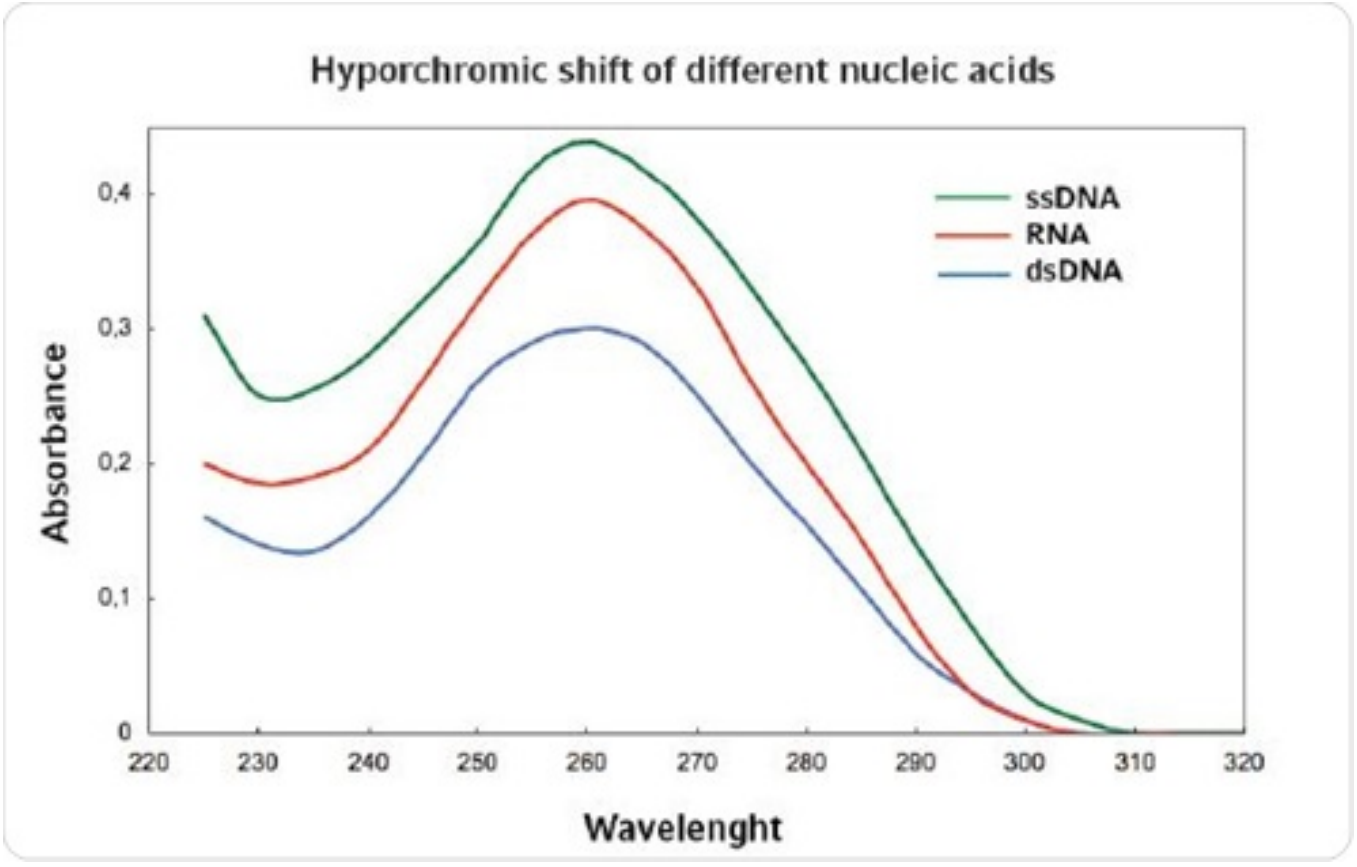
-2.4 kcal/mol

		$(i+1, j-1)$					
		CG	GC	GU	UG	AU	UA
(i, j)	CG	-2.4	-3.3	-2.1	-1.4	-2.1	-2.1
	GC	-3.3	-3.4	-2.5	-1.5	-2.2	-2.4
	GU	-2.1	-2.5	1.3	-0.5	-1.4	-1.3
	UG	-1.4	-1.5	-0.5	0.3	-0.6	-1.0
	AU	-2.1	-2.2	-1.4	-0.6	-1.1	-0.9
	UA	-2.1	-2.4	-1.3	-1.0	-0.9	-1.3

For comparison: Thermal energy $RT \approx 0.6$kcal/mol at 37C.

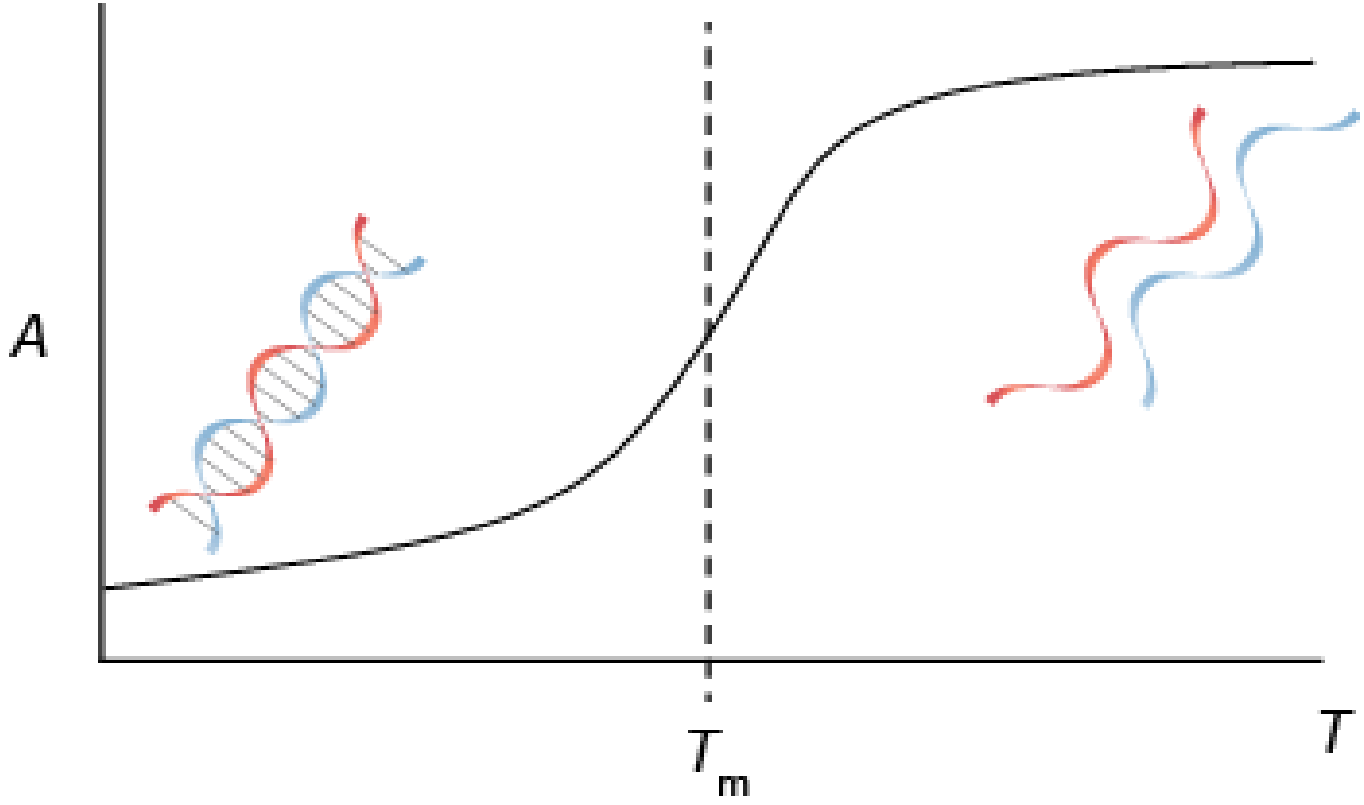
Determining Nearest Neighbor Parameters

UV absorption can distinguish between double and single stranded nucleic Acids — Hyperchromicity



Determining Nearest Neighbor Parameters

UV absorption (at 260nm) can be used to follow the unfolding transition as a function of temperature.



Analyzing UV Melting Curves

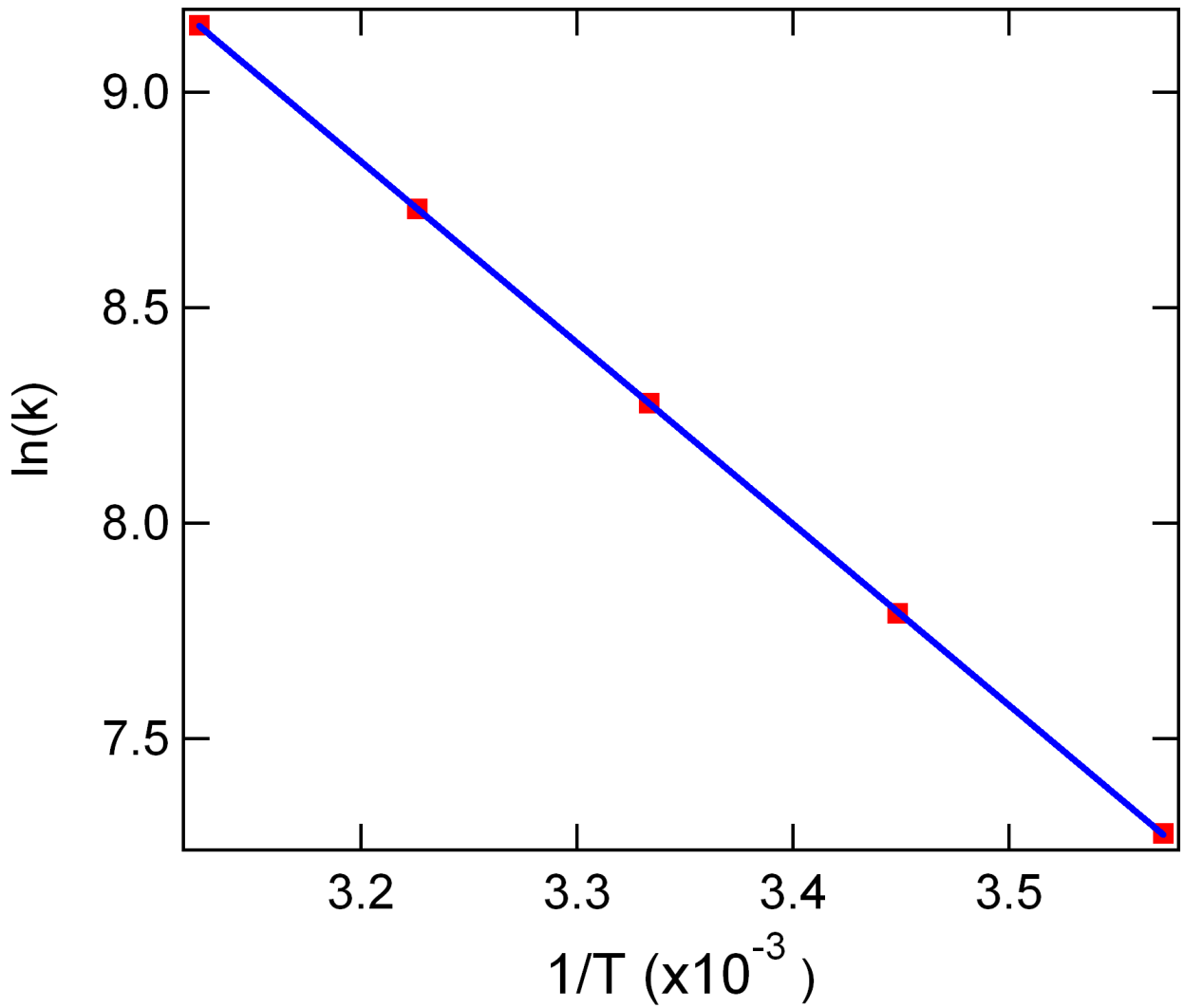
Assume a 2-state model

$$f_{unfolded}(T) = \frac{A(T) - A(T_{min})}{A(T_{max}) - A(T_{min})}$$

$$K_{eq} = \frac{f_{unfolded}}{f_{folded}} = e^{-\Delta G/RT}$$

Van't Hoff Analysis

$$\ln K_{eq} = -\frac{\Delta G}{RT} = -\frac{\Delta H}{RT} + \frac{\Delta S}{R}$$



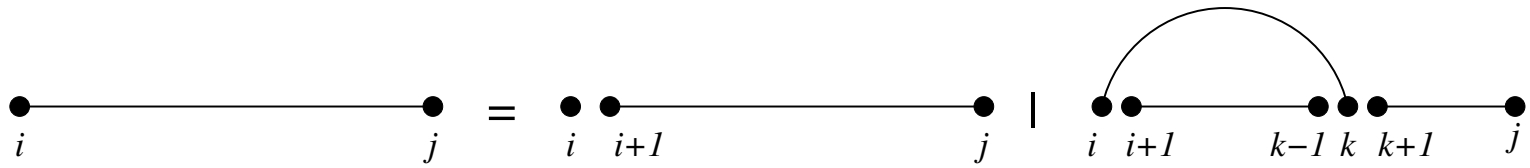
Turner Nearest Neighbor Parameters

- Up-to-date parameter sets available at the NNDB `rna.urmc.rochester.edu/NNDB`
- Up top 10000 entries per table
- Many tied entries, only **294** *independent* parameters
- Derived from **802** optical melting experiments
- Similar sets available for single stranded DNA

Parameters can also be *learned* from RNAs with known structure
Danger of *overfitting*, since most known structures come from few families

The RNA Conformation Space

The number of secondary structures for a sequence $x = x_1 \dots x_n$ can be enumerated recursively:



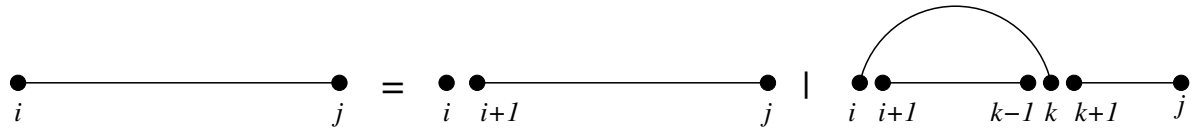
$$S_{ij} = S_{i+1,j} + \sum_{k=i+m}^{j} S_{i+1,k-1} S_{k+1,j}\, pair(x_k, x_j)$$

$pair(x_k, x_j) = 1$ if $x_k x_j$ is a canonical pair
(GC, CG, AU, UA, GU, UG)
otherwise $pair(x_k, x_j) = 0$.
For typical sequences the number of conformations grows as

$$\overline{S}_{1n} \sim n^{-\frac{3}{2}} 1.85^n$$

Solving the Folding Problem

Toy model for RNA folding: assign energies to base pairs $\varepsilon(x, y)$.
Easily solved by **Dynamic Programming**, i.e.:
Recursive computation with tabulation intermediate results.



$$E_{ij} = \min_{i<k\leq j} \left\{ E_{i+1,j} \; ; \quad \left(E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k) \right) \right\}$$

- E_{1n} is the best possible energy for our sequence x.
- Backtracing through the E table yields the corresponding structure.
- The Algorithm requires $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ CPU time.

In practice this toy model is not good enough!
For serious predictions, we need to use loop dependent energies.

Folding using Nussinov's Algorithm

$\varepsilon(C, G) = \varepsilon(G, C) = -3 \qquad \varepsilon(A, U) = \varepsilon(U, A) = -2$
$\varepsilon(G, U) = \varepsilon(U, G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

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

Folding using Nussinov's Algorithm

$\varepsilon(C, G) = \varepsilon(G, C) = -3 \qquad \varepsilon(A, U) = \varepsilon(U, A) = -2$
$\varepsilon(G, U) = \varepsilon(U, G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0							A
	0	0	0	0	0						G
		0	0	0	0	0					C
			0	0	0	0	0				A
				0	0	0	0	-3			C
					0	0	0	0	0		A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C, G) = \varepsilon(G, C) = -3 \qquad \varepsilon(A, U) = \varepsilon(U, A) = -2$
$\varepsilon(G, U) = \varepsilon(U, G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0						A
	0	0	0	0	0	-3					G
		0	0	0	0	0	0				C
			0	0	0	0	0	-3			A
				0	0	0	0	-3	-3		C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C, G) = \varepsilon(G, C) = -3 \qquad \varepsilon(A, U) = \varepsilon(U, A) = -2$
$\varepsilon(G, U) = \varepsilon(U, G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3					A
	0	0	0	0	0	-3	-3				G
		0	0	0	0	0	0	-3			C
			0	0	0	0	0	-3	-3		A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3	-3				A
	0	0	0	0	0	-3	-3	-3			G
		0	0	0	0	0	0	-3	-6		C
			0	0	0	0	0	-3	-3	-3	A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3	-3	-3			A
	0	0	0	0	0	-3	-3	-3	-6		G
		0	0	0	0	0	0	-3	-6	-6	C
			0	0	0	0	0	-3	-3	-3	A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3	-3	-3	-6		A
	0	0	0	0	0	-3	-3	-3	-6	-9	G
		0	0	0	0	0	0	-3	-6	-6	C
			0	0	0	0	0	-3	-3	-3	A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C, G) = \varepsilon(G, C) = -3 \qquad \varepsilon(A, U) = \varepsilon(U, A) = -2$
$\varepsilon(G, U) = \varepsilon(U, G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3	-3	-3	-6	-9	A
	0	0	0	0	0	-3	-3	-3	-6	-9	G
		0	0	0	0	0	0	-3	-6	-6	C
			0	0	0	0	0	-3	-3	-3	A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	-3	-3	-3	-6	-9	A
	0	0	0	0	0	-3	-3	-3	-6	-9	G
		0	0	0	0	0	0	-3	-6	-6	C
			0	0	0	0	0	-3	-3	-3	A
				0	0	0	0	-3	-3	-3	C
					0	0	0	0	0	0	A
						0	0	0	0	0	C
							0	0	0	0	A
								0	0	0	G
									0	0	G
										0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

	A	G	C	A	C	A	C	A	G	G	C	
	0	0	0	0	0	0	-3	-3	-3	-6	-9	A
		0	0	0	0	0	-3	-3	-3	-6	-9	G
			0	0	0	0	0	0	-3	-6	-6	C
				0	0	0	0	0	-3	-3	-3	A
					0	0	0	0	-3	-3	-3	C
						0	0	0	0	0	0	A
							0	0	0	0	0	C
								0	0	0	0	A
									0	0	0	G
										0	0	G
											0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

	.	(							)	)	
	A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	0	-3	-3	-3	-6	-9	A
		0	0	0	0	0	-3	-3	-3	-6	-9	G
			0	0	0	0	0	0	-3	-6	-6	C
				0	0	0	0	0	-3	-3	-3	A
					0	0	0	0	-3	-3	-3	C
						0	0	0	0	0	0	A
							0	0	0	0	0	C
								0	0	0	0	A
									0	0	0	G
										0	0	G
											0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

	.	(	(					)	)	)	
	A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	0	-3	-3	-3	-6	-9	A
		0	0	0	0	0	-3	-3	-3	-6	-9	G
			0	0	0	0	0	0	-3	-6	-6	C
				0	0	0	0	0	-3	-3	-3	A
					0	0	0	0	-3	-3	-3	C
						0	0	0	0	0	0	A
							0	0	0	0	0	C
								0	0	0	0	A
									0	0	0	G
										0	0	G
											0	C

Folding using Nussinov's Algorithm

$\varepsilon(C,G) = \varepsilon(G,C) = -3 \qquad \varepsilon(A,U) = \varepsilon(U,A) = -2$
$\varepsilon(G,U) = \varepsilon(U,G) = -1$

$$E_{ij} = E_{i+1,j} \mid E_{i+1,k-1} + E_{k+1,j} + \varepsilon(x_i, x_k)$$

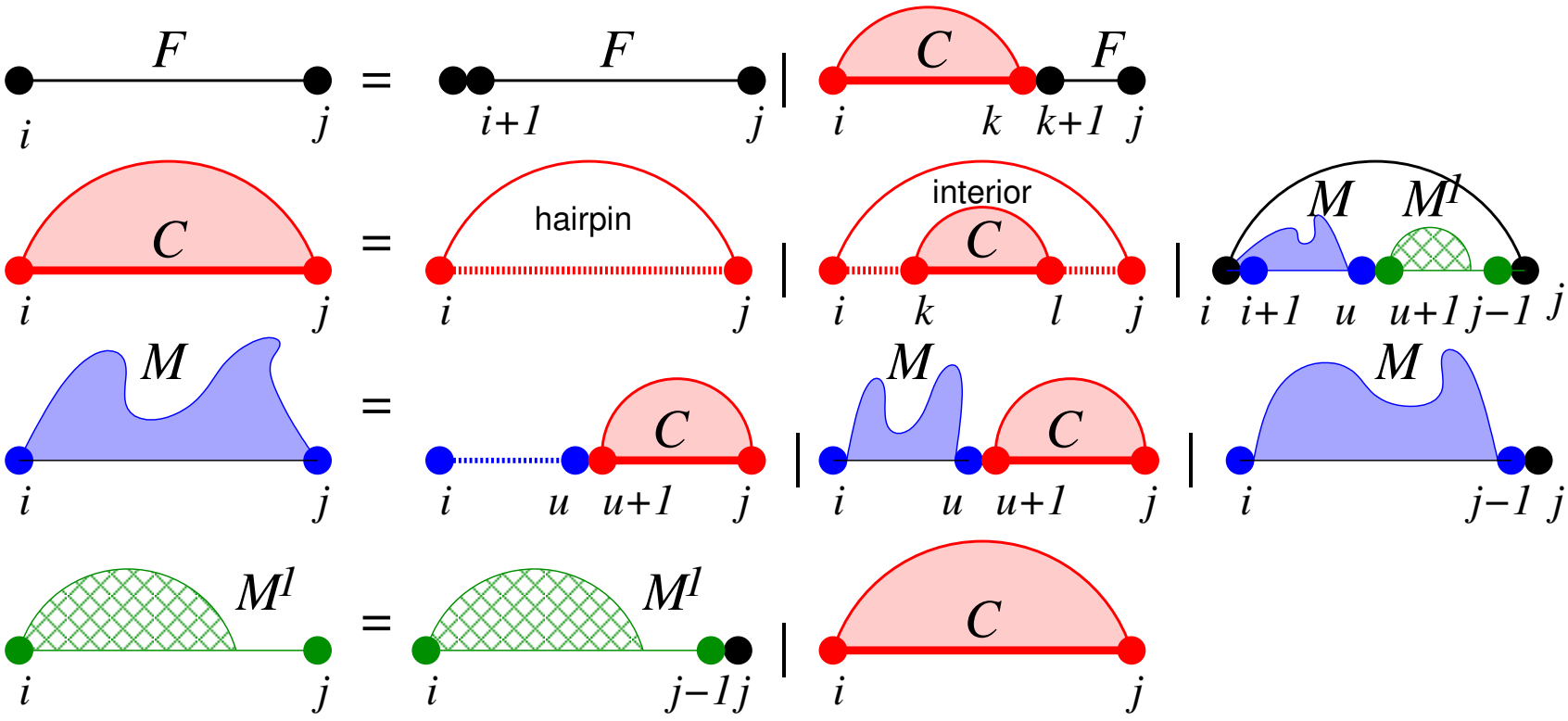
	.	(	(	.	(	.	.	.	)	)	)	
	A	G	C	A	C	A	C	A	G	G	C	
0	0	0	0	0	0	0	-3	-3	-3	-6	-9	A
		0	0	0	0	0	-3	-3	-3	-6	-9	G
			0	0	0	0	0	0	-3	-6	-6	C
				0	0	0	0	0	-3	-3	-3	A
					0	0	0	0	-3	-3	-3	C
						0	0	0	0	0	0	A
							0	0	0	0	0	C
								0	0	0	0	A
									0	0	0	G
										0	0	G
											0	C

Algorithms for Secondary Structure Prediction

The principle of *Dynamic programming* is used in many algorithms for RNA structure prediction in many different flavors.

- Minimum free energy structure (Zuker & Stiegler '81)
- Optimal and *certain* suboptimal structures (Zuker '89)
- All structures within an energy range (Wuchty et al. '99)
- Partition function and base pair probabilities (McCaskill '90)
- Stochastic suboptimals (Ding & Lawrence '01)
- Maximum expected accuracy structures (Do et al '06)
- Consensus structure prediction from alignment
  (Knudsen & Hein '99, Hofacker et al. '02)
- Minimum free energy with pseudo-knots (Rivas & Eddy '99)
- *Extended* secondary structures with non-canonical pairs
  (Parisien & Major '08, Höner et al '11)

Folding with Loop based Energies



F_{ij} free energy of the optimal substructure on the subsequence $x[i..j]$.

C_{ij} optimal free energy on $x[i..j]$, where (i, j) pair.

M_{ij} $x[i..j]$ is part of a multiloop and contains at least one pair.

M^1_{ij} same as M_{ij} but contains exactly one component closed by (i, h).

Minimum Free Energy Folding and Accuracy

MFE folding predicts, the optimal, most probable, structure.

+ easy to program, calculate, and interpret
− one structure to represent the equilibrium ensemble
− no indication of reliability

Accuracy measured by comparison with (phylogenetic) model structures.

- about 40-70% accuracy with current parameters
- accurate structures can usually be found within small energy increment of the MFE

How accurate is it?

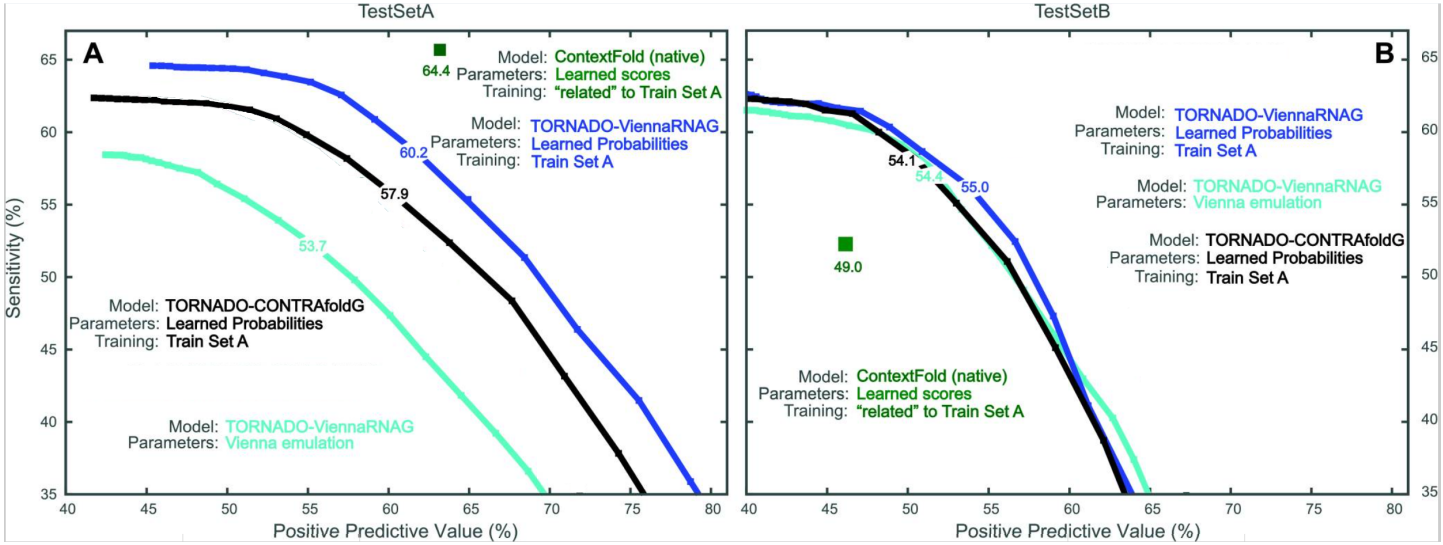
- Accuracy varies widely between RNA families
- Better for shorter RNAs with less complex structures
- Hardly any difference between programs

Limits to prediction accuracy

A number of different factors contribute to the inaccuracy of our predictions:

- Limited accuracy of energy parameters
- Beyond secondary structure:
  Pseudo-knots, 3D structure, G-quadruplexes, non-canonical pairs
- Deviation from "standard" conditions
  Ion concentrations (esp. Mg^{2+}), temperature
- Interaction with other molecules:
  RNA interact with proteins other RNAs and metabolites
- Posttranscriptional RNA modifications
- Folding kinetics:
  native structure $\neq$ ground state structure

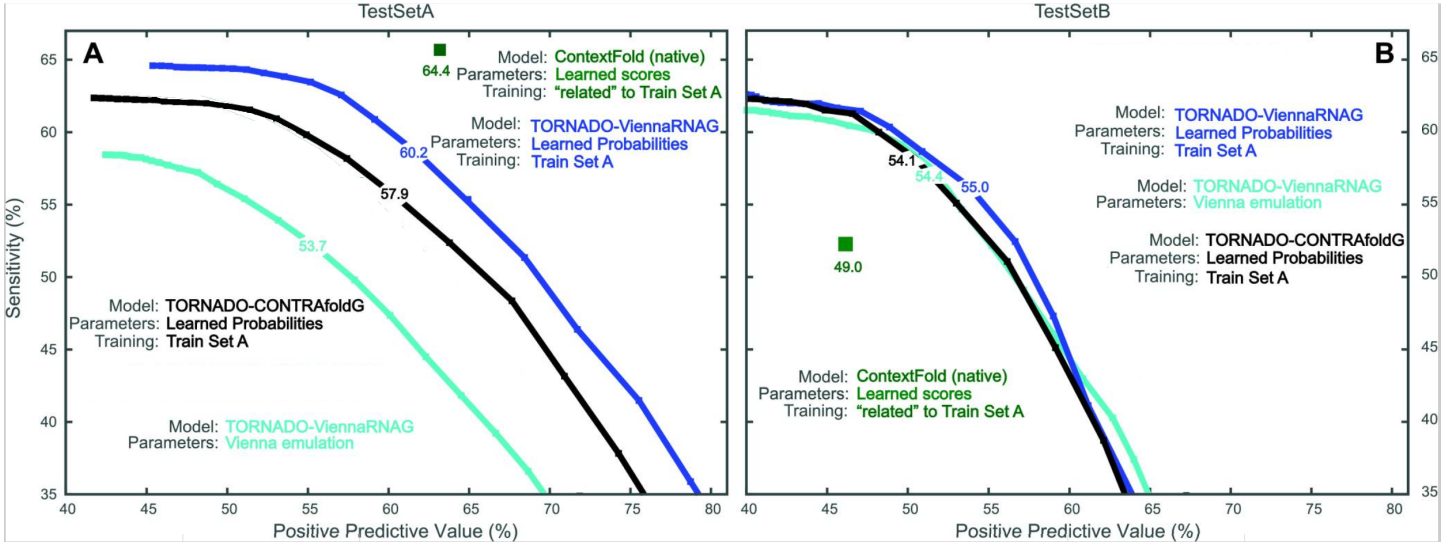
Does Training Parameters help?



modified from Rivas et al. 2012

- Improved performance on RNA families used for training
- No improvement for unrelated RNAs

Does Training Parameters help?



modified from Rivas et al. 2012

- Improved performance on RNA families used for training
- No improvement for unrelated RNAs
- Never the use the same RNA family in training and testing
- Limiting sequence similarity is not enough!

Deep Learning for Structure prediction

- Early deep learning methods reported huge accuracy improvements
- . . . but training and test sets are drawn from the same families
- $\rightarrow$ Neural Networks recognize homolog structures rather than predict structures *de novo*

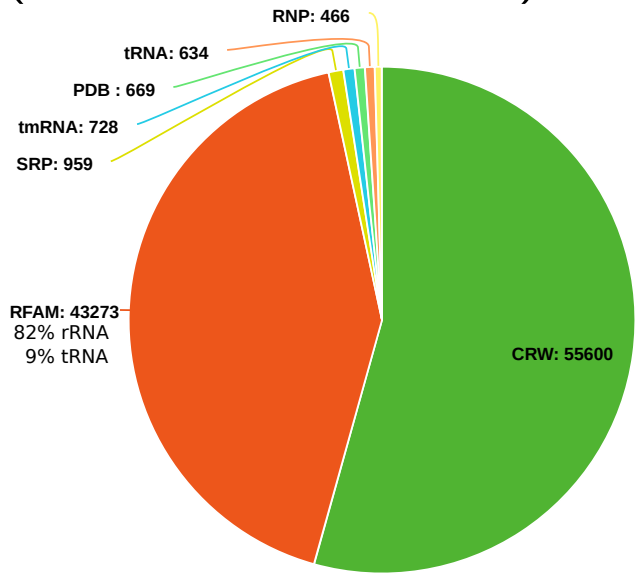
Deep Learning for Structure prediction

- Early deep learning methods reported huge accuracy improvements
- . . . but training and test sets are drawn from the same families
- $\rightarrow$ Neural Networks recognize homolog structures rather than predict structures *de novo*
- DNNs can generalize to new sequences
- Don't generalize well to new structures

Datasets for Training

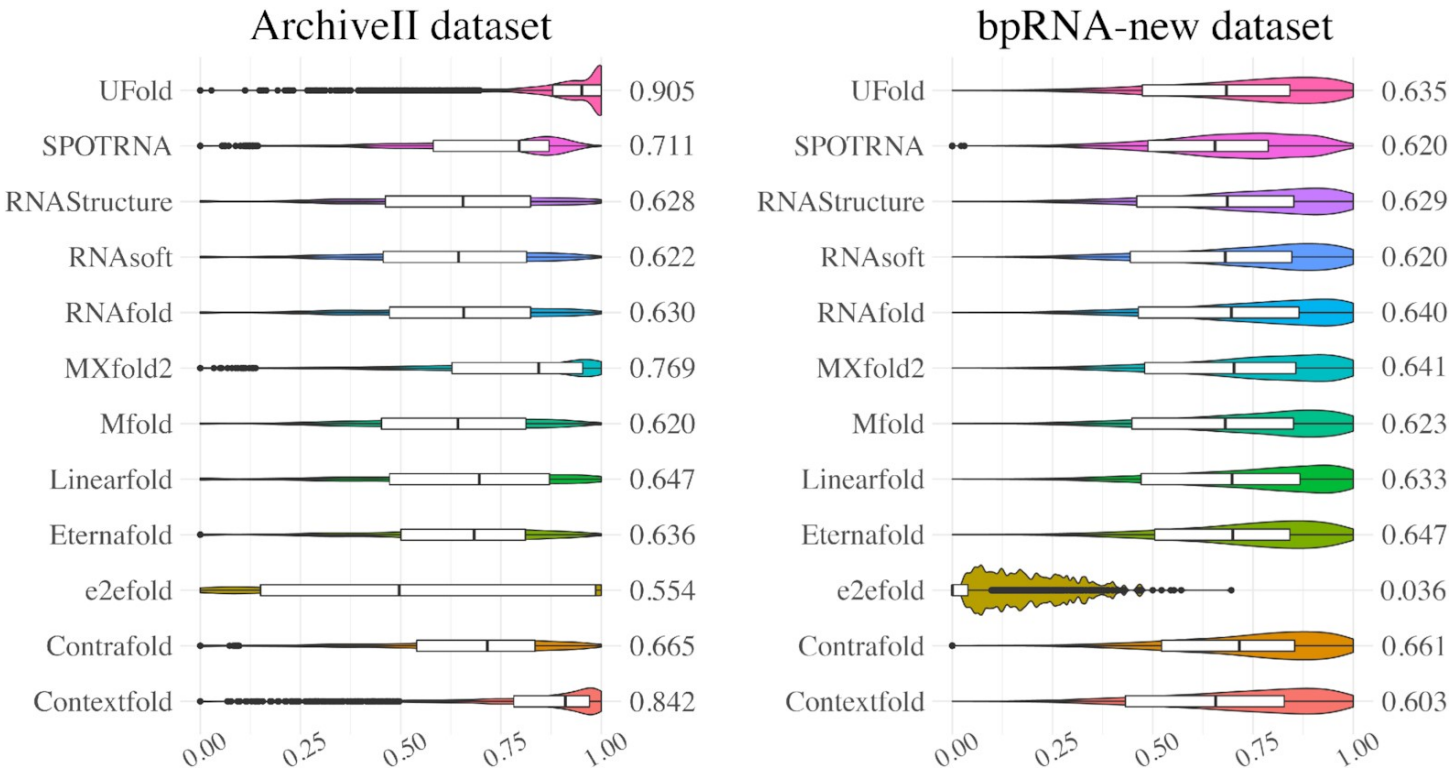
Largest and widely used data set: bpRNA (Danaee et al., 2018)

- 100 000 sequence/structure pairs
- Collected from 7 databases
- CRW: SSU, LSU, and 5S rRNAs
- RFAM 12.2: 2588 families
  but 82% rRNA, 9% tRNA



$\rightarrow$ Many sequences, but low structural diversity!

Effect of Biased Training Sets



Fu et al, 2022

Synthetic Data sets to Explore the Effect of Biases

How can we mimic training data with low structural diversity using synthetic data?

- Take all structures from bpRNA length ≤ 120
- For each structure design a sequence using RNAinverse

Synthetic Data sets to Explore the Effect of Biases

How can we mimic training data with low structural diversity using synthetic data?

- Take all structures from bpRNA length ≤ 120
- For each structure design a sequence using RNAinverse
- Resulting data set has same *structure* distribution as bpRNA
- No homology, very low sequence similarity

Synthetic Data sets to Explore the Effect of Biases

How can we mimic training data with low structural diversity using synthetic data?

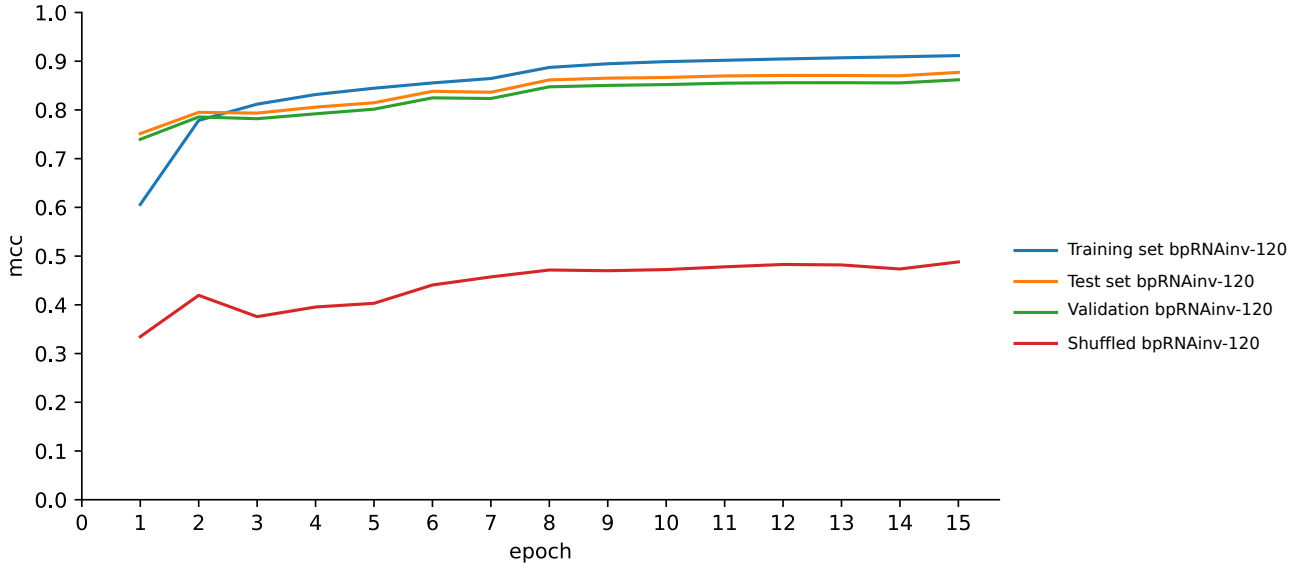
- Take all structures from bpRNA length ≤ 120
- For each structure design a sequence using RNAinverse
- Resulting data set has same *structure* distribution as bpRNA
- No homology, very low sequence similarity

After training with these data set, test performance on two test sets:

1. A test set produced in the same way by RNAinverse:
   unrelated sequences, but same structure bias
2. Di-nucleotide shuffling of training sequences:
   same sequence composition, but unrelated structures

Performance with the bpRNAinv data set

Reimplementation of SPOT-RNA trained on synthetic data



- Performance on novel sequences with known structures as good as training set
- Poor performance when structures are dis-similar to training structures

Limits to prediction accuracy

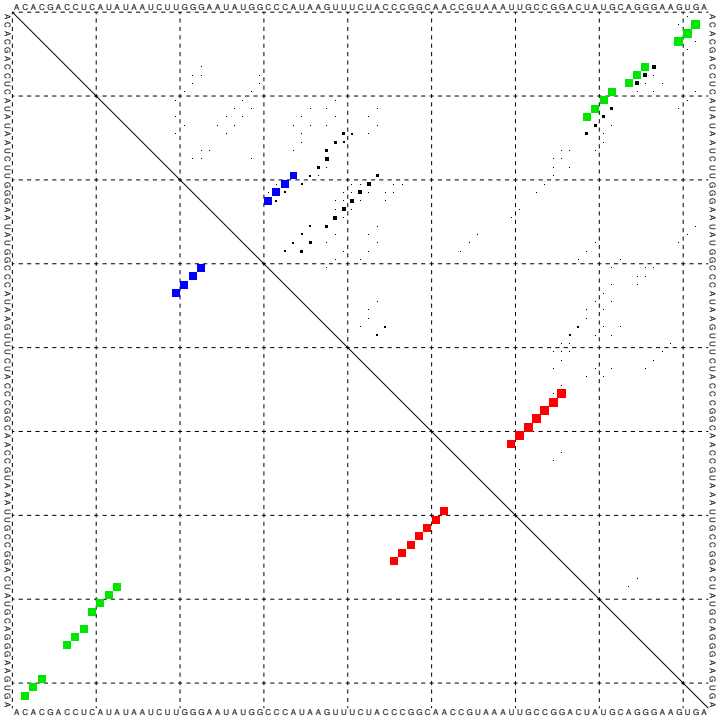
Limited prediction accuray is due to

- Model simplifications
- not energy parameters

How deal with it?

- Don't rely on a single structure
- Don't assume there is a single native structure
- Include external information

Representing Ensembles of Structures



```
 ACACGACCUCAUAUAAUCUUGGGAAUAUGGCCCAUAAGUUUCUACCCGGCAACCGUAAAUUGCCGGACUAUGCAGGGAAGUGA -22.60 2.00
.(((..(((((((.....((((.......))))...........(((((((.......)))))))..))))..)))...))).  -22.60
.(((..(((((((.....((((.......))))...((....))(((((((.......)))))))..))))..)))...))).  -22.00
.(((..(((((((.....((((.......))))...........(((((((.........)))))))..))))..)))...))).  -21.70
.(((..(((((((.....(((.......))).............(((((((.......)))))))..))))..)))...))).  -21.70
.(((..(((((((...(..((((.......))))..).......(((((((.......)))))))..))))..)))...))).  -21.40
.(((..(((((((....(((((.......))))..))).......(((((((.......)))))))..))))..)))...))).  -21.30
.(((..(((((.......(((((.......))))...........(((((((.......)))))))...)))..)))...))).  -21.30
.(((..(((((((.....((((.......))))...........(((((((.......)))))))..))))..))...))).  -21.20
.(((..(((((((.....(((((((.((.....)).)))))))).(((((((.......)))))))..))))..)))...))).  -21.10
.(((..(((((((.....((((.......))))...((....))(((((((.........)))))))..))))..)))...))).  -21.10
.(((..(((((((.....(((.......))))....((....))(((((((.......)))))))..))))..)))...))).  -21.10
.(((..(((((((.(((..((((.......))))...)))).....(((((((.......)))))))..))))..)))...))).  -20.90
.(((..(((((((...(..((((.......))))..).((....))(((((((.......)))))))..))))..)))...))).  -20.80
.(((..(((((((.....(((.......))).............(((((((.........)))))))..))))..)))...))).  -20.80
.(((..(((((((.((.((((((.......))))..)).)).)....(((((((.......)))))))..))))..)))...))).  -20.70
.(((..(((((((.....((((((((((...))).))))))).(((((((.......)))))))..))))..)))...))).  -20.70
.(((..(((((((.....((((.......))))..((...)).(((((((.......)))))))..))))..)))...))).  -20.70
...
```

Partition Function, Boltzmann and Pair Probabilities

The partition function Z is the fundamental quantity of statistical mechanics. All thermodynamic properties can be derived from it.

Partition Function, Boltzmann and Pair Probabilities

The partition function Z is the fundamental quantity of statistical mechanics. All thermodynamic properties can be derived from it.

- Partition function $Z = \sum_s \exp\left(\frac{-E(s)}{RT}\right)$
- Boltzmann probability of a structure s: $p(s) = \frac{1}{Z} \exp\left(\frac{-E(s)}{RT}\right)$
- Expected value of any quantity A: $\langle A \rangle = \sum_s A(s)p(s)$
- Free energy $F = -RT \ln Z$
- Entropy $S = -\frac{\partial F}{\partial T}$

Partition Function, Boltzmann and Pair Probabilities

The partition function Z is the fundamental quantity of statistical mechanics. All thermodynamic properties can be derived from it.

- Partition function $Z = \sum_s \exp\left(\frac{-E(s)}{RT}\right)$
- Boltzmann probability of a structure s: $p(s) = \frac{1}{Z} \exp\left(\frac{-E(s)}{RT}\right)$
- Expected value of any quantity A: $\langle A \rangle = \sum_s A(s)p(s)$
- Free energy $F = -RT \ln Z$
- Entropy $S = -\frac{\partial F}{\partial T}$

Constrained Parition Functions

Probability of some feature A. Compute the constraint partition function

$$Z^A = \sum_{s \text{ has feature } A} \exp\left(\frac{-E(s)}{RT}\right), \qquad p(A) = \frac{Z^A}{Z}$$

- Probability of forming the pair (i, j)
- probability of forming an aptamer structure
- probability of presenting a binding site
- ...

Computing Partition Function and Pair Probabilities

For simplicity back to the Nussinov model:
Computing the partition function $Z = \sum_{\Psi} \exp(-E(\Psi)/RT)$ is simple:

$$Z_{ij} = Z_{i+1,j} + \sum_{k,\,(i,k)\ \text{pairs}} Z_{i+1,k-1} Z_{k+1,j} \exp(-\varepsilon_{ik}/RT)\,.$$

In conjunction with the partition function $\widehat{Z}_{ij}$ for structures *outside* the subsequence $x[i..j]$ we can compute pair probabilities:

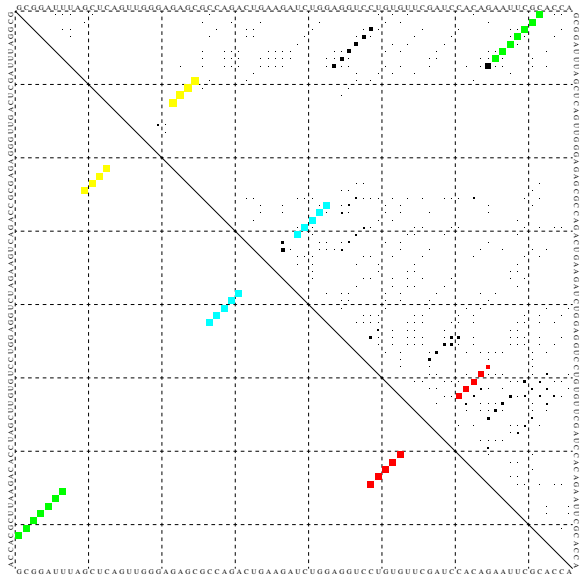
$$p_{ij} = \widehat{Z}_{ij} Z_{i+1,j-1} \exp(-\varepsilon_{ij}/RT)/Z\,.$$

Pair Probabilities

Equilibrium probabilities for all possible pairs can be calculated via McCaskill's partition function algorithm:

- provides a rigorous description of structures in thermodynamic equilibrium
- probability dot plots contain entropic terms not included in energy dot plots
- starting point for measures of reliability and well-definedness

– not as easy to interpret as a small set of structures



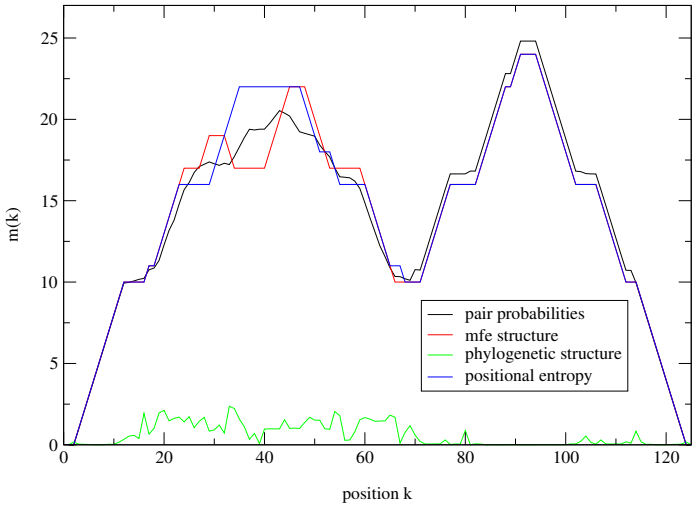
Well-defined Regions

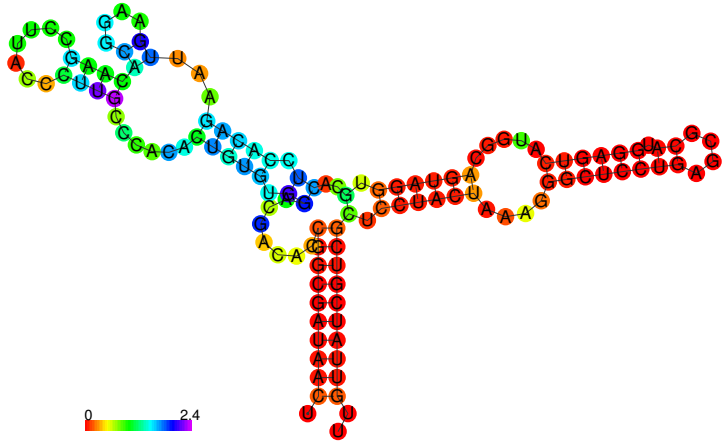
Pair probabilities can help judge the *reliability* of a prediction.

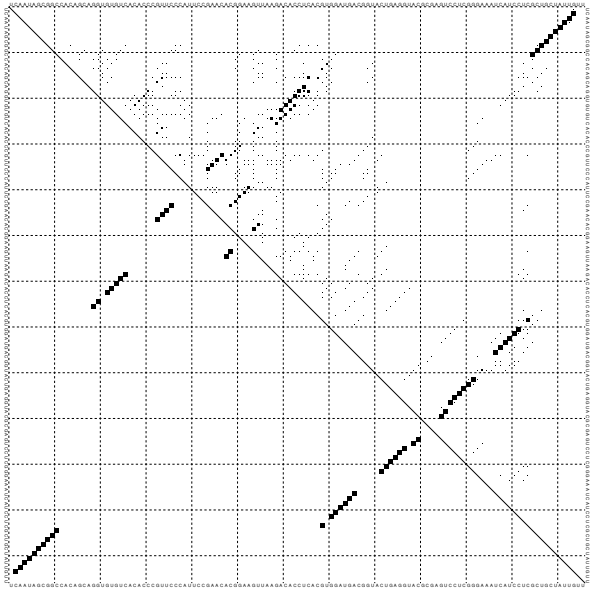
Well-definedness: Are there many structural alternatives?
e.g. "ensemble diversity" (returned by RNAfold) measures dissimilarity of structural alternatives
Computed directly from pair probabilities $\langle d \rangle = \sum_{i,j} p_{ij} \cdot (1 - p_{ij})$

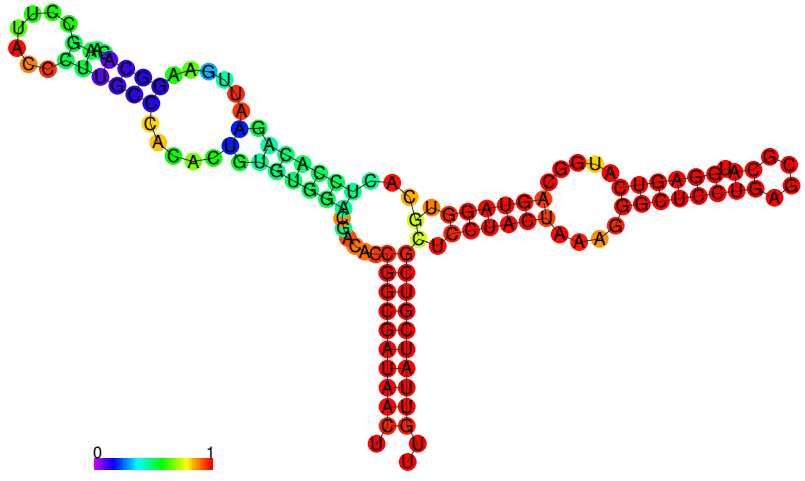
Local reliability: Which parts the prediction we can trust?
High probability pairs are almost always correct.
Secondary structure plots can be colored by pair probability or "positional entropy" to highlight reliable and unreliable regions.
positional entropy is computed from pair probabilities as $S(k) = -\sum_i p_{ik} \ln p_{ik}$

Structures with reliability annotation









Alternatives to the MFE as "best" Structure

Maximum Expected Accuracy (MEA)

- Select the structure expected to have the most base pairs correct
- $\rightarrow$ Maximize the sum of pair probabilities $< A >= \sum_{(i,j) \in S} p_{ij}$
- Computed by "Nussinov"-like dynamic programming

Centroid Structure

- Choose the structure that is "nearest" to all other structure in the Boltzmann ensemble
- Minimize $< d(S) >= \sum_{(i,j) \in S} p_{ij} + \sum_{(i,j) \notin S} (1 - p_{ij})$
- Trivial solution: just pick all pairs with probability > 0.5

No free lunch: Centroid and MEA structure may correspond to a very unlikely structure!

Suboptimal Structures

Zuker's p-suboptimal structures (`mfold`)

For each pair (i, j) generate the best structure containing that pair.

- \+ easy to compute by multiple backtracking
- \+ present a user with few, representative alternatives
- – important alternatives can be missed
- – somewhat arbitrary selection of representatives

Complete suboptimal folding

generates **all** structures within given energy range from the mfe.

- \+ Calculate any thermodynamic average
- \+ Investigate the energy landscapes, e.g. find low-lying local minima
- – huge amount of data, only for moderately short sequences

Stochastic suboptimal structure

generate Boltzmann weighted sample through stochastic backtracking

Stochastic Sampling

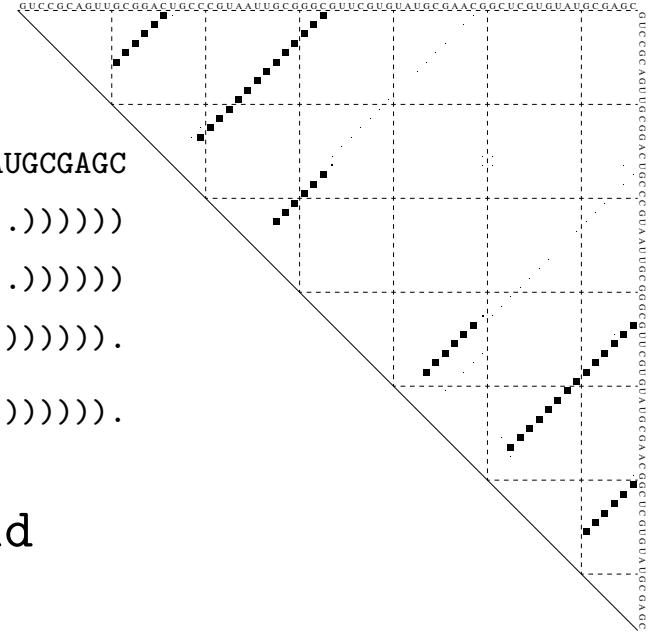
- Almost any quantity of interest can be approximated by Boltzmann sampling
- $\langle A \rangle = \sum_s A(s)p(s)$ is impractical compute
- Much easier:

$$\langle A \rangle \approx \frac{1}{N} \sum_{s \in \text{sample}} A(s)$$

- Sampling errors proportional $\frac{1}{\sqrt{N}}$
- $N \approx 1000$ is often sufficient

Dot Plots and suboptimal Structures

Zuker suboptimal structures may miss important alternatives



```
GUCCGCAGUUGCGGACUGCCCGUAAUUGCGGGCGUUCGUGUAUGCGAACGGCUCGUGUAUGCGAGC
((((((....))))))..((((((....))))))((((((....))))))..((((((....))))))
((((((((((((((.....))))))))))))))((((((....))))))..((((((....))))))
((((((((((((((.....))))))))))))))..((((((((((((.....))))))))))))).
((((((....))))))..((((((....))))))..((((((((((((.....))))))))))))).
```

Only the first 3 structures are found by `mfold`

Variations on Backtracing

What we need:

- Let π be a partial structure consisting of
  - Ω_π a set of already known base pairs
  - Υ_π a list of sequence intervals with unknown structure
- $E(\pi)$ best energy that can be obtained by completing π
- $\mathfrak{S}$ a stack of partial structures to be processed

Variations on Backtracing

MFE backtracing:

$\varnothing \rightarrow \mathfrak{S}$.
while $\mathfrak{S} \neq \emptyset$ **do**
  $\pi \leftarrow \mathfrak{S}$;
  if π is complete **then** output π
  $[i, j] = I \in \Upsilon_\pi$.
  $\pi' = \pi \blacktriangleleft (i)$
  if $E(\pi') = E_{\text{opt}}$
    then $\pi' \rightarrow \mathfrak{S}$; **next**;
  for all $k \in [i, j]$ **do**
    $\pi' = \pi \blacktriangleleft (i, k)$
    if $E(\pi') = E_{\text{opt}}$
      then $\pi' \rightarrow \mathfrak{S}$; **next**;

Variations on Backtracing

Wuchty suboptimals

$\varnothing \to \mathfrak{S}$.
while $\mathfrak{S} \neq \emptyset$ **do**
  $\pi \leftarrow \mathfrak{S}$;
  if π is complete **then** output π
  $[i,j] = I \in \Upsilon_\pi$
  $\pi' = \pi \blacktriangleleft (i)$
  if $E(\pi') \leq E_{\mathrm{opt}} + \Delta E$
    then $\pi' \to \mathfrak{S}$;
  for all $k \in [i,j]$ **do**
    $\pi' = \pi \blacktriangleleft (i,k)$
    if $E(\pi') \leq E_{\mathrm{opt}} + \Delta E$
      then $\pi' \to \mathfrak{S}$;

Variations on Backtracing

Stochastic backtracing

$\varnothing \to \mathfrak{S}$.
while $\mathfrak{S} \neq \emptyset$ **do**
  $\pi \leftarrow \mathfrak{S}$;
  if π is complete **then** output π
  $[i, j] = I \in \Upsilon_\pi$
  $r = Z(\pi) \cdot \mathrm{rand}()$;
  $\pi' = \pi \blacktriangleleft (i)$; $r = r - Z(\pi')$
  if $r \leq 0$ **then** $\pi' \to \mathfrak{S}$; **next**;
  for all $k \in [i, j]$ **do**
    $\pi' = \pi \blacktriangleleft (i, k)$; $r = r - Z(\pi')$
    if $r \leq 0$ **then** $\pi' \to \mathfrak{S}$; **next**;

SCFG based Methods

Stochastic Context Free Grammars

- An extension to HMMs that can handle long range correlations
- Formally, a CFG is a tuple $G = (V, \alpha, S, R)$ of Alphabet α, nonterminals V, Start symbol S, and rewrite rules R
- *Stochastic* CFG adds probabilities to each rule
- Generates sequences using a series of rewriting rules
- Well known repertoire of algorithms that work on any SCFG

Nussinov Algorithm as SCFG

$$
\begin{aligned}
S &\rightarrow aS|cS|gS|uS \\
S &\rightarrow PS \\
S &\rightarrow \varnothing \\
P &\rightarrow aSu|uSa|cSg|gSc|gSu|uSg
\end{aligned}
$$

- Applying grammar rules produces an RNA sequence
- The parse tree of productions used corresponds to the structure
- A *stochastic* CFG assigns probabilities to each production rule
- CYK algorithm finds the parse tree (structure) most likely to produce the given sequence

The Pfold Grammar

Nussinov grammar neglects stacking $\rightarrow$ poor accuracy
Can we improve this?

$$
\begin{aligned}
S &\rightarrow LS \mid L \\
L &\rightarrow U \mid P \\
F &\rightarrow P \mid LS \\
P &\rightarrow aFu \mid uFa \mid cFg \mid gFc \mid gFu \mid uFg \\
U &\rightarrow a \mid c \mid g \mid u
\end{aligned}
$$

- Distinguishes helix initiation, elongation, termination
- Only slightly worse prediction than full Turner model

The Pfold Grammar

Nussinov grammar neglects stacking $\rightarrow$ poor accuracy
Can we improve this?

$$
\begin{aligned}
S &\rightarrow LS(0.87) \,|\, L(0.13) \\
L &\rightarrow U(0.89) \,|\, P(0.11) \\
F &\rightarrow P(0.79) \,|\, LS(0.21) \\
P &\rightarrow aFu \,|\, uFa \,|\, cFg \,|\, gFc \,|\, gFu \,|\, uFg \\
U &\rightarrow a \,|\, c \,|\, g \,|\, u
\end{aligned}
$$

- Distinguishes helix initiation, elongation, termination
- Only slightly worse prediction than full Turner model

SCFGs vs Energy directed folding

Many RNA related algorithms come in two flavors, based on either

energy directed folding or *stochastic context free grammars*

physics based model
parameters from experiment

probabilistic inference
parameters learned from data

Algorithms are mostly analogous, but terminology is different!

MFE folding
pair probabilities

CYK algorithm
Inside/Outside algorithm

Beyond classical secondary structure

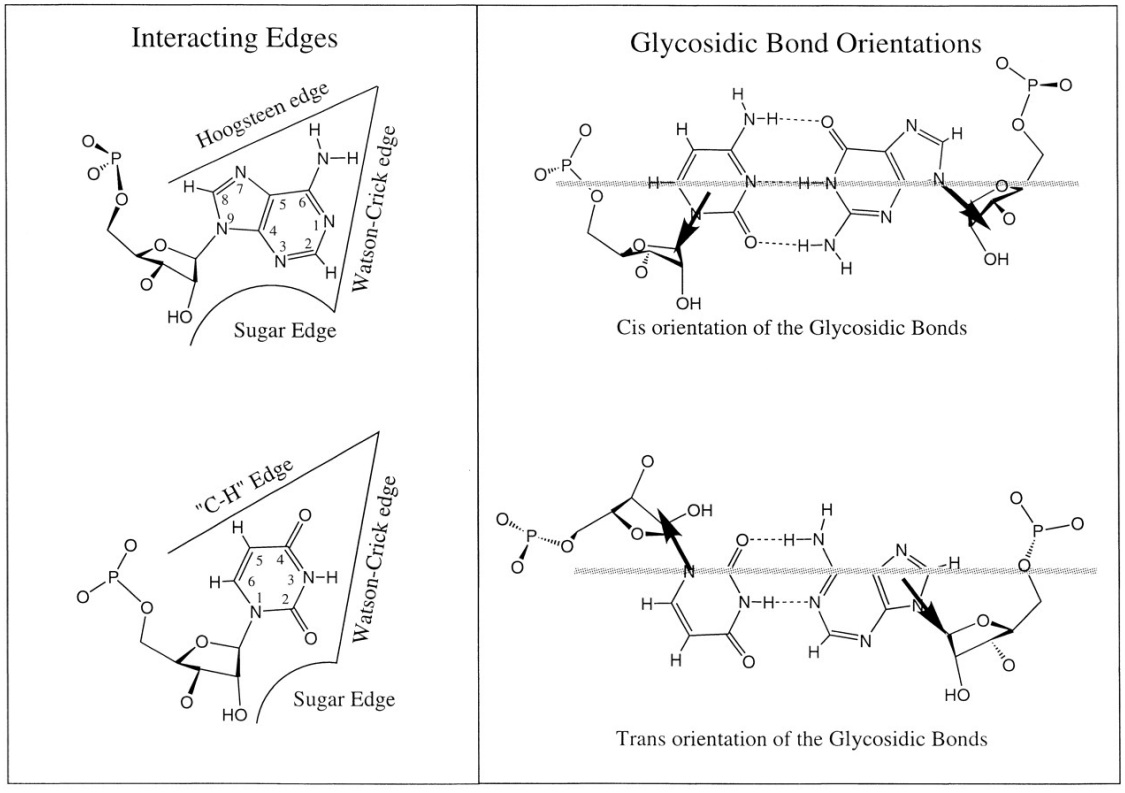
State of the art in RNA secondary structure prediction:

- Classical secondary structure considers only 6 types of pairs CG,GC,AU,UA,GU,UG
- All base pairs are assumed to be Watson-Crick type
- Loops are drawn as unstructured bubbles
- Energy model assigns free energies to each loop

Can we improve the level of detail?

Leontis-Westhof classification

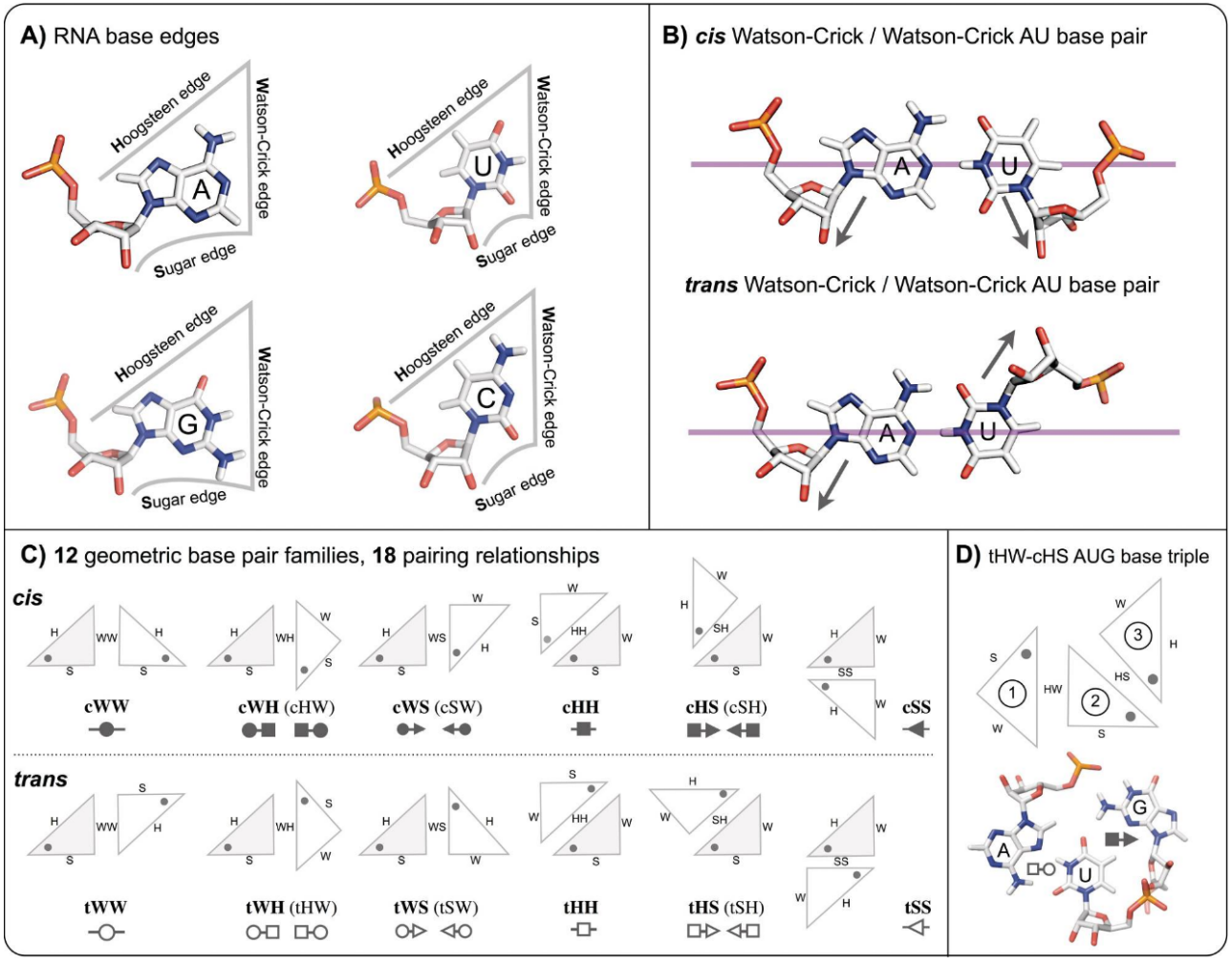
Watson-Crick base pairing is only part of the story!



Overall 12 different pairing types for any two bases

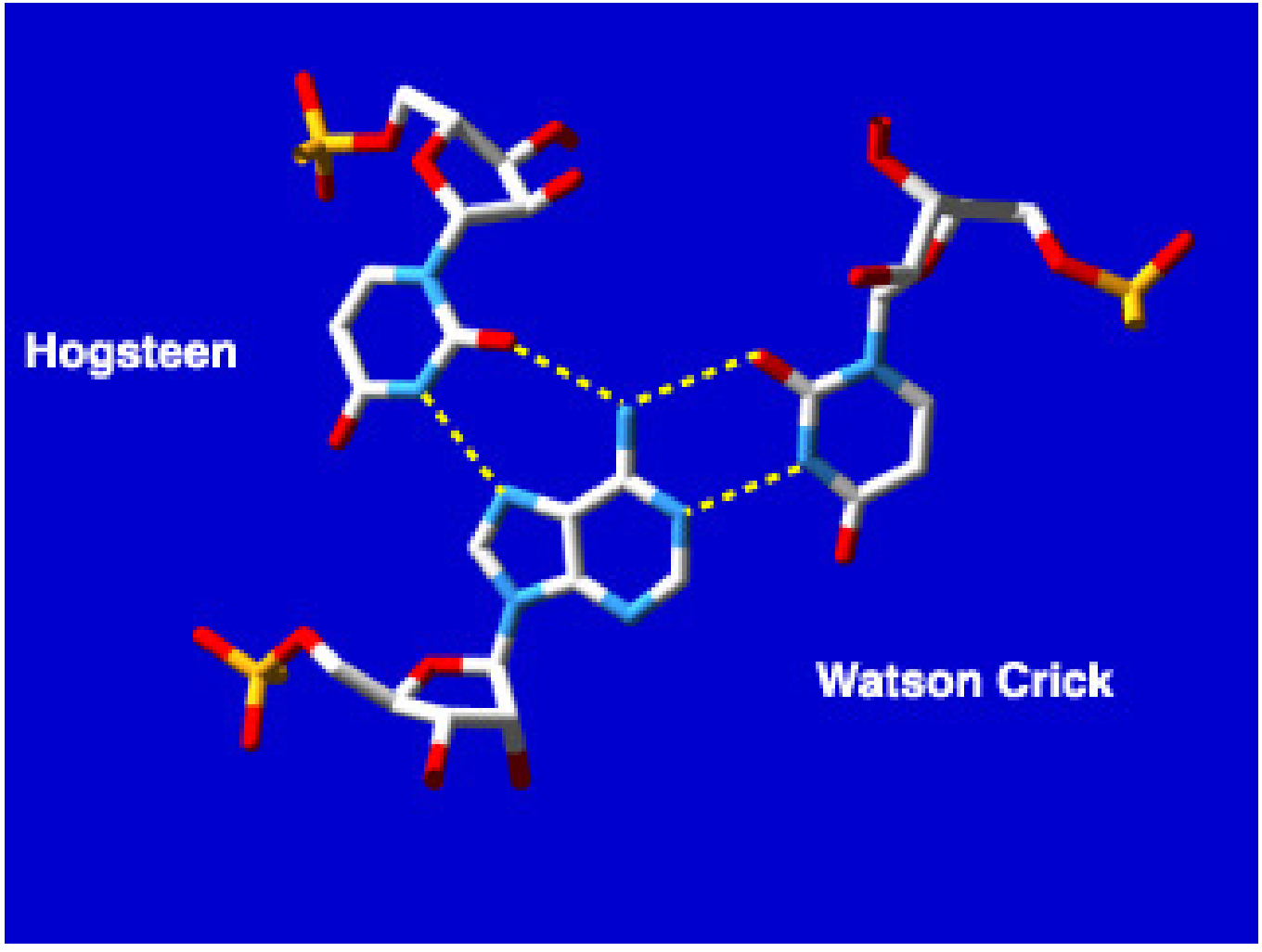
Starting point for defining recurring tertiary structure *motifs*

Leontis-Westhof classification



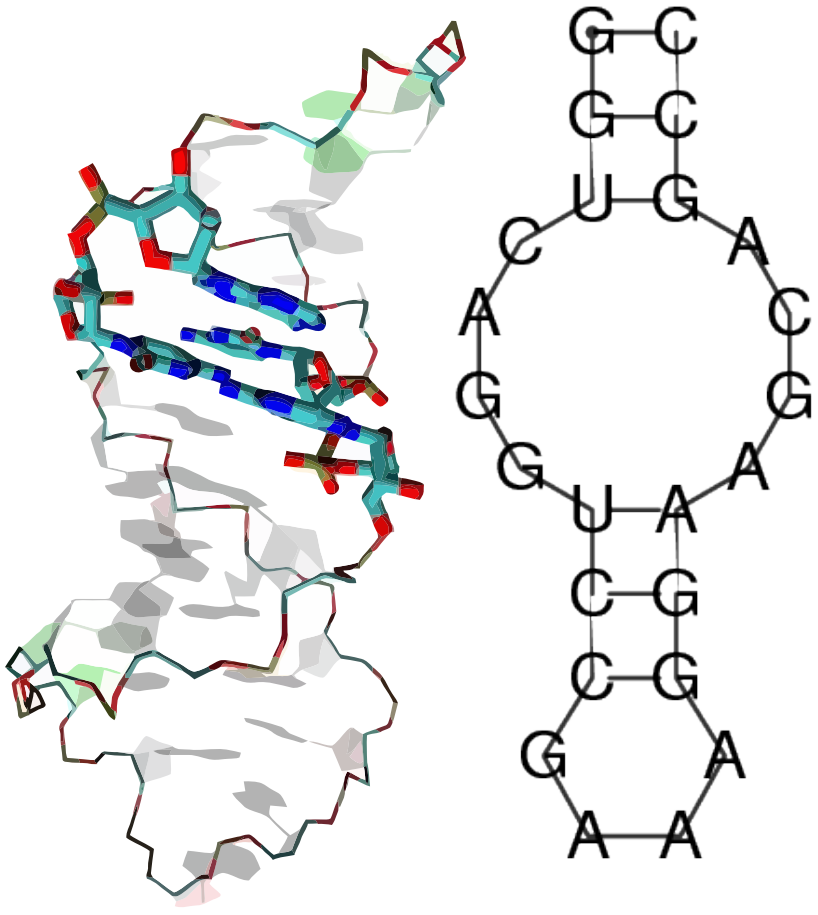
In structure drawings the interacting edge (Watson-Crick, Hoogsteen, Sugar) is represented by a circle, square, or triangle

Base triples



A base can pair at two different edges forming *base-triples*

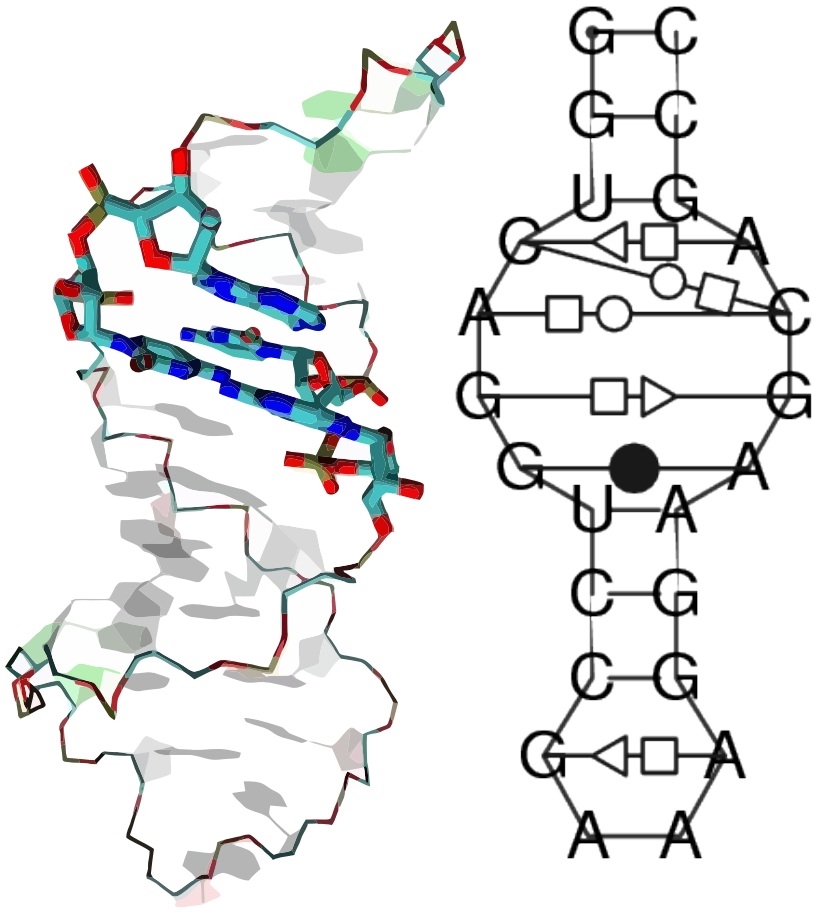
2D and 3D Structures



Looking at known tertiary structures we find that:

- "Loops" are not unstructured
- full of non-canonical base pairs
- all 16 [ACGU]×[ACGU] combinations form pairs
- the same two nucleotides can pair in many different ways

2D and 3D Structures



Looking at known tertiary structures we find that:

- "Loops" are not unstructured
- full of non-canonical base pairs
- all 16 [ACGU]×[ACGU] combinations form pairs
- the same two nucleotides can pair in many different ways

Predicting extended secondary structures

MC-Fold by Parisien & Major (Nature 2008) Can we include non-canonical pairs in prediction?

- Folding algorithms can be easily extended
- More DP matrices (for each of the 12 pairing types) still same $\mathcal{O}(n^3)$ complexity
- Implemented in MC-Fold by Parisien & Major (originally inefficient with exponential runtime)

Predicting extended secondary structures

MC-Fold by Parisien & Major (Nature 2008) Can we include non-canonical pairs in prediction?

- Folding algorithms can be easily extended
- More DP matrices (for each of the 12 pairing types) still same $\mathcal{O}(n^3)$ complexity
- Implemented in MC-Fold by Parisien & Major (originally inefficient with exponential runtime)

. . . causes parameter explosion

- $12x12x16x16 \approx 36000$ stacked pair parameters
- No free energies from experiments
- Limited training data (3D structures)