Computational Methods for RNA Secondary Structure

Michael Zuker

June 8, 2006

Introduction

What is RNA secondary structure?

RNA secondary structure is similar to an alignment of protein and nucleic acid sequences, except that the sequence folds back on itself and "complementary bases" pair rather than identical or similar bases. Also, an alignment of 2 or more bio-sequences is a statement about an inferred evolutionary history. In contrast, an RNA secondary structure is a simplification of a complex 3 dimensional folding of a biopolymer. (It should be noted here that an alignment of protein sequences also has structural implications, since the aligned residues should be superimposable in 3 dimensions.)

An RNA molecule is composed of 4 types of (ribo)nucleotides. Each nucleotide contains a phosphate group, a sugar group (ribose) and a base. The polymer is formed by the linkage of the phosphate groups. The non-planar 5 member ribose ring connects the phosphate to the base. Finally, the bases are connected to the ribose group. Only the bases differ. The 4 different bases, adenine (A), cytosine (C), guanine (G) and uracil (U) are illustrated below.

The complementary bases, C-G and A-U form stable "base pairs" with each other through the creation of hydrogen bonds between donor and acceptor sites on the bases. These are called "Watson-Crick (W-C)" base pairs. In addition, we consider the weaker G-U "wobble pair", where the bases bond in a skewed fashion. All of these are called "canonical base pairs". Other base pairs occur, some of which are stable. They are called non-canonical base pairs.

The "secondary structure" of an RNA molecule is the collection of base pairs that occur in its 3 dimensional structure. An RNA sequence will be represented as

$$\mathbf{R} = r_1, r_2, r_3, \ldots, r_n,$$

where r_i is called the i^{th} (ribo)nucleotide. Each r_i belongs to the set $\{A, C, G, U\}$. We will refer to i as the i^{th} base of the sequence. A secondary structure, or folding, on $\mathbf{R}$ is a set $\mathbf{S}$ of ordered pairs, written as $i.j$, $1 \leq i < j \leq n$ satisfying:

1. $j - i > 3$
2. If $i.j$ and $i'.j'$ are 2 base pairs, (assuming without loss in generality that $i \leq i'$), then either:
   (a) $i = i'$ and $j = j'$ (they are the same base pair),
   (b) $i < j < i' < j'$ ($i.j$ precedes $i'.j'$), or

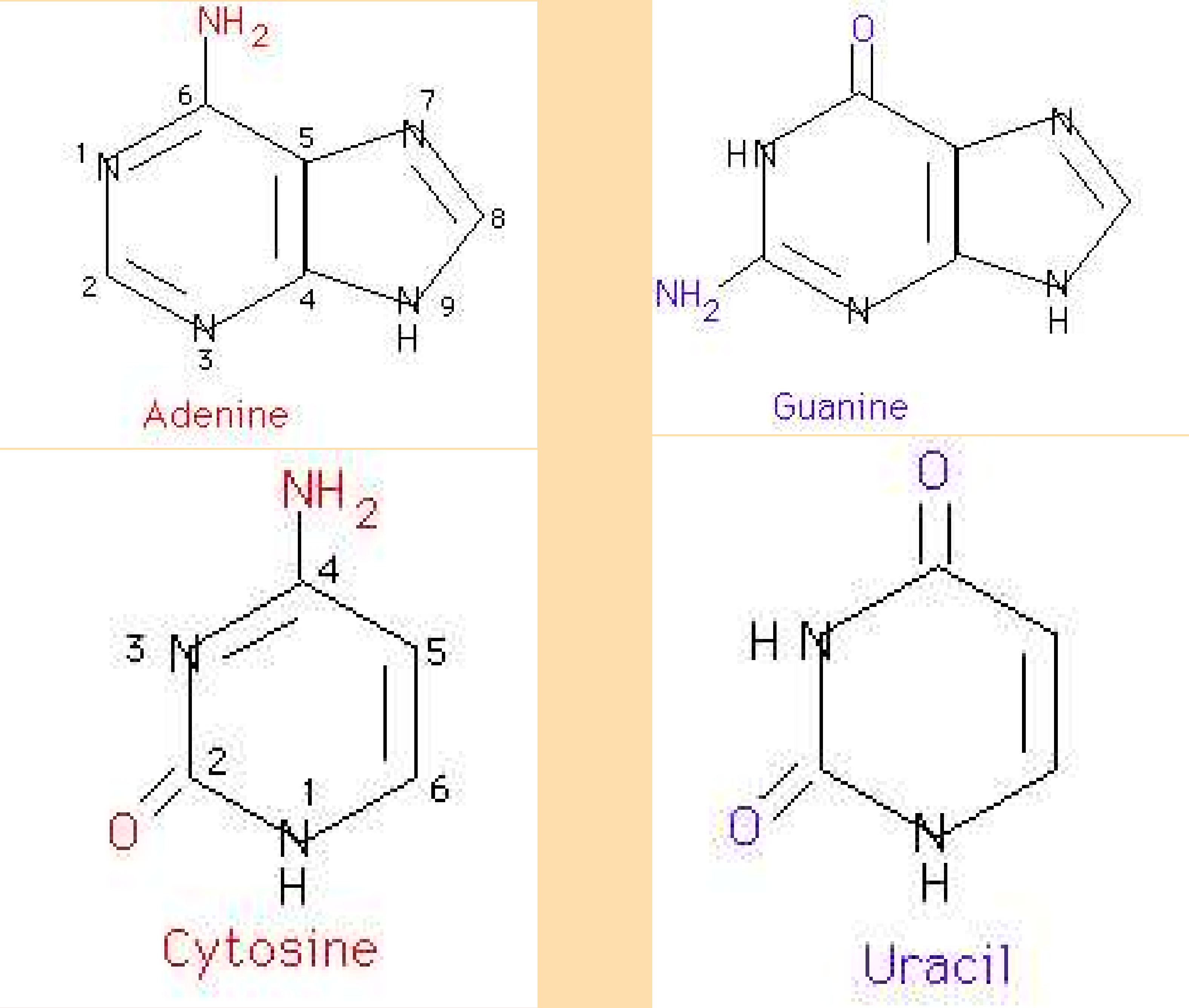


Figure 1: The four unmodified RNA bases.

(c) $i < i' < j' < j$ ($i.j$ includes $i'.j'$).

The last condition excludes "pseudoknots". These occur when 2 base pairs, $i.j$ and $i'.j'$ satisfy $i < i' < j < j'$. Pseudoknots are excluded because energy minimizing methods cannot deal with them. It is not known how to assign energies to the loops created by pseudoknots and dynamic programming methods that compute minimum energy structures break down. For this reason, pseudoknots are often considered as belonging to tertiary structure. However, pseudoknots are real and important structural features. Covariance methods are able to predict them from aligned, homologous RNA sequences. Featured in Figure 2 is a small pseudoknot model.

The RNA component of this course deals with structure prediction from sequence data. There are 2 routes. The first attempts structure prediction of single sequences based on minimizing the free energy of a folding. The second computes common foldings for a family of aligned, homologous RNAs. Usually, the alignment and secondary structure inference must be performed simultaneously, or at least iteratively.

Display of RNA Secondary Structures

Figure 3 represents the usual display of an RNA secondary structure. In this case, it is a computer prediction for the RNA component of *Bacillus subtilis* RNase P. The nucleotides are laid out in such a way that paired bases are proximal. In this representation, W-C pairs are denoted by "-" and GU pairs by "•". A group of at least 2 consecutive base pairs is called a helix. A helix of k base pairs contains $k-1$ stacking interactions. We assign energies to these *between base pair* regions, although the energies contain terms from both hydrogen bonding and base pair stacking.

The open regions surrounded by single stranded bases are called loops. Various types of loops are illustrated in the figure. A more formal definition is given below.

Figure 4 represents the same *B. subtilis* folding. The nucleotides are stretched out uniformly along the circumference of a circle and the base pairs are represented by circular arcs that link paired bases and meet the circle at right angles. In this representation, a structure is free of pseudoknots if, and only if, no 2 base pair arcs intersect.

The triangular image in Figure 6 is referred to as an RNA structure dot plot. A dot is placed in the i^{th} row and j^{th} column of a triangular array to represent the base pair $i.j$. This figure once more represents the same *B. subtilis* folding. Many RNA secondary structures may be superimposed on a single dot plot, and this fact makes them useful for comparative analyses. Figures 7, 8 and 9 give some details on how secondary structure looks in dot plot format.

RNA Folding by Energy Minimization

Base pair dependent energy rules.

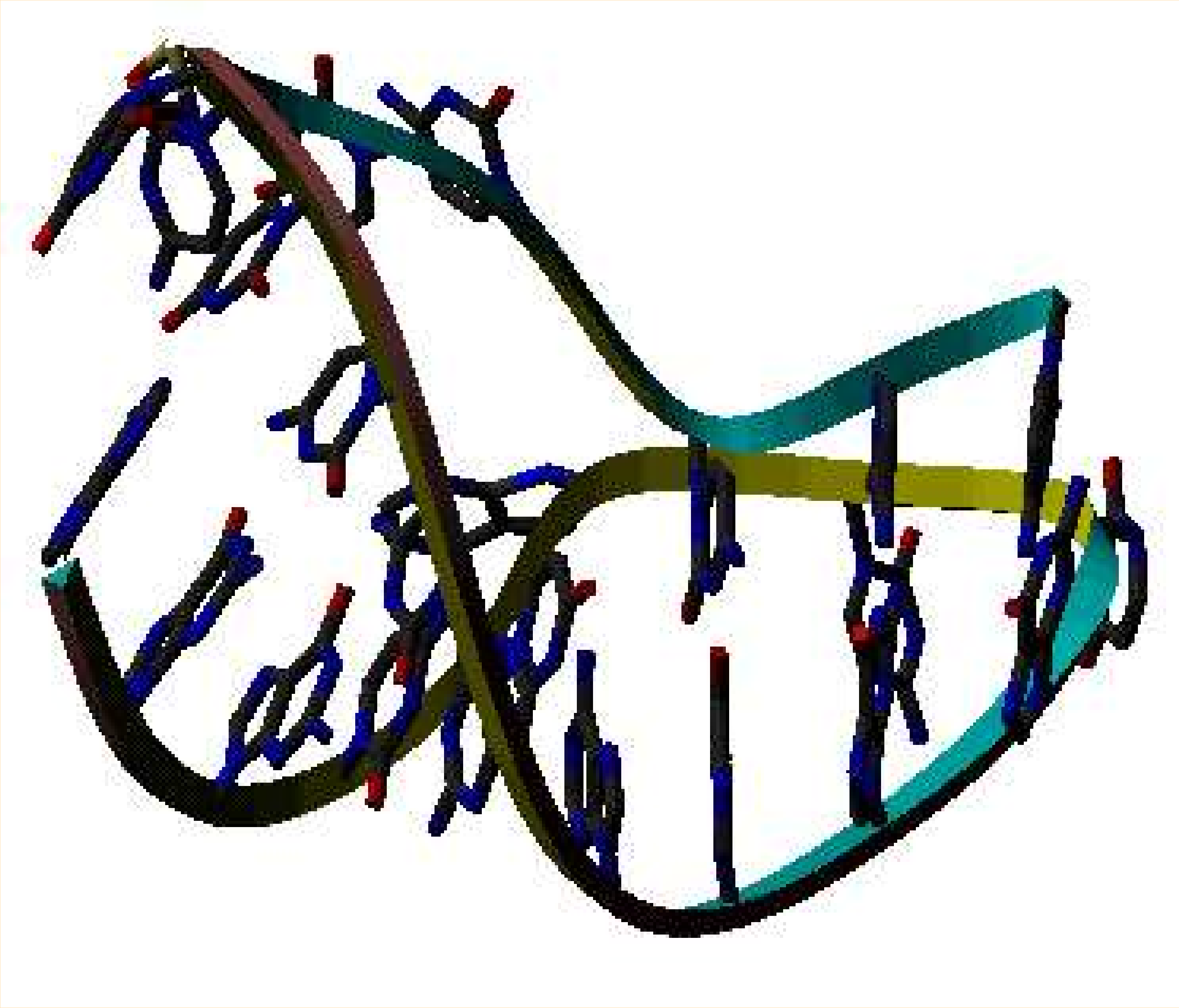
The quickest and easiest route to RNA structure prediction is through the use of simple energy rules. One way is to assign an energy to each base pair in a secondary structure. That is, there is a function e such that $e(r_i, r_j)$ is the energy of a base pair. The energy, $E(\mathbf{S})$, of the entire structure, is then given by:

$$E(\mathbf{S}) = \sum_{i.j \in \mathbf{S}} e(r_i, r_j). \tag{1}$$

Reasonable values of e at 37° are -3, -2 and -1 kcal/mole for GC, AU and GU base pairs, respectively. Unfortunately, such simple minded rules are insufficient to capture the destabilizing effects of various loops, or the nearest neighbor interactions in helices and loops. More sophistication is required.

For base pair dependent energy rules, a particularly simple recursive algorithm is available that computes minimum energy foldings.

Figure 2: A 3D model of a pseudoknot. The 2 helices in this structure are stacked coaxially.



The corresponding secondary structure is:

```
              A-C
3'- A-G-G-C-U/   U
    U-C-C-G-A-G-G-G
     U        C-C-C - 5'
      C--U--C/
```

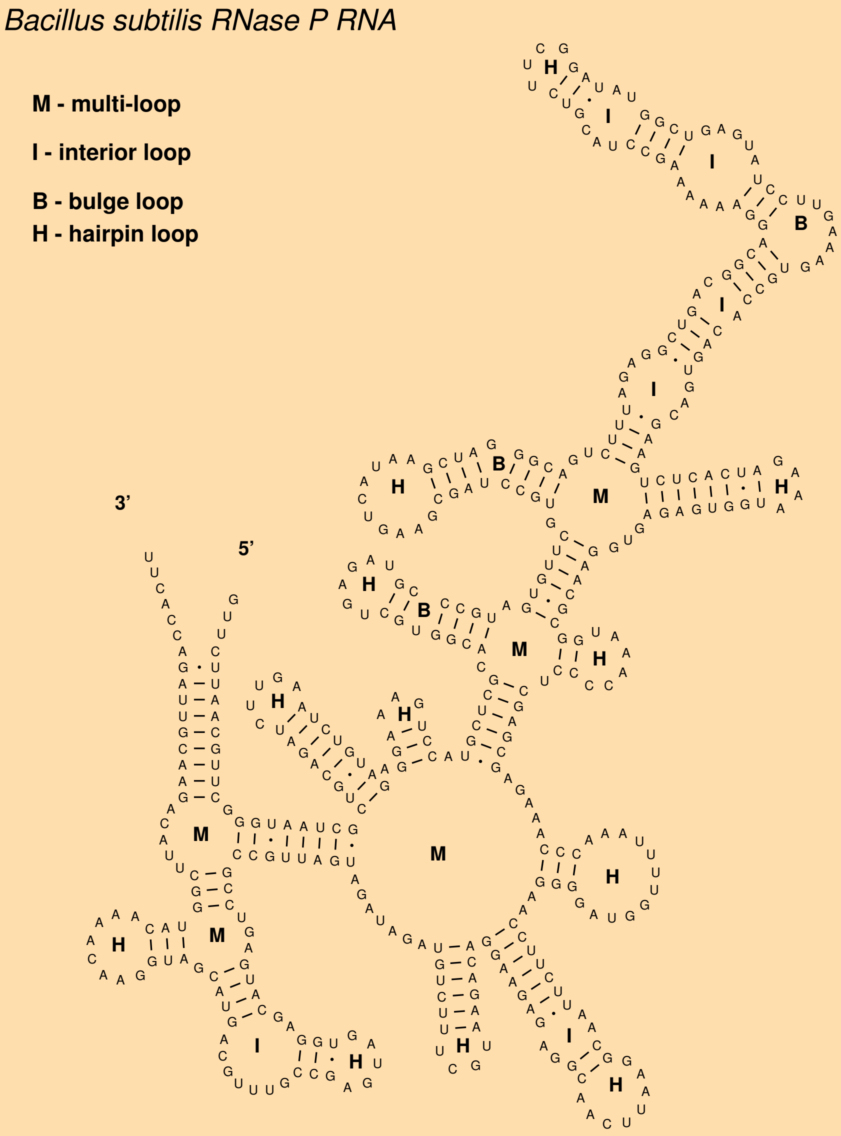


Figure 3: A computer predicted folding of *Bacillus subtilis* RNase P RNA.

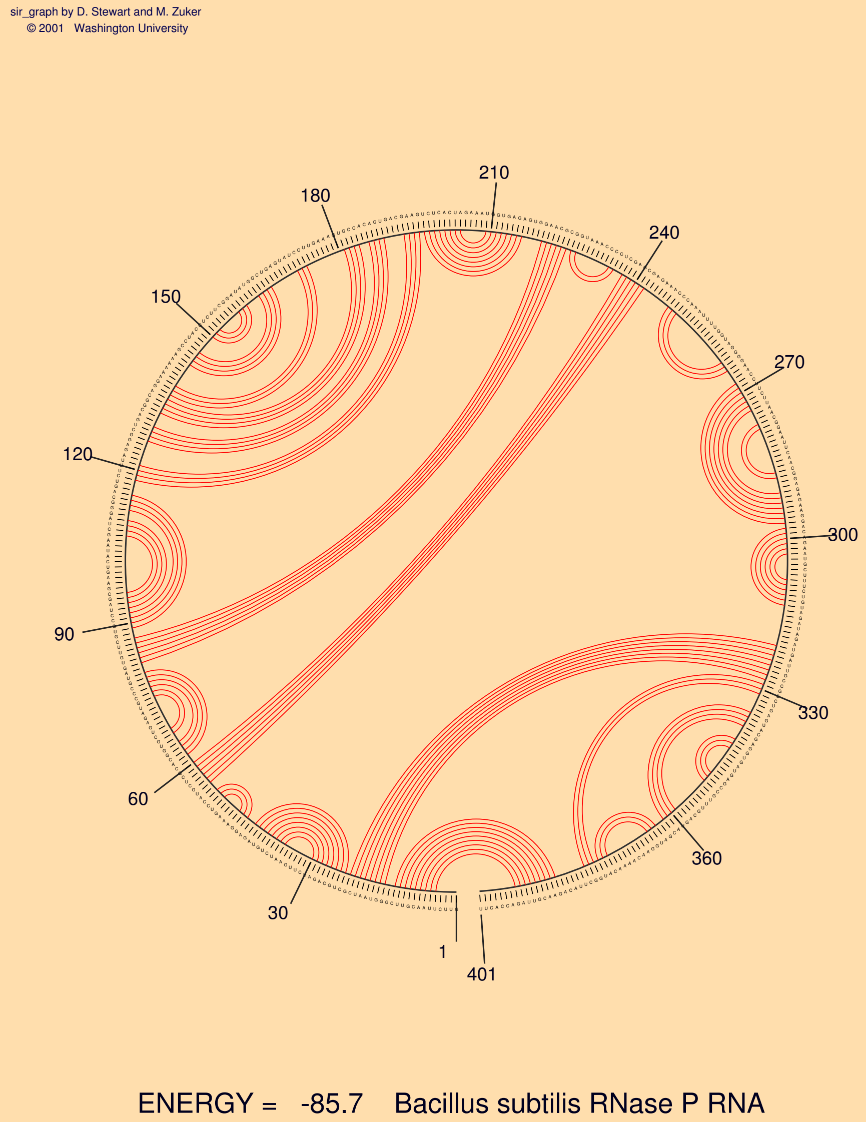

sir_graph by D. Stewart and M. Zuker
© 2001 Washington University
210
180
240
150
270
120
300
90
330
60
360
30
1
401
ENERGY = -85.7 Bacillus subtilis RNase P RNA

Figure 4: A circular representation of the *B. subtilis* folding.

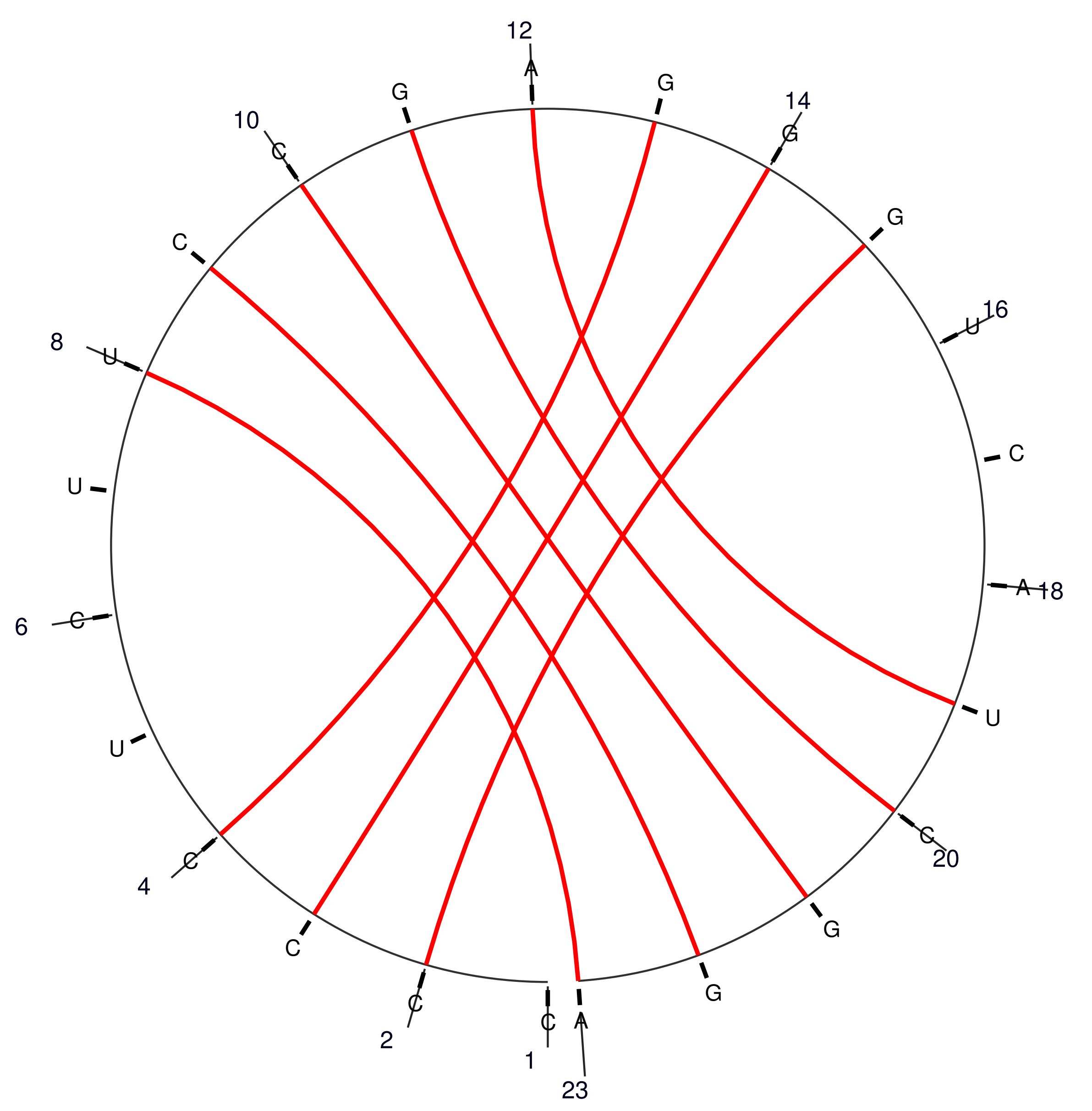


Figure 5: Circular plot for the Pseudoknot structure in Figure 2

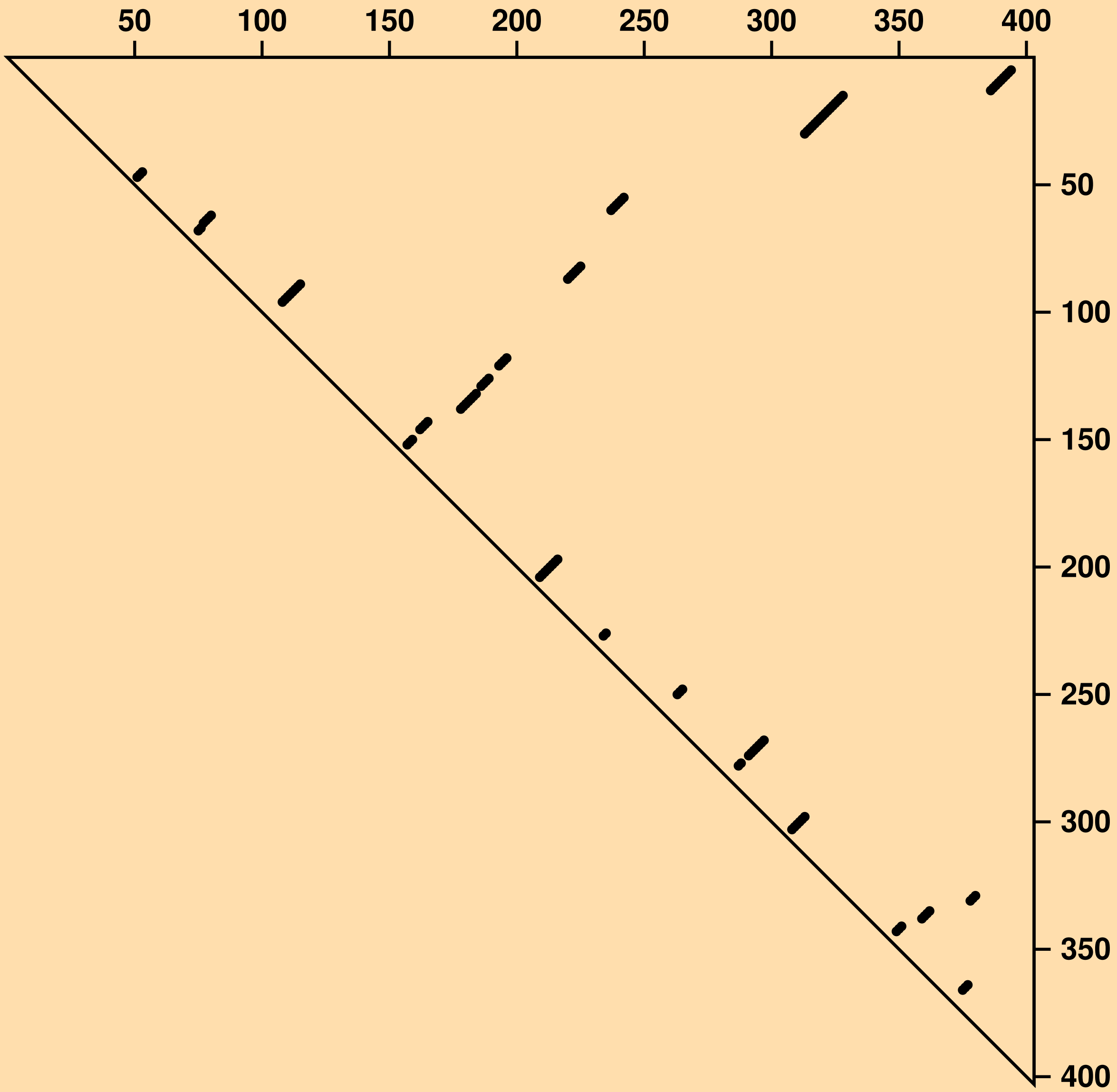
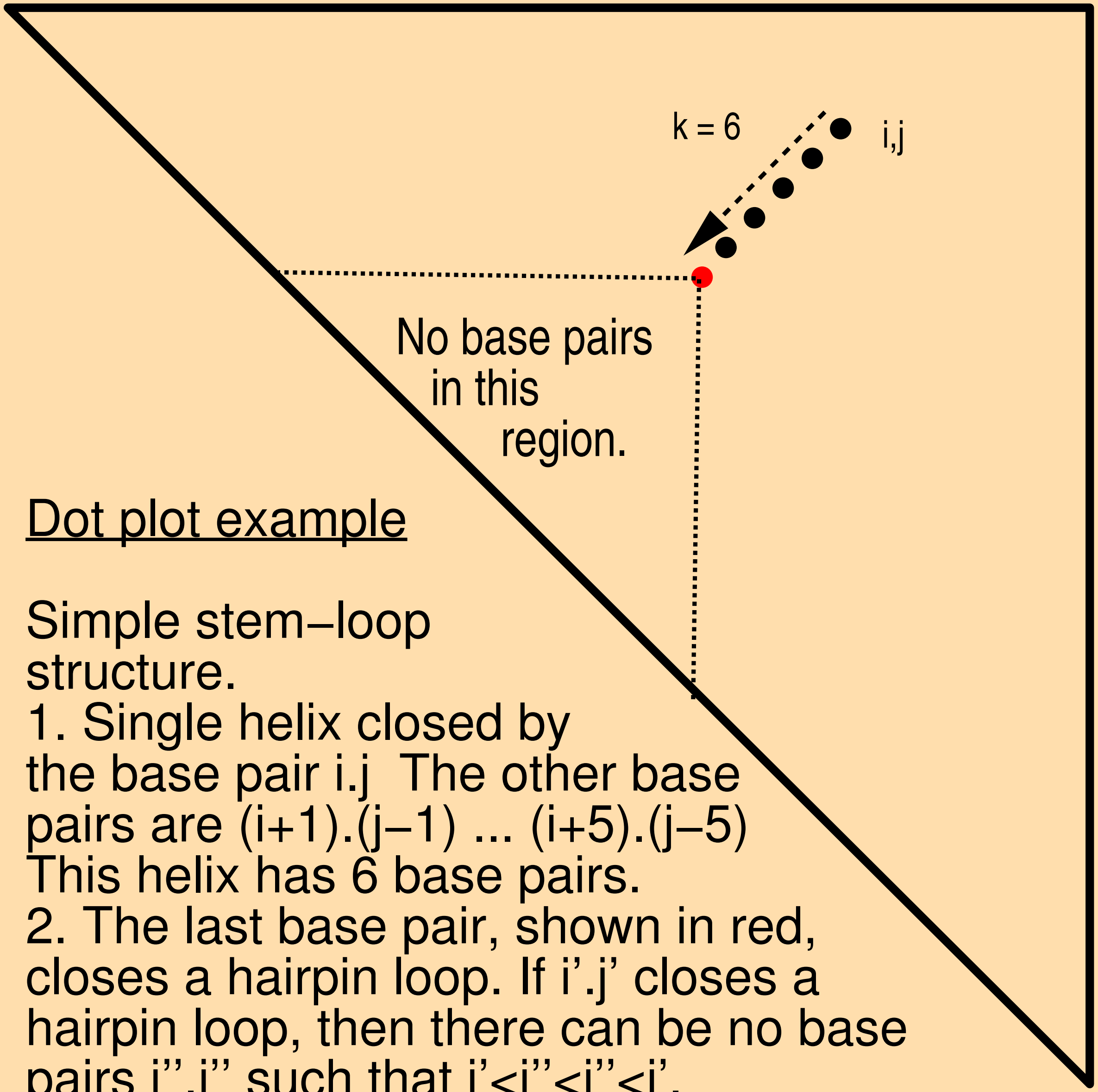


Figure 6: Dot plot representation of the *B. subtilis* folding.

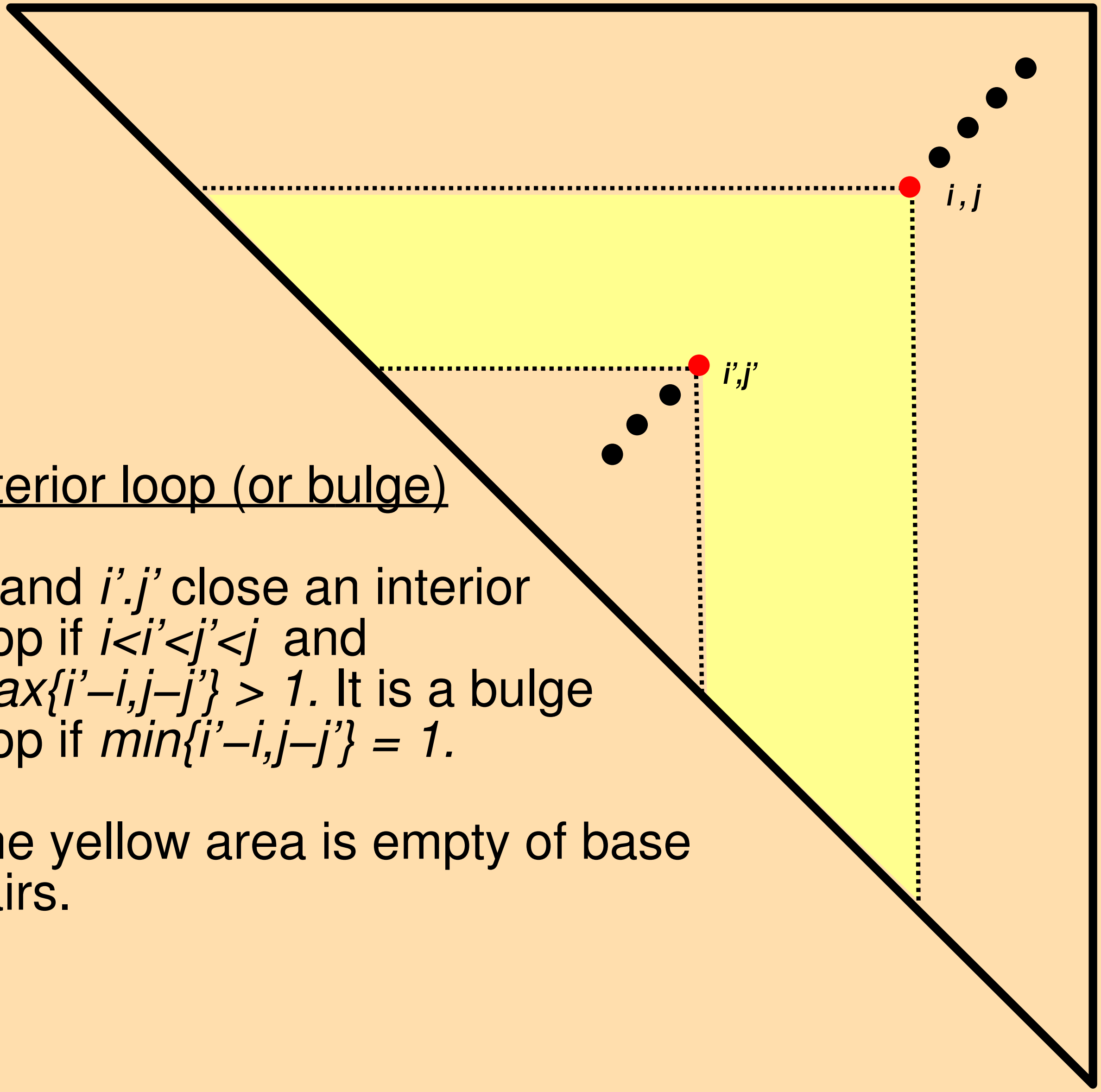


Dot plot example

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’.

Figure 7: An illustration of a helix and hairpin loop in dot plot format.

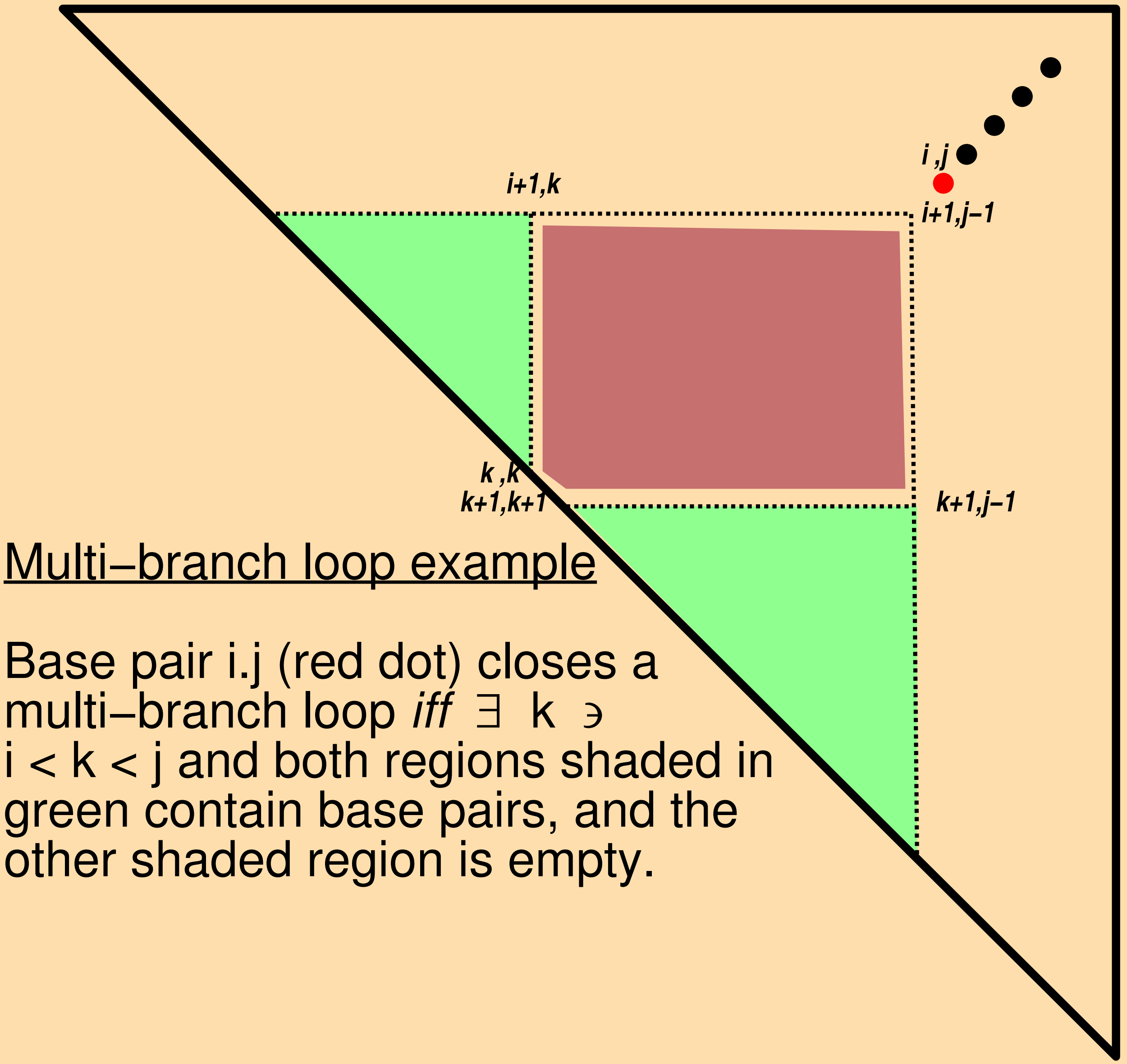


Interior loop (or bulge)

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.

Figure 8: An illustration of an interior loop in dot plot format.



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.

Figure 9: An illustration of a multi-branch loop in dot plot format. The division of the main triangle with right angle at $(i+1).(j-1)$ into 2 sub-triangles is a minimum requirement. There can be $m-1$ sub-triangles is the multi-branch loop has m adjacent stems.

Let

$$E = \min E(\mathbf{S})$$

where **S** ranges over all secondary structures. The energy for pairing r_i with r_j is given by $e(i,j)$. Auxiliary numbers $E_{i,j}$ are computed for all fragments, $r_i \dots r_j$ of the RNA.

$$\begin{aligned} E_{ij} &= 0 \text{ for } j-i<4, \qquad \text{otherwise} \\ E_{ij} &= \min\left\{E_{i+1,j}, E_{i,j-1}, e(i,j)\right. \\ &+ \left. E_{i+1,j-1}, \min_{k=i+1}^{j-1}\left(E_{i,k}+E_{k+1,j}\right)\ \right\} \end{aligned} \tag{2}$$

That is:

- Fragments of length ≤ 4 have 0 folding energy, since they cannot fold. Otherwise,
- r_i is unpaired, or
- r_j is unpaired, or
- r_i and r_j pair with each other, or
- r_i and r_j both pair, but **not** with each other. In this case, r_i pairs with r_{k1} and r_j pairs with r_{k2}, where $i < k1 < k2 < j$. The k in the recursion can be any integer satisfying $k1 \le k < k2$.

A folding with minimum energy is computed using a *traceback* algorithm. Start: Set $i = 1$ and $j = n$. Put i and j on to the "traceback stack".
Recursion:

1. If the traceback stack is empty, the traceback terminates. Otherwise, take i and j from the traceback stack.
2. If $E_{i+1,j} = E_{i,j}$, then i is not paired.
   (a) If $j-i>3$, set $i = i+1$ and continue with 2.
   (b) If $j-i \le 3$, continue with 1.

   Otherwise, continue with 3.
3. If $E_{i,j-1} = E_{i,j}$, then j is not paired.
   (a) If $j-i>3$, set $j = j-1$ and continue with 3.
   (b) If $j-i \le 3$, continue with 1.

   Otherwise, continue with 4.
4. If $E_{i,j} = e(i,j) + E_{i+1,j-1}$, then r_i pairs with r_j. Add $i.j$ to the list of base pairs, set $i = i+1$ and $j = j-1$ and continue with 2. Otherwise, continue with 5.
5. If $E_{i,j} = E_{i,k} + E_{k+1,j}$, for some $k \in (i,j)$, put the fragment $k+1 \dots j$ on to the traceback stack (i is $k+1$ and j is the current j) and deal with $i \dots k$ by setting $j = k$ and continue with 2. (Note that some k must exist if this point is reached.)

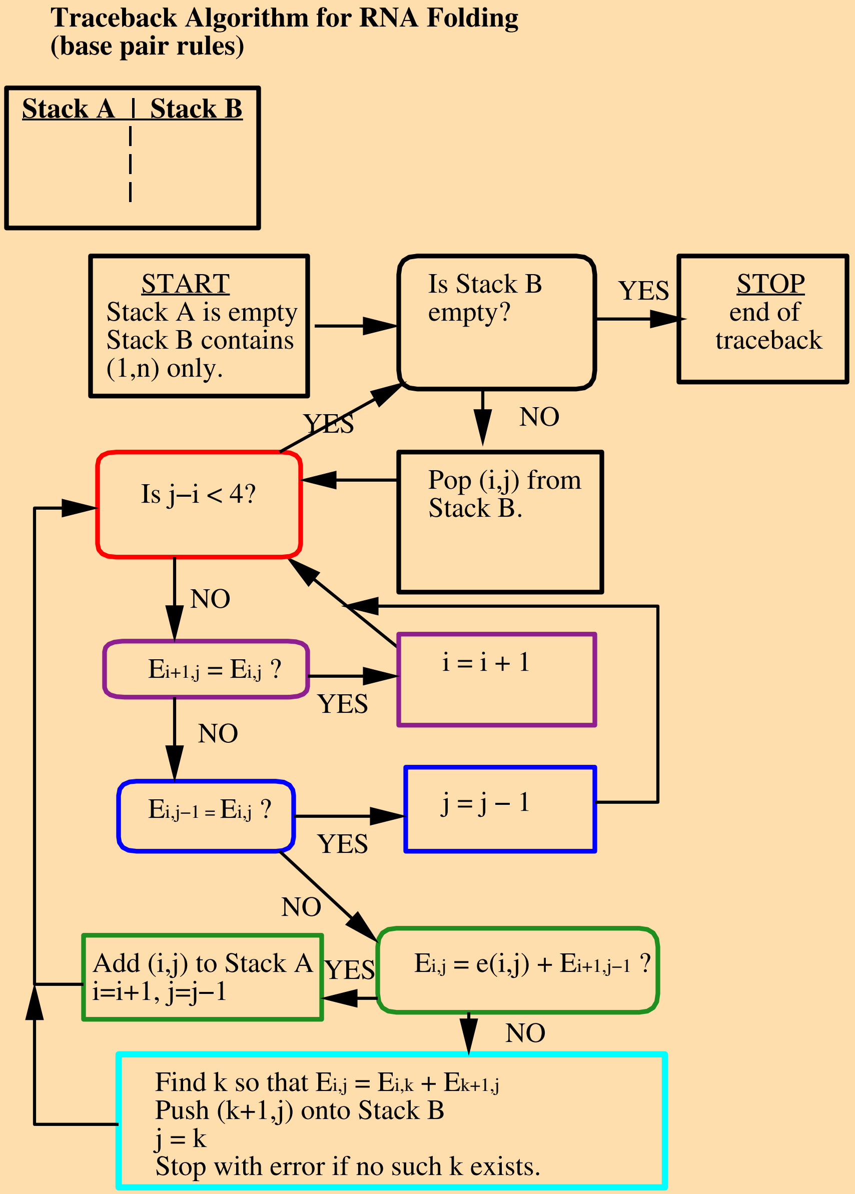


Figure 10: An illustrated traceback chart for RNA folding using base pair dependent energy rules.

Fill and Traceback Example

Given the sequence **R** =
GCAGCACCCAAAGGGAAUAUGGGAUACGCGUA. The base pair folding energies are e(i,j) = -3, -2 and -1 for GC (CG), AU (UA) and GU (UG) base pairs, respectively.

The matrix E is given below. $E_{i,j}$ appears in row i and column j of the triangular array.

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
A	G	C	A	C	C	C	A	A	A	G	G	G	A	A	U	A	U	G	G	G	A	U	A	C	G	C	G	U	A		
0	0	-3	-3	-3	-3	-6	-6	-6	-6	-6	-9	-12	-12	-12	-14	-14	-14	-16	-17	-17	-17	-18	-18	-20	-21	-24	-24	-25	-25	G	1
0	0	0	0	0	-3	-3	-3	-3	-3	-6	-9	-9	-9	-9	-11	-11	-13	-16	-17	-17	-17	-17	-17	-20	-21	-22	-24	-24	-24	C	2
0	0	0	0	0	-3	-3	-3	-3	-3	-6	-6	-9	-9	-9	-11	-11	-13	-14	-14	-14	-14	-16	-16	-18	-18	-21	-21	-23	-23	A	3
	0	0	0	0	-3	-3	-3	-3	-3	-6	-6	-9	-9	-9	-11	-11	-12	-14	-14	-14	-14	-15	-15	-18	-18	-21	-21	-22	-22	G	4
		0	0	0	0	0	0	0	0	-3	-6	-9	-9	-9	-11	-11	-11	-14	-14	-14	-14	-15	-15	-17	-18	-20	-21	-21	-21	C	5
			0	0	0	0	0	0	0	-3	-6	-9	-9	-9	-11	-11	-11	-11	-11	-13	-13	-15	-15	-15	-18	-18	-18	-20	-20	A	6
				0	0	0	0	0	0	-3	-6	-9	-9	-9	-9	-9	-11	-11	-11	-13	-13	-13	-15	-15	-18	-18	-18	-18	-20	C	7
					0	0	0	0	0	-3	-6	-6	-6	-6	-6	-6	-8	-8	-10	-10	-10	-10	-12	-15	-15	-15	-18	-18	-18	C	8
						0	0	0	0	-3	-3	-3	-3	-3	-4	-4	-5	-7	-7	-7	-7	-8	-9	-12	-12	-15	-15	-15	-15	C	9
							0	0	0	0	0	0	0	0	-2	-2	-4	-4	-4	-4	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	10
								0	0	0	0	0	0	0	-2	-2	-4	-4	-4	-4	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	11
									0	0	0	0	0	0	-2	-2	-3	-3	-3	-3	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	12
										0	0	0	0	0	-1	-1	-2	-2	-2	-2	-3	-5	-6	-9	-9	-12	-12	-13	-13	G	13
											0	0	0	0	-1	-1	-2	-2	-2	-2	-3	-5	-6	-9	-9	-12	-12	-12	-12	G	14
												0	0	0	0	0	-2	-2	-2	-2	-2	-4	-6	-9	-9	-9	-9	-9	-11	G	15
													0	0	0	0	-2	-2	-2	-2	-2	-4	-6	-6	-6	-8	-8	-9	-11	A	16
														0	0	0	0	0	-1	-1	-2	-4	-6	-6	-6	-6	-7	-9	-11	A	17
															0	0	0	0	-1	-1	-2	-4	-6	-6	-6	-6	-7	-9	-11	U	18
																0	0	0	0	0	-2	-4	-4	-4	-4	-6	-7	-9	-9	A	19
																	0	0	0	0	-2	-2	-3	-3	-4	-6	-7	-7	-9	U	20
																		0	0	0	0	-1	-1	-3	-3	-6	-6	-7	-7	G	21
																			0	0	0	0	0	-3	-3	-6	-6	-6	-6	G	22
																				0	0	0	0	-3	-3	-3	-3	-3	-4	G	23
																					0	0	0	0	0	0	-1	-3	-4	A	24
																						0	0	0	0	0	-1	-2	-4	U	25
																							0	0	0	0	0	-2	-2	A	26
																								0	0	0	0	0	0	C	27
																									0	0	0	0	0	G	28
																										0	0	0	0	C	29
																											0	0	0	G	30
																												0	0	U	31
																													0	A	32

One possible traceback route is:

$E_{1,32} = E_{1,31}$

$E_{1,31} = e(1,31) + E_{2,30}$ Base pair (1,31)

$E_{2,30} = e(2,30) + E_{3,29}$ Base pair (2,30)

$E_{3,29} = E_{4,29}$

$E_{4,29} = e(4,29) + E_{5,28}$ Base pair (5,28)

$E_{5,28} = e(5,28) + E_{6,27}$ Base pair (6,27)

$E_{6,27} = E_{7,27} = E_{7,26}$

At this point, the “simple” methods to move down or left fail. The structure bifurcates.

$E_{7,26} = E_{7,15} + E_{16,26}$

Place (16,26) on the stack. Continue with (7,15)

$E_{7,15} = e(7,15) + E_{8,14}$ Base pair (7,15)

$E_{8,14} = e(8,14) + E_{7,13}$ Base pair (8,14)

$E_{9,13} = e(9,13)$ Base pair (9,13).

Take (16,26) from the stack.

$E_{16,26} = E_{17,26} = E_{18,26}$ Two dangling As.

$E_{18,26} = e(18,26) + E_{19,26}$ Base pair (18,26)

$E_{19,25} = e(19,25) + E_{20,25}$ Base pair (19,25)

$E_{20,24} = e(20,24)$ Base pair (20,24)

The traceback route is:

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
C	A	C	C	C	A	A	A	G	G	G	A	A	U	A	U	G	G	G	A	U	A	C	G	C	G	U	A		
-3	-3	-3	-3	-6	-6	-6	-6	-6	-9	-12	-12	-12	-14	-14	-14	-16	-17	-17	-17	-18	-18	-20	-21	-24	-24	-25	-25	G	1
0	0	0	-3	-3	-3	-3	-3	-6	-9	-9	-9	-9	-11	-11	-13	-16	-17	-17	-17	-17	-17	-20	-21	-22	-24	-24	-24	C	2
0	0	0	-3	-3	-3	-3	-3	-6	-6	-9	-9	-9	-11	-11	-13	-14	-14	-14	-14	-16	-16	-18	-18	-21	-21	-23	-23	A	3
0	0	0	-3	-3	-3	-3	-3	-6	-6	-9	-9	-9	-11	-11	-12	-14	-14	-14	-14	-15	-15	-18	-18	-21	-21	-22	-22	G	4
0	0	0	0	0	0	0	0	-3	-6	-9	-9	-9	-11	-11	-11	-14	-14	-14	-14	-15	-15	-17	-18	-20	-21	-21	-21	C	5
	0	0	0	0	0	0	0	-3	-6	-9	-9	-9	-11	-11	-11	-11	-11	-13	-13	-15	-15	-15	-18	-18	-18	-20	-20	A	6
		0	0	0	0	0	0	-3	-6	-9	-9	-9	-9	-9	-11	-11	-11	-13	-13	-13	-15	-15	-18	-18	-18	-18	-20	C	7
			0	0	0	0	0	-3	-6	-6	-6	-6	-6	-6	-8	-8	-10	-10	-10	-10	-12	-15	-15	-15	-18	-18	-18	C	8
				0	0	0	0	-3	-3	-3	-3	-3	-4	-4	-5	-7	-7	-7	-7	-8	-9	-12	-12	-15	-15	-15	-15	C	9
					0	0	0	0	0	0	0	0	-2	-2	-4	-4	-4	-4	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	10
						0	0	0	0	0	0	0	-2	-2	-4	-4	-4	-4	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	11
							0	0	0	0	0	0	-2	-2	-3	-3	-3	-3	-4	-6	-6	-9	-9	-12	-12	-14	-14	A	12
								0	0	0	0	0	-1	-1	-2	-2	-2	-2	-3	-5	-6	-9	-9	-12	-12	-13	-13	G	13
									0	0	0	0	-1	-1	-2	-2	-2	-2	-3	-5	-6	-9	-9	-12	-12	-12	-12	G	14
										0	0	0	0	0	-2	-2	-2	-2	-2	-4	-6	-9	-9	-9	-9	-9	-11	G	15
											0	0	0	0	-2	-2	-2	-2	-2	-4	-6	-6	-6	-8	-8	-9	-11	A	16
												0	0	0	0	0	-1	-1	-2	-4	-6	-6	-6	-6	-7	-9	-11	A	17
													0	0	0	0	-1	-1	-2	-4	-6	-6	-6	-6	-7	-9	-11	U	18
														0	0	0	0	0	-2	-4	-4	-4	-4	-6	-7	-9	-9	A	19
															0	0	0	0	-2	-2	-3	-3	-4	-6	-7	-7	-9	U	20
																0	0	0	0	-1	-1	-3	-3	-6	-6	-7	-7	G	21
																	0	0	0	0	0	-3	-3	-6	-6	-6	-6	G	22
																		0	0	0	0	-3	-3	-3	-3	-3	-4	G	23
																			0	0	0	0	0	0	-1	-3	-4	A	24
																				0	0	0	0	0	-1	-2	-4	U	25
																					0	0	0	0	0	-2	-2	A	26
																						0	0	0	0	0	0	C	27
																							0	0	0	0	0	G	28
																								0	0	0	0	C	29
																									0	0	0	G	30
																										0	0	U	31
																											0	A	32

The structure that was computed is shown in Figure 11 This optimal structure is not unique. Another possibility is:

```
            10
-  A  -- -    A GG
 GC GC  A CCCA A
 UG CG  U GGGU U  G
A  -  CA A    A AA
```

```
 30          20
```

This would be found by following the "-15"s to position (6,25).

Loop Dependent Energy Rules

What are loops?

K-LOOP DECOMPOSITION: If $i.j$ is a base pair in $\mathbf{S}$ and $i < k < j$, we say that k is *accessible* from $i.j$ if there is no $i'.j' \in \mathbf{S}$ such that $i < i' < k < j' < j$. Similarly, we say that the base pair $k.l$ is accessible if both k and l are accessible. The set of $(k-1)$ base pairs and k' single-stranded bases accessible from $i.j$ is called the k-loop closed by $i.j$. The "null" k-loop, $\mathbf{L}_0$ (or $\mathbf{L}_e$) consists of those single- and double-stranded bases accessible from no base pair. This is usually referred to as the "exterior loop", and its bases and base pairs are called "free" bases and "free" base pairs, respectively.
Then any secondary structure $\mathbf{S}$ partitions the sequence, $\mathbf{R}$, uniquely into k-loops $\mathbf{L}_0, \mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_m$ where $m > 0$ iff $\mathbf{S} \neq \emptyset$. Note that m is the number of base pairs in the secondary structure. Since each loop is uniquely determined by its closing base pair (except for $\mathbf{L}_0$), we can write $\mathbf{L}_{i.j}$ to denote the loop closed by the base pair $i.j$. We can then write:

$$\mathbf{R} = \mathbf{L}_0 \bigcup \left(\bigcup_{i.j \in \mathbf{S}} \mathbf{L}(i.j) \right) \tag{3}$$

This decomposition was first introduced by Sankoff[1]. The present definition follows Zuker and Sankoff[2] and Zuker[3], where the closing base pair is not contained in the k-loop. k-loops are also called k-cycles. k' is called the *size* of the k-loop. Energies are assigned to the k-loops, and the energy of a structure S is given by:

$$\begin{aligned} E(S) &= \sum_{i=0}^{m} e(\mathbf{L}_i) \\ &= e(\mathbf{L}_0) + \left(\sum_{i.j \in \mathbf{S}} e(\mathbf{L}(i.j)) \right) \end{aligned} \tag{4}$$

Note that e is now a function of k-loops instead of a function of base pairs.
Biochemists had developed their own nomenclature for k-loops long before any formal definition was given. The various cases and sub-cases are as follows:

1. $k = 1$: A 1-loop is called a *hairpin* loop.
2. $k = 2$: Let $i'.j'$ be the base pair accessible from $i.j$. Then the 2-loop is called a
   (a) *stacked pair* if $i' - i = 1$ and $j - j' = 1$, a
   (b) *bulge loop* if $i' - i > 1$ or $j - j' > 1$, but not both, and an
   (c) *interior loop* if both $i' - i > 1$ and $j - j' > 1$.
3. $k \geq 3$: These k-loops are called *multi-branched* or *multi-loops*.

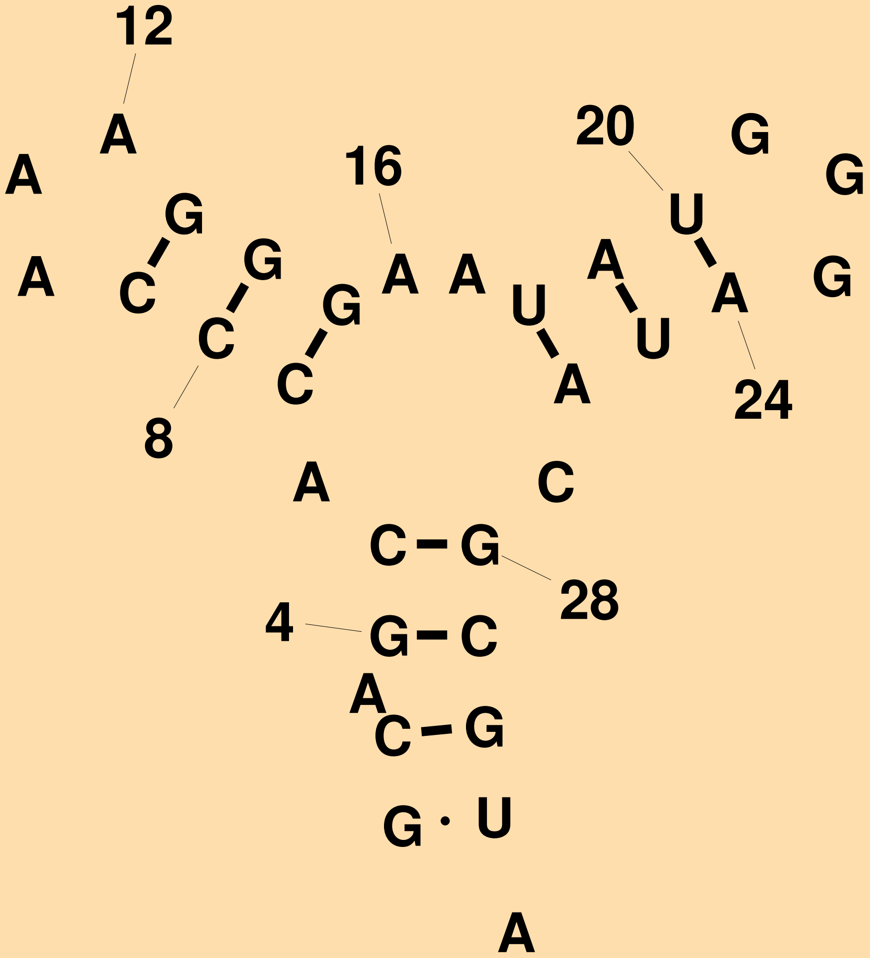


Figure 11: Computed folding for the sample sequence.

```
DESTABILIZING ENERGIES BY SIZE OF LOOP
SIZE  INTERNAL   BULGE     HAIRPIN
----------------------------------
1        .         3.8        .
2        .         2.8        .
3        .         3.2       5.6
4       1.7        3.6       5.5
5       1.8        4.0       5.6
6       2.0        4.4       5.3
7       2.2        4.6       5.8
8       2.3        4.7       5.4
             ...
30      3.7        6.1       7.7
```

Figure 12: The loop.dg or loop.TC contains size based free energy increments for hairpin, bulge and interior loops up to size 30. Entries with '.' are undefined.

We introduce some additional notation here, by defining some special cases for e. We let $eh(i,j)$ be the free energy of the hairpin loop closed by the base pair $i.j$. For 2-loops, we let $ebi(i,j,i',j')$ be the free energy of the bulge or interior loop closed by the 2 base pairs $i.j$ and $i'.j'$. As a special case $es(i,j)$ is defined to be $ebi(i,j,i{+}1,j{-}1)$, the base pair and stacking energy of the 2 adjacent base pairs, $i.j$ and $i{+}1.j{-}1$. The terms $l_s(\mathbf{L})$ and $l_d(\mathbf{L})$ denote the number of single-stranded bases and base pairs in a loop, respectively. The size of a 1 or 2-loop is defined as $l_s(\mathbf{L})$.

Hairpin loops

Polymer theory predicts that the free energy increment, $\delta\delta G$, of a hairpin loop should be

$$\delta\delta G = 1.75 \times RT \times \ln(l_s), \tag{5}$$

where T is absolute temperature and R is the universal gas constant (1.9872 cal/mol/K). The factor 1.75 would be 2 if the chain were not self-avoiding in space. This free energy is entirely entropic. In reality, we use tabulated values for $\delta\delta G$ for l_s from 3 to 30. These values are based on measurements and interpolations of measurements, and are stored in a file named loop.dg, loop.dat or loop.TC, where TC is a temperature (integral) in °C. The suffix "dat" is used for free energies at 37 degrees and refers to version 3.0 of mfold, for which we have only these free energies. The suffixes "dg" and "dh" refer to free energies at 37 degrees and enthalpies, respectively. The enthalpies are assumed to be constant over the range of folding temperatures that are used. The "dg" and "dh" files are combined to produce free energies at any temperature. Thus loop.dg and loop.37 refer to the same file. The same suffix convention holds for other files defined below. Equation 5 is used to extrapolate beyond size 30. Thus, for $l_s > 30$,

$$\delta\delta G = \delta\delta G_{30} + 1.75 \times RT \times \ln(l_s/30). \tag{6}$$

Figure 12 shows the information stored in the loop file.
In addition, the effects of *terminal mismatched pairs* are taken into account for hairpin loops of size greater than 3. For loops of size 4 and greater closed by a base pair $i.j$, an extra $\delta\delta G$ is applied. This is referred to as the *terminal mismatch* free energy for hairpin loops. These parameters are stored in a file named

```
        5′ --> 3′            5′ --> 3′
           WX                   CX
           ZY                   GY
        3′ <-- 5′            3′ <-- 5′
    Y: A   C   G   U     A    C    G    U
      ---------------    ------------------
X: A | AA  AC  AG  AU  -1.5 -1.5 -1.4 -1.8
   C | CA  CC  CG  CU  -1.0 -0.9 -2.9 -0.8
   G | GA  GC  GG  GU  -2.2 -2.0 -1.6 -1.2
   U | UA  UC  UG  UU  -1.7 -1.4 -1.9 -2.0
```

Figure 13: On the left, a typical 4 × 4 table. The pairs WX and YZ are covalently linked. WZ is assumed to be the closing base pair of a hairpin loop, and XY is the mismatched pair. ‘X’ refers to row , and ‘Y’ to column, in order A, C, G and U. Thus ‘GU’ is the same as ‘34’ and is the mismatch free energy for a GU mismatch (X=G and Y=U). On the right is a sample table for W=C and Z=G.

```
 Seq    Energy
 -------------
  GGGGAC -3.0
   ...
  CGAAGG -2.5
  CUACGG -2.5
   ...
  GUGAAC -1.5
  UGGAAA -1.5
```

Figure 14: Sample “distinguished” tetraloops together with the free energy bonuses, in kcal/mole, attached to them. These entries include the closing base pair of the loop. Triloops are not shown since they are not currently in use for RNA folding.

tstackh.dg or tstackh.TC, as above. The data are arranged in 4 × 4 tables that each comprise 4 rows and columns. Figure 13 illustrates how the parameters are stored.

Both the loop and tstackh files treat hairpin loops in a generic way, and assume no special structure for the bases in the loop. We know that this is not true in general. For example, the anti-codon loop of tRNA is certainly not unstructured. For certain small hairpin loops, special rules apply. Hairpin loops of size 3 are called triloops and those of size 4 are called tetraloops. Files of “distinguished” triloops and tetraloops have been created to store the free energy bonus assigned to those loops. These parameters are stored in files triloop.dg and tloop.dg, respectively (or triloop.TC and tloop.TC for a specific temperature, TC). Some typical entries are given in Figure 14

Finally, there are some special hairpin loop rules derived from experiments that will be defined explicitly here. A hairpin loop closed by r_i and r_j ($i < j$) called a “GGG” loop if $r_{i-2} = r_{i-1} = r_i = G$ and $r_j = U$. Such a loop receives a free energy bonus that is stored in the miscloop.dg or miscloop.TC file, which contains a variety of miscellaneous, or extra free energy parameters. Another special case is the “poly-C” hairpin loop, where all the single stranded bases are C. If the loop has size 3, it is given a free energy penalty of $c3$. Otherwise, the penalty is $c_2 + c_1 \times l_s$. The constants c_1, c_2 and c_3 are all stored in the miscloop file.

```
                5′ --> 3′
                   CX
                   GY
                3′ <-- 5′
         Y:  A    C    G    U
        ----------------------
     X: A |  .    .    .   -2.1
        C |  .    .   -3.3  .
        G |  .   -2.4  .   -1.4
        U | -2.1  .   -2.1  .
```

Figure 15: Sample free energies in kcal/mole for CG base pairs stacked over all possible base pairs, XY. X refers to row and Y refers to column, in the order A, C, G and U respectively. Entries denoted by an isolated period, ‘.’, are undefined, and may be considered as $+\infty$.

To summarize, we can write the free energy, $\delta\delta G_H$ of a hairpin loop as:

$$\delta\delta G_H = \delta\delta G_H^1 + \delta\delta G_H^2 + \delta\delta G_H^3 + \delta\delta G_H^4, \tag{7}$$

where

1. $\delta\delta G_H^1$ is the size dependent contribution from the loop file, or from equation 6 for sizes > 30,
2. $\delta\delta G_H^2$ is the terminal mismatch stacking free energy, taken from the tstackh file (0 for hairpin loops of size 3),
3. $\delta\delta G_H^3$ is the bonus free energy for triloops or tetraloops listed in the TRILOOP or TLOOP files. This value is 0 for loops not listed in the TRILOOP or TLOOP files and for loop sizes > 4,
4. $\delta\delta G_H^4$ is the bonus or penalty free energy for special cases not covered by the above.

Stacks, Bulges and Interior Loops

A 2-loop of size 0 is called a *stacked pair*. This refers to the stacking between the $i.j$ and immediately adjacent $i+1.j-1$ base pair contained in the loop. Free energies for these loops are stored in a file named stack.dg, or stack.TC, where TC is a temperature, as defined above. The layout is the same as for the tstackh file. A portion of such a file is given in Figure 15. A group of 2 or more consecutive base pairs is called a *helix*. The first and last are the closing base pairs of the helix. They may be written as $i.j$ and $i'.j'$, where $i < i' < j' < j$. Then $i.j$ is called the external closing base pair and $i'.j'$ is called the internal closing base pair. This nomenclature is used for circular RNA as well, even though it depends on the choice of origin.

Only Watson-Crick and wobble GU pairs are allowed as *bona fide* base pairs, even though the software is written to allow for any base pairs. The reason is that nearest neighbor rules break down for non-canonical, even GU base pairs, and that mismatches must instead be treated as small, symmetric interior loops. Note that the stacks $\begin{array}{ccc} 5' & --> & 3' \\ & WX & \\ & ZY & \\ 3' & <-- & 5' \end{array}$ *and* $\begin{array}{ccc} 5' & --> & 3' \\ & YZ & \\ & XW & \\ 3' & <-- & 5' \end{array}$ are identical, and yet formally different for $W \neq Y$ and $X \neq Z$. These stacked pairs are stored twice in the

file, and the “mfold” software checks for symmetry. This is an example of built in redundancy as a check on precision.
A 2-loop, $\mathbf{L}$, of size > 0 is called a *bulge loop* if $l_s^1(\mathbf{L}) = 0$ or $l_s^2(\mathbf{L}) = 0$, and an interior loop if **both** $l_s^1(\mathbf{L}) > 0$ and $l_s^2(\mathbf{L}) > 0$.
Bulge loops up to size 30 are assigned free energies from the loop file (See Figure 12). For larger bulge loops, equation 6 is used. When a bulge loop has size 1, the stacking free energy for base pairs $i.j$ and $i'.j'$ are used (from the stack file).
Interior loops have size ≥ 2. If $l_s^1(\mathbf{L}) = l_s^2(\mathbf{L})$, the loop is called *symmetric*; otherwise, it is *asymmetric*, or lopsided. The asymmetry of an interior loop, a($\mathbf{L}$) is defined by:

$$a(\mathbf{L}) = |l_s^1(\mathbf{L}) - l_s^2(\mathbf{L})|. \tag{8}$$

The free energy, $\delta\delta G_I$, of an interior loop is the sum of 4 components:

$$\delta\delta G_I = \delta\delta G_I^1 + \delta\delta G_I^2 + \delta\delta G_I^3 + \delta\delta G_I^4. \tag{9}$$

1. $\delta\delta G_I^1$ is the size dependent contribution from the loop file, or from equation 6 for sizes > 30.

2. $\delta\delta G_I^2$ and $\delta\delta G_I^3$ are terminal mismatch stacking free energies, taken from the tstacki file. The format of this file is identical to the format of the tstackh file. There are 2 terms because of the terminal stacking of both r_{i+1} and r_{j-1} on the $i.j$ base pair, and of both $r_{i'-1}$ and $r_{j'+1}$ on the $i'.j'$ base pair. This may be visualized as

$$\begin{array}{ccccccccccc} 5'- & r_i & - & r_{i+1} & -3' & & 5'- & r_{j'} & - & r_{j'+1} & -3' \\ & \bullet & & \circ & & \text{and} & & \bullet & & \circ & \\ 3'- & r_j & - & r_{j-1} & -5' & & 3'- & r_{i'} & - & r_{i'-1} & -5', \end{array}$$

   where $\bullet$ denotes a base pair and $\circ$ denotes a mismatched pair.

3. $\delta\delta G_I^4$ is the asymmetry penalty, and is a function of a($\mathbf{L}$) defined in equation 8. The penalty is 0 for symmetric interior loops. The asymmetric penalty free energies come from the miscloop.dg or miscloop.TC file.

Equation 9 is now used only for loops of size > 4 or of asymmetry > 1. This means that special rules apply to 1×1, 1×2 and 2×2 interior loops. Free energies for these symmetric and almost symmetric interior loops are stored in files sint2.dg, asint1x2.dg and sint4.dg, respectively. As above, the suffix TC is used in place of dg when explicit attention is paid to temperature. These files list all possible values of the single stranded bases, and all possible Watson-Crick and GU base pair closings. The sint2 file comprises a 6×6 array of 4×4 tables. There is a table for all possible 6×6 closing base pairs. The free energy values for each choice of closing base pairs are arranged in 4×4 tables. The term “closing base pairs” refers to the closing base pair of the loop and the contained base pair of the loop, as in the strict definition of a loop. An example of such a table is given in Figure 16.
The asint1x2 file comprises a 24 row by 6 column array of 4×4 tables. There is a 4×4 table for all possible $6 \times 6 \times 4$ closing base pairs and choice of one of the single stranded bases. The free energy values for each choice of closing base pairs and a single stranded base are arranged in 4×4 tables. An example of these tables is given in Figure 16.
Finally, the sint4 file contains 36 16×16 tables, 1 for each pair of closing base pairs. A 2×2 interior loop can have 4^4 combinations of single stranded bases. If, for example, the loop is closed by a GC base pair and an AU base pair, we can write it as:

```
5′ ------> 3′
 G \/ \_/ A
 C /\  |  U
3′ <------ 5′
```

```
        5' --> 3'               5' --> 3'
            X                       X
           C A                     C  A
           G T                     G  U
            Y                       YA
        3' <-- 5'               3' <-- 5'
 Y:  A    C    G    T    Y:  A    C    G    U
 ---------------------   ---------------------
A | 1.1  2.1  0.8  1.0  A | 3.2  3.0  2.4  4.8
C | 1.7  1.8  1.0  1.4  C | 3.1  3.0  4.8  3.0
G | 0.5  1.0  0.3  2.0  G | 2.5  4.8  1.6  4.8
T | 1.0  1.4  2.0  0.6  U | 4.8  4.8  4.8  4.8
```

Figure 16: Left: Free energies for all 1 × 1 interior loops in DNA closed by a CG and an AT base pair. Right: Free energies for all 1 × 2 interior loops in RNA closed by a CG and an AU base pair, with a single stranded U 3' to the double stranded U. As in similar Figures, X refers to row and Y to column.

Both the large 'X' and large 'Y' refer to an unmatched pair of bases that are juxtaposed. They can each take on 16 different values, from 'AA','AC', ..., to 'UU', or 1 to 16, respectively. The number in row 'X' and column 'Y' of the table is the free energy of the 2 × 2 interior loop with the indicated single stranded bases. Figure 17 shows the full table for the CG and AU closing base pairs.

Some special rules apply to 2-loops. A stacked pair that occurs at the end of a helix has a different free energy than if it were in the middle of a helix. Because of the availability and precision of data, we distinguish between GC closing and non-GC closing base pairs. In particular, a penalty (terminal AU penalty) is assigned to each non-GC closing base pair in a helix. The value of this penalty is stored in the MISCLOOP file.

NON-GC CLOSING PENALTY

Because free energies are assigned to loops, and not to helices, there is no *a priori* way of knowing whether or not a stacked pair will be terminal or not. For this reason, the terminal AU penalty is built into the TSTACKH and TSTACKI tables. For bulge, multi-branch and exterior loops, the penalty is applied explicitly. In all of these cases, the penalty is formally assigned to the adjacent loop, although it really belongs to the helix.

HIGHLY ASYMMETRIC INTERIOR LOOPS

A "Grossly Asymmetric Interior Loop (GAIL)" is an interior loop that is $1 \times n$, where $n > 2$. The special "GAIL" rule that is used in this case substitutes AA mismatches next to both closing base pairs of the loop for use in assigning terminal stacking free energies from the TSTACKI file.

EXPEDIENT RULES FOR MULTI-LOOPS

Because so little is known about the effects of multi-branch loops on RNA stability, we assign free energies in a way that makes the computations easy. This is the justification for the use of an *affine* free energy penalty for multi-branch loops. The free energy, $\delta\delta G(\mathbf{L})$, is given by:

$$\delta\delta G(\mathbf{L}) = a + b \times l_s(\mathbf{L}) + c \times l_d(\mathbf{L}) + \delta\delta G_{stack}, \tag{10}$$

where a, b and c are constants that are stored in the miscloop file and $\delta\delta G_{stack}$ includes stacking interactions that will be explained below. This simple energy function allows the dynamic programming algorithm used by "mfold" to find optimal multi-branch loops in time proportional to n^3. It would take exponentially increasing time (with sequence length) to use a more appropriate energy function derived from Jacobson-Stockmeyer theory [4] that grows logarithmically with

```
5' -------> 3'
 G \/ \_/ C
 C /\  |  G
3' <------- 5'
```

Y:	A	A	A	C	C	C	G	G	U	U	U
	A	C	G	A	C	U	A	U	C	G	U
AA	1.5	1.2	-0.5	1.2	1.8	0.80	0.10	-0.7	1.9	-0.3	1.5
AC	1.2	0.9	-0.8	0.9	0.9	0.00	-0.20	-2.0	1.0	-1.6	0.2
AG	0.1	-0.1	-1.9	-0.2	0.9	-0.10	-1.30	-1.3	0.9	-0.9	0.9
CA	1.2	1.0	-0.8	0.9	1.0	0.00	-0.10	-1.9	1.0	-1.5	0.2
CC	1.8	1.0	0.2	0.9	1.0	0.00	0.90	-0.9	1.0	-0.5	0.2
X CU	1.9	1.0	0.3	1.0	1.0	0.00	0.90	-0.9	1.1	-0.5	0.3
GA	-0.5	-0.8	-2.6	-0.8	0.2	-0.80	-1.90	-1.9	0.3	-1.5	0.3
GG	1.1	0.9	-0.9	0.8	1.5	0.50	-0.20	-1.0	1.5	-0.6	1.1
GU	-0.3	-1.5	-1.5	-1.6	-0.5	-1.50	-0.90	-4.5	-0.5	-4.1	-0.5
UC	0.8	0.0	-0.8	0.0	0.0	-1.00	-0.10	-1.9	0.0	-1.5	-0.7
UG	-0.7	-1.9	-1.9	-2.0	-0.9	-1.90	-1.30	-4.9	-0.9	-4.5	-0.9
UU	1.5	0.2	0.3	0.2	0.2	-0.70	0.90	-0.9	0.3	-0.5	-0.5

Figure 17: Free energies for all 2×2 interior loops in RNA closed by a GC and a CG base pair. Values of ‘X’ or ‘Y’ that correspond to bases that could form Watson-Crick pairs have been removed for brevity.

```
        X                     X
------------------    ------------------
 A    C    G    U      A    C    G    U
------------------    ------------------
    5′ --> 3′             5′ --> 3′
       CX                    C
       G                     GX
    3′ <-- 5′             3′ <-- 5′
-1.7 -0.8 -1.7 -1.2   -0.2 -0.3  0.0  0.0
```

Figure 18: Free energies for all possible single stranded bases that are adjacent to a CG base pair. 'X' refers to column. Note that the $3'$ dangling free energies are larger in magnitude than the $5'$ dangling free energies.

$l_s(\mathbf{L})$. In the "efn2" program that recalculates folding free energies using more realistic rules (defined below), equation 10 is replaced by:

$$\begin{aligned}\delta\delta G(\mathbf{L}) = \quad & a + 6b + 1.75 \times RT \times \ln(l_s(\mathbf{L})/6) + \\ & c \times l_d(\mathbf{L}) + \delta\delta G_{stack}.\end{aligned} \tag{11}$$

That is, the linear dependence on l_s changes to a logarithmic dependence for more than 6 single stranded bases in a multi-branch loop.

Single base stacking free energies, $\delta\delta G_{stack}$ are computed for multi-branch and exterior loops. In the folding algorithm these are single strand stacking free energies, also known as *dangling base* free energies, because they are applied to single stranded bases adjacent to a base pair that is either in the loop, or closes the loop. This single stranded base may "dangle" from the $5'$ or $3'$ end of the base pair. These parameters are stored in a file named dangle.dg or dangle.TC, as above.

Figure 18 shows some single strand stacking free energies.

If $i.j$ and $j+2.k$ are 2 base pairs, then r_{j+1} can interact with both of them. In this case, the stacking is assigned to only 1 of the 2 base pairs, whichever has a lower free energy (usually the $3'$ stack). If $k.l$ is a base pair and both r_{k-1} and r_{l+1} are single stranded, then both the $5'$ and $3'$ stacking are permitted. The value of $\delta\delta G_{stack}$ is then the sum of all the single base stacking free energies associated with the base pairs and closing base pair of the loop.

It has been evident for some time that to make the free energy rules more realistic for multi-branch and exterior loops, and to improve folding predictions, we would be compelled to take into account the stacking interactions between adjacent helices. Two helices, $\mathbf{H_1}$ and $\mathbf{H_2}$ in a multi-branch or exterior loop are adjacent if there are 2 base pairs $i.j$ and $j+1.k$, $i.j$ and $i+1.k$ or $i.j$ and $k.j-1$ that close $\mathbf{H_1}$ and $\mathbf{H_2}$, respectively. The last 2 cases can only occur in a multi-branch loop. In addition, we define *almost adjacent* helices as 2 helices where the addition of a single base pair (usually non-canonical), results in an adjacent pair. The concept of adjacent helices is important, since they are often coaxial in 3 dimensions, with a stacking interaction between the adjacent closing base pairs. The concept of almost adjacent comes from tRNA where, in many cases, the addition of a GA base pair at the base of the anti-codon stem creates a helix that is adjacent to, and stacks on, the D-loop stem.

Algorithms for loop dependent rules

No single base stacking and constant energy for multi-branch loops.

The first algorithm will assume that multi-branch loops all have constant energy, a. In addition, single base stacking will be ignored. For $i < j$, let $W(i,j)$ be the minimum folding energy of all non-empty foldings on the sub-sequence $r_i, \ldots, r_j$. To aid the algorithm, we must define an auxiliary quantity, $V(i,j)$. This is the minimum energy of all foldings on the sub-sequence $r_i,\ldots,r_j$ that contain the base pair $i.j$. For this model, it is crucial to note the obvious fact that $\forall\, i,j$,

$$W(i,j) \le V(i,j).$$

This notation is the same as that used by Zuker and Stiegler[5] and by Zuker[3, 6]. Sankoff *et al.*[1] and Zuker and Sankoff[2] use F and C in place of W and V respectively. Sankoff[1] define the quantity $F(i,j)$ over all possible foldings *including* the empty folding.

Boundary conditions for W and V are $W(i,j) = V(i,j) = +\infty$ if $j-i < 4$. Recursions for W and V are dependent on the nature of the energy rules for loops. Define $eh(i,j)$ to be the energy of the hairpin loop closed by the base pair $i.j$, $es(i,j)$ the energy of the stacked pair $i.j$ and $i{+}1.j{-}1$, and $ebi(i,j,i',j')$ the energy of the bulge or interior loop closed by $i.j$ with $i'.j'$ accessible from $i.j$. In this case we can write, for $1 \le i < j \le n$:

$$\begin{aligned} W(i,j) &= \min\{W(i{+}1,j), W(i,j{-}1), V(i,j), \\ &\quad \min_{i\le k<j}\{W(i,k)+W(k{+}1,j)\}\} \end{aligned} \tag{12}$$

and

$$\begin{aligned} V(i,j) &= \min\{eh(i,j), es(i,j)+V(i{+}1,j{-}1), \\ &\quad VBI(i,j), VM(i,j)\}, \end{aligned} \tag{13}$$

where

$$VBI(i,j) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{ebi(i,j,i',j') + V(i',j')\} \tag{14}$$

and

$$VM(i,j) = \min_{i<k<j-1}\{W(i{+}1,k)+W(k{+}1,j{-}1)\} + a. \tag{15}$$

The minimum folding energy, E_{min} is given by $W(1,n)$. Note that there are $O(n^2)$ pairs (i,j) satisfying $1 \le i < j \le n$. The computation of VM in 15 takes $O(n^3)$ steps. However, the computation of $VBI(i,j)$ requires $O(n^4)$ steps as written. The most practical way of reducing this search time to $O(n^3)$ steps is to limit the maximum size of a bulge or interior loop to some fixed number, d, usually taken to be 30. The search in equation 14 is now limited by $2 < i'-i+j-j'-2 \le d$.

Single base stacking and affine energies for multi-branch loops.

I will introduce one further level of complexity for loop dependent rules. Let the multi-loop energy be of the form

$$e(\mathrm{multi-loop}) = a + b \times k' + c \times k \tag{16}$$

for constants a, b and c. This is equivalent to affine gap penalties for gaps in sequence alignment, and a cubic order algorithm remains possible.

Further, we now add single base stacking energies to the folding model. Let $ed(i,j,k)$ be the single base stacking energy of the k^{th} base on the $i.j$ base pair. The “d” stands for “dangle”, and k is adjacent to $i.j$. If $k = i{+}1$ or $j{+}1$, we call this a $3'$-dangling base. For $k = i{-}1$ or $j{-}1$, k is called a $5'$-dangling base.

Single stranded bases adjacent to helices are assumed to stack if the stacking is favorable. For this reason, the algorithm assumes that all dangling energies are non-positive. When a single stranded base is adjacent to 2 helices, it is allowed to stack on 1 of them. Usually a $3'$ stack is more energetically favorable. The W array now contains penalties for all "exterior" bases and base pairs for foldings on $r_i \ldots r_j$. However, these penalties are only valid if these "exterior" bases end up enclosed in multi-branch loops. For this reason, we compute 2 other simple array, $W_5(i)$ and $W_3(j)$, where $W_5(i)$ is the best folding energy on $r_1 \ldots r_i$ without penalties on exterior base pairs and $W_3(j)$ is the equivalent on $r_j \ldots r_n$.
Equation (12) is replaced by

$$W(i,j) = \min \left\{ \begin{array}{l} b+c+ed(i{+}1,j,i)+V(i{+}1,j), \\ b+W(i{+}1,j), b+W(i,j{-}1), \\ b+c+ed(i,j{-}1,j)+V(i,j{-}1), \\ 2b+c+ed(i{+}1,j,i)+ed(i,j{-}1,j) \\ +V(i{+}1,j{-}1), V(i,j), WM(i,j) \end{array} \right\}, \tag{17}$$

where

$$WM(i,j) = \min_{i \le k < j} \{W(i,k)+W(k{+}1,j)\}. \tag{18}$$

Equation (17) expands on (12) by considering single-base stacking. Of the 7 cases in (17), 3 are new:

1. Base i stacks over the base pair $i{+}1.j$,
2. base j stacks over the base pair $i.j{-}1$,
3. both i and j stack over the base pair $i{+}1.j{-}1$. This is allowed to happen even if $i.j$ can form a valid base pair.

The single base stacking rules are not used in interior and hairpin loops. In these cases, the mismatched pair adjacent to the closing helix or helices is considered as a *terminal mismatch* and is treated according to special rules that do not affect the algorithm.
Equation (15) for $VM(i,j)$ becomes complicated both by the addition of single base stacking and by the "affine" energy rules defined in equation (16).

$$\begin{array}{l} VM(i,j) = a+c+ \\ \min\{VM_1(i,j), VM_2(i,j), VM_3(i,j), VM_4(i,j)\}, \end{array} \tag{19}$$

where

$$\begin{array}{rcll} VM_1(i,j) & = & \min\limits_{i<k<j-1} & \{W(i{+}1,k)+W(k{+}1,j{-}1)\}, \\ VM_2(i,j) & = & b+\min\limits_{i+1<k<j-1} & \{ed(i,j,i{+}1)+W(i{+}2,k)+W(k{+}1,j{-}1)\}, \\ VM_3(i,j) & = & b+\min\limits_{i<k<j-2} & \{ed(i,j,j{-}1)+W(i{+}1,k)+W(k{+}1,j{-}2)\}, \\ VM_4(i,j) & = & 2b+\min\limits_{i+1<k<j-2} & \{ed(i,j,i{+}1)+ed(i,j,j{-}1)+ \\ & & & W(i{+}2,k)+W(k{+}1,j{-}2)\}. \end{array} \tag{20}$$

The usual order of the fill algorithm is to compute for j increasing from 1 to n, and, for each j, for i decreasing from j to 1. In the computer, the values of each row of W and V are stored consecutively. Once the n^{th} column has been filled, auxiliary dynamic programming algorithms are executed to compute W_5 and W_3. We set $W_5(0) = W_3(n+1) = 0$, and then

$$W_5(i) = \min\{W_5(i-1), W_5^1(i), W_5^2(i), W_5^3(i), W_5^4(i)\}, \tag{21}$$

where

$$\begin{aligned}
W_5^1(i) = &\min_{0\le k<i} \{W_5(k)+V(k+1,i)\},\\
W_5^2(i) = &\min_{0\le k<i} \{W_5(k)+ed(k+2,i,k+1)+V(k+2,i)\},\\
W_5^3(i) = &\min_{0\le k<i} \{W_5(k)+ed(k+1,i-1,i)+V(k+1,i-1)\},\\
W_5^4(i) = &\min_{0\le k<i} \{W_5(k)+ed(k+2,i-1,k+1)+\\
&ed(k+2,i-1,i)+V(k+2,i-1).
\end{aligned} \tag{22}$$

The recursion for W_3 is similar.

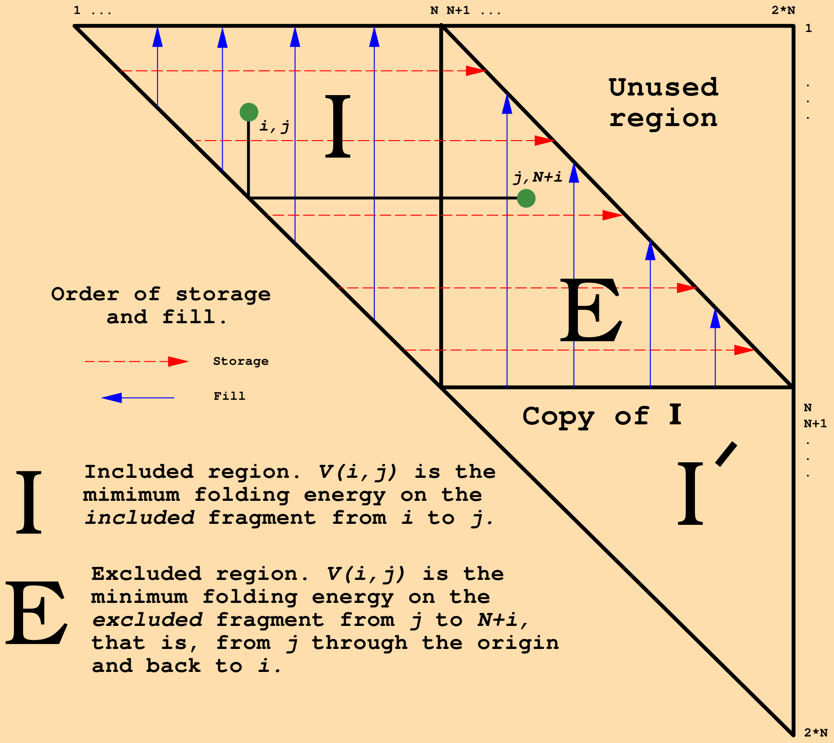
$$W_3(j) = \min\{W_3(j-1), W_3^1(j), W_3^2(j), W_3^3(j), W_3^4(j)\}, \tag{23}$$

where

$$\begin{aligned}
W_3^1(j) = &\min_{j<k\le n} \{V(j,k+1)+W_3(k)\},\\
W_3^2(j) = &\min_{j<k\le n} \{ed(j,k+2,k+1)+V(j,k+2)+W_3(k)\},\\
W_3^3(j) = &\min_{j<k\le n} \{ed(j-1,k+1,j)+V(j-1,k+1)+W_3(k)\},\\
W_3^4(j) = &\min_{j<k\le n} \{ed(j-1,k+2,k+1)+ed(j-1,k+2,j)+\\
&V(j-1,k+2)+W_3(k)\}.
\end{aligned} \tag{24}$$

The overall minimum folding energy is $W_5(n) = W_3(1)$, not $W(1,n)$.

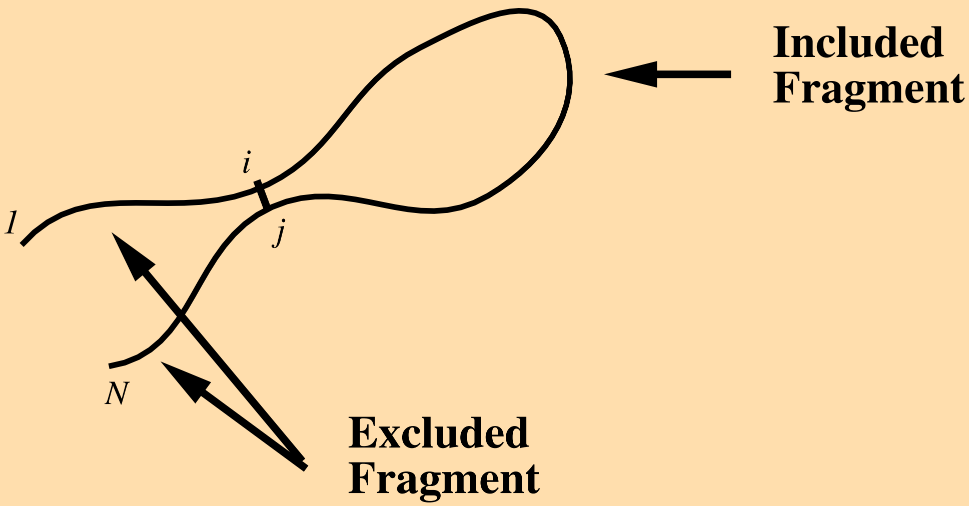
Figure 19: Fill region for the suboptimal algorithm.



Included region. *V(i,j)* is the mimimum folding energy on the *included* fragment from *i* to *j*.

Excluded region. *V(i,j)* is the minimum folding energy on the *excluded* fragment from *j* to *N+i*, that is, from *j* through the origin and back to *i*.

Apparent Lack of Symmetry:

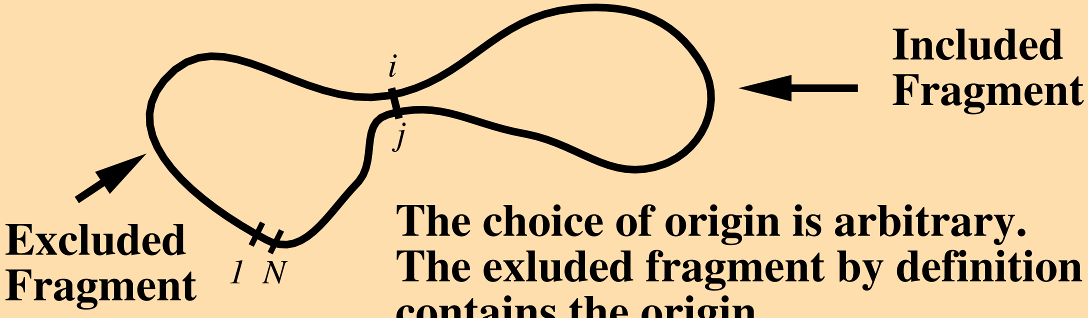


Included Fragment

Excluded Fragment

For any *i,j* pair, *V(i,j)* gives the best folding energy for the included region only.

Solution: "Circularize" the sequence.



Included Fragment

Excluded Fragment

**The choice of origin is arbitrary.
The exluded fragment by definition contains the origin.**

Figure 20: The basic idea behind the suboptimal folding algorithm and an elegant way to compute partition functions. Circularize the sequence. This is equivalent to doubling the sequence and folding *modulo n*.

Suboptimal Algorithm

The suboptimal folding algorithm uses a trick that is identical in spirit to the "forward-backward" dynamic programming used in suboptimal sequence alignment [7, 8]. In this case, the sequence is **doubled**, as illustrated in Figure 20.

That is, we perform the algorithms described above on $\mathbf{R}' = r_1, r_2, r_3, \ldots, r_{2n}$, where $r_{n+i} = r_i$ for $1 \le i \le n$. This defines a large triangular region that is 4 times the size of the original region. The region where $j - i \ge n$ is not used. That is, the algorithm is not performed for i and j in this region. The region defined by $1 \le i < j \le n$, the original dynamic programming region, is called the *included region*, **I**. That is because $V(i,j)$ contains the energy of the best folding in the fragment included between the $i.j$ base pair. The region defined by $n+1 \le i < j \le 2n$, is an exact copy of **I**, and is called $\mathbf{I}'$. It is useful conceptually to be able to consider base pairs in this region. The region defined by $i \le n$, $j > n$ and $j - i < n$ is called the *excluded region*, or **E**.

There is a 1:1 correspondence between **I** and **E**. This mapping is given by $i.j \mapsto j.i+n$. The reverse mapping is $i.j \mapsto j-n.i$. If $i.j \in \mathbf{I}$, then $j, n+i \in \mathbf{E}$, and $V(j, n+i)$ is the energy of the best folding of the fragment *excluded* by the $i.j$ base pair. This is the best folding energy on the 2 fragments from j to the $3'$ end (n) and from the $5'$ end $(n+1)$ to i $(n+i)$, assuming that the RNA is circular. For circular RNA, the algorithm is exact. Otherwise, corrections have to be made because the ends are not spliced together. Figure 19 shows both the *included* and *excluded* regions.

Thus, when a loop contains r_n and r_{n+1}, it becomes an exterior loop instead of a hairpin, stack, bulge, interior or multi-loop. For this reason, it is safe to give an energy of $+\infty$ to any hairpin, bulge or interior loop that contains an end of the sequence. When $1 \le i \le n < j \le 2n$, one special case must be considered for the optimal multi-loop closed by $i.j$. In the equations for $VM_1(i,j)$ to $VM_4(i,j)$, (20), the summands on the right become:

$$\begin{array}{ll} VM_1(i,j): & W_3(i{+}1) + W_5(j{-}1), \\ VM_2(i,j): & ed(i{+}2, j{-}1, i{+}1) + W_3(i{+}2) + W_5(j{-}1), \\ VM_3(i,j): & ed(i{+}1, j{-}2, j{-}1) + W_3(i{+}1) + W_5(j{-}2)\}, \\ VM_4(i,j): & ed(i{+}2, j{-}1, i{+}1) + ed(i{+}1, j{-}2, j{-}1) + \\ & W_3(i{+}2) + W_5(j{-}2), \end{array} \tag{25}$$

when $k = n$. Thus the algorithm considers the case that $i.j$ is an exterior base pair of the folding.

For any possible base pair $i.j$, the quantity $V_E(i,j) = V(i,j) + V(j, i+n)$ is the minimum folding energy of a folding **constrained to contain the $i.j$ base pair**. If E_{min} is the minimum folding energy, and ΔE a (small) energy increment, then a quick scan of all the at most $\frac{n(n-1)}{2}$ base pairs quickly reveals which ones can be in foldings within ΔE from the minimum folding energy. These base pairs can be plotted in a single dot plot known as the *energy dot plot*. This is the superposition of all possible foldings within ΔE of the minimum folding energy. Figure 21 depicts the 9.5 kcal (*i.e.*, $\Delta E = 9.5$ kcal/mole) "energy dot plot" for the cdk2 gene of *Xenopus leavis* (Bases 731-1193 of gb—U07979—XLU07979).

The energy increment is divided into 3 regions of roughly 3.1 kcal/mole. Base pairs in optimal foldings are plotted as black dots in the upper and lower triangular regions. Base pairs within 3.1 kcal/mole from optimal are plotted in red. Then the blue (3.2-6.2 kcal/mole) and finally the yellow (6.3-9.5 kcal.mole) are plotted. The clutter of base pairs in the region roughly defined by the purple triangle indicates the lack of a well defined folding in that region of the sequence (bases 50 to 210). In contrast, the long and stable stem loop region running from 249 to 391 is very well determined. The green lines in the dot plot mark the region where base pairs involving these base pairs can occur. Note that it is almost completely free of alternative base pairs to the stem loop structure in black.

Structures within ΔE of E_{min} can be computed by selecting a base pair, $i.j$, from the energy dot plot and performing a double traceback. That is, a first traceback

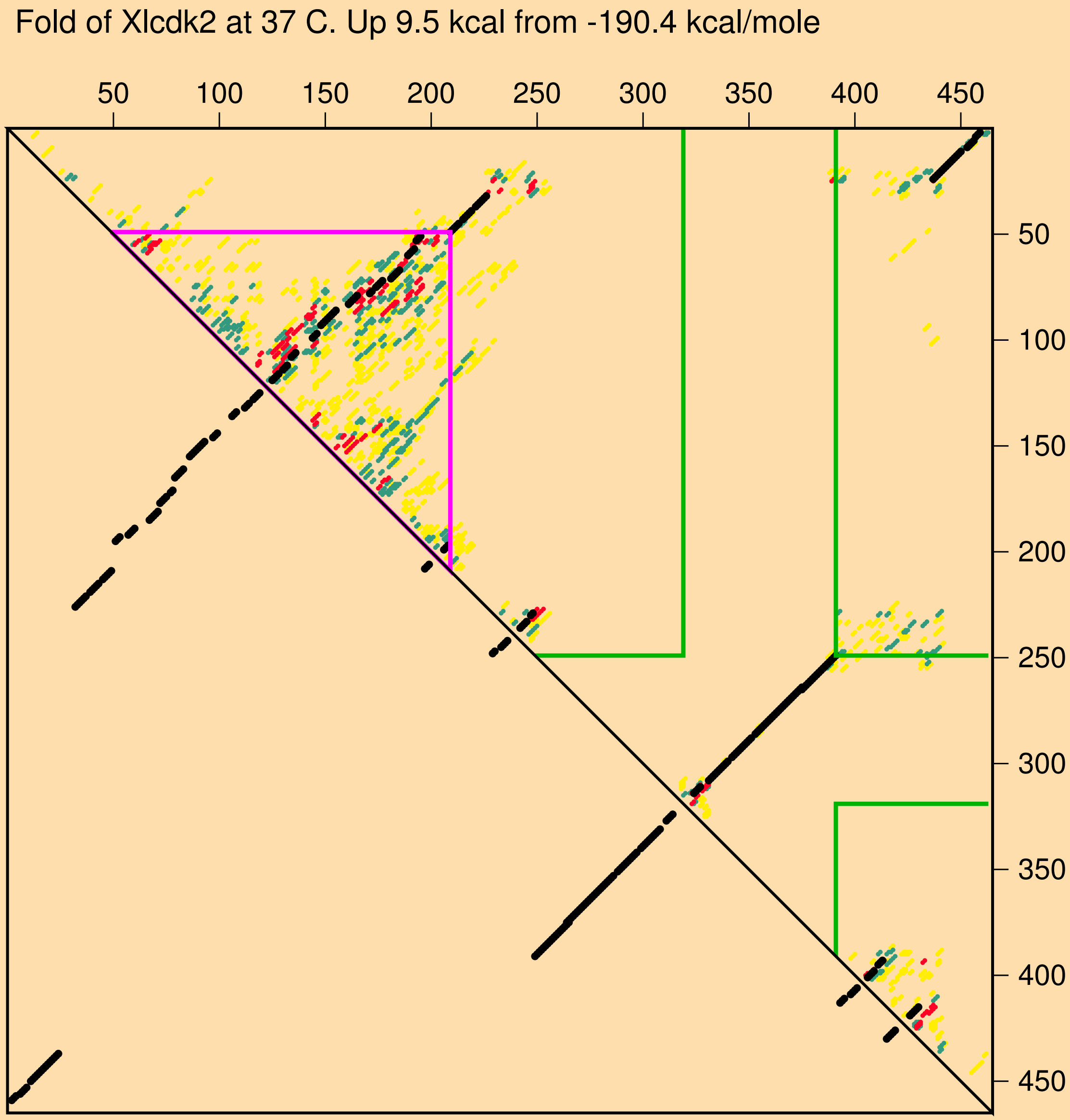


Figure 21: Illustration of clear and cluttered areas in an energy dot plot.

computes an optimal folding in the included fragment from r_i to r_j with energy $V(i,j)$. The second traceback computes an optimal folding in the excluded fragment for r_j through the origin to r_{i+n}. The 2 foldings are combined into a folding of the entire sequence. In practice, base pairs can be selected interactively or the program can select them automatically. In the latter case, all possible base pairs are sorted by $V_E(i,j)$. A structure is computed using the first on the list. The next base pair selected is the first base pair that is not within a distance of d of a base pair that has occurred in a computed folding. The distance between 2 base pair $i.j$ and $i'.j'$ is defined to be $max\{|i-i'|,|j-j'|\}$. Larger values of d ensure that fewer foldings are computed.

The Equilibrium Partition Function

Another attractive feature of RNA secondary structure modeling is that one can compute partition functions exactly. What is this?

Counting foldings.

The first step is to count foldings, and we'll start in a simple way. Let $T(n)$ be the number of foldings on a sequence of length n, where any base may pair with any other base, and where hairpin loops need only contain a single base pair. Then $T(1) = T(2) = 1$ and we'll define $T(0) = 1$ for convenience.

To compute $T(n+1)$ for $n \geq 2$, we consider all cases for r_{n+1}. Either it is unpaired, which may occur in $T(n)$ ways, or it pairs with some base r_{k+1}, for $0 \leq k \leq n-2$. For each of these cases, the foldings on $r_1 \ldots r_k$ and $r_{k+1} \ldots r_{n+1}$ are independent because no pseudoknots are allowed. There are $T(k)T(n-k-1)$ possibilities for each k. The final result is that for $n \geq 2$,

$$T(n+1) = T(n) + \sum_{k=0}^{n-2} T(n-k-1)T(k). \tag{26}$$

Figure 22 illustrates this recursion by using different colors to represent the various cases. We can introduce a "generating function", f, for T by $f(x) = \sum_{k=0}^{\infty} T(k)x^k$. The convolution sum in equation 26 suggests f^2, and by computing the series for $f^2(x)$, one readily derives

$$x^2 f^2(x) - (x^2 - x + 1) f(x) + 1 = 0. \tag{27}$$

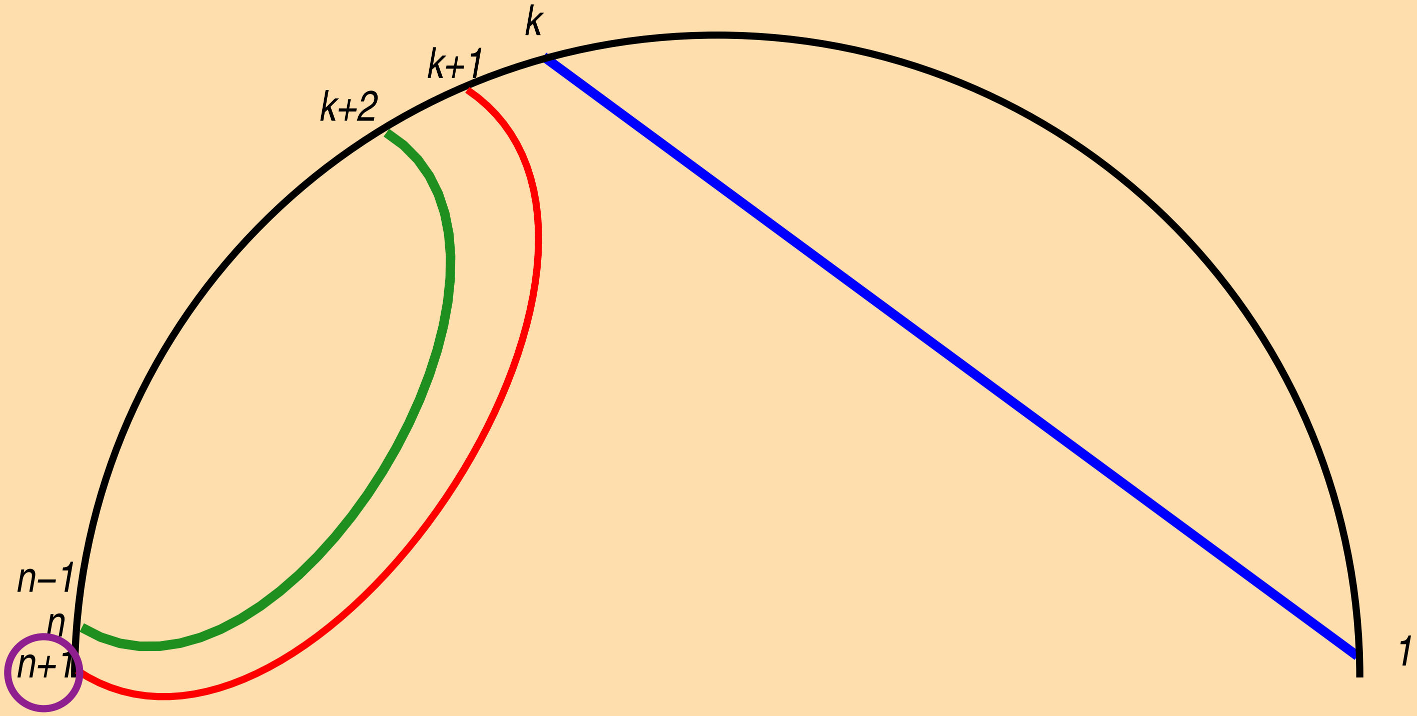
From this, $f(x)$ can be readily computed, and standard asymptotic methods can be used to show that

$$T(n) \sim \sqrt{\frac{15+7\sqrt{5}}{8\pi}} \frac{1}{n^{3/2}} \left(\frac{3+\sqrt{5}}{2} \right)^n \tag{28}$$

as $n \to \infty$. Various generalizations are possible, such as the expected number of foldings of a random sequence where only valid base pairs are allowed.

If we count foldings so easily, then we can count foldings weighted by their energies. For teaching purposes, this will be developed here only for the simpler base pair dependent energy rules. As in the previous derivation for base pair dependent rules, let $e(i,j)$ be the free energy assigned to the base pair $i.j$.

Counting foldings. All pairs and sharp turns are allowed.



$$T(n+1) = T(n) + \sum_{k=0}^{n-2} T(n-k-1)\, T(k)$$

where $T(0) = T(1) = T(2) = 1$.

Figure 22: The simplest formula for counting unknotted foldings.

Partition functions.

Definition.

The definition of the partition function is:

$$Q = \sum_{S \in \mathcal{S}} e^{-\frac{\Delta G_S}{RT}}. \tag{29}$$

This is a weighted counting of all structures. Note that the lower the free energy, the higher the weighting. According to statistical mechanical theory, this Boltzmann weighting gives the probability density for every folding. That is, the probability of any particular folding, S, is give by $exp(-\Delta G_S/RT)/Q$.

The number of secondary structures grows roughly as 1.8^n, but the computation can be performed in reasonable time using a recursion similar to the original dynamic programming algorithm for computing an optimal folding. We need to introduce new terms.

Algorithm for base pair dependent rules.

AUXILIARY PARTITION FUNCTIONS:

Q_{ij} : defined for $r_i \dots r_j$

Q'_{ij} : *restricted*, must contain $r_i - r_j$ base pair

That is, we consider partition functions for every fragment of the original sequence. The Q's are required for loop dependent energy rules, but not for the simpler base pair energy rules. Simple recursions exist for base pair dependent energy rules.

$$\begin{aligned} Q_{ij} &= 1, \quad \text{for} \quad j-i<4, \quad \text{otherwise} \\ Q_{ij} &= Q_{i+1,j} + \sum_{k=i+4}^{j} \exp\left(-\frac{e(i,k)}{RT}\right) Q'_{i+1,k-1} Q_{k+1,j}, \end{aligned} \tag{30}$$

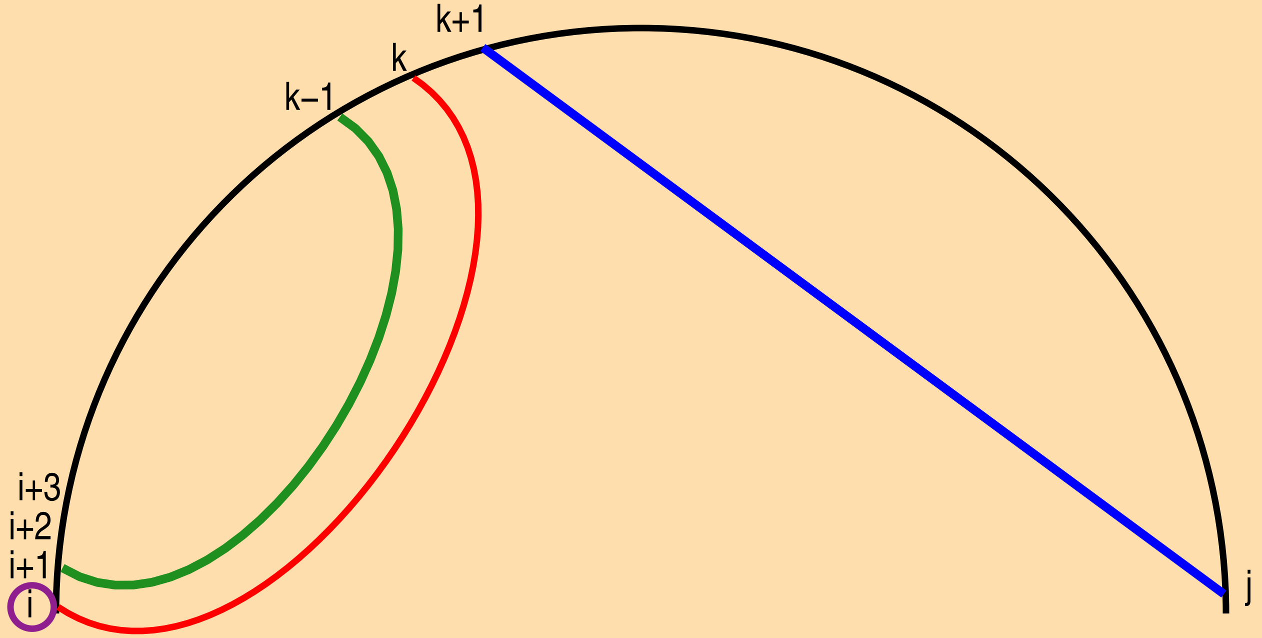
where $Q_{j+1,j}$ is defined to be 1 for convenience.

This algorithm is illustrated in Figure 23.

When the recursion is finished, we know the probability, P, of any structure S. That is:

$$\Pr(S) = \frac{e^{-\frac{E(S)}{RT}}}{Q_{1,n}} \tag{31}$$

Partition function – base pair dependent rules



$$Q_{ij} = Q_{i+1,j} + \sum_{k=i+4}^{j} exp(-e(i,k)/RT)\; Q_{i+1,k-1}\; Q_{k+1,j}$$

Cases:

1. i is unpaired
2. i pairs with a base k, where $Q_{j+1,j} = 1$

Figure 23: This figure illustrates the recursion step for computing the partition function for base pair energy rules. For each summand, the contribution is broken into 3 components, as shown by the colors.

In fact, the algorithm is run on the **doubled sequence**! That is, we extend $r_1, r_2, r_3, \ldots, r_n$ up to r_{2n}, where $r_i = r_{n+i}$. We only compute the recursions for $j - i \leq n$. When $1 \leq i < j \leq n$, Q'_{ij} is the partition function for the *included fragment* from i to j. Then $Q'_{j,i+n}$ is the partition function for the excluded fragment from j through the origin and back to to i. Thus the probability of observing the base pair $i.j$ is given by:

$$\Pr(r_i - r_j) = \frac{Q'_{ij} Q'_{j,i+n}}{e^{-\frac{e(i,j)}{RT}} Q_{1,n}}. \tag{32}$$

The exponential term in the denominator corrects for the fact that both Q'_{ij} and $Q'_{j,i+n}$ count the energy contribution of the base pair $i.j$.

Because of these recursions, which generalize to loop dependent energy rules, it is possible to compute a mathematically rigorous solution where almost all energy states are taken into account. For the base pair dependent model above, all energy states are considered. For loop dependent energy rules, structures with very large interior loops are ignored.

Partition function computations for loop dependent energy rules were first described by McCaskill [9]. A suite of programs, known as the “Vienna package” [10], has been developed around these ideas.

Heat Capacity and Melting

The heat capacity, C_p, can be derived from the partition function directly without examining a representative class of foldings. The formulae presented below are general. They work for any partition function calculations versus temperature. The effective Gibbs free energy of the ensemble, G, is given by:

$$G = -RT \ln Q. \tag{33}$$

The enthalpy, H, is given by:

$$H = kT^2 \frac{\partial \ln Q}{\partial T} = G - T \frac{\partial G}{\partial T}. \tag{34}$$

From this, we derive:

$$C = \frac{dH}{dT} = -T \frac{\partial^2 G}{\partial T^2}. \tag{35}$$

One still has to compute Q for varying temperatures over the desired range. To compute C at some temperature T_i, the Vienna group fits a least squares parabola to G at $2m+1$ points:
$T_{i-m}, T_{i-m+1}, \ldots, T_i, T_{i+1}, \ldots, T_{i+m}$.
The second derivative of this polynomial is the estimate used for the second partial derivative of G with respect to temperature at T_i.

EXAMPLE: The *RNAheat* program from the Vienna package was used to compute the heat capacity curve for the 5S rRNA of *R. rattus*. Note, in Figure 24 the many transitions that take place during melting. These indicate conformational changes as the molecule is heated.

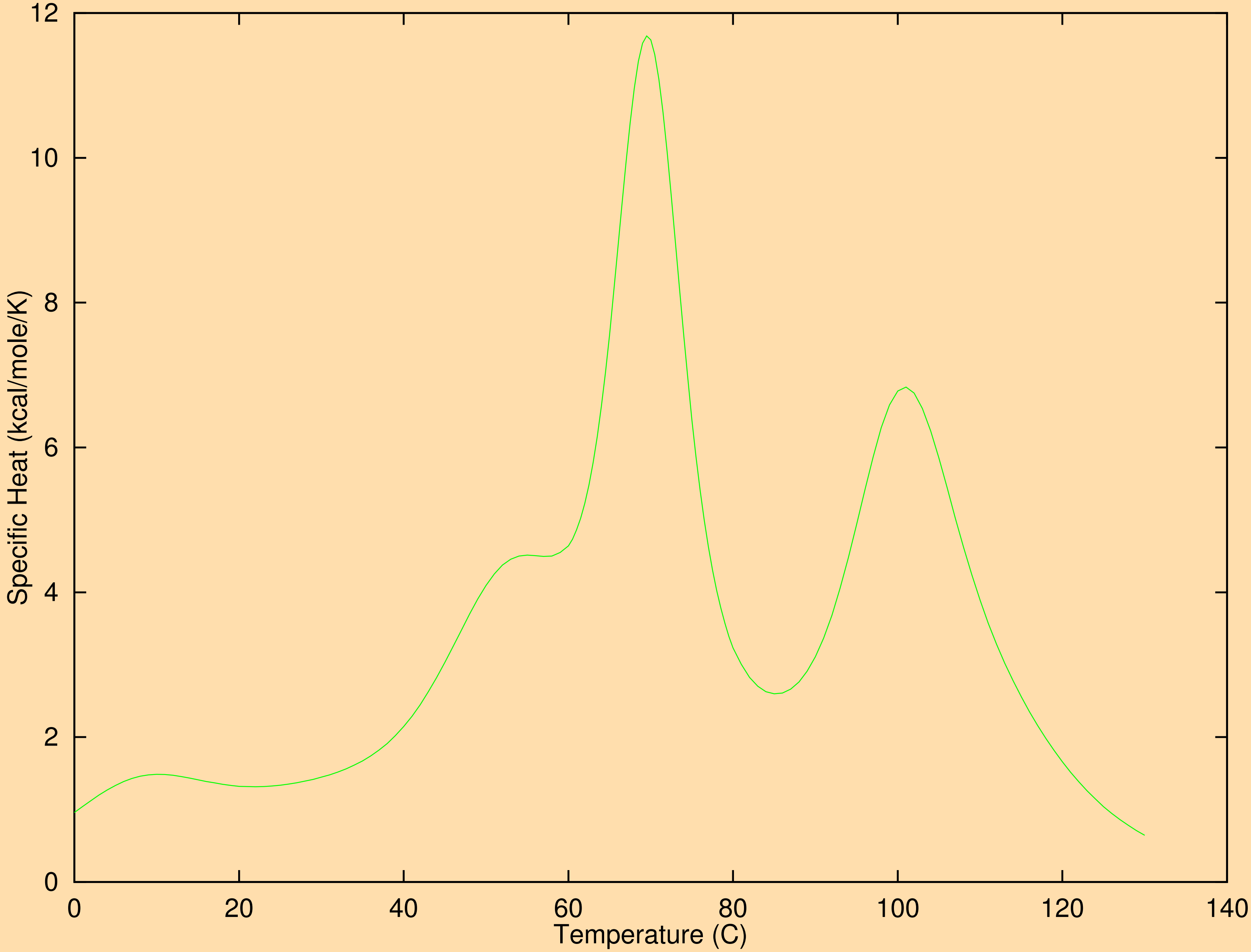


Figure 24: Heat capacity curve for *R. rattus* 5S rRNA. Temperature is in °C and heat capacity is in kcal/mole/°K.

Sample foldings

This section illustrates the appearance of “mfold”
PostScript output files, including both the “energy dot plot” and structures. A “Vienna boxplot” is also shown. There are some results that consider the reliability and accuracy of foldings.

Simple Cases

Example 1

The “energy dot plot” is an integral part of the folding prediction. Consider the folding of a short RNA sequence:
`ACCCCCUCCU UCCUUGGAUC AAGGGGCUCA A,`
using default parameters, except for setting ‘W’=2. $\Delta G = -9.7$ kcal/mole at 37°, so $\Delta\Delta G = 1.0$ rather than 5% of ΔG. A single, optimal folding is computed. A glance of the “energy dot plot” , shown in Figure 25, reveals the optimal folding in black dots (symbols), but another set of yellow dots, indicating base pairs in at least 1 other suboptimal folding. The value of ‘W’ is too large for this other folding to be predicted, but a glance at the dot plot shows that something else is there. When the sequence is refolded with the default value of ‘W’=1, a second, totally different folding is predicted. Figure 27 displays these foldings with individual bases drawn.

Example 2

Important alternative foldings might not appear in the “energy dot plot” if $\Delta\Delta G$ is too small. This is especially true in the folding of short sequences. When the short sequence:
`AAGGGGUUGG UCGCCUCGAC UAAGCGGCUU GGAAUUCC,`
is folded, also with default parameters, a single optimal folding is computed. However, the “energy dot plot” contains only the optimal, black dots from Figure 28. Changing the window size would not reveal anything new. When the value of P is increased to 25 (25%), the “energy dot plot” now reveals a very distinct alternate folding as shown in Figure 28. The “mfold” program now computes 2 foldings, plotted in Figure 29, using the default value of W.

Some real examples, good and bad.

The Potato Spindle Tuber Virus

The Potato Spindle Tuber Virus is a small circular RNA virus (viroid) that infects potatoes. It contains 359 base pairs. In these examples, it has been folded as linear RNA, so that the “Vienna package” could also be used. We used the KF440-2 isolate with GenBank accession number X58388. The locus is PTVTVD440. We call the sequence PSTVD (or PSTVd).

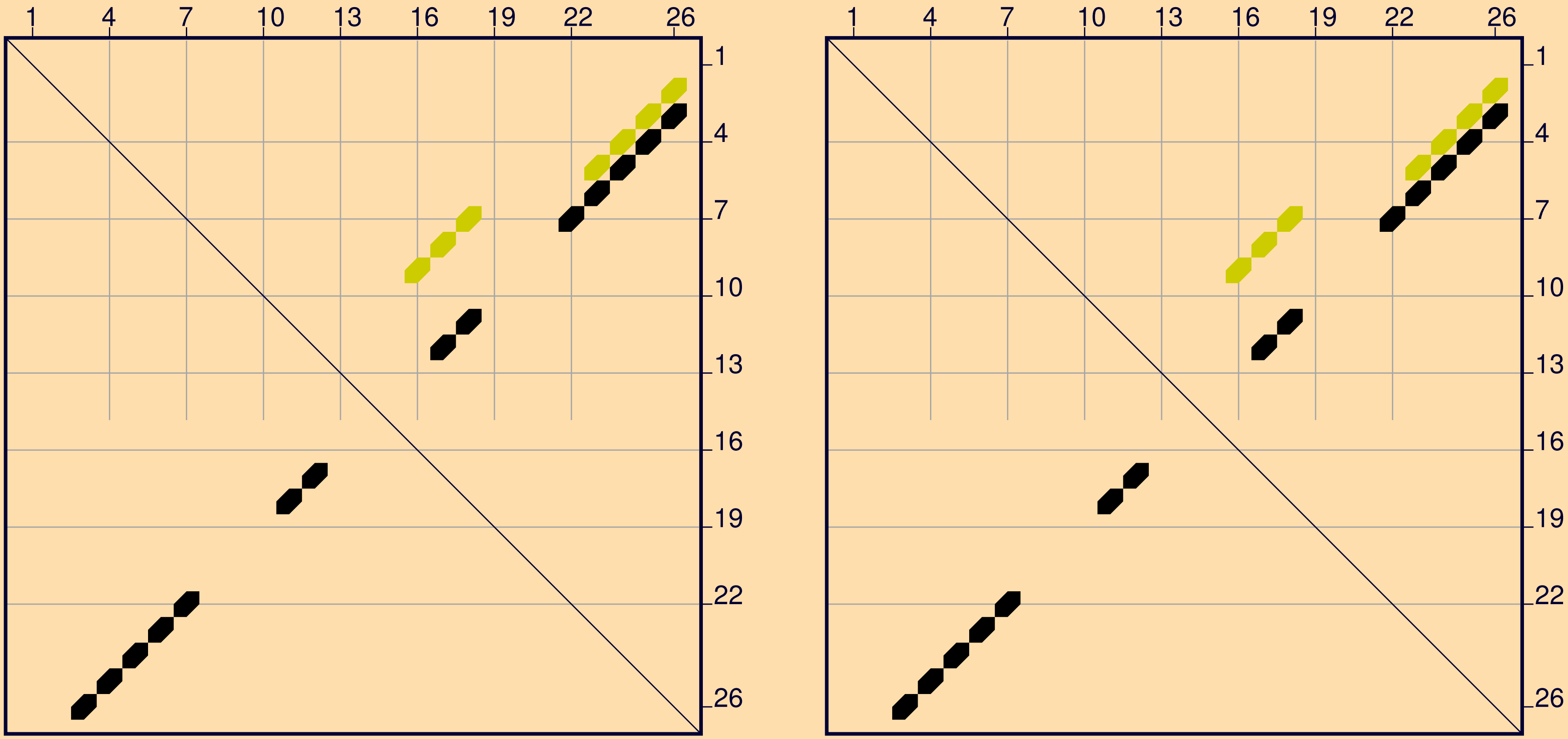


Figure 25: The “energy dot plot” for the “Example 1” sequence. Surrounding annotation, which would not be legible at this scale, has been removed. The yellow dots indicate base pairs in foldings within 0.6 kcal/mole of the optimal folding free energy of -9.7 kcal/mole.

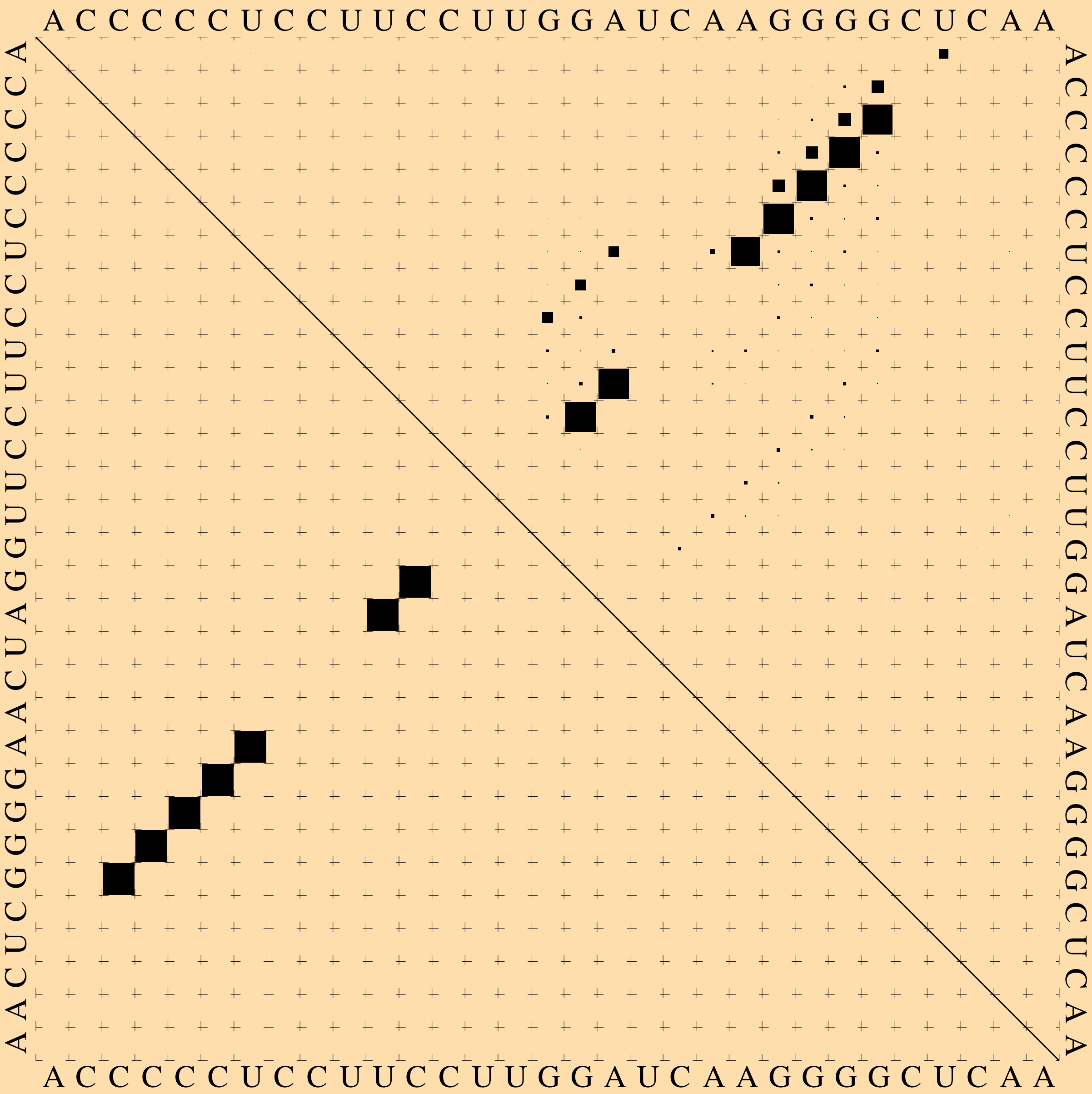


Figure 26: The probability dot plot for the "Example 1" sequence. The probabilities of base pairs are proportional to the area of the black "dots" (squares).

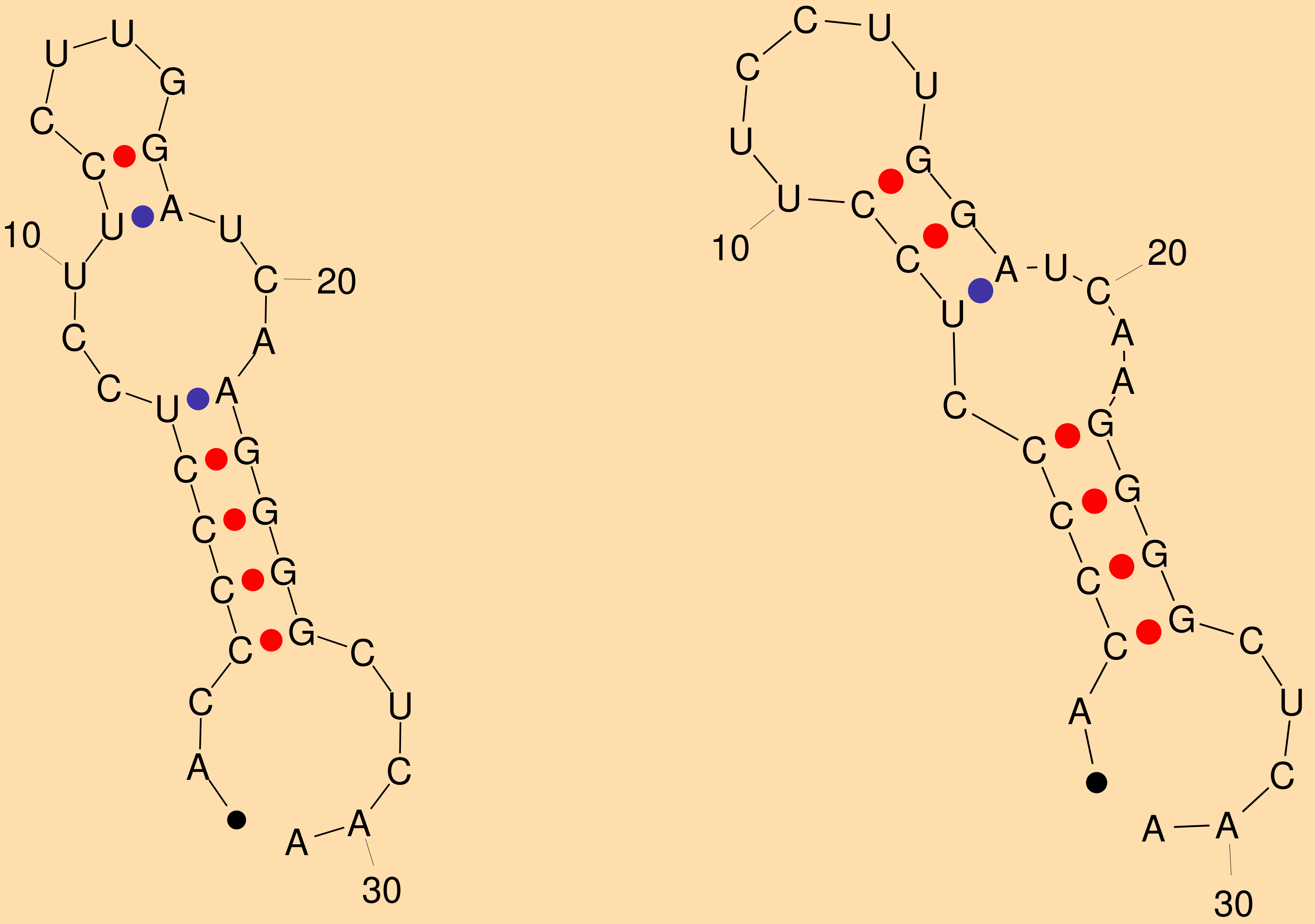


Figure 27: The 2 predicted foldings for the “Example 1” sequence. (Left) The optimal folding with $\Delta G = -9.7$ kcal/mole. (Right) The suboptimal fold ($\Delta G = -9.1$ kcal/mole) found after refolding with ‘W’=0.

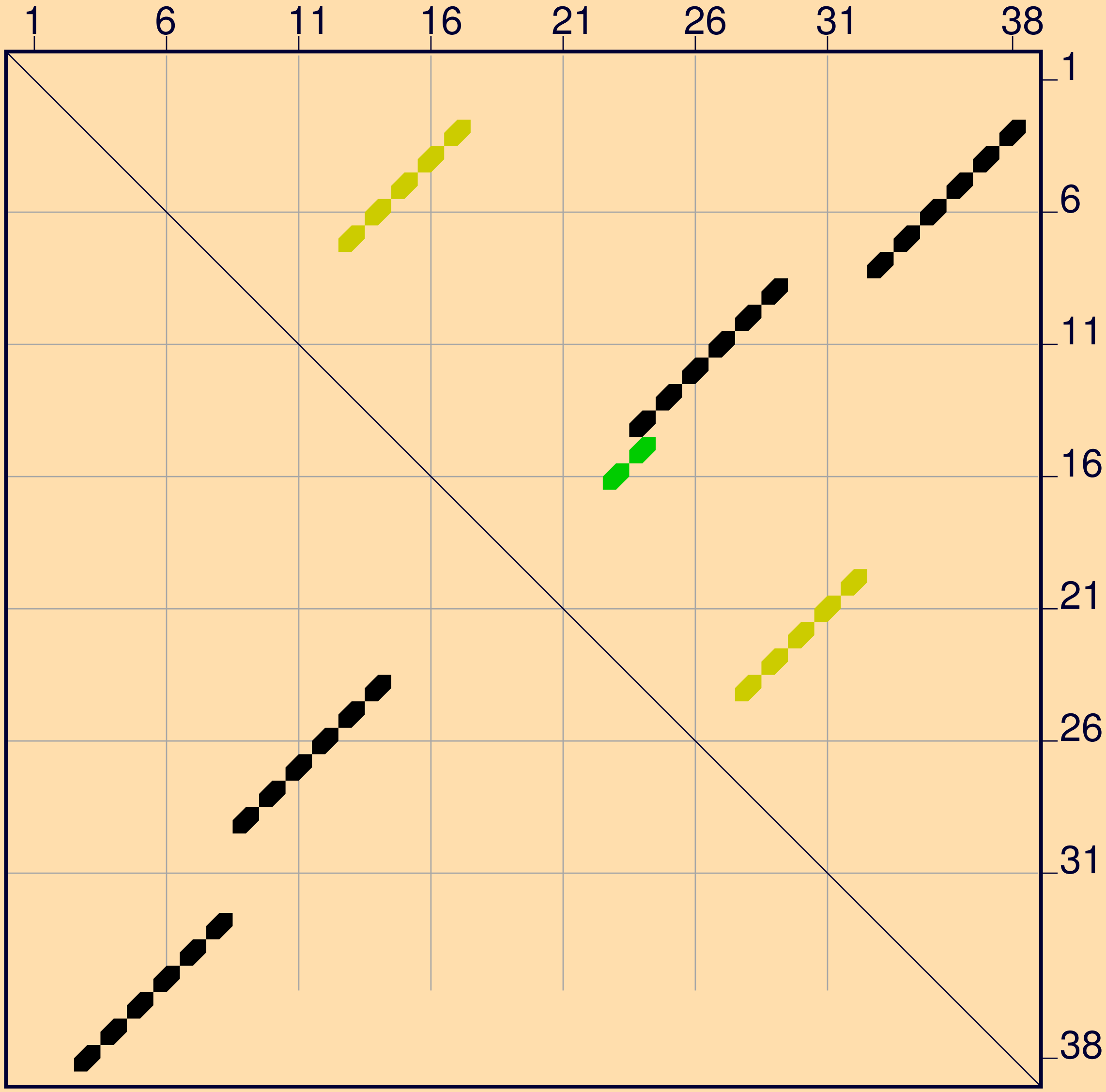


Figure 28: The “energy dot plot” for “Example 2” sequence with $\Delta\Delta G$ increased to 25% of 10.1, or 2.5 kcal/mole. The value of $\Delta\Delta G$ in the plot may be less than this maximum value, since there may be no base pairs in foldings that are **exactly** $\Delta\Delta G$ from the minimum free energy. The 2 green dots represent base pairs that can be in foldings with ΔG between -9.4 and -8.6 kcal/mole. These numbers are -8.6 and -7.6 for the yellow dots. In this case, the black dots comprise the optimal folding, and the yellow dots comprise the single suboptimal folding that is computed. The green dots would only be found in a folding if the value of W were lowered sufficiently.

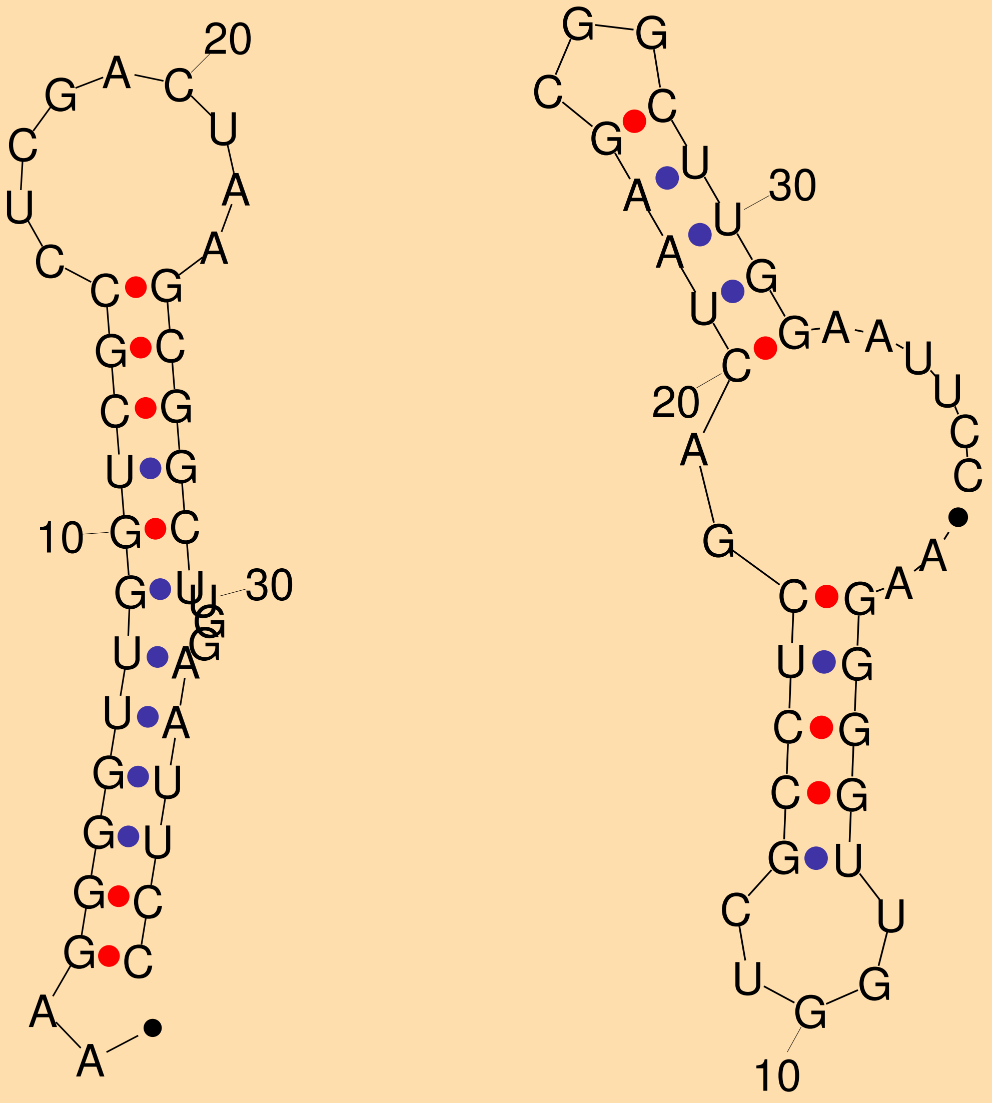

20
10
30
AAGGGGUUGGUCGCCUCGACUAAGCGGCUGGAAUUCC
20
30
10
AAGGGGUUGGUCGCCUCGACUAAGCGGCUGGAAUUCC

Figure 29: The 2 predicted foldings for the “Example 2” sequence. (Left) The optimal folding with $\Delta G = -10.1$ kcal/mole. (Right) The suboptimal fold ($\Delta G = -7.8$ kcal/mole) found after refolding with ‘P’=25.

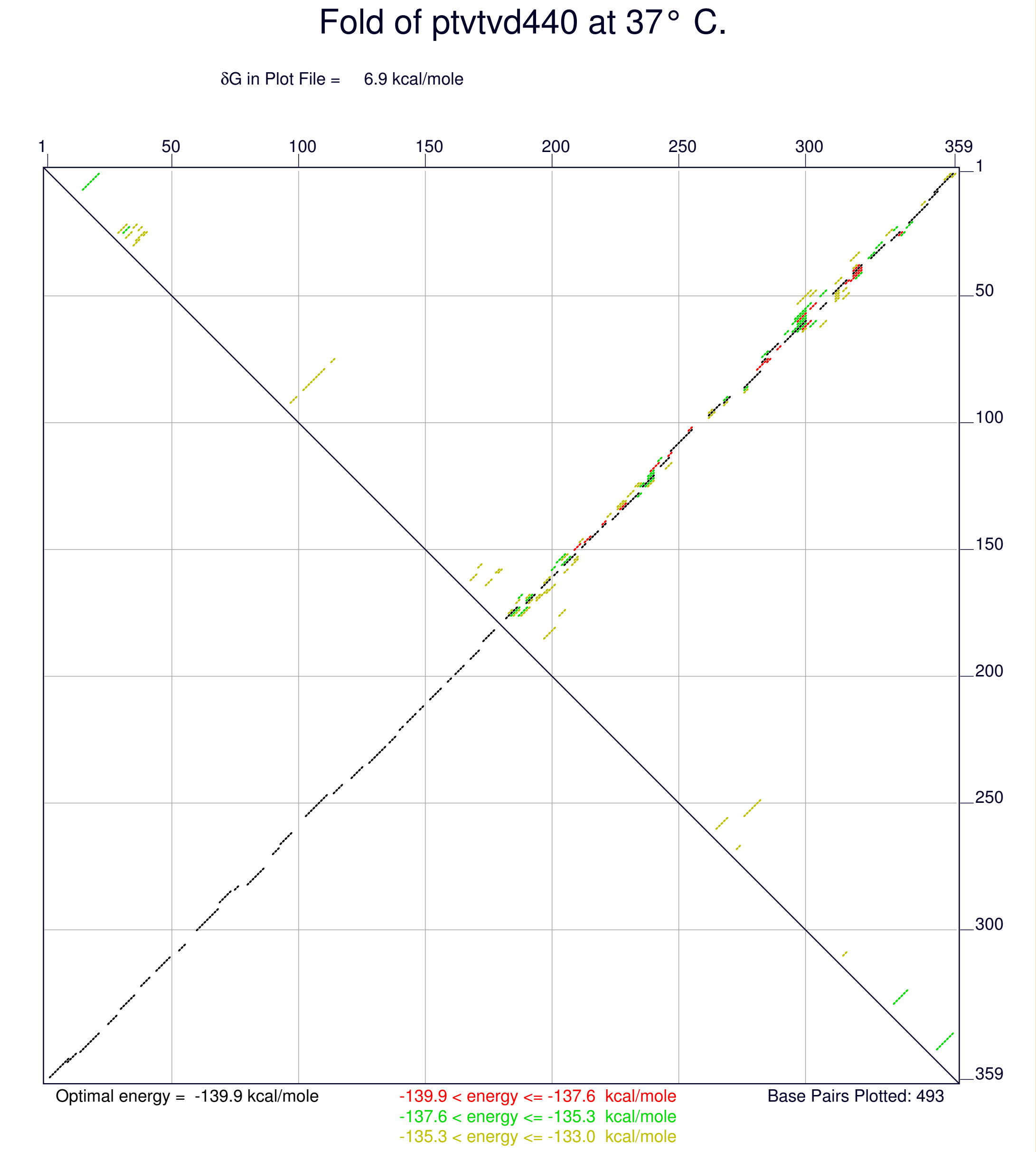


Figure 30: The "energy dotplot" for PSTVd at 37° C. Note the strong tendency to fold into a rod-like structure with few alternatives.

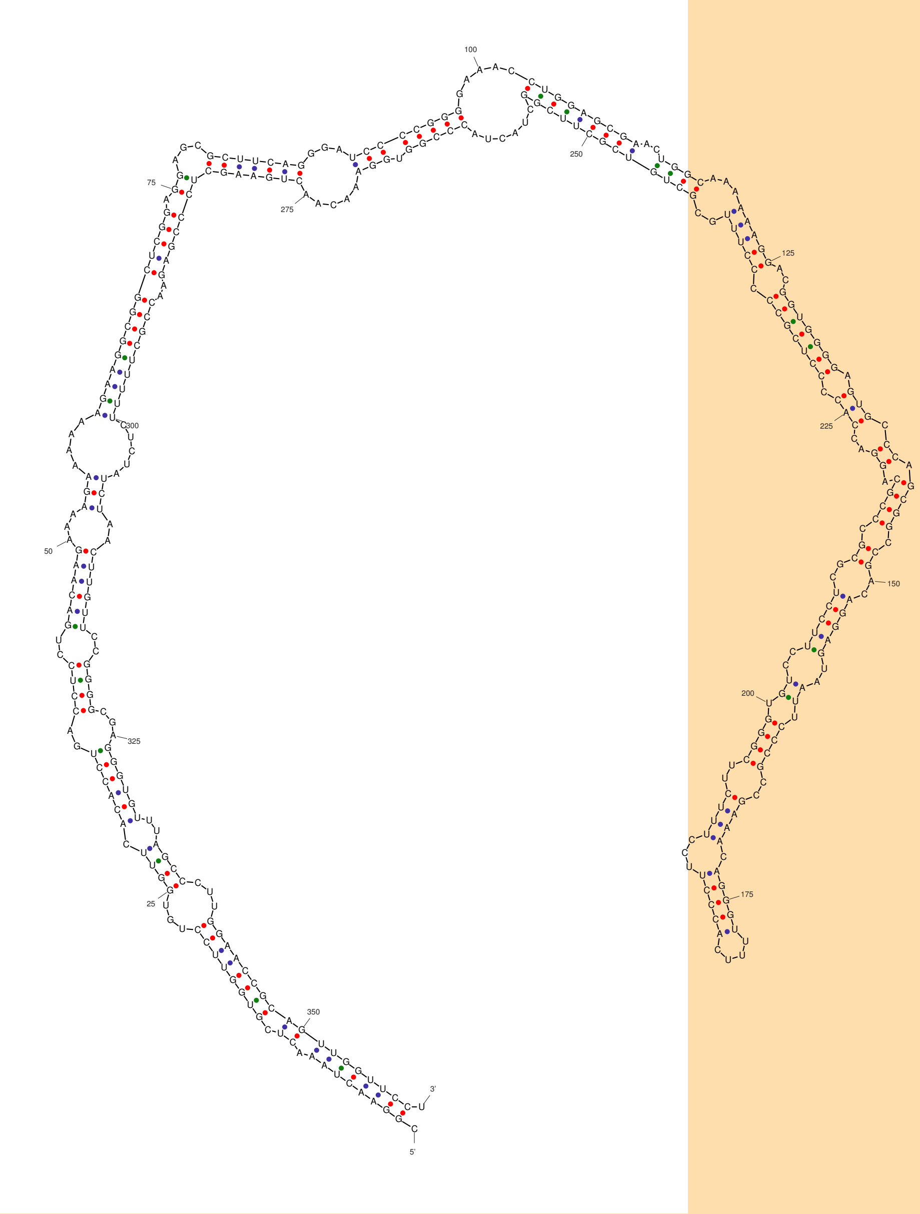


Figure 31: Optimal folding of PSTVd into the correct rod-like structure. $\Delta G = -139.9$ kcal/mole.

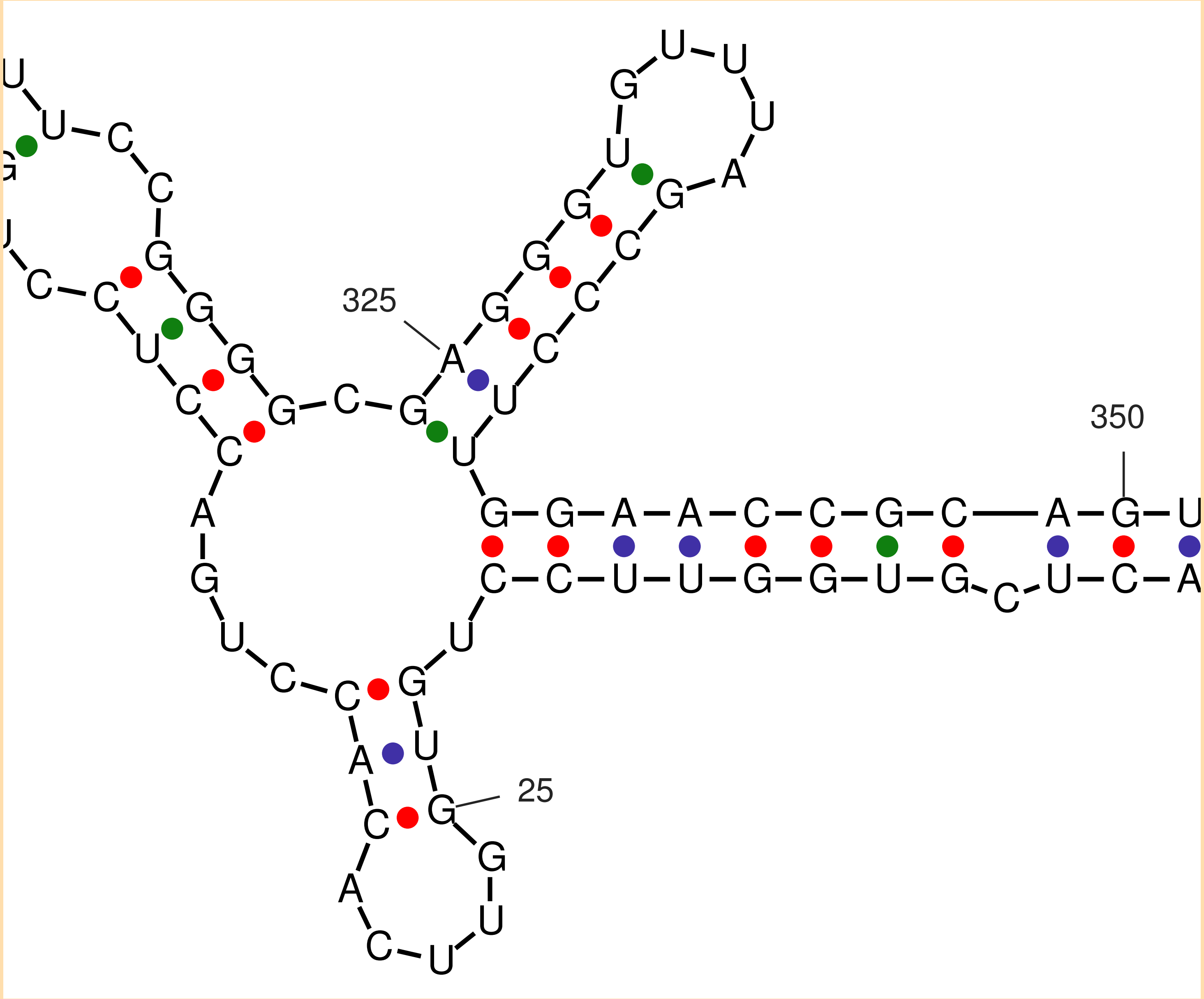


Figure 32: A suboptimal folding of PSTVd showing a slight rod separation to form a local cruciform structure. $\Delta G = -136.2$ kcal/mole, which is a large increment from the minimum free energy folding.

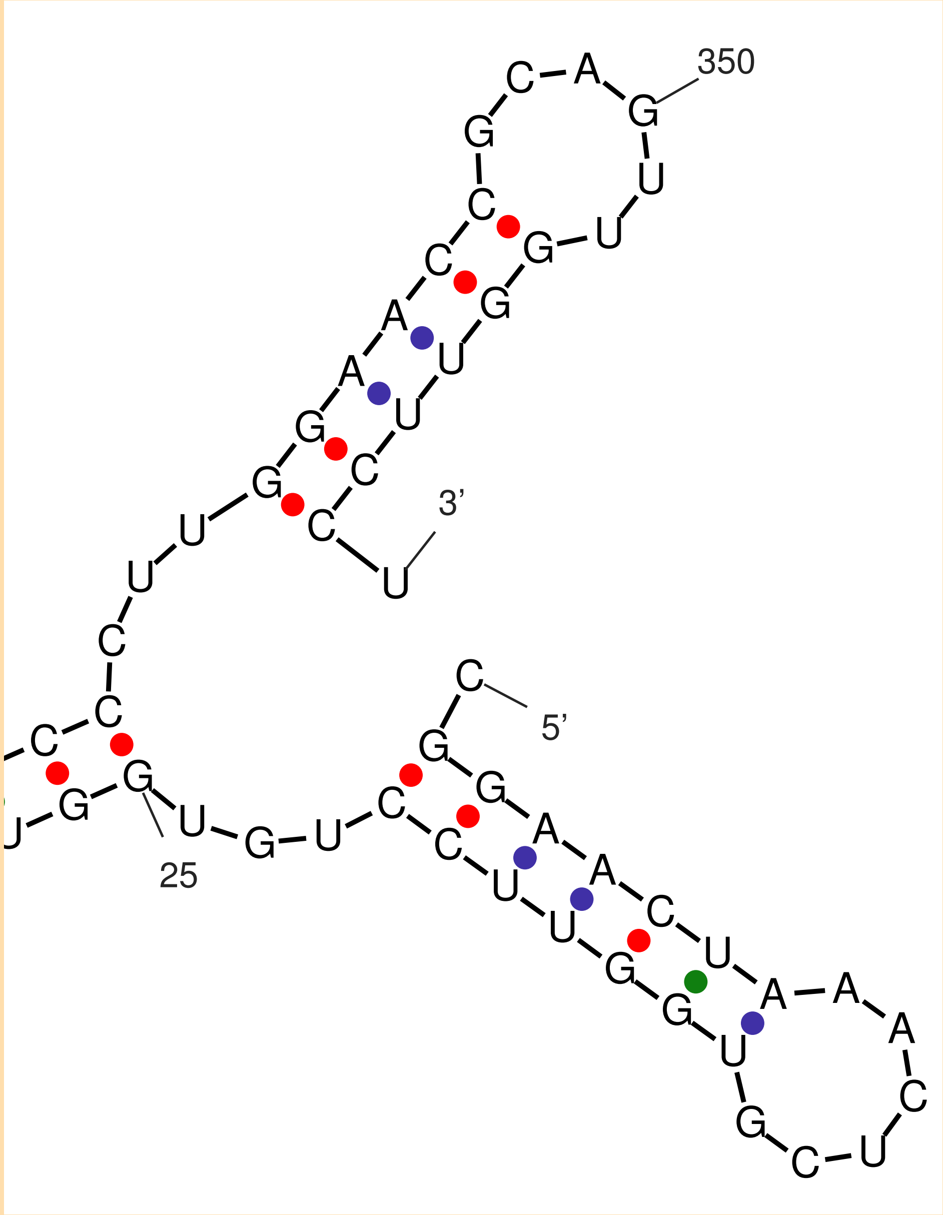


Figure 33: A suboptimal folding of PSTVd showing rod separation at one end. $\Delta G = -135.8$ kcal/mole.

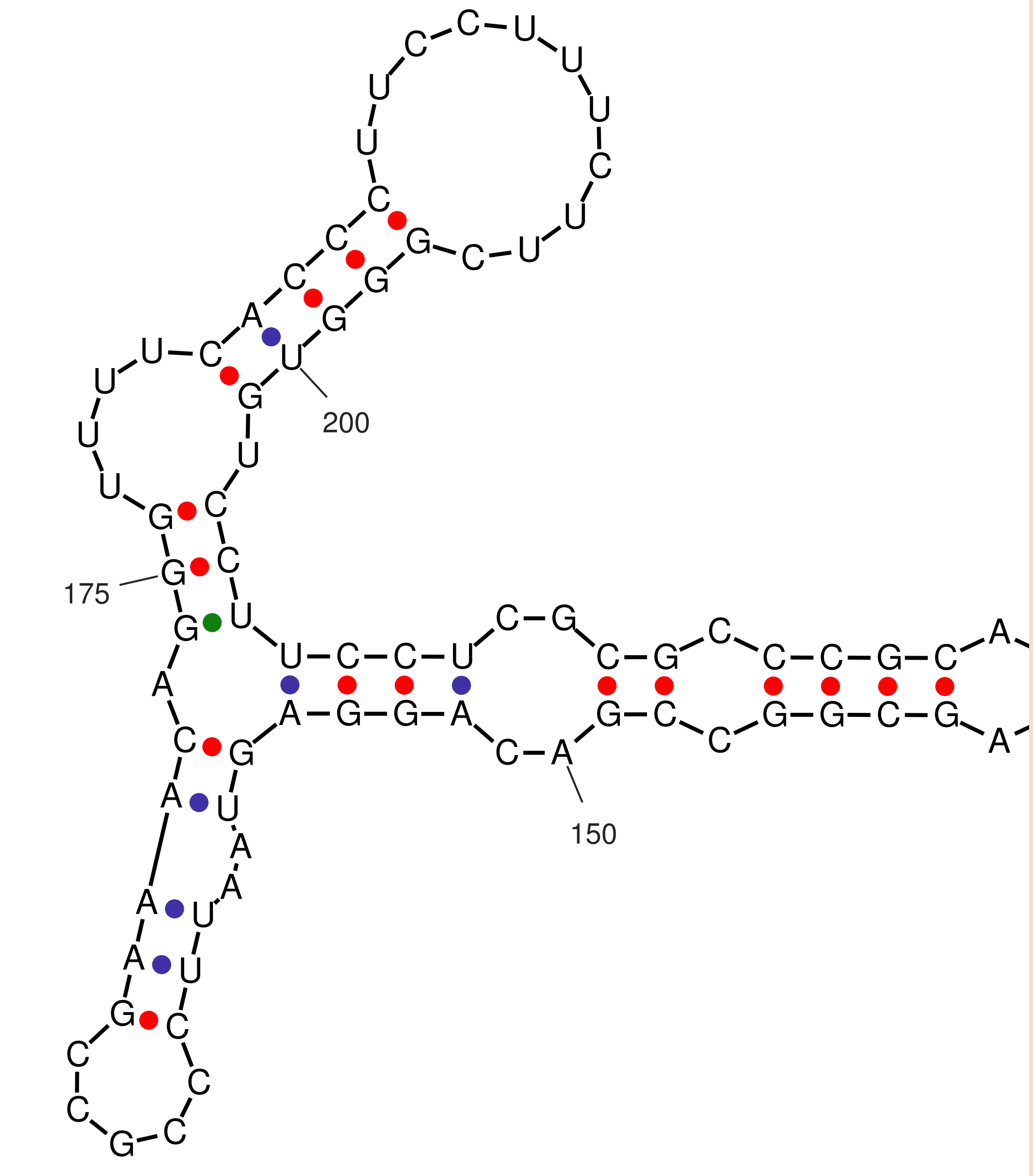

U-C-C-U-U-U-C-U-U-C-G
C-C-C-A-C
G-G-G-U
U-U-C
U-U-U-G
G-U-C-C-U
G-G-G
200
175
U-C-C-U-C-G-C-G-C-C-C-G-C-A
A-G-G-A-C-A-G-C-C-G-G-C-G-A
150
A-C-A-A-A-G-C-C-G-C-C-C-U-U-A-A-U-G

Figure 34: A suboptimal folding of PSTVd showing rod separation at the opposite end of the structure compared for Figure 33. $\Delta G = -134.5$ kcal/mole.

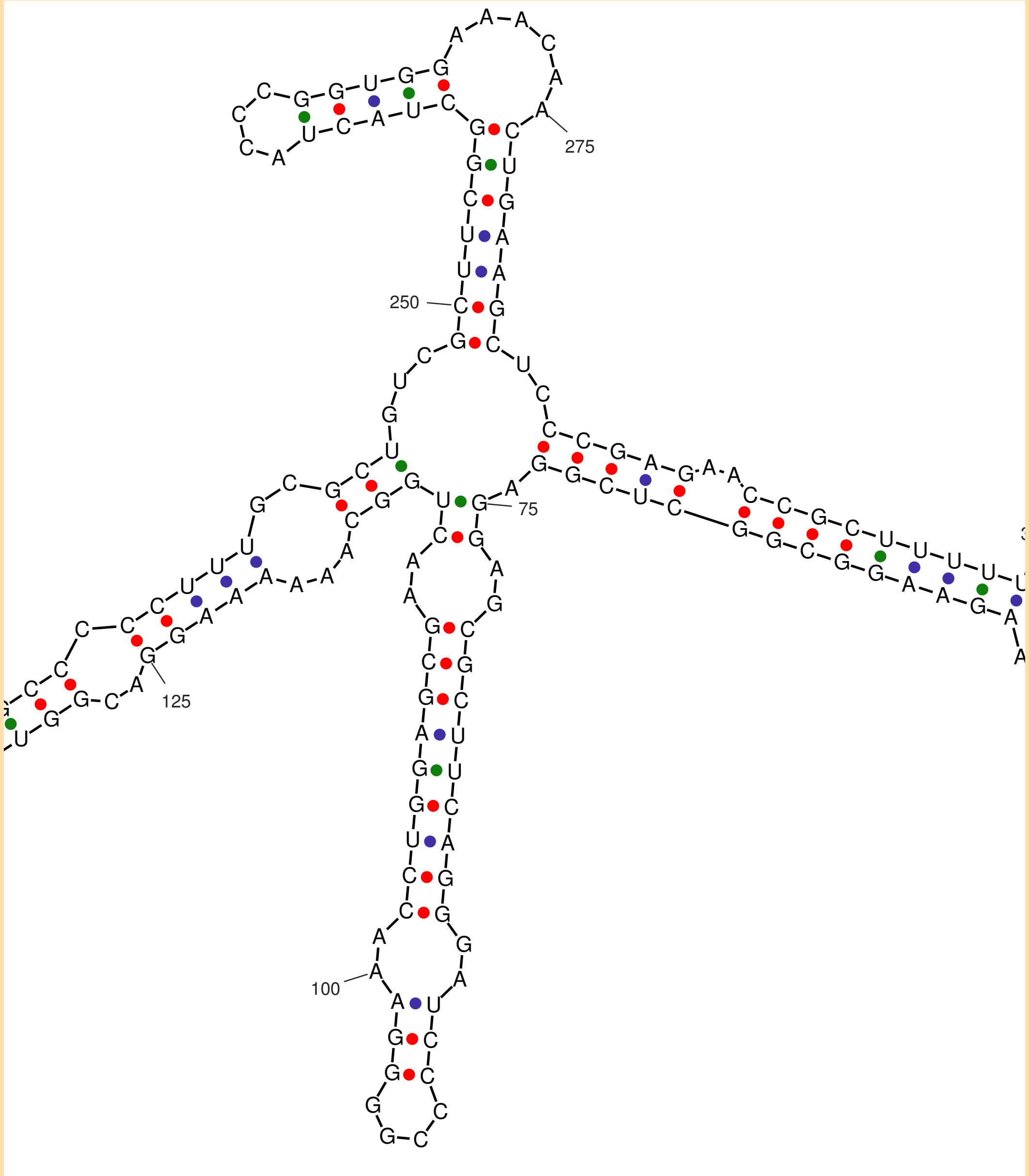


Figure 35: A final suboptimal folding of PSTVd showing a more substantial strand separation and cruciform formation in the middle of the rod. Compare with Figure 32. $\Delta G = -133.2$ kcal/mole.

This virus is a “good” one in terms of RNA folding prediction. Not only is the minimum energy folding overwhelmingly correct, but the dot plot show little propensity to form suboptimal foldings close to the minimum free energy.

At 75° C, the rod-like structure has fallen apart. See Figures 36 and 37.

Figures 38 and 39 show base pair probabilities by computing partition functions. The Vienna package was used, together with “mfold” software to display the “boxplot” (which I call a “probability dot plot”).

E. coli 16S rRNA.

Small subunit rRNA (SSU rRNA) is poorly predicted by energy minimizing algorithms. Some Archaea SSU rRNAs are rather well predicted, but many others are not. Eukaryotic SSU rRNA is quite poorly predicted. The worst examples are from mitochondrial rRNA (mt rRNA). Figures 40 and 41 show the energy dot plot for *E. coli* 16S rRNA. The second representation shows the same dotplot, using shades of grey instead of different colors to represent base pairs in suboptimal foldings. Overlayed on this plot are the base pairs in the correct folding. the color code is described in the caption of Figure 41.

Folding *E. coli* and *Zea Mays* chloroplast 16S rRNA together.

The uncertainties of minimum energy folding can sometimes be mitigated by folding a sequence using energy minimization, but allowing only those base pairs that can also exist in corresponding positions in “homologous” RNAs. An example of this is to refold *E. coli* 16S rRNA while allowing only those base pairs that also exist in *Zea Mays* chloroplast 16S rRNA. One of the great difficulties of using such methods is that “corresponding positions” are defined by a sequence alignment (multiple alignment), and it is not clear *a priori* which alignment is correct. I used a “default” alignment giving scores of 1s and 0s to matches and mismatches (respectively) and a simple gap penalty of 5. Figures 42 and 43 illustrate the dramatics improvement when a single sequence is added to “constrain” the folding.

Thermococcus celer, a “well behaved” Archaeon

Some 16S rRNAs fold well on their own. a good example is *T. celer* 16S rRNa. The optimal computed folding contains 326 out of 462 phylogenetically determined base pairs (71%). This rises to 352 (76%) when near misses are included. See Figure 44.

Annotation of Foldings

The information in the “energy dot plot” or probability dot plot (boxplot) can be used to annotate individual foldings to indicated the “well-definedness” of the base pairs and or bases.

For the “energy dot plot” , some crude statistics are used. For a given free energy increment, $P\!-\!num(i)$ is defined to be the total number of dots (base pairs)

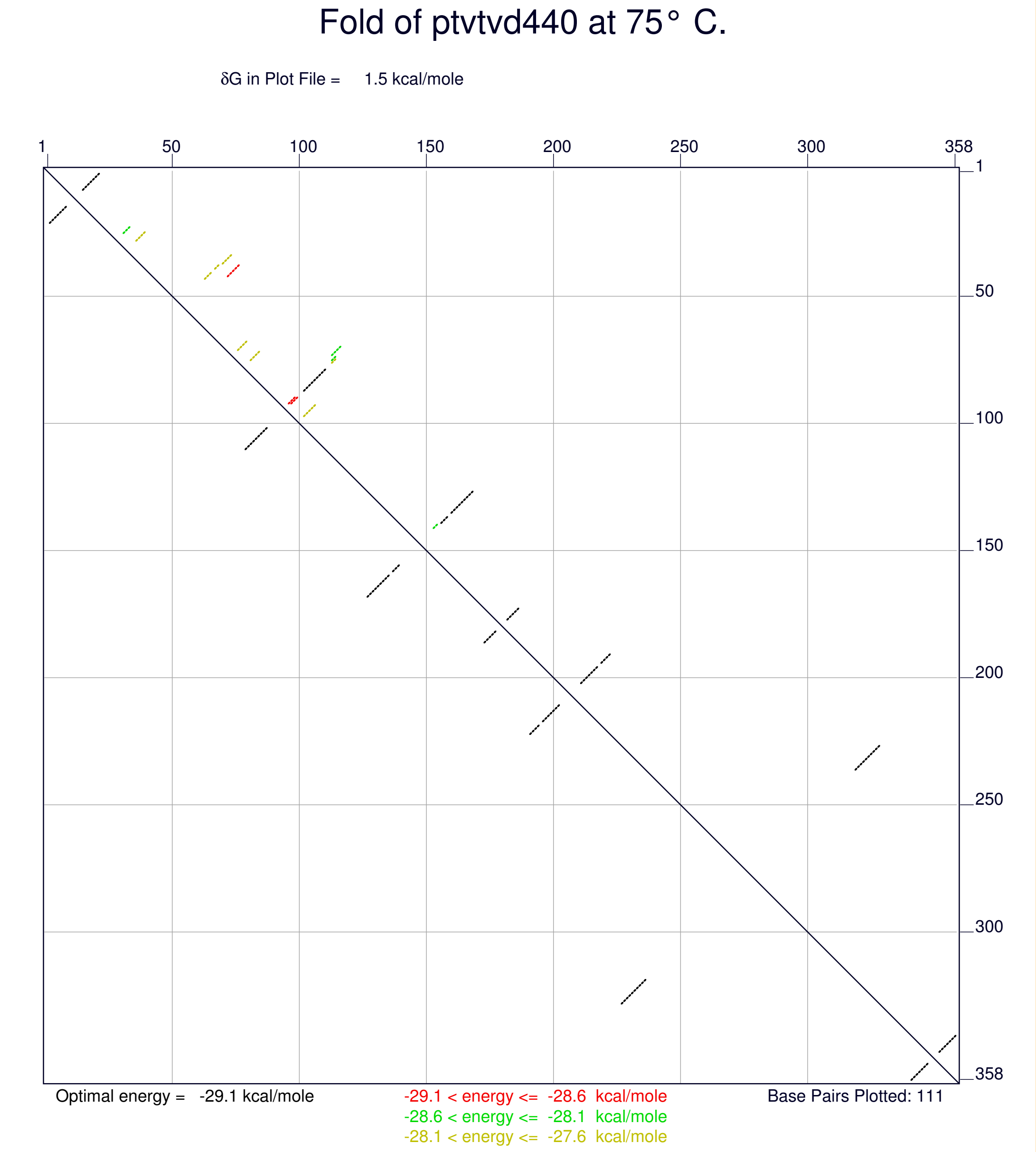


Figure 36: The "energy dotplot" for PSTVd at 75° C. Note how the rod-like structure has disintegrated into a number of local stem-loop structures.

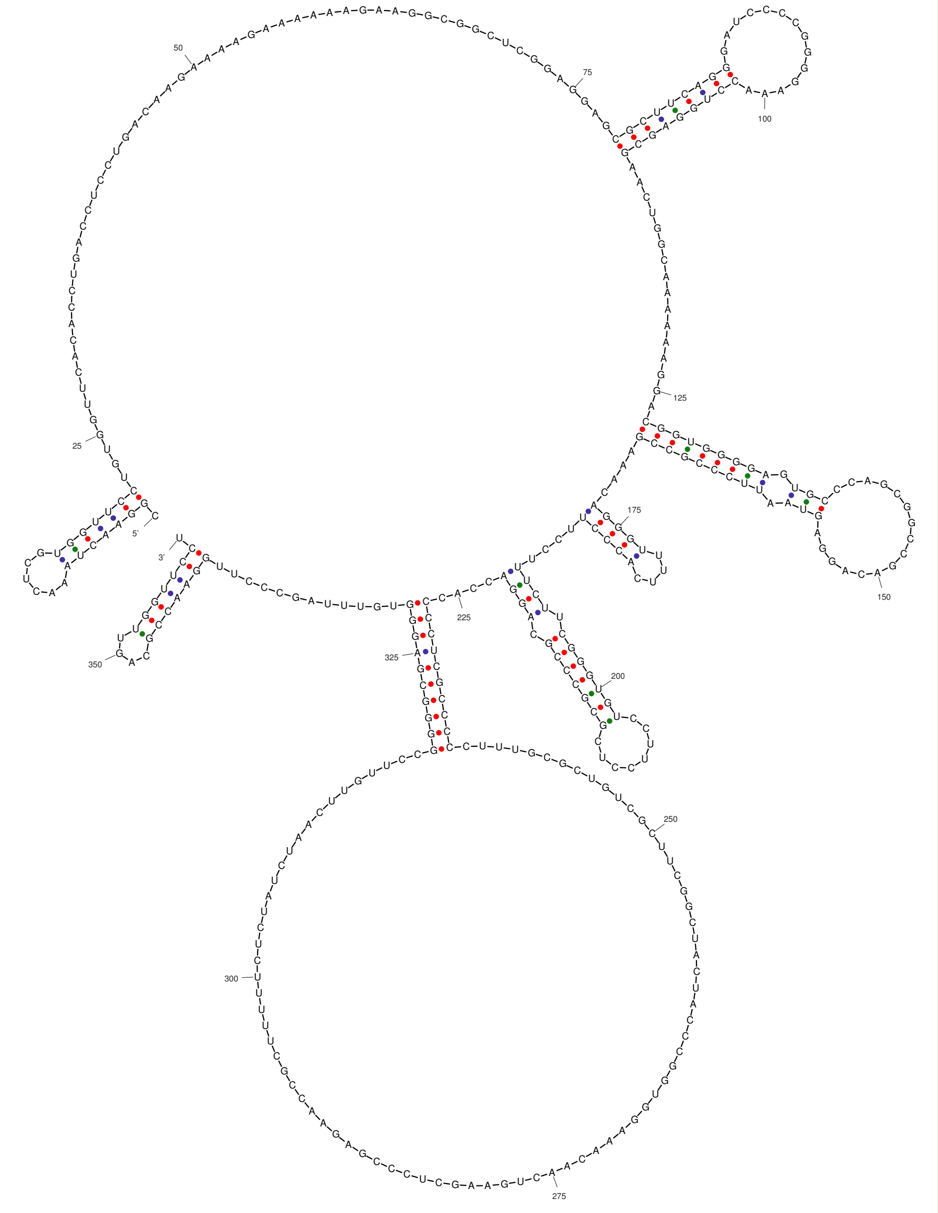


Figure 37: The optimal folding of PSTVD at 75° C. $\Delta G = -29.1$ kcal/mole.

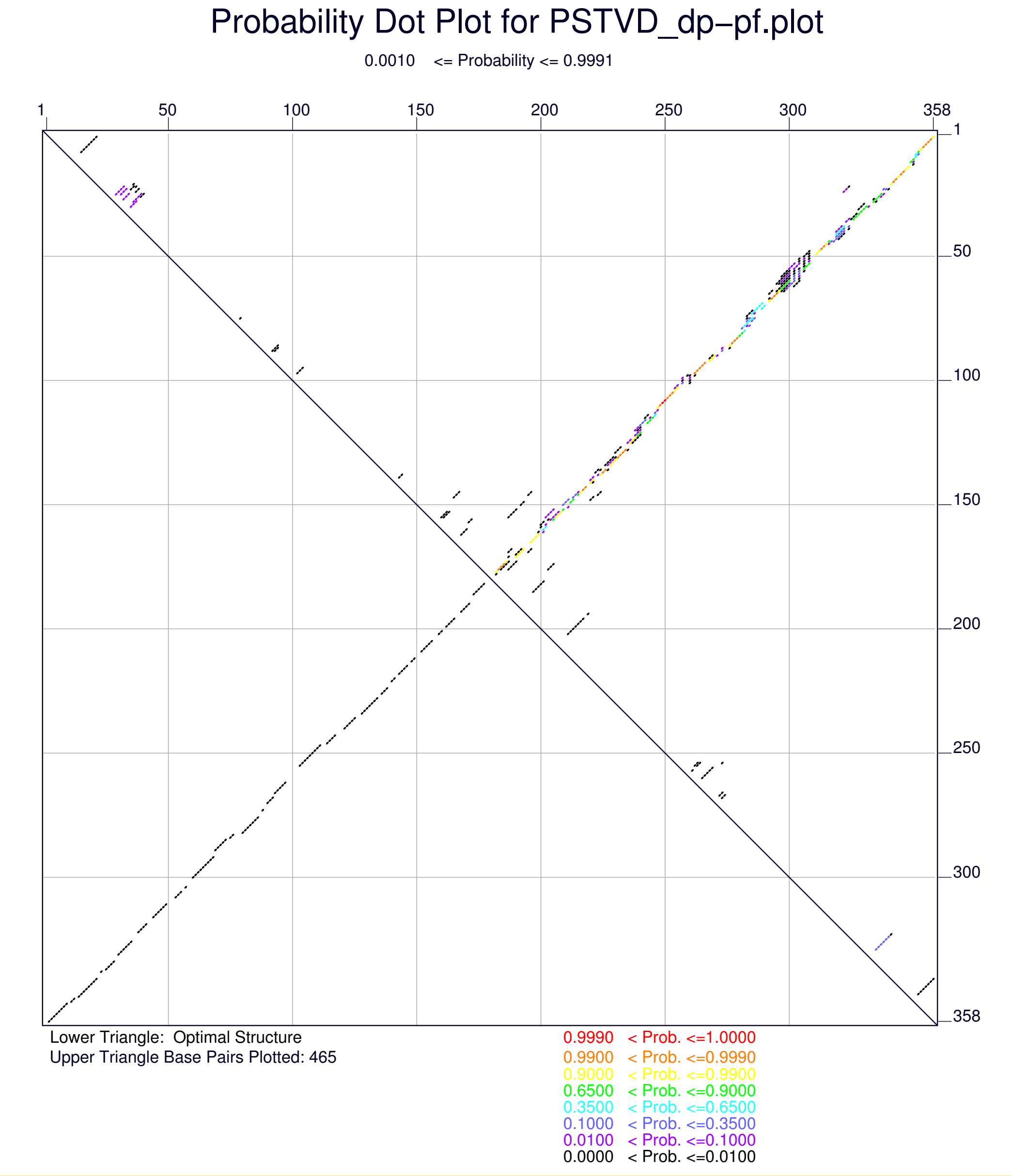


Figure 38: The “probability dotplot” for PSTVd at 37° C. Note the similarity to the “energy dot plot”. The probability cutoff is 0.001

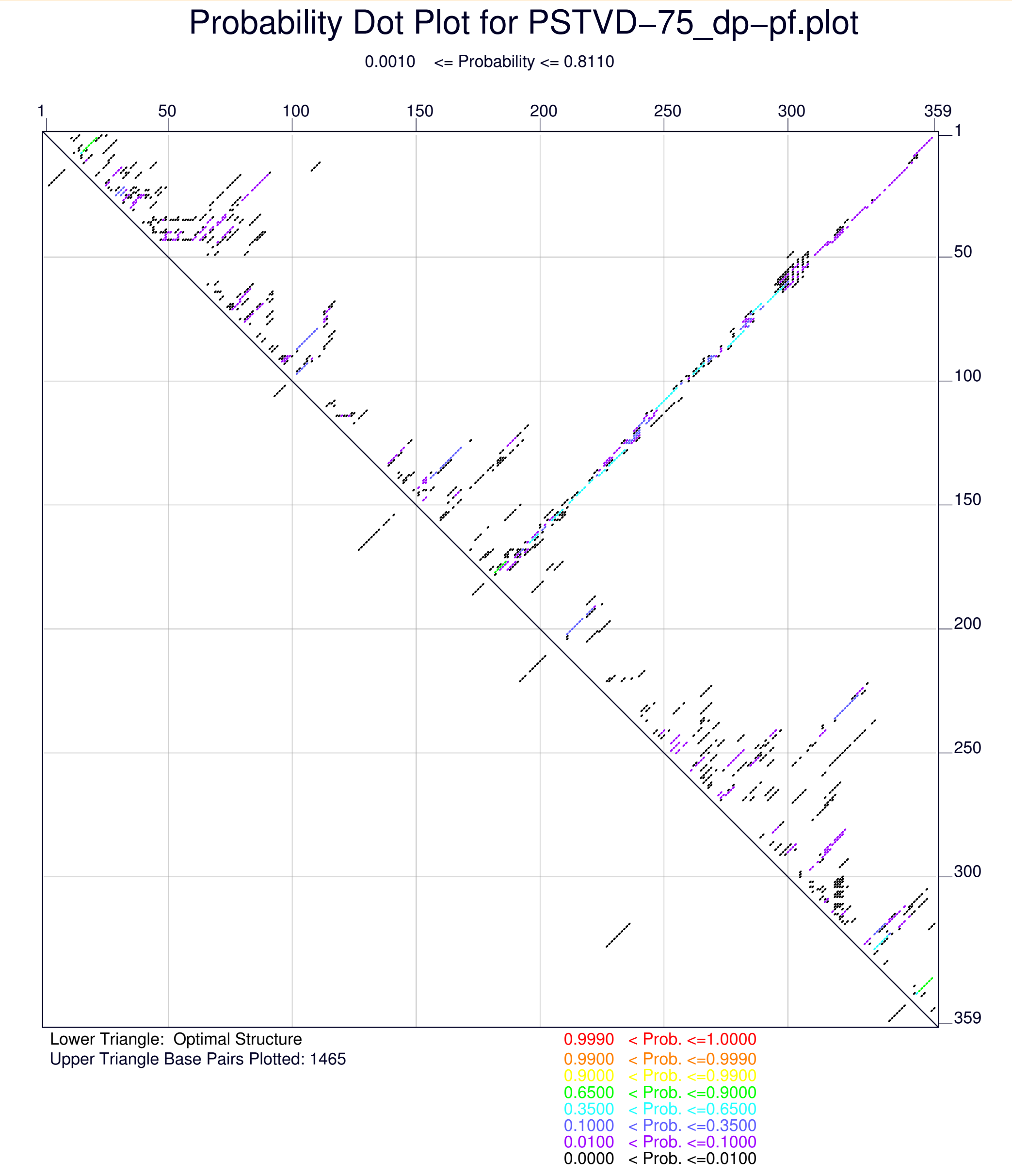


Figure 39: The "probability dotplot" for PSTVd at 75° C. The rod base pairs still show, but at low probabilities. Many more base pairs appear, mostly with low probabilities.

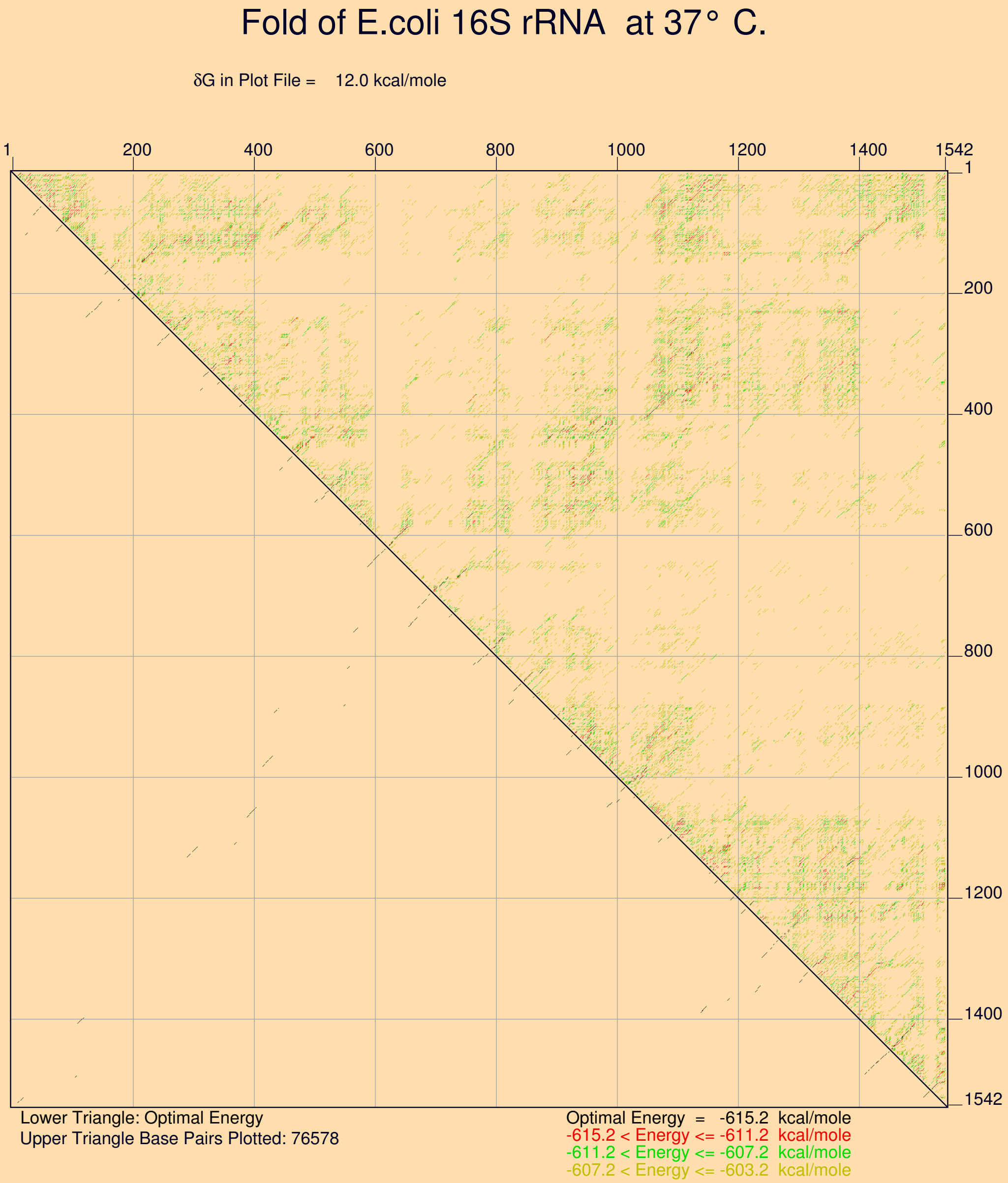


Figure 40: The energy dot plot for *E. coli* 16S rRNA. Although only 240 out of 477 base pairs are predicted in the optimal folding (50%), a cautious user cannot complain that the results are misleading. The enormous clutter in the dot plot shows that energy minimization is insufficient for folding prediction in this RNA.

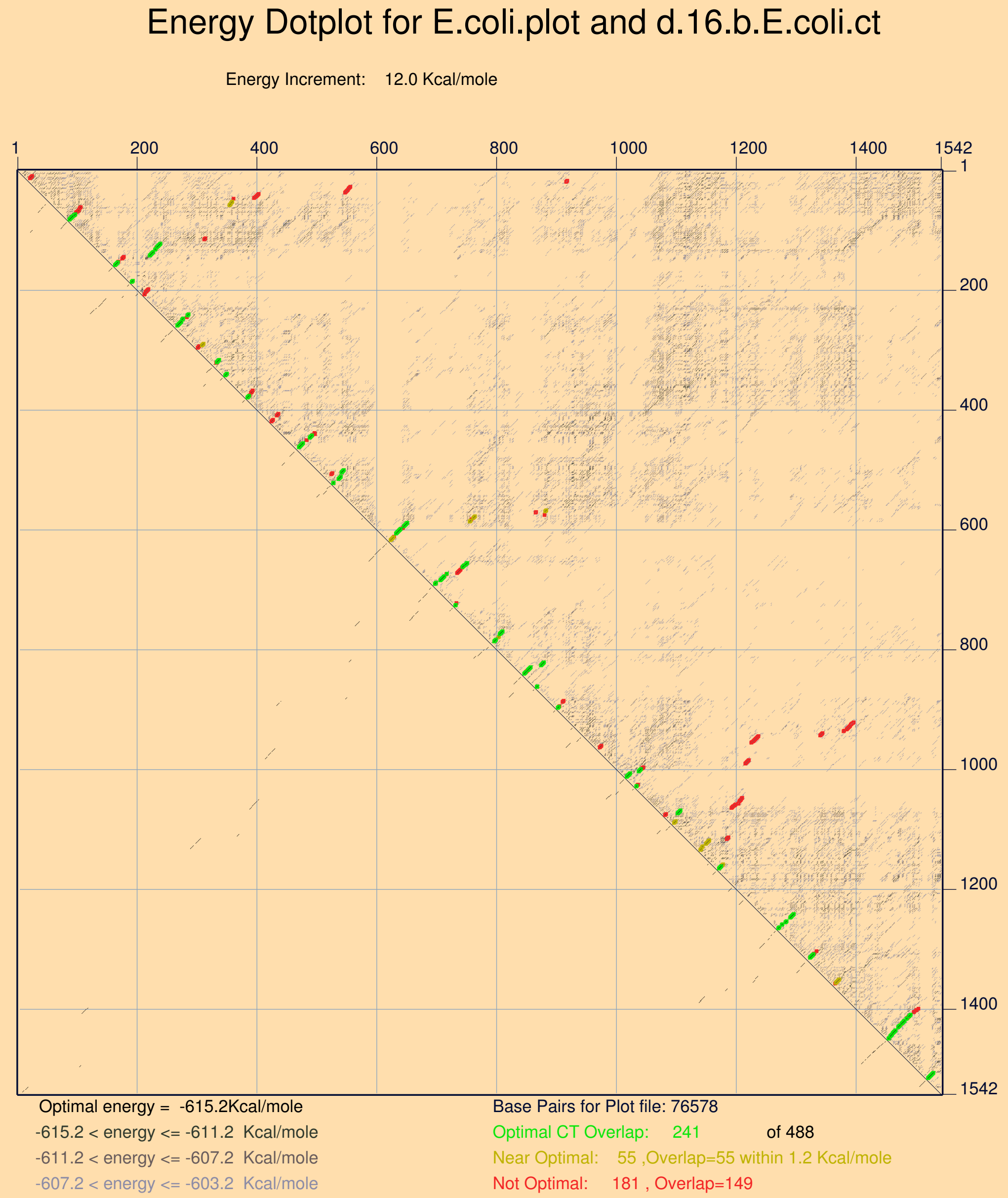


Figure 41: The "overlay" energy dot plot for *E. coli* 16S rRNA. The suboptimal base pairs are indicated in shades of gray. The correct secondary structure is overlayed on this dot plot. Green dots indicate correct base pairs that are in an optimal folding. The 55 yellow dots indicate "near misses". These correct base pairs are in foldings only 1.2 kcal/mole higher than the minimum. The red dots indicate the (other) correct base pairs that were not predicted.

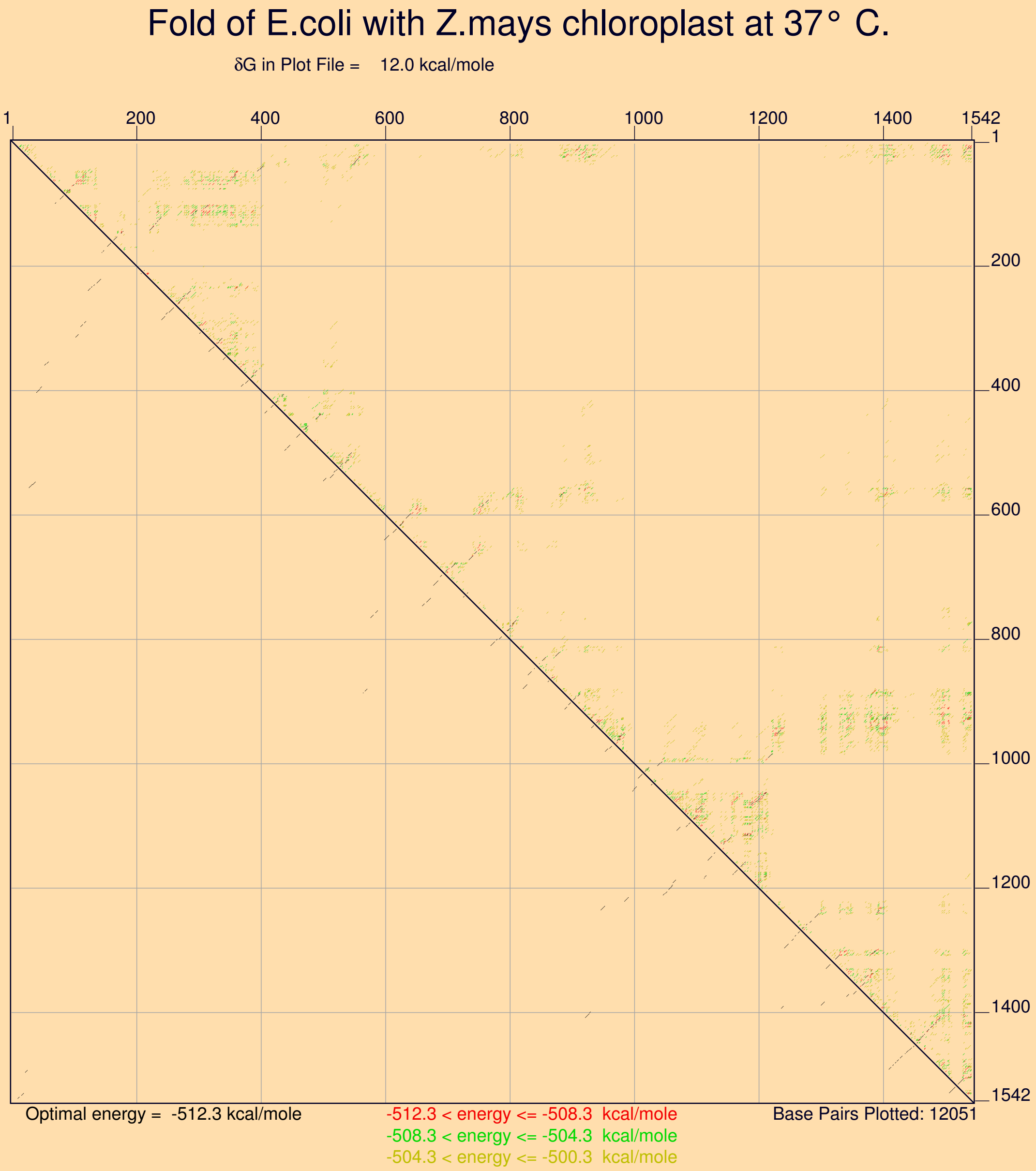


Figure 42: The energy dot plot for *E. coli* 16S rRNA folded together with *Zea mays* chloroplast 16S rRNA. Note how much less cluttered this dot plot is, compared to folding *E. coli* 16S rRNA alone (Figure 40). This suggests that the folding prediction might be better (more accurate).

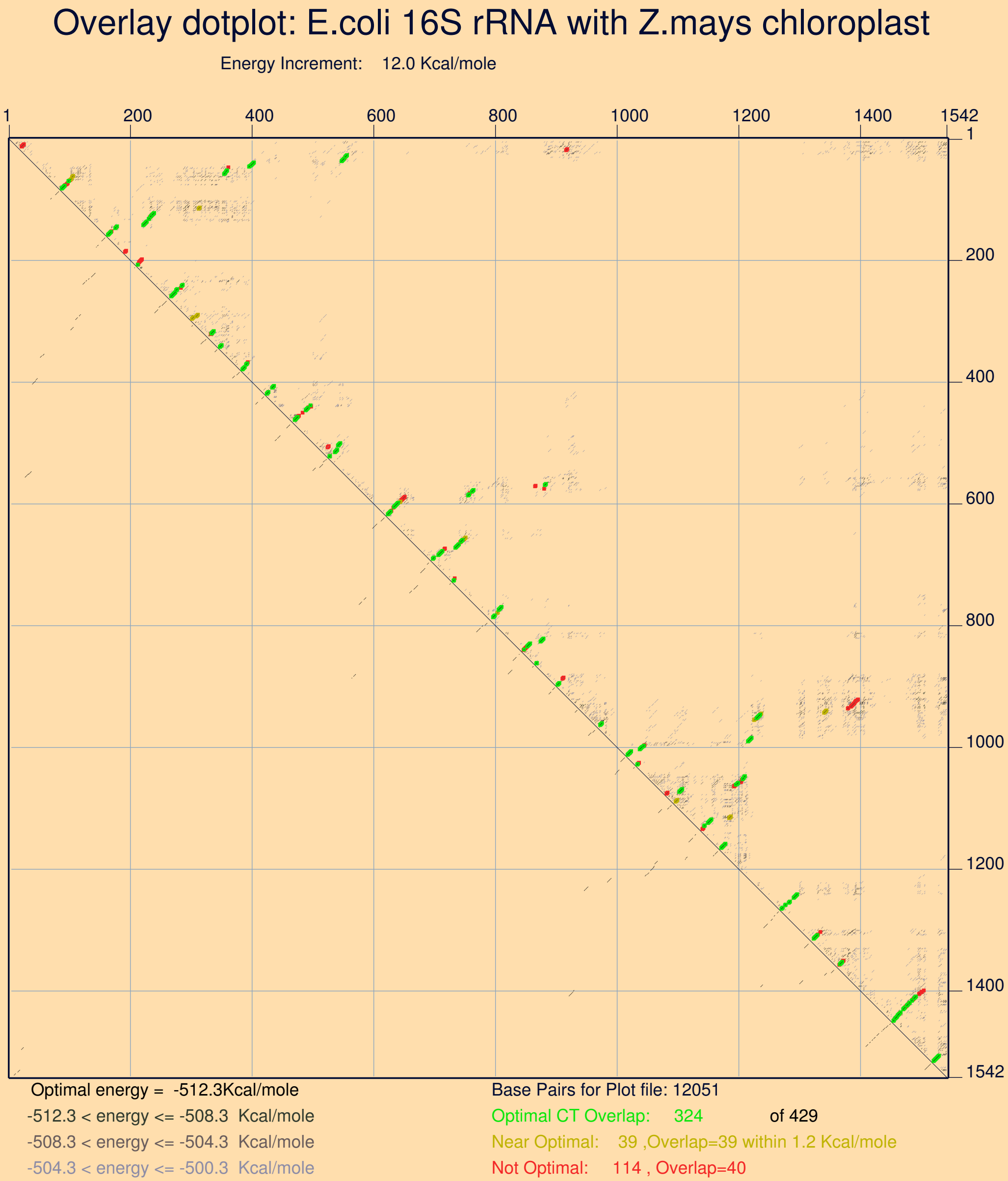


Figure 43: The "overlay" energy dot plot for *E. coli* 16S rRNA folded together with *Zea mays* chloroplast 16S rRNA. The optimal folding now has 324 out of the 477 base pairs in the correct secondary structure (68%). Adding the 39 "near misses" bring this total to 363 (76%).

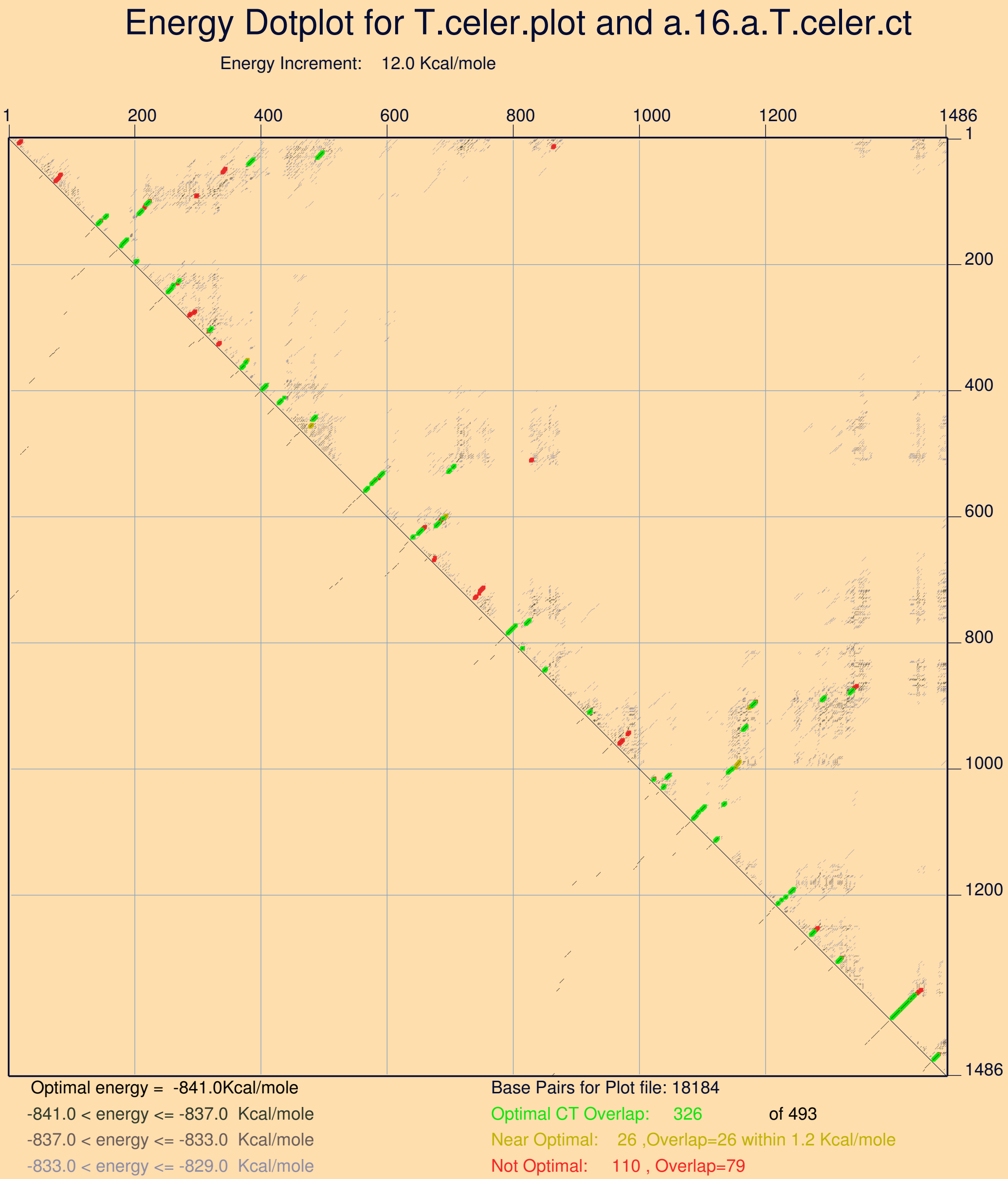


Figure 44: The "overlay" energy dot plot for *Thermococcus celer* 16S rRNA. Note the relative lack of clutter and the better outcome when this SSU rRNA is folded using energy rules alone.

in the i^{th} row and column of the "energy dot plot" . A value of 0 indicates that the i^{th} base must be single stranded in all foldings within the given free energy increment. A value of 1 indicates that the i^{th} base must be paired to a single partner in all foldings within the given free energy increment. A value of 1. Bases with (relatively) high $P-num$ values are said to be "poorly defined" in terms of secondary structure. They can pair with many other bases in allowable structures.

The probability dot plot gives precise information based on a statistical mechanical model. Each base pair can be labeled by it's probability, and each base can be labeled by the probability that it is paired. These $P-num$ values or probability values can be used to annotated individual foldings by using different. The "rainbow" colors red through violet indicate "good" through "bad". For $P-num$, this translates into low (well-defined) to high (poorly defined). For probabilities, base pairs with high probability values are colored in the red range. while those with low probabilities are colored in violet. Single stranded bases are colored according to their probability of being single-stranded. The coloring schemes are shown in Figure 45

RNA Folding by Comparative Sequence Analysis

One of the guiding principles in molecular biology is that structure is much more conserved than sequence. In proteins, 2 serine proteases, for example, can drift so far apart over time that sequence alignment is impossible. Nevertheless, greater than 50% structural similarity can remain. In RNA, the same principle is true. The implications for secondary structure are that secondary structure is conserved even though sequence drift occurs. Thus when 1 base of a pair changes, we usually find that its partner also changes so as to conserve that base pair. This phenomenon is called a *compensatory base change*. How can we detect them?

Suppose that we have an alignment of a group of homologous RNAs and that we are quite sure of the alignment. This is often possible by taking, for example, a group of tRNAs or 5S RNA's that are the same length. Formally, we are given a multiple alignment of m RNA sequences:

$$\begin{array}{rcl} R_1 & = & r_1(1), r_1(2), r_1(3), \ldots, r_1(n), \\ R_2 & = & r_2(1), r_2(2), r_2(3), \ldots, r_2(n), \\ R_3 & = & r_3(1), r_3(2), r_3(3), \ldots, r_3(n), \\ \vdots & & \vdots \\ R_m & = & r_m(1), r_m(2), r_m(3), \ldots, r_m(n). \end{array} \tag{36}$$

They are all the same length because they have already been aligned and some of the characters may be gaps ("-"). Suppose that base pairs $r_k(i) - r_k(j)$ can form in sequence k for all m sequences. In particular, this means that there are no gaps in columns i or j. This is not necessarily evidence for a conserved base pair.

What is needed is extra evidence in the form of compensatory base changes. Suppose that there are n_1 G-C, n_2 C-G, n_3 A-U and n_4 U-A base pairs between the 2 columns. Then the minimum number of compensatory changes that must have occurred during evolution is given by 1 less than the number of the n_ls that are not 0. This sort of evidence becomes more convincing when all the consecutive base pairs in a helix are conserved and have 1 or 2 compensatory base changes. To some, this is sufficient evidence for base pairing. The actual number of compensatory base changes can be much larger than the minimum. If the evolutionary past can be reconstructed, then it is observed that compensatory base changes occur over and over again during evolution.

More quantitative measures of the interdependence between pairs of columns. We can map A, C, G and U to the numbers 1, 2, 3 and 4, respectively. We can let e_i be the sample mean of the ith column. Then assuming independence of the columns, we would expect the covariance between distinct columns, i and j, to be

	Hex	% P-num	Probability		Hex	% P-num	Probability
🟥	ff0000	0.0-2.5	0.999	🟦	00ffff	50.0-52.5	0.500
🟥	ff1f00	2.5-5.0	0.998	🟦	00bfff	52.5-55.0	0.366
🟥	ff3f00	5.0-7.5	0.997	🟦	007fff	55.0-57.5	0.269
🟧	ff5f00	7.5-10.0	0.997	🟦	003fff	57.5-60.0	0.197
🟧	ff7f00	10.0-12.5	0.995	🟦	0000ff	60.0-62.5	0.144
🟧	ff9f00	12.5-15.0	0.994	🟦	1f00ff	62.5-65.0	0.106
🟧	ffbf00	15.0-17.5	0.991	🟦	3f00ff	65.0-67.5	0.077
🟨	ffdf00	17.5-20.0	0.988	🟪	5f00ff	67.5-70.0	0.057
🟨	ffff00	20.0-22.5	0.984	🟪	7f00ff	70.0-72.5	0.042
🟨	dfff00	22.5-25.0	0.978	🟪	9f00ff	72.5-75.0	0.031
🟩	bfff00	25.0-27.5	0.969	🟪	bf00ff	75.0-77.5	0.022
🟩	9fff00	27.5-30.0	0.958	🟪	df00ff	77.5-80.0	0.016
🟩	7fff00	30.0-32.5	0.943	🟪	af00cf	80.0-82.5	0.012
🟩	5fff00	32.5-35.0	0.923	🟪	7f009f	82.5-85.0	0.009
🟩	3fff00	35.0-37.5	0.894	🟪	5f007f	85.0-87.5	0.006
🟩	1fff00	37.5-40.0	0.856	🟪	3f005f	87.5-90.0	0.005
🟩	00ff00	40.0-42.5	0.803	⬛	1f003f	90.0-92.5	0.003
🟩	00ff3f	42.5-45.0	0.731	⬛	09001f	92.5-95.0	0.003
🟩	00ff7f	45.0-47.5	0.634	⬛	040009	95.0-97.5	0.002
🟩	00ffbf	47.5-50.0	0.500	⬛	000000	97.5-100.0	0.001

Figure 45: The coloring schemes for $P-num$ and probability annotation. $P-num$ values are normalized so that the highest value for a certain free energy increment is 100%. Forty different colors are used. Our software can easily used other color tables.

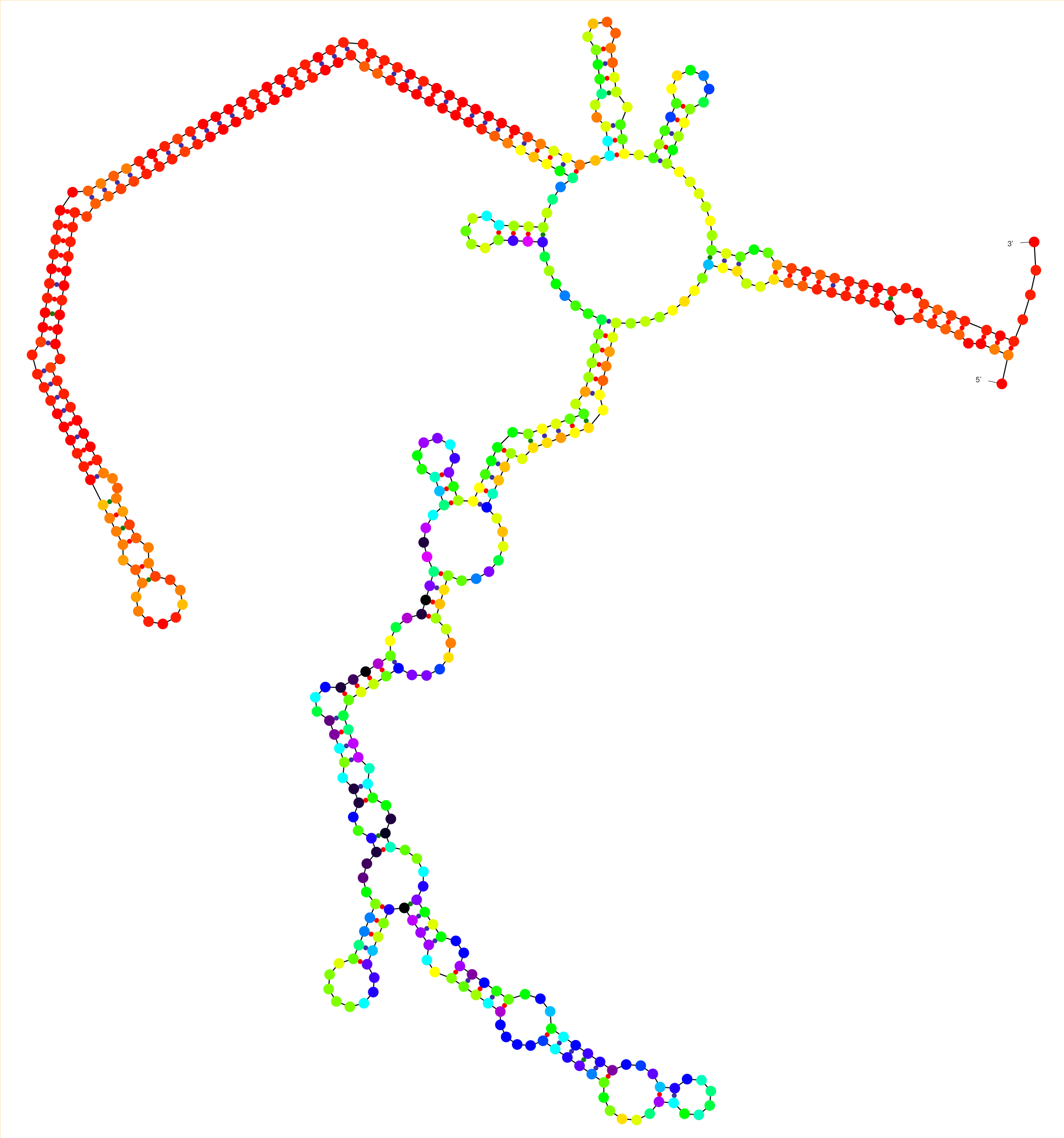


Figure 46: The optimal folding of the *Xenopus Laevus* sequence whose “energy dot plot” was shown in Figure 21. The annotation uses $P-num$ values at a free energy increment of 11.1 kcal/mole.

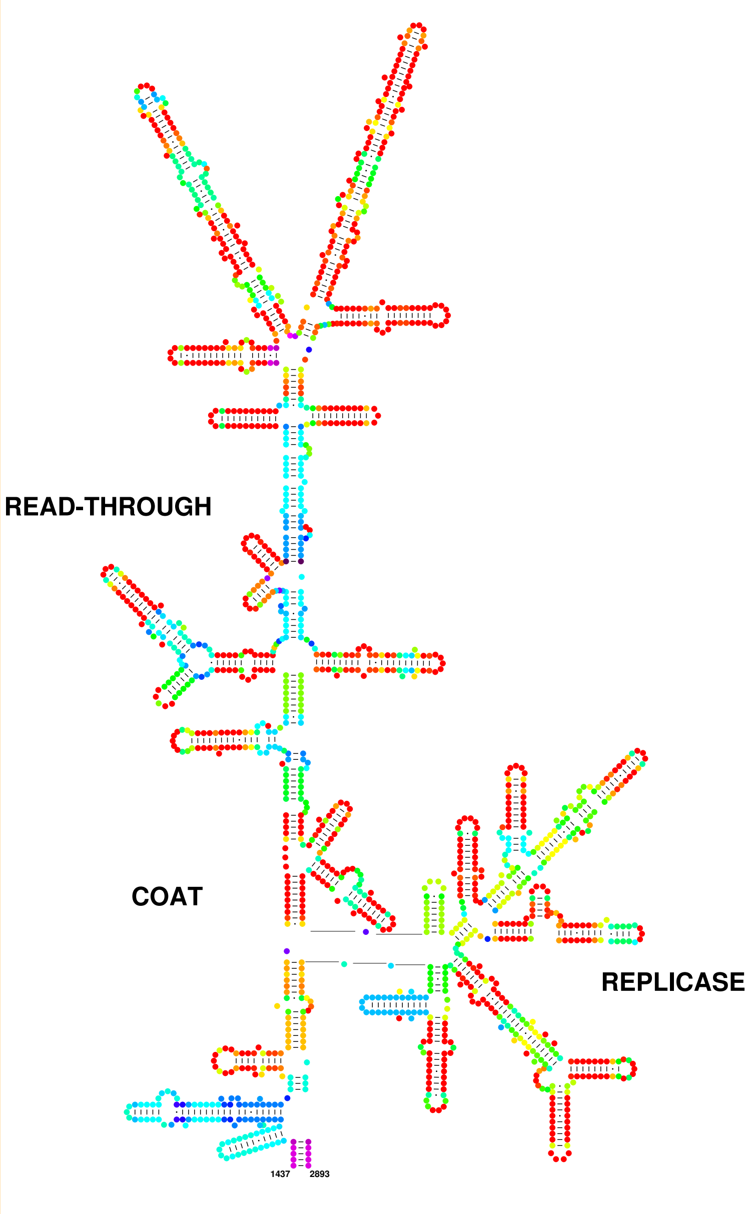


Figure 47: A proposed folding of the “central hairpin” of coliphage Qβ using probability annotation.

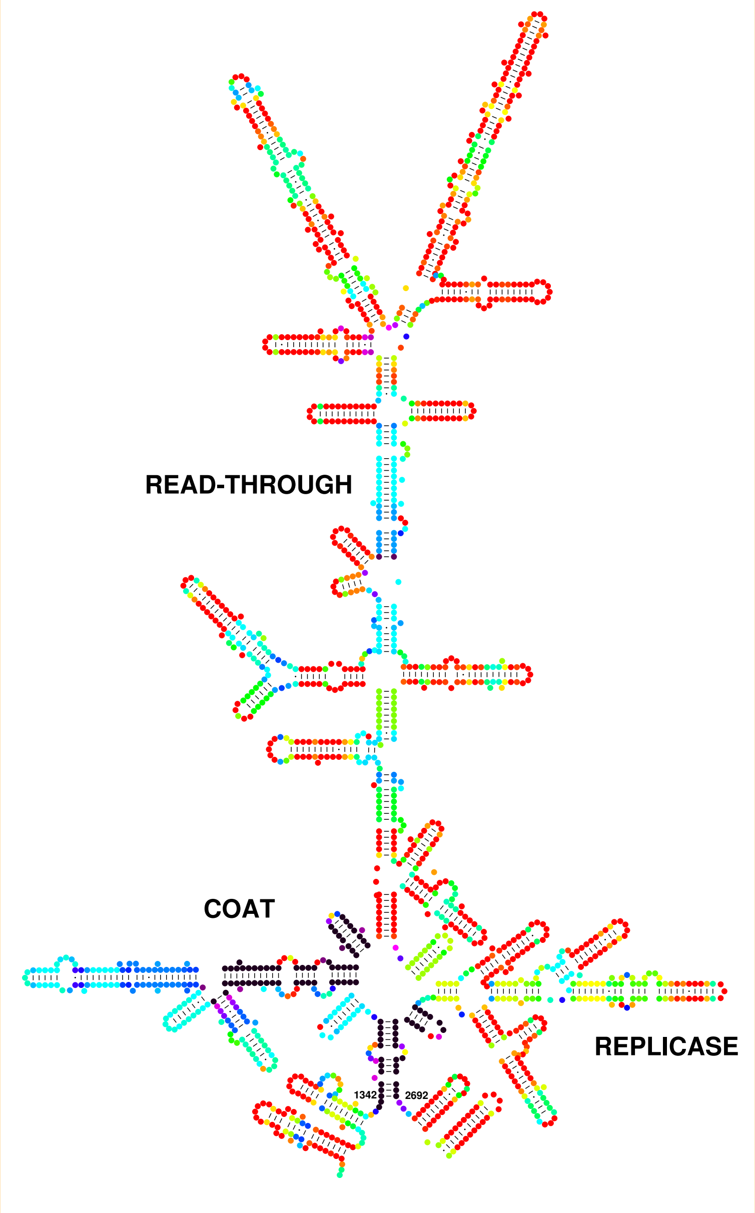


Figure 48: A alternative folding of the “central hairpin” of coliphage Qβ using probability annotation. Note the low probabilities for some of the different base pairs in the base of this folding.

0. The covariance, is given by:

$$cov(i,j) = \sum_{h=1}^{m} (r_h(i) - e_i)(r_h(j) - e_j), \tag{37}$$

where $e_l = \frac{1}{m} \sum_{h=1}^{m} r_h(l)$.

The method that is actually used today is the *mutual information content* of 2 columns. For a single column, i, let $f_i(N)$ be the frequency in the i^{th} column of the nucleic acid, N, where $N \in \{A,C,G,U\}$. Looking at another column, j, let $f_{i,j}(N_1,N_2)$ be the joint frequency of the 2 nucleotides, N_1 from the i^{th} column and N_2 from the j^{th} column. On the hypothesis that any 2 columns are independent, we would expect $f_{i,j}(N_1,N_2)$ to be roughly $f_i(N_1)f_j(N_2)$. Thus we would expect $\log \frac{f_{i,j}(N_1,N_2)}{f_i(N_1)f_j(N_2)}$ to be roughly 0. This leads to the definition of mutual information content, $H(i,j)$ between 2 different columns, i and j:

$$H(i,j) = \sum_{N_1,N_2 \in \{A,C,G,U\}} f_{i,j}(N_1,N_2) \log_2 \frac{f_{i,j}(N_1,N_2)}{f_i(N_1)f_j(N_2)}. \tag{38}$$

COMMENT

The logarithm is taken to the base 2 so that the answer is in bits. When columns i and j covary perfectly, one might expect

$$f_{i,j}(N_1,N_2) = f_i(N_1) = f_j(N_2) \tag{39}$$

for all pairs of nucleotides. This quantity would be 0 when N_1 and N_2 cannot pair. Thus we expect a maximum value of 2 bits when there is perfect correlation and 0 for complete randomness.

Examples

Figure 49 shows the alignment of 20 5S rRNA sequences. They were deliberately chosen to have length 120, so that the alignment would be trivial and not in doubt. Figure 50 illustrates the limited amount of information on secondary structure that can be deduced from just 20 sequences. Using an alignment of the entire Eukaryote 5S rRNA database with gaps (Figure 51) gives a mutual information dot plot where the noise (low MI with many isolated base pairs) has been eliminated and the signal amplified. Figure 53 illustrates the consensus folding models for Prokaryote and Eukaryote 5S rRNA.

In 1990, Winker *et al.* [11] reported a method that finds conserved base pairs through covariance analysis. This includes adjustment of alignments. Chan *et al.* [12] have a program that will find conserved helices in pre-aligned sequences. A certain number of compensatory changes are required per helix. The method of Han and Kim [13] starts with pre-aligned sequences and predicts common foldings compatible with the covariation evidence.

```
                  10        20        30        40        50        6
Salmo gcuuacGgcCAuAccAgccugaauacgCCcgaUCuCgUccGAuCucgGaAGcuAagCag
Misgu gcuuacGgcCAuAccAcccugagcacgCCcgaUCuCgUccGAuCucgGaAGcuAagCag
Misgu gcuuacGgcCAcAccAaccugagcaagCCcgaUCuCgUcuGAuCucgGaAGccAagCag
Chrys gccuacGacCAuAccAccaugaguauaCCgguUCuCgUccGAuCaccGgAGucAagCau
Aurel gccuacGacCAuAccAccaugaauacaCCgguUCuCgUccGAuCaccGaAGuuAagCau
Nemop gucuacGacCAuAccAcaaugaacacaCCgguUCuCgUccGAuCaccGaAGuuAagCau
Antho gucuacGgcCAuAccAccgggaaaaaaCCgguUCuCgUccGAuCaccGaAGucAagCcc
Halic gccugcGgcCAuAccAcguugaaugcaCCgguUCcCaUcuGAaCaccGaAGuuAagCaa
Halic gccuacGgcCAuAccAcguugaaaacaCCgguUCuCgUcuGAuCaccGaAGuuAagCaa
Brach gccuagGacCAuAucAcguugaaugcaCCgguUCuCgUccGAuCaccGaAGuuAagCaa
Acyrt ggcaacGacCAuAccAcguugaauacaCCaguUCuCgUccGAuCacuGaAGuuAagCaa
Bomby gccaacGucCAuAccAuguugaauacaCCgguUCuCgUccGAuCaccGaAGucAagCaa
Plano gauagcGucCAuAccAcacugaaaacaCCgguUCuCgUccGAuCaccGcAGuuAagCag
Artem accaacGgcCAuAccAcguugaaaguaCCcagUCuCgUcaGAuCcugGaAGucAcaCaa
Duges gucgacGcuCAuAcuAgguuggguccaCCcgaUCuCgUucGAuCucgGcAGuuAaaCaa
Tetra guugucGgcCAuAcuAaggugaaaacaCCggaUCcCaUucGAaCuccGaAGuuAagCgc
Param guugguGgcCAuAcuAagccuaaagcaCCggaUCcCaUucGAaCuccGaAGuuAagCgg
Bress guuaucGgcCAuAcuAagccaaaagcaCCggaUCcCaUucGAaCuccGaAGuuAagCgg
Euplo gcuaucGgcCAuAcuAagccaaaugcaCCggaUCcCaUccGAaCuccGaAGuuAagCgg
Bleph guugucGgcCAuAcuAugccuaacgcaCCagaUCcCaUccGAaCucuGaAGuuAagCgg
            -  -- -  -           --   -- - -  -- -   - --  -  -
```

Figure 49: A trivial alignment of 20 5S rRNA sequences chosen for their common length.

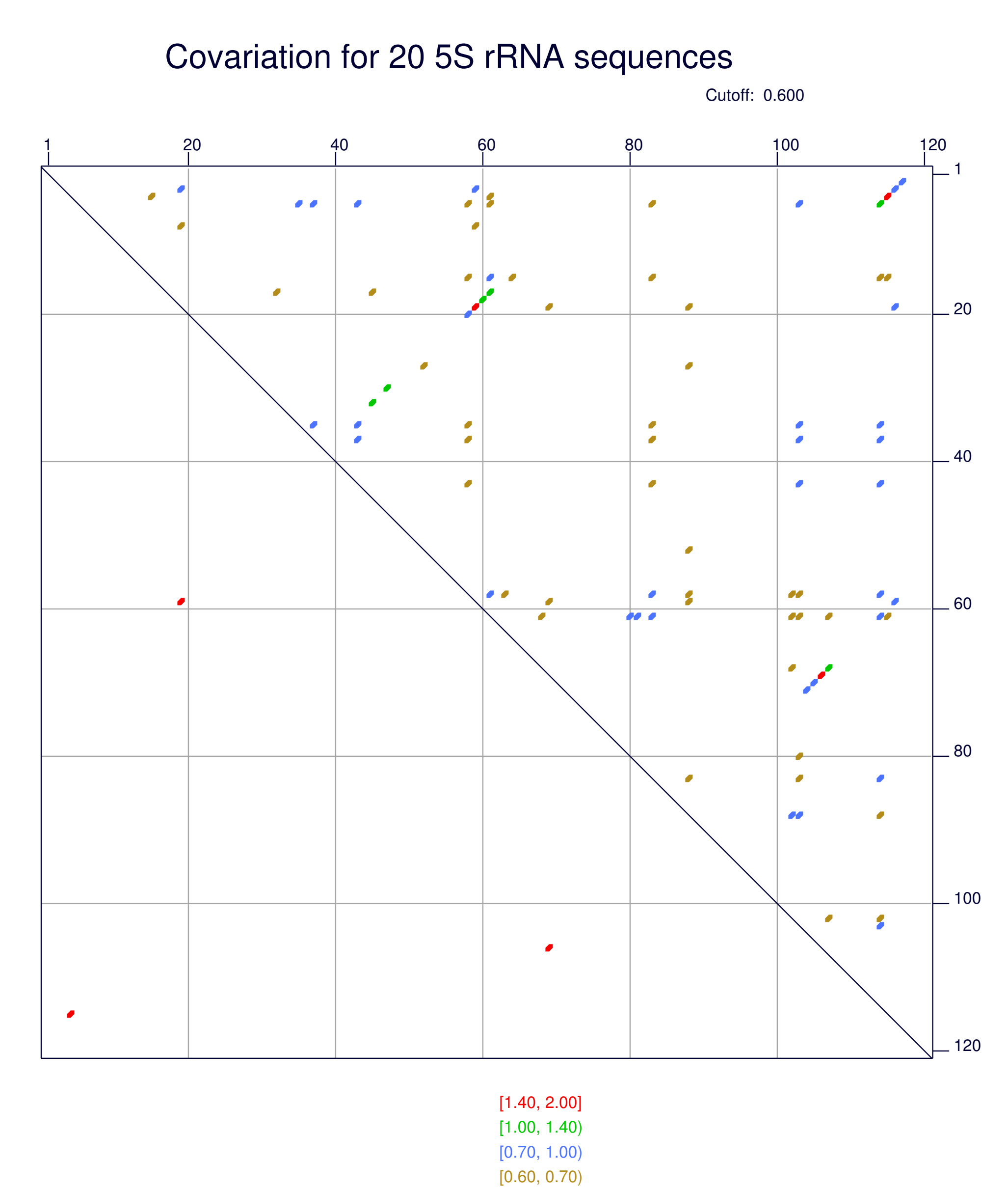


Figure 50: Mutual information plot for the 20 5S rRNA sequences from Figure 49.

```
A_kurodai  --G-GCU-ACGGCCAUACCAC-G-U-UG-AACCUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACGUCGGGCGUG-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAACACC---A---CGUGCUGUAGGCG------
A_rufus    --G-UCU-ACGGCCAUACCAC-G-U-UG-AAAAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACGUCGGGCGUG-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAACACC---A---CGUGUUGUAGACA------
H_pomatia  --G-UCU-ACGGCCAUACCAC-G-U-UG-AAAAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACGUCGGGCGUG-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAACACC---A---CGUGUUGUAGAUA------
I_illeceb  --G-CUU-ACGGCCAUAUCAC-G-C-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUAGAGCCUA-GU-CAGUACUUGGAUGGGUGACC-GCCUGGGAAUACU---A---GGUGCUGUAAGCAU-----
O_vulgaris --G-CUU-ACGGCCAUAUGAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACAUCGAGCCUA-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAACCCU---A---GGUGCUGUAAGCAU-----
S_officina --G-CUU-ACGGCCAUAUCAC-G-C-UG-AAAACACCGGUUCUCGUCC-GAACACCGA-AGUUAAGCAACGUAGAGCCUG-GU-GAGUACUUGGAUGGGUGACC-GCUUGGGAAUACC---A---GGUGCUGUAAGCAU-----
L_polyphe  --G-UCA-UCGUUCAUACCAC-G-U-UG-AAAGCGCCGGUUCUCGUCU-GAUCCCCGA-AGCUAAGCAACGUCGGGCCCG-GU-UAGUACUUGGGAGGGUGACC-ACCUGGGAAUACC---G---GGUGAUGAUGACAU-----
A_diadema  --G-CCA-ACGGCCAUACCAU-G-C-UG-AAAGCACCGGUUCUCGUCU-GAUCACCGC-AGUUAAGCAGCAUCGGGCGCG-GU-CAGUACUUGGGAGGGUGACC-ACCUGGGAACACC---G---CGUGCUGUUGGCAU-----
E_califor  --G-CCA-ACGGCCAUACCAU-G-C-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAGCAUCGGGCGCG-GU-CAGUACCUGGGAGGGUGACC-ACCUGGGAACACC---G---CGUGCUGUUGGCAU-----
D_magna    --G-UCA-ACGGCCAUACCAC-G-U-AG-AACUCACCCGGUCUCGUCA-GCUCCCGGA-AGUUAAGCUACGUCGGGUCCC-GU-UAGUACUUGGAUGGGUGACC-GCUUGGGAAUACG---G---GAUGCUGUUGGCAU-----
L_oceanica --A-UCA-ACGGCCAUACCAC-G-U-UG-AAAACACCGCUUCUCGUCC-GAUCAGCGA-AGUUAAGCAACGUAGGGUCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAACACC---A---GAUGCUGUUGGCUU-----
H_gammarus --G-UCA-ACGGCCAUACCAC-G-U-UG-AAAACACCGCUUCUCGUCC-GAUCAGCGA-AGUUAAGCAACAUUGGGUCUG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---A---GAUGCUGUUGGCUU-----
C_pagurus  --G-CCA-ACGACCAUACCAC-G-U-UG-AAUGCACCGCUUCUCGUCC-GAUCAGCGA-AGUUAAGCAACGUUGGGUCUG-GA-UAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---A---GAUGUUGUUGGCAU-----
Spirobolus --G-UCA-ACGGCCAUACUAC-G-U-UG-AAAACACCAGUUCUCGUCU-GAUCACUGA-AGUUAAGCAACGUCGGGCCCA-GU-CAGUACCUGGAUGGGUGACC-GCCUGGGAAUACU---G---GGUGCUGUUGGCAUU----
L_migrato  --G-CCA-ACGGCCAUACCAC-G-U-UG-AAUACACCGGUUCUCGUCC-GAUCAGCGA-AGUUAAGCAACGUUGGGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---G---GGUGCUGUUGGCCC-----
D_melanog  --G-CCA-ACGACCAUACCAC-G-C-UG-AAUACAUCGGUUCUCGUCC-GAUCACCGA-AAUUAAGCAGCGUCGGGCGCG-GU-UAGUACUUAGAUGGGGGACC-GCUUGGGAACACC---G---CGUGUUGUUGGCCU-----
A_domestic --G-CCA-ACGUCCAUACCAC-G-U-UG-AAAGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAGCGUCGGGCGCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAACCCC---G---CGUGACGUUGGCAU-----
A_magnolia --G-GCA-ACGACCAUACCAC-G-U-UG-AAUACACCAGUUCUCGUCC-GAUCACUGA-AGUUAAGCAACGUCGGGCGUA-GU-UAGUACUUGGAUGGGUGACC-GCUUGGGAACACU---A---CGUGCCGUUGGCAU-----
P_cynthia  --G-CCA-ACGUCCAUACCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCGCG-GU-CAGUACUUGGGUGGGUGACC-ACCUGGGAACACC---G---CGUGCCGUUGGCU------
B_mori1    --G-CCA-ACGUCCAUACCAU-G-U-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACAUCGGGCGUG-GU-CAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---A---CGUGAUGUUGGCUU-----
B_mori2    --G-CCA-ACGUCCAUACCAU-G-U-UG-AAUACACUGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACAUCGGGCGUG-GU-CAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---A---CGUGAUGUUGGCUU-----
B_mori3    --G-CCA-ACGUCCAUACCAU-G-U-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACAUCGGGCGUA-GU-CAGUACUUGGAUGGGUGACC-GCCUGGGAACACU---A---CGUGAUGUUGGCUU-----
A_pernyi   --G-ACA-ACGUCCAUACCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCGCG-GU-CAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---G---CGUGACGUUGGCUU-----
T_molitor  --G-ACA-ACGGCCAUAGCAC-G-C-UG-UAAAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAGCGUUGCGCGCG-GU-UAGUACUUGGAUGGGUGACC-GCUUGGGAACACC---G---CGUGCCGUUGUCAU-----
H_rufipes  --G-UCA-ACGGCCAUAGCAC-G-C-UG-UAAAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAGCGUUGCGCGCG-GU-UAGUACCUGGAUGGGUGACC-GCUUGGGAACACC---G---CGUGCCGUUGACAU-----
A_vulgaris --G-UCU-ACAGCCAUACUAC-G-C-UG-AAAACACCGGUUCUCGUCU-GAUCACCGA-AGUUAAGCAGCAUCAGGCCCA-GU-CAGUACUUGGAUGGGAGACC-UCCUGGGAAUACU---G---GGUGCCGUAGACUU-----
A_pectini  --G-UUU-ACGACCAUACUAC-G-U-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCUUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUCGUAGGCUU-----
H_pulcher  --G-UUU-ACGACCAUACCAU-G-C-UG-AAUAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAGCAUAGGGCCCG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGUGUUGUAGGCAU-----
P_depress  --G-CCU-ACGACCAUACCAU-G-C-UG-AAUAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAGCAUAGGGCCCG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGUGUUGUAGGCUU-----
S_kowalev1 --G-CCU-ACGGCCAUACCAC-G-U-AG-AAUGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCUGCGUCGGGCGUG-GU-UAGUACUUGCAUGGGAGACC-GGCUGGGAAUACC---A---CGUGCUGUAGGCUU-----
S_kowalev2 --G-CCU-ACGGCCAUACCAC-G-U-AG-AAUGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCUGCGUCGGGCGUG-GU-UAGUACUUGCAUGGGAGACC-GGCUGGGAAUACC---A---CGUGCCGUAGGCUU-----
H_roretzi  --A-UCU-ACGGCCAUACCAC-A-U-UG-AGAGCACCCGAUCUCGUUC-GAUCUCGGA-AGUUAAGCAAUGUAGGGAUCA-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACU---G---GUUGCUGUAGAAA------
E_japonic1 --G-CCU-ACGACCAUAUCAC-C-C-UG-AAUACGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGAGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUUGUAGGCUU-----
E_japonic2 -GG-CCU-ACGACCAUAUCAC-C-C-UG-AAUGCGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGAGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUUGUAGGCUU-----
E_japonic1 --G-CCU-ACGACCAUAUCAC-C-C-UG-AAUACGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGAGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUUGUAGGCUU-----
S_canicul  --G-CCU-ACGGCCAUACUAG-U-C-UG-AAAACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGAUUCAGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGCAGUAGGCUUU----
M_fossilS  --G-CUU-ACGGCCAUACCAC-C-C-UG-AGCACGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACU-GCCUGGGAAUACC---A---GGUGUUGUAAGCUU-----
M_fossilO  --G-CUU-ACGGCCACACCAA-C-C-UG-AGCAAGCCCGAUCUCGUCU-GAUCUCGGA-AGCCAAGCAGGGUUGGGCCUG-GU-UAGUACUUGGAUGGGAGACU-GCCUGGGAAUACC---A---GGUGUUGUAAGCUU-----
T_tincaS   --G-CUU-ACGGCCAUACCAC-C-C-UG-AGCACGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGCUGUAAGCUU-----
T_tincaO   --G-CUU-ACGGCCAUACCAC-C-U-UG-AGAACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGCUGUAAGCUU-----
X_boreal2  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAAGUGCCCGAUAUCGUCU-GAUCUCGGA-AGCCAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUCGUAGGCUU-----
S_oshimae  --G-UUU-ACGACCAUAUCAC-G-U-UG-AAUAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAGCGUCGAGCCUA-GU-NAGUACUUGGAUGGGUGACC-GCCUGGGAAUACU---A---GGUGCCGUAGACUU-----
X_laevis3  --G-CCU-ACGGCCACACCAC-C-C-UG-AAAGUGCCCGAUCUCGUCU-GAUCUCGGA-AGCCAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUCGUAGGCUUU----
X_laevis2  --G-CCU-ACGGCCACACCAC-C-C-UG-AAAGUGCCUGAUCUCGUCU-GAUCUCAGA-AGCGAUACAGGGUCGGGCCUG-GU-UAGUACCUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUCGUAGGCUU-----
X_laevis1  --G-CCU-ACGGCCACACCAC-C-C-UG-AAAGUGCCCGAUCUCNUCU-GAUCUCGGA-AGCGAUCCAGGGCCGGGCCUG-GU-UAGUACCUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGUGUCGUAGGCUU-----
X_tropic3  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAAGCGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGACGGGCUUG-GU-UAGUACCUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUUGUAGGCUU-----
B_taurus   --G-UCU-ACGGCCAUACCAC-C-C-UG-AACGCGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGUGCUGUAGGCUU-----
X_tropic2  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAAGCGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGACGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---U---GGUGCUGUAGGCNU-----
X_tropic1  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAAGCGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGACGGGCUUG-GU-UAGUACCUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGUUGUAGGCNU-----
I_iguana   --G-CCU-ACGGCCAUACCAC-C-C-UG-AACACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGUGCUGUAGGCUU-----
T_carolina --G-UCU-ACGGCCAUACCAC-C-C-UG-AACACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-UCCUGGGAAUACU---G---GGUGCUGUAGGCUU-----
G_gallus1  --G-CCU-ACGGCCAUCCCAC-C-C-UG-GUAACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-UCCUGGGAAUACC---G---GGUGCUGUAGGCUUU----
G_gallus2  --G-CCU-ACGGCCAUACCAC-C-C-UG-GAAACGCCCGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCUCG-GU-UAGUACUUGGAUGGGAGACU-GCCUGGGAAUACC---G---AGUGUCGUAGGCGU-----
P_walt1O1  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAUGCGCCUGAUCUCGUCU-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACU-GCCUGGGAAUACC---A---GGUGCUCUAGGCAU-----
P_walt1S   --G-CCU-ACGGCCAUACCAC-C-C-UG-AAUGCGCCCGAUCUCGUCC-GAUCUCGGA-AGCUAAGCAGGGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---A---GGUGCUGUAGGCAU-----
P_walt1O2  --G-CCU-ACGGCCAUACCAC-C-C-UG-AAUGCGCCUGAUCUCGUCU-GAUCUCAGA-AGCUAAGCAGNGUCGGGCCUG-GU-UAGUACUUGGAUGGGAGACU-GCCUGGGAAUACC---A---GGUGCUGUAGGCAU-----
A_castell1 --G-GAU-ACGGCCAUACUGC-G-C-AG-AAAGCACCGCUUCCCAUCC-GAACAGCGA-AGUUAAGCUGCGCCAGGCGGU-GU-UAGUACUGGGGUGGGCGACC-ACCCGGGAAUCCA---C---CGUGCCGUAUCCU------
A_castell2 --G-GAU-ACGGCCAUACUGC-G-C-AG-AAAGCACCGCUUCCCAUCC-GAACAGCGA-AGUUAAGCUGCGCGAGGCGGU-GU-UAGUACUGGGGUGGGCGACC-ACCCGGGAAUCCA---C---CGUGCCGUAUCCU------
Ac_crinita --A-GGA-ACGGCCAUACCAC-G-U-CG-AUCGCACCACAUCCCGUCC-GCUCUGUGA-AGUUAAGCGACGUCGGGCCAG-GC-UAGUACUACGGUGGGGGACC-ACGUGGGAAGCCC---U---GGUGCUGUUCCU-------
B_japonicu --G-UUG-UCGGCCAUACUAU-G-C-CU-AACGCACCAGAUCCCAUCC-GAACUCUGA-AGUUAAGCGGCAUAAGGCGAG-GU-UAGUACUUGGGUGGGGGACC-GCCAGGGAAGCCC---U---CGUGCUGACAGCUA-----
B_vorax    --G-UUA-UCGGCCAUACUAA-G-C-CA-AAAGCACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGGCUUAAGGCAUG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGCCC---A---UGUGCUGAUAGCUU-----
C_campylu  --G-CUG-UCGGCCAUACUAA-G-G-UG-AACACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGCCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---C---AGUGUCGACAGCCU-----
C_colpoda  --G-CUG-UCGGCCAUACUAA-G-A-UG-AACACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGUCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---C---AGUGUCGACAGCCU-----
C_fascic   --G-AGU-ACGACCAUACUUG-A-G-UG-AAAACACCAUAUCCCGUCC-GAUUUGUGA-AGUUAAGCACCCACAGGCUUA-GU-UAGUACUGAGGUCAGUGAUG-ACUCGGGAACCCU---G---AGUGCCGUACUCCC-----
C_paradox  --G-UGU-ACGGCUAUACUAC-C-G-GA-AAAGCGCCCGUUCCCGUCC-GAUUACGGA-AGCCUAGCCCGGUCAGGCCCG-AC-UAGUACUAGGGUGGGGGACC-ACCUGGGAACAUC---G---GGUGCUGUACACU------
D_discoid  --G-UAU-ACGGCCAUACUAG-G-U-UG-GAAACACAUCAUCCCGUUC-GAUCUGAUA-AGUAAAUCGACCUCAGGCCUU-CC-AAGUACUCUGGUUGGAGACA-ACAGGGGAACAUA---G---GGUGCUGUAUACU------
E_gracil1  --G-CGU-ACGGCCAUACUAC-C-G-GG-AAUACACCUGAACCCGUUC-GAUUUCAGA-AGUUAAGCCUGGUCAGGCCCA-GU-UAGUACUGAGGUGGGCGACC-ACUUGGGAACACU---G---GGUGCUGUACGCUU-----
Eu_gracil2 --G-AGU-ACGGCCAUACUAC-C-G-GG-AAUACACCUGAACCCGNUC-GAUUUCAGA-AGUUAAGCCGGGUUAGGCCCA-GU-UAGUACUGAG-UGGGCGACC-ACUUGGGAACACU---G---GGUGCUGUACGCUU-----
E_woodruf  --G-CUA-UCGGCCAUACUAA-G-C-CA-AAUGCACCGGAUCCCAUCC-GAACUCCGA-AGUUAAGCGGUUUAAGGCCUG-UU-AAGUACUGAGGUGGGGGACC-ACUCGGGAACUUC---A---GGUGCUGAUAGCUU-----
T_patula   --G-CUG-UCGGCCAUACUAA-G-G-UG-AACACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGCCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---U---AGUGUCGACAGCCU-----
M_inverta  -AA-CUA-ACGACCAUACGCG-C-C-AU-AAUGCACCCGAAACCUUCA-GAACUCGGA-AGUCAAGCUGGUGUCGGCCUG-CC-UAGUACUGCGGAGGGGGACC-ACGCGGGAACAUC---A---GGUGUUGUUAGUU------
P_tetraure --G-UUG-GUGGCCAUACUAA-G-C-CU-AAAGCACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGGCUUAAGGCGAG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---U---CGUGUUGACAACCC-----
P_polyceph --G-GAU-GCGGCCAUACUAA-G-G-AG-AAAGCACCUCAUCCCGUCC-GAUCUGAGA-AGUUAAGCUCCUUCAGGCGUG-GU-UAGUACUGGGGUGGGGGACC-ACCUGGGAAUCCC---A---CGUGCUGCAUUCUU-----
P_berghei  --U-GAC-UCGUUCAUACUAC-A-G-UG-GCCACACCAGAUCCCAUCA-GAACUCUGA-AGUUAAGCACUGUAAGGCUUG-GC-UAGUACUGAGGUGGGAGACC-GCUCGGGAACACU---A---AGUGAUGAGUCAUU-----
T_vorax    --G-UUG-UCGGCCAUACUAA-G-G-UG-AAAACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGCCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---C---AGUGUCGANANCCU-----
T_paravor  --G-CUG-UCGGCCAUACUAA-G-G-UG-AAAACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGCCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---C---AGUGUCGACAGCCU-----
T_thermoph --G-NUN-UCGGCCAUACUAA-G-G-UG-AAAACACCGGAUCCCAUUC-GAACUCCGA-AGUUAAGCGCCUUAAGGCUGG-GU-UAGUACUAAGGUGGGGGACC-GCUUGGGAAGUCC---C---AGUGUCGANANCCU-----
T_cruzi    -AG-AGU-ACGACCAUACUUG-A-G-UG-AAUACACCAUAUCCCGUCC-GAUUUGUGA-AGUUAAGCGCCCACAGGCUUC-GU-CAGUACGGCGAUCAGUGAUG-GCGCUGGAACCCC---G---GGUGCCGUACUCCC-----
S_carestia --A-UCU-GGGGCCAUACCAC-A-G-UG-AACACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCACUGUAGGGCCGA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAAUACU---CU--GGUGCCCCAGGU-------
T_encephal --A-UUU-UCGGCCACAGGAU-U-G-UG-AAAACGCUGCAUCCCGUCC-GAUCUGCGC-AGCCAAGCACAACACCGCUCA-GU-CAGUAGGAAGGUGGGGGACC-AUUUCCGAAUCCU---G---GGUGCCGAAGUU-------
T_mesent1  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAAUACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGCCGAGUACCGCUCA-GU-UAGUACCACGGUGGGGGACC-ACGUGGGAAUCCU---G---GGUGCUGUGGUU-------
T_mesent2  --A-UUC-ACGGCCACAGGAU-U-A-AG-AAAACACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGAUGAAUACCGCCCA-GU-CAGUACCAUGGUGGGGGACC-ACAUGGGAAUGCU---G---GGUGCUGUGGUU-------
G_primul   --A-UCU-GCGGCCAUAGAAC-C-U-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAAGCAAGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGCAGUUU------
T_papilion --A-UCC-UCGGCCAUAGAAU-G-A-CG-AAAACGACGCGUCCCGUCC-GAUCUGCGA-AUCUAAGCGUCGUAUCGCUAG-GU-UAGUACCAAGGUGGGGGACC-ACUUGGGAAUCCC---U---AGUGCCGAGGU-------
T_violea   --A-UCU-UCGGCCAUAGGAC-A-G-AG-AAAAUACCGCAUCCCGUCC-GAUCUGCGC-AGUCAAGCUCUGUACCGCUUA-GU-UAGUACCAUAGUGGGGGACC-AUAUGGGAAUCCU---G---AGUGCUGAAGUUU------
U_fusispor --A-UCC-ACGGCCAUAGGAC-U-U-CG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCGGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUUU------
A_edulis   --A-UCC-ACGGCCAUAGGAC-U-G-UG-AAAGCACCGCAUCCCGUCU-GAUCUGCGC-AGUUAAACACAGUGCCGCCUA-GU-UAGUACCAUGGUGGGGGACC-ACAUGGGAAUCCU---G---GGUGCUGUGGUU-------
B_adusta   --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
C_pallida  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
S_commune  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCUCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
C_radiatus --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
E_semperv  --A-UCC-ACGGCCAUAGGAC-C-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
G_phoenic  --A-UCU-GCGGCCAUAGAAC-C-G-UG-AAAAUACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGCACGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGCAGUU-------
G_applana  --A-UCC-ACGGCCAUAGGAC-U-C-CG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCGGAGUGCCGCCCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
P_ostreat  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCU-GAUCUGCGC-AGUUAACCAGAGUGCCGCUCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
R_cyanoxa  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUUC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUUU------
S_commune  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCUCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
S_paradoxa --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
C_tussilag --A-UCC-ACGGCCAUAGGAC-C-U-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
C_apobasid --A-UNC-ACGGCCAUAGGAC-U-U-AG-AAAACGCCGCAUCCCGUCC-GAUCUGCGA-AGCUAAGCUGGGUACCGCCUA-GU-UAGUACCAUGGUGGGGGACC-ACAUGGGAAUCCU---G---GGUGCUGUGGUU-------
G_clavaria --A-UCC-ACGGCCAUAGGAC-C-U-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
R_malvinel --A-GUU-GCUACCAUACGAA-G-A-UG-AAAGCACUGCAUCCCGUCC-GAUCUGCAA-AGUUAAGCAUCUUACGGCCCA-GU-CAGUACUAUGGUGGGGGACC-ACGUGGGAAUACU---GU--GGUGUAGCAAUU------
G_sabinae  --A-UUU-GGGGCCACACCAC-A-G-UG-AACUCACCGCAUCCCGUCU-GAUCUGCGC-AGUCAAUCACUGUAGGGCCUC-GU-AAGUAGUAGUGUGGGGGACC-AACUGCGAACACG---AU--GGUGCCCCAGGU-------
P_suaveol  --A-UCU-GGGGCCACACCAC-A-G-UG-AACUCACCGCAUCCCGUUC-GAUCUGCGC-AGUCAAACACUGUAGGGCCAA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAACACU---UU--GGUGCCCCAGGU-------
S_salmonic --A-UCU-GCGGCCAUACCGC-G-A-UG-AACCCUCCGCGUCUCGUCC-GAUCCGCGC-AGAUAAGCAUCGCAGGGGCCA-GA-GAGUAUUGACGUGGGUGACC-AGUCGAGAACACU---GU--GCUGCCGCAGGU-------
U_scabios  --A-UCU-GCGGCCAUACCGC-G-A-UG-AACCUUCCGCGUCUCGUCC-GAUCCGCGC-AGACAAGCAUCGCAGGGGCCA-GA-GAGUAUUGACGUGGGUGACC-AGUCGAGAACACU---GU--GCUGCCGCAGGU-------
S_polygoni --A-UCA-GCGGCCAUACCGC-G-A-UG-AACCUUCCGCGUCUCGUCC-GAUCCGCGC-AGACAAGCAUCGCAGGGGCCA-GA-GAGUAUUGACGUGGGUGACC-AGUCGAGAACACU---GU--GCUGCUGCAGGU-------
R_toruloid --A-UCU-GCGGCCAUACCGC-G-A-UG-AACACACCGCGUCUCGUCC-GAUCCGCGA-AGUUAAGCAUCGCAGGGGCCA-GA-GAGUAUUGCCGUGGGUGACC-AGGCGAGAACACU---GU--GCUGCCGCAGGU-------
U_fluitans --A-UUU-GGGGCCACACCGC-G-A-UG-AACCUUCCGCGUCCCGUCA-GAUACGCGC-AGACAAGCAUCGCAGGGGUAA-GA-GAGUAUCGGCGUGGGGGACC-AGCCGAGAACACU---UU--ACUGCUCCAGGU-------
S_vulgare  --A-GAU-UGCACCACACUAA-G-C-GA-CAAACACUGCGUCCCGUCC-GAUCCGCAA-AGUCAAAUCGCUUAAGGCUCC-GU-CAGUACCGAAGUGGGGGACC-AUUCGGGAAUCCG---G---AGUGUGUAAUUU------
L_pyriform --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGAGGGGGACC-ACGCGGGAAUCUU---G---GGUGCUGUGGUU-------
P_poarum   --A-UCC-ACGGCCAUAGGAC-C-U-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
G_phoenic  --A-UCU-GCGGCCAUAGAAC-C-G-UG-AAAAUACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGCACGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGCAGUU-------
T_controv  --A-UCU-GCGGCCAUAGAAC-C-U-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCAAGGUAUCGCUCA-GU-UAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---G---AGUGCUGCAGUU-------
T_anomala  --A-UCC-GCGGCCAUAGGAU-C-A-GG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAAGCCUGAUACCGCCGA-GU-UAGUACUAAGGUAGGGGACU-ACUUGGGAAUCCU---C---GGUGCUGCGGUU-------
U_hordei   --A-UCU-GCGGCCAUAGAAC-C-U-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAAGCAAGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---A---GGCGCUGCAGUUU------
G_primul   --A-UCU-GCGGCCAUAGAAC-C-U-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAAGCAAGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGCAGUUU------
U_maydis   --A-UCU-GCGGCCACAGAGA-C-U-UG-AAAAUACCGCAUCCCGUCC-GAUCUGCGC-AGUCAAGCAAGUCGUCGCCUA-GC-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGCAGUU-------
F_thuemen  --A-UCU-GCAACCAUAGGAU-C-A-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCAUGAUACCGGCUU-GG-AAGUACUGCGGUGGGGGACC-ACGCGGGAAUGCG---A---GCUGUUGCAGUU-------
G_primul   --A-UCU-GCGGCCAUAGAAC-C-U-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAAGCAAGGUAUCGCCUA-GU-CAGUACUGCGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGCAGUUU------
I_perplex  --A-UCC-ACGGCCAUAGGAC-A-U-CG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCGGUGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
M_acetabut --A-UCC-GCGGCCAUAGAAC-U-G-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCACAGUAUCGCCUA-GU-CAGUACUACGGUGGGGGACC-ACGUGGGAAUCCU---G---GGUGCCGCGGUU-------
T_oedoceph --A-UCC-GCGGCCACAGAAC-U-G-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGA-AGUUAACCACAGUGUCGCCUA-GU-UAGUACUACGGUGGGGGACC-ACGUGGGAAUCCU---A---GGUGCCGCGGUU-------
S_aggrega  --A-CAG-CCGUUCAUACCAC-A-C-GG-AGAAUACCGGAUCUCGUUC-GAACUCCGC-AGUCAAGCCGUGUCGGGCGUG-CU-CAGUACUACCAUAGGGGACU-GGGUGGGAAGCGU---G---CGUGACGGCUGUU------
T_visurgen --A-UGA-GCCCUCAUAUCAU-G-U-GG-AGUGCACCGGAUCUCAUCC-GAACUCCGU-AGUUAAGCCACAUAGAGCGCG-UC-UAGUACUGCCGUAGGGGACU-AGGUGGGAAGCAC---G---CGUGGGGCUCAUU------
C_reinha1  --A-UGGAUCGUUCAAACCUU-C-A-AG-GCCCCUCCCCAUCCCAUCA-GCACUGGGA-AGAUAAGCCUGAAUGGGCUGA-AC-UAGUAGUACGGUGGGGGACC-ACGUGCGAAUCCU---C---AGUGACGACCUGGUU----
C_reinha2  --A-UGGAUUGCUUAUACCUU-U-A-UG-AAAACUCCCCAUCCCAUUA-GCACUGGGA-AGAUAAGUAUGAAUGGGCUGA-AU-CAGUAGUACGGUGGGGGACC-ACGUGCGAACCCU---C---AGUGACGACCUGUU-----
Chlorella  --AUGCU-ACGUUCAUACCAC-C-A-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAAACGUGGUUGGGCUCG-AC-UAGUACUGGGUUGAGGGAUU-ACCUGGGAACCCC---G---AGUGACGUAGUGU------
Chlamydom  --A-UGG-UCGUUCAUACUAG-C-A-CU-ACUGCACCCUAACCCGUCA-GAUCUAGGA-AGUUAAGCGUGCUCAGGCGAG-GC-CAGUAGUACGGUGGGUGACC-ACGUGCGAAGCCC---U---CGUGACGAUCGU------
C_pyrenoi  --A-UGCUACGUUCAUACCAC-C-A-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAAACGUGGUUGGGCUCG-AC-UAGUACUGGGUUGAGGGAUU-ACCUGGGAACCCC---G---AGUGACGUAGUGU------
C_scutata  --G-UGGUUCCAUCAUACCAU-G-C-CU-ACUACGCCGCAUCCCAUCA-GAACUGCGA-AGCUAAGCGGCAUUGGGCCAG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---GGUGUGGAACC--------
K_flaccidu --G-UGGUUCGUUCAUACCUA-G-G-CU-AUUGCGCCGGAUCCCAUCA-GACCUCCGA-AGCUAAAGGCCUGUGGGCGAG-AA-UAGUACUGGAAUGGGUGACC-UUCCGGGAAGUCC---U---CGUGACGAACCCA------
N_flexilis --A-UGGUACGGUCAUACCAC-G-G-CU-AAUGCGCCCGAUCCCAUCC-GAACUCGGA-AGCCAAGCGCCGUUGGGCCGG-AA-UAGUACUGGGAUGGGUGACC-UCCUGGGAAGUCC---C---GGUGCUGUACCUAU-----
C_monilig  --G-CGCUACGGCCAUACCAG-C-G-AG-AAAGCUUCAGAUCCCAUCA-GAACUCUGC-AAAUAAGCUCGUUUGGGCGCC-AC-UAGUAACAGGGUUAGUAAUA-ACCUGUGAACCUG---G---UGUGCUGUAGUGC------
P_minor    --G-UGAUGCGUUCAUACUAC-U-U-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAAGCGGAGUUAGGCCGG-UC-UAGUACUGAGUUGAGUGAUC-ACUCGGGAAUCAC---C---GGUGACGCAUUGC------
P_subcord  --A-UGCUACGAUCAUACCAC-U-U-AG-AAAGCACCCGGUCCCAUCA-GACCCCGGA-AGUUAAGCUGAGUUGGGCUGG-AC-UAGUACUGGAUUCAGGAAUG-AUCUGGGAAUCCC---C---AGUGUCGUAGUGU------
Prasinocla --A-UGCUACGUUCAUACCAC-U-C-AG-AAAACGCCGGGUCCCAUCA-GAUCCCCGA-AGCUAAGCUGAGUUGGGCUGG-AU-UAGUACUGGAUUCAGGAAUG-AUCUGGGAAUCCC---C---AGUGACGUAGUGU------
S_obliquus --AUGCU-ACGUUCAUACCAC-C-A-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAAACGUGGUUGGGCUUG-AU-UAGUACUGGGUUGAGGGAUC-ACCUGGGAACCCC---G---AGUGACGUAGUGU------
S_quadric  --AUGCU-ACGUUCAUACCAC-C-A-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAAACGUGGUUGGGCUCG-AC-UAGUACUGGGUUGAGGGAUC-ACCUGGGAACCCC---G---AGUGACGUAGUGU------
Spirogyra  --AUGCU-ACGGUCAUACCAC-C-A-CG-AAAGCACCCGAUCCCAUCA-GAACUCGGA-AGUUAGACGUGGUUGGGCCAG-AU-UAGUACUGGGUUGAGGGAUC-ACUUGGGAACCCC---U---GGUGCUGUAGUGU------
U_pertusa  --GUGAU-ACGGUCAUACCAC-C-A-GG-AAAACAGGCGAUCCCAUCA-GAACUCGCA-ACUUAAGCCUGGUUGGGCAGG-AU-UAGUACUGGGCUGAGUGAUC-UCCUGGGAAUCCC---C---UGUGCUGUAUCGC------
S_grevill  --A-GGU-ACGAUCAUACCCG-G-G-UU-ACUACGCCGGAUCCCAUCC-GAACUCCGC-AGCUAAGUCCCCGUGGGCUCG-AA-UAGUACUGGCUAGGGUGACC-CGCCGGGAAGUCC---G---AGUGUCGUACCUU------
C_cohnii   --G-CUG-ACGGCCAUACCGU-G-U-CG-AAUGCACCGGAUCUCUUCU-GACCUCCGA-AGUUAAGCGGCACAGGGCCCG-GA-UAGUACUGGGGUGGGGGACC-GCCCGGGAAGUCCUUAG---GGUGCUGUCAGCU------
D_tenue    --A-GGA-ACGACCAUACCAC-G-A-UG-CCAAUACCGCUUCCCGUCU-GCUCAGCGA-AGUCAAGCAUCGUCGGGCCCG-GU-UAGUACUACGGUGGGGGACC-ACGUUGGAAUCCC---G---GGUGUUGUUCUU-------
H_foetidus --A-UGA-ACGGUCAUAUCAC-G-U-AA-ACUGCACCCGGUCUCGUCC-GAUCCCGGA-AGUUAAGCUGCGUCGAGUCCA-GC-CAGUAGUACGGUGCGUGAGC-ACGUGCGAAGCCU---G---GGUACUGUUCGU------
C_parameci UUC-UGU-ACGGUCAUACCUG-G-U-UG-GAAACGGCGGAUCCCGUCC-GAUCUCCGA-AGCUAAGCAACCAUGGGCGUG-UC-UAGUACUCAUGGGGGGGACC-ACUGGGGAAGCGC---A---CGUACUGUACAGC------
A_lubricum --A-GGA-ACGGCCAUACCAC-G-C-CG-AUCGCACCACAUCCCGUCC-GCUCUGUGA-AGUUAAGCGGCGUCGGGCCAG-GC-UAGUACUACGGUGGGGGACC-ACGUGGGAAGCCC---U---GGUGCUGUUCCU-------
C_flagelli --A-GGA-ACGGCCAUACCAC-G-C-CG-AUCGCACCAUAUCCCGUCC-GCUCUGUGA-AGUUAAGCGGCGUCGGGCCAG-GC-UAGUACUACGGUGGGGGACC-ACGUGGGAAGCCC---U---GGUGCUGUUCCU-------
E_bicyclis --A-GGA-ACGGCCAUACCAC-G-U-CG-AUCGCACCACAUCCCGUCC-GCUCUGUGA-AGUUAAGCGGCGUCGGGCCAG-GC-UAGUACUACGGUGGGGGACC-ACGUGGGGGACCC---U---GGUGCUGUUCCU-------
S_fulvell  --A-GGA-ACGGCCAUACCAC-G-U-CG-AUCGCACCACAUCCCGUCC-GCUCUGUGA-AGUUAAGCGGCGUCGGGCCGG-GC-UAGUACUACGGUGGGGGACC-ACGUGGGAAGCCC---C---GGUGCUGUUCCU-------
C_caldari  --A-CGU-UCGGUCAUACUAGUG-A-CA-CAUGCGCCUGAACCCAUUUCGAAUUCAGA-AGCUAAGCGCCCUCAGGCUUG-GU-UAGUACUGAGGUGAGGGAUCCACUCGGGAACCCC---A---AGUGCCGUACGUUUU----
P_palmata  --A-CAU-GCGGCCAUAGUGCAG-G-AA-CAUGCGCCGAAACCCAUCCCGAAUUUCGA-AGCUAAGCUCCGCCACGCACG-AU-UAGUACUCUGGAGGGGGACC-ACAGUGGAAUCUC---G---UGUGCCGCAUGUU------
P_umbilica --A-CGU-GCGGCCAUAGUGCAG-G-AA-CAUGCGCCGAAACCCAUCCCGAAUUUCGA-AGCUAAGCUCCGCCACGCACG-AU-UAGUACUCUGGAGGGGGACC-UCAGUGGAAUCUC---G---UGUGCCGCAUGUU------
P_yezoenzi --A-CGU-ACGGCCAUAUCCGAG-A-CA-CGCGUACCGGAACCCAUUCCGAAUUCCGA-AGUCAAGCGUCCGCGAGUUGG-GU-UAGUAAUCUGGUGAAAGAUC-ACAGGCGAACCCC---C---AAUGCUGUACGUC------
P_tenera   --A-CGU-ACGGCCAUAUCCGAG-A-CA-CGCGUACCGGAACCCAUUCCGAAUUCCGA-AGUCAAGCGUCCGCGAGUUGG-GU-UAGUAGUCUGGUGAGGGAUC-ACAGGCGAACCCC---C---AAUGCUGUACGUC------
B_ectocarp --A-CAU-ACGACCAUAGUAC-GAG-GA-CAUGCGCAGGAGCCCAUCUCGAACUCCUC-AGCUAAAUCCCGUCACGCACG-CU-UAGUACUCCGGAGGGGGACC-ACGGUGGAAUCGC---G---UGUGUCGUAUGUU------
G_amansi2  --A-CAU-UCGGCCAUACUUG-GUA-GA-UAUGCGCCUUAUCCCAUCCCGAACUGAGA-AGCUAAGUCUCCAUAGGCCCA-GG-GAGUACUGCGGUCAGGGAUG-ACGCUGGAAAUCU---G---GGUGCUGAAUGUU------
G_amansiS  --A-CAU-UCGGCCAUACUAA-GUC-GA-CAUGUACCCGAUCCCAUCCCGAACUCGGA-AGUCAAGUCGCUUCAGGCCGG-GU-UAGUACUGAGGUGAGGGAUC-ACUCGGGAAUCCC---C---GGUGCCGAAUGU-------
```

```
G_compres  --A-CAU-UCGGCCAUAUCAA-GUU-GA-CAAACGCCGUAUCCCAUCCCGAACUACGA-AGCUAAGUCACUUUGAGCUGG-GC-UAGUACUGAGAUGAGGGAUC-GGUCUGGAAUCCC---C---AGUGCUGAAUGUU------
C_crispata --A-CAU-UCGGCCAUACCAG-G-ACGA-CAAAUACCCCAUCCCAUCUCGAACUGGGC-AGUUAAGUCUCCUCGGGCGCG-CU-UAGUACUGAGGUCAGGGAUG-ACUCGGGAAUCGC---G---CGUGCUGAAUGUU------
G_complan  --A-CAU-GCGGCCAUACUAA-A-U-GAUGAUGUACCGGAUCCCAUCUCGAACUCCGA-AGUCAAGGCAUUUCAGGCAGG-GC-UAGUACUGACGAUAGAGAUG-AGUCCGGAACCCC---C---UGUGCCGCAUGU-------
H_sanguin  --G-UCU-ACGACCAUACCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUAGGGCCUG-CC-CAGUACUUGGAUGGGUGACC-GCCUGGGAACAGC---A---GGUGUUGUAGACUU-----
H_panicea  --G-CCU-ACGGCCAUACCAC-G-U-UG-AAAACACCGGUUCUCGUCU-GAUCACCGA-AGUUAAGCAACGUAGGGCCUG-CC-CAGUACUUGGAUGGGUGACC-GCCUGGGAACAGC---A---GGUGUUGUAGGCUU-----
H_oculata  --G-CCU-GCGGCCAUACCAC-G-U-UG-AAUGCACCGGUUCCCAUCU-GAACACCGA-AGUUAAGCAACGUCGGGCCAG-CU-UAGUACCUGGAUGGGUGACC-GCCUGGGAAUCGC---U---GGUGCUGCAGGCUU-----
H_oculata  --G-CCU-GCGGCCAUACCAC-G-U-UG-AAUGCACCGGUUCCCAUCU-GAACACCGA-AGUUAAGCAACGUCGGGCCAG-CU-UAGUACCUGGAUGGGUGACC-GCCUGGGAAUCGC---U---GGUGCUGCAGGCUU-----
T_adhaeren --C---U-ACGACUAUAUGAC-G-U-UG-AAUAUACCCGUUCUCGUCU-GAUUACGGA-AGUUAAGUAACGUCGAGUGGG-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAAUACC---U---CAUGCCGUAGGC-------
D_misakie  --G---U-ACGAUCAUACUUG-G-C-CG-UAAGCACCCGUUCUCAGCC-GACCACGGA-AGUUAAGCGGUUAUAGGCUGU-GU-UAGUACUUGGCUGGGUGACC-GCCUGGGAACUCA---C---AGUGUCGUACUUU------
B_plicatil --G-CCU-AGGACCAUACUAC-G-U-UG-AAUGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGAGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGUUCUAGGCCUU----
A_suum     --G-CUU-ACGACCAUACCAC-G-U-UG-AAAGCACGACAUCCCGUCU-GAUCUGUCA-AGUUAAGCAACGUUGGGCCUA-GU-UAGUACUUGGAUCGGAGACG-GCUUGGGAAUCCC---A---GGUGCUGUAAGCU------
C_elegans  --G-CUU-ACGACCAUAUCAC-G-U-UG-AAUGCACGCCAUCCCGUCC-GAUCUGGCA-AGUUAAGCAACGUUGAGUCCA-GU-UAGUACUUGGAUCGGAGACG-GCCUGGGAAUCCU---G---GAUGUUGUAAGCU------
A_suum     --G-CUU-ACGACCAUACCAC-G-U-UG-AAAGCACGACAUCCCGUCU-GAUCUGUCA-AGUUAAGCAACGUUGGGCCUA-GU-UAGUACUUGGAUCGGAGACG-GCUUGGGAAUCCC---A---GGUGCUGUAAGCU------
R_tokai    --G-CUU-ACGACCAUAUCAC-G-U-UG-AANNCACGNCAUCCCGUCC-GAUCUGNCA-AGUUAAGCAACGUUGNGNNCN-GU-UAGUACUUGGAUCGGAGACG-GCNUGGGAAUCCN---G---NNUGUNGUAAGCU------
L_anatina  --G-UCU-ACGACCAUACCAC-G-U-UG-AAANCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCCAG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---U---GGUGCCGUAGACA------
U_unicinct --G-UCU-ACGGCCAUACCAC-G-U-UG-AAAACACCAGUUCUCGUCC-GAUCACUGA-AGUUAAGCAACGUCGGGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGCUGUAGACUU-----
B_neritina --G-UUU-ACGGCCAUAUCAC-G-U-UG-AAAACGCCAGUUCUCGUCC-GAUCACUGA-AGCUAAGCAACGUCGAGCCUG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---A---GGUGCUGUAGACUU-----
P_gouldii  --G-CCU-ACGACCAUACCAC-G-U-UG-AAAACACCAGUUCUCGUCC-GAUCACUGA-AGUUAAGCAACGUCGGGCCAG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---U---GGUGUUGUAGGCUU-----
R_pachypt  --G-UCU-ACGGCCAUAUCAC-G-U-UG-AAAACACCGGUUCNCGUCC-GAUCACCGA-AGUUAAGCAACNUCGAGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGCUGUAGNCUN-----
E_gracile1 --G-UCU-ACAACCAUACCAC-G-U-UG-AACACACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCAACGUAGGGCUCG-GU-UAGUACUUAGAUGGGUGACC-GCUUGGGAAUACC---G---AGUGUUGUAGNCUU-----
E_gracile2 --G-GCA-ACGACCAUACCAU-G-C-UG-AAAAUACCAGUUCCCGUCC-GAUCACUGA-AGUCAAGCAGCAUCGGGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAACACC---G---GGUGUCGUUGUCA------
L_genicul  --G-UCU-ACGACCAUAUCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGAGCGUG-GU-UAGUACGUGGAUGGGUGACC-GCCUCGGAAUACC---A---CGUGUUGUAGGCUU-----
D_japonic1 --G-UCG-ACGCUCAUACUAG-G-U-UG-GGUCCACCCGAUCUCGUUC-GAUCUCGGC-AGUUAAACAACCUUAGGCCUC-GU-UAGUACUUGAAUGCGUGAGC-GUCGGGAAUACG---A---GGUGGUGUCGACUU-----
D_japonic2 --G-UCG-ACGCUCAUACUAG-G-U-UG-GGUACACCCGAUCUCGUUC-GAUCUCGGAAUGUUAAGCAGCCUGAGGCCUC-GU-UAGUACUUGAAUGCGUGAGC-GUCGGGAAUACG---A---GGUGGUGUCGACUU-----
P_reticula --G-AUA-GCGUCCAUACCAC-A-C-UG-AAAACACCGGUUCUCGUCC-GAUCACCGC-AGUUAAGCAGUGUCGGGCCCA-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACU---G---GGUGUCGCUACCUU-----
A_equina   --G-UCU-ACGGCCAUACCAC-C-G-GG-AAAAUACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCCCGGUAGGGCCAG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---U---GGUGCUGUAGACUU-----
A_aurita1  --G-CCU-ACGACCAUACCAC-C-A-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAUGGUCAGGCCGG-GU-CAGUACCUGGAGUGGUGACC-GCCUGGGAACACC---C---GGUGUUGUAGGCCU-----
A_aurita2  --G-CCU-ACGACNAUACCAC-C-A-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGNAUGGUAGGGCCGG-GU-CAGUACCUGGAUGGGUGACC-GCCUGGGAACACC---C---GGUGUUGUAGGCNU-----
A_japonica --G-UCU-ACGGCCAUACCAC-C-G-GG-AAAAAACCGGUUCUCGUCC-GAUCACCGA-AGUCAAGCCCGGUAGGGCCAG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---U---GGUGCUGUAGACUU-----
C_quinque. --G-CCU-ACGACCAUACCAC-C-A-UG-AGUAUACCGGUUCUCGUCC-GAUCACCGG-AGUCAAGCAUGGUCGGGCCGG-GU-CAGUACCUGGAUGGGUGACC-GCCUGGGAACACC---U---GGUGUUGUAGGCCU-----
N_dofleini --G-UCU-ACGACCAUACCAC-A-A-UG-AACACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAUUGUCGGGCCAG-GA-UAGUACCUGGAUGGGGGACC-GCCUGGGAACGCC---U---GGUGUCGUAGACUU-----
S_saltatr  --G-UCU-ACGGCCAUACCAC-G-A-UG-AAUACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAUUGUCGGGCCAG-GA-UAGUACUUGGAUGGGGGACC-GCCUGGGAACUCC---U---GGUGCCGUAGACUUU----
B_belcheri --G-CCU-ACGGCCAUACCAC-G-U-UG-AGUACACCCGAUCUCGUUC-GAUCUCGGA-AGUUAAGCAACGUCGAGUCCG-GU-CAGUACUUGGAUGGGAGACC-GCCUGGGAAUACC---G---GGAGUUGUAGGCAU-----
E_albidus  --G-UCU-ACGGCCAUACCAC-G-U-UG-AAAGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGCUGUAGACUU-----
P_brevicir --G-CCU-ACGGCCAUACUAC-G-U-UG-AAAACACCGGUUCUCGUCU-GAUCACCGA-AGUUAAGCAACGUCGGGCCUG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---A---GGUGCUGUAGGUUU-----
S_japonica --G-CCU-ACGGCCAUACCAU-G-C-UG-AAUACACCCGUUCUCGUCC-GAUCACGGA-AGUUAAGCAGCAUCGGGCACG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---UGUGCAGUAGGCUU-----
C_magnifi  --G-UCU-ACGGCCAUAUCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGAGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGCUGUAGACUU-----
S_velum    --G-UCU-ACGGCCAUACCAC-G-U-UG-AAAGCACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGGGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGCAGUAGACUU-----
M_edulis   --G-UCU-ACGACCAUAUCAC-G-U-UG-AAAACACCGGUUCUCGUCC-GAUCACCGA-AGUUAAGCAACGUCGAGCCCG-GU-UAGUACUUGGAUGGGUGACC-GCCUGGGAAUACC---G---GGUGUUGUAGACA------
M_polymor  --G-GAU-GCGGUCAUACCAG-G-G-CU-ACUACACCAGAUCCCAUCA-GAACUCUGC-AGUUAAGCGCCCUUGGGCCGG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGCUGCAUCCA------
P_trichoma --G-GAU-GCGGUCAUACCAA-G-G-CU-ACUACACCAGAUCCCAUCA-GAACUCUGA-AGUUAAGCGCCUUUGGGCCGG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGCUGCAUCCA------
L_heterop  --G-GAU-GCGGUCAUACCAN-G-G-CU-ACUACACCAGAUCCCAUCA-GAACUCUGC-AGUUAAGCGCCUUUGGGCCGG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGCUGCAUCCA------
A_punctat  ----GGU-GCGGUCAUACCAG-G-G-CU-ACUACACCGGAUCCCAUCA-GAACUCCGU-AGUUAAGCGCCCUUGGGCCGG-AU-CAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGCUGCACCCU------
P_nudum    --GUGGU-GCGGUCAUACCAC-C-G-UU-AAUGCACCGGAUCCCGUCG-GAACUCCGU-AGUUAAGCGCGCUUGGGCCGG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGCCGCGCCCAC-----
D_acumina  --GUGGU-GCGGUCAUACCAG-C-G-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGCGCUUGGGCCAG-AA-CAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---GGUGCUGCACCCUU-----
P_aquilin  --GUGGU-GCGGUCAUACCAG-C-G-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCCAG-AA-CAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---GGUGCUGCACCCUU-----
E_arvense  --GUGGU-GCGGUCAUACCAG-C-G-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCCAG-AA-CAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---GGUGCCGCACCCC------
L_clavatum --GUGGU-GCGGUCAUACCAG-C-A-CU-ACUAGACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCCUG-AA-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---G---GGUGCCGCACCCUC-----
G_biloba   --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCACGCUUGGGCUGG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---C---AGUGUUGCACCCUC-----
G_gnemon   --G-GGU-GCGAUAAUACCAC-C-G-CU-AACGUAUCGGAUCCGAUCA-GAACUCCGU-AAUUAAGCGCGCUUGGGCUAG-AG-UAGUACUGGGAUGGGUGACC-UCUCGGGAAGUCC---U---AGUGUUGCACCCAC-----
C_revoluta --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGUUGG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---AAUAUUGCACCCUU-----
E_kokanica --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCUAG-AG-UAGUACUGGGAUGGGUGACC-UCCUGGGAAGUCC---U---AGUGUUGCACCCUC-----
A_thaliana --G-GAU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCAUCCCU-----
S_oleracea --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCCU-----
M_glyptos  --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCCGG-AG-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUAUUGCACCCUU-----
T_baccata  --G-AGU-GCGAUCAUACCAG-C-G-UU-UGUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCCAG-AG-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---C---GGUGUCGCACCCUUU----
J_media    --G-GGU-GCGAUCAUACCAG-C-G-UU-AGUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCCGG-AG-UAGUACUGGGGGAGUUGACC-UCCCGGGAAGUCC---C---GGUGUUGCACCCUU-----
P_contorta --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCUAG-AG-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---AGUGUUGCACCCUCC----
P_silvest  --G-GGU-GCGAUCAUACCAG-C-G-UU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGCGCUUGGGCUAG-AG-UAGUACUGGGAUGGGUGACC-UCCCGGGAAGUCC---U---AGUGUUGCACCCC------
B_napus    --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCCU-----
G_hirsutum ----GGU-UCAAUCAUACCGA-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-CAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGAACCCU------
H_annuus   --G-GUU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-CCCUGGGAAGUCC---U---CGUGUUGCAACCCC-----
L_sativa   --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-CCCUGGGAAGUCC---U---UGUGUUGCACCCC------
L_minor    --G-GGU-GCGAUCAUACCAG-C-A-CU-AGAGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-CAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCU------
L_luteus   --A-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCUC------
N_tabacum  --G-GAU-GCGAUCAUACCAG-C-A-CU-AACGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-CCCUGGGAAGUCC---U---CGUGUUGCAUCCCU-----
Magnolia   --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCC------
M_sativa   --A-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---UGUGUUGCACCUC------
O_sativa1  --G-GAU-GCGAUCAUACCAG-C-A-CU-AAAGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---UGUGUUGCAUCCC------
O_sativa2  --G-GAU-GCGAUCAUACGAG-G-G-CG-AAAGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---UGUGUUGCAUCCC------
P_vulgaris --A-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACUUU------
V_faba     --A-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---UGUGUUGCACCUCU-----
N_tabacum  --G-GAU-GCGAUCAUACCAG-C-A-CU-AACGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-CCCUGGGAAGUCC---U---CGUGUUGCAUCCCU-----
S_alba     --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCC-------
R_acetosa  --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCCU-----
S_oleracea --G-GGU-GCGAUCAUACCAG-C-A-CU-AAUGCACCGGAUCCCAUCA-GAACUCCGC-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCACCCCU-----
S_cereale  --G-GAU-GCGAUCAUACCAG-C-A-CU-AAAGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCAUUCCC-----
Z_mays     --G-GAU-GCGAUCAUACCAG-C-A-CU-AAAGCACCGGAUCCCAUCA-GAACUCCGA-AGUUAAGCGUGCUUGGGCGAG-AG-UAGUACUAGGAUGGGUGACC-UCCUGGGAAGUCC---U---CGUGUUGCAUUCC------
B_simplex  --A-GCU-ACGGCCAUACCAC-C-C-AG-AAAGCUCCGCAUCCCGUCC-GAUCUGCGA-AGAUAAUCUGGUGCUGGCGCG-GU-CAGUACUGUCGGUGGGAGACC-AGAUGGGAAUACC---G---CGUGCUGUAGUU-------
P_irregula --A-GCU-ACGGCCAUACAUA-G-A-UG-AAAAUACCGGAUCCCGUCC-GAUCUCCGA-AGUCAAGCAUCUAAUGGCGAC-GU-CAGUACUGUGAUGGGGGACC-GCACGGGAAUACG---U---CGUGCUGUAGUU-------
C_albicans --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUCCCCGUUC-GAUCAACCGUAGUUAAGCUGCUAAGAGCAAU-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUA---U---UGUGCUGCAAUCU------
C_azyma    --G-CCU-ACGGUCAUAUCAA-U-G-CG-AAUAUACGAUUUCCCGUCU-GAUCAAUCAUAGUCAAACGCGUUAGAGCCAC-AUUCAGUAUUACGGUGGGAGACC-ACGUGAGAACGGG---U---GGUACUGAAGGU-------
C_cylindra --A-GCU-ACGGCCAUAUCUA-G-C-AG-UCAGCACCGUUUCCCGUCC-GAUCAACAGUAGUGAAUCUGCUAAGAGCCUG-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUC---A---GGUGCUGUAGUUU------
C_diversa  --G-GUU-GCGGCCAUAUCUA-G-C-GG-AAAGCACCGUUUCCCGUCC-GAUCAACUGCAGUUAAGCCGCUGAGAGCCUG-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUC---A---GGUGCUGCAAUC-------
C_magnolia --A-GCU-GCGGCCAAACCCA-G-G-UG-AAUACAAGACUUCCCGUCC-GAUCAGUCCUAUUUAAACACCUGAGGGCCUU-ACCAAGUAUUGUAGUGGGAGACC-AUACGAGAACAGA---A---GGUGCUGCAGUU-------
C_melibios --G-GUU-GCGGCCAUACCAA-G-U-GG-AAAAUACUGUUUCCCGUCC-GAUCAACGUUAGUCAAGCCACUUAGGGCCAC-AGUGAGUAAUACAGUGGGAGACC-AUGUGUGAAAUUG---U---GGUGCUGCAAUCU------
C_parapsil --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUUCCCGUCC-GAUCAACAGUAGUUAAGCUGCUAAGAGCGAG-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUC---U---CGUGCUGCAAUCU------
C_rugopell --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUUCCCGUCC-GAUCAACUGCAGUUAAGCUGCUAAGAGCCUG-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUC---A---GGUGCUGCAAUC-------
C_rugosa   --G-GUU-GCGGCCAUACCUA-G-U-AG-AAAGCACCAUUUCCCGUCC-GAUCAAUUGCAGUUAAGCUACUAAGGGUCUG-AUUGAGUAGUGUAGUGGGAGACC-AUACGCGAAAUUC---A---GAUGCUGCAAUCU------
C_zeylanoi --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUUCCCGUCC-GAUCAACUGUAGUUAAGCUGCUAAGAGCGAG-ACCGAGUAGUGUAGUGGGAGACC-AUACGCGAAACUC---U---CGUGCUGCAAUCU------
Z_helleni  --A-GCU-ACGGCCAUAUCCA-G-A-UG-AAAAUAAGACUUCCCGUCC-GAUCAGUCCUAUUCAAGCAUCUGAGAGCCUU-ACUAAGUACUAUAGUGGGUGACC-AUAUGGGAACUGA---A---GGUGCUGUAGUU-------
R_crocorum --A-UCU-GGGGCCAUACCAC-A-G-CC-GAAUCACCGCAUCCCGUUC-GAUCUGCGC-AGUUAAAUGCUGUAGGGCCGA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAAUACU---CU--GGUGCCCCAGGU-------
R_globular --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACUGCAUCCCGUCC-GAUCUGCAA-AGUUAACCAGAGUGCCGCCCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---C---GGUGCUGUGGUU-------
R_hiemalis --A-UCU-GGGGCCAUACCAC-A-G-UU-AAAUCACCGCAUCCCGUCU-GAUCUGCGA-AGUUAAGCACUGUAGGGCCGA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAAUACU---CU--GGUGCCCCAGGU-------
S_penicill --A-UCC-ACGGCCAUAGGAC-A-C-UG-AAAACAUCGCAUCCCGUCC-GAUCUGCGC-AAUUAAGCAGUGUGCCGCUUA-GU-UAGUACCAUGGUGGGGGACC-ACAUGGGAAUCCU---A---GGUGCUGUGGUU-------
K_microst  --A-CNU-ACGGCCACAGUCA-G-C-UG-AAAACUGGGCAUCCCGUCC-GCUCUGCCAUACAUAAGCAGUUGAACGGCAG-AU-UAGUACUACGGUGGGUGACC-ACGUGGGAAUCCC---U---GCUGCUGUAUGUU------
C_laurent  --A-UCC-ACGGCCACAGGAC-U-C-AG-AAAACACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGCUGAGUACCGCCUA-GU-UAGUACCAACGAGGGGGACC-ACGUGGGAAUCCU---A---GGUGCCGUGGUU-------
C_albidus  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUACCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGUGGGAAUCCU---A---GGUGCUGUGGUU-------
C_humicol  --A-UCC-ACGGCCAUAGGAC-U-C-AG-AAAAUACCGCAUCCCGUCC-GAUCUGCGA-AGUCUAGCUGAGUACCGCCUA-GU-UAGUACCAUGGUGGGGGACC-ACAUGGGAAUCCU---G---GGUGUCGUGGUU-------
F_capsuli  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
P_poarum   --A-UCC-ACGGCCAUAGGAC-C-U-UG-AAAACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUU-------
F_florifoW --A-UCC-ACGGCCAUAGGAC-C-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGGGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---A---GGUGCUGUGGUUU------
P_rhodozy  --A-UCC-ACGGCCAUAGGAU-A-U-CG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCGAUAUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
B_magnus   --AAGCA-UCGGCCAUACAUA-C-C-AG-AAAAUACCGGAUCCCGUCC-GAUCUCCGA-AGUCAAACUGGUAAUGGCCUA-GU-CAGUACUACGGUGGGGGACC-ACGUGGGAAUACU---A---GGUGCUGAUGUUU------
B_trispora --AAGCU-ACGGCCAUAUAAU-A-C-CG-AAAGCACCGGAUCCCGUCC-GAUCUCCGC-AGUUAAACGGUAUAUAGAUCA-GU-CAGUACUAUGGUGGGGGACC-ACAUGGGAAUACU---G---GUUGCUGUAGUUU------
C_elegans  --AAGUU-ACGGCCACAUCAU-C-G-UG-AAAAGACCAAAUCCCGUCC-GAUCUUUGC-AGUCAAACACGAUCGAGCUUC-GU-CAGUACUAUGGUGAGAGAUC-ACAUGGGAAUACG---A---AGUGCUGUAAUUU------
M_formosen --AAAUU-ACGGCCAUACACA-G-G-AG-AAAGCUCCGCAUCCCGUUC-GAUCUGCGC-AGAUAACCUCCUGAUGGCAGG-GU-CAGUACUAUGGUGGGGGACC-ACAUGGGAAUACC---C---UGUGCUGUAAUUU------
P_blakesle --AAUCU-ACGGCCAUACAGA-U-A-GU-AACACACCGGAUCCCGUCU-GAUCUCCGC-AGUUAAGUCUCUCCUGGUAGC-GU-CAGUACUAUGGUGGGGGACC-ACAUGGGAAUACG---C---UAUGUCGUAGGUU------
S_culiseta --AUCCU-ACGGCCAUACACA-C-C-AG-AAAGCACCAAAUCCCGUCC-GAUCUUUGA-AGUUAAGCUGGUGAUGGCGUC-GU-CAGUACUGUGGUGGGGGACC-ACACGGGAAUACG---A---UGUGCCGUAAGUUU-----
C_stellat  --GUCUU-ACGGCCAUUCACA-C-C-AG-AAAGCACCAAAUCCCGUCC-GAUCUUUGA-AGUUAAGCUGGUGAUGGCGUC-GU-CAGUACUGUGGUGGGGGACC-ACACGGGAAUACG---G---CGUGCCGUAGGUCA-----
G_hibernus --GUCUU-ACGGCCAUUCACA-C-C-AG-AAAGCACCAAAUCCCGUCC-GAUCUUUGA-AGUUAAGCUGGUGAUGGCGUC-GU-CAGUACUGUGGUGGGGGACC-ACACGGGAAUACG---G---CGUGCCGUAAGUCA-----
C_mojaven  --A-GCU-GGGGCCAUAUGAA-C-C-UA-UAAGUACCGCAUCCCGUCCCGAUCUGCGA-AGUCAAGCAGGUUACAGCUCG-GU-UAGUACUAUGAUGGGAGACC-ACAUGGGAAUACC---G---GGUGUCCCAGCUU------
D_acumino  --A-GUU-ACGACCAUACCAA-C-U-UG-AAAGCACCACACCCCGUCA-GAUCUGUGA-AGUUAAGCAAGUUAGAGCUUC-GU-UAGUAGUUUGGUGGGGGACC-ACAAGCGAAUACG---A---AGUGCCGUAGCUU------
L_macrosp  --A-GUA-ACGGCCAUAUCAG-C-U-UG-AAAGCACCGCAACCCGUCA-GAUCUGCGA-AGUUAAGCAAGCUAGAGCUAG-GU-UAGUACUAUGGUGGGGGACC-ACAUGGGAAUACU---U---AGUGUCGUUAUUU------
A_parasiti --G-GUU-ACGGCCAUACCUA-G-C-AG-AAAGCACCCCAUCUCGUCC-GAUCUGGGA-AGUUAAACUGCUAUGGGCGUG-GU-UAGUACUGCGGUGGGGGACC-ACGCGGGAAUACC---A---CGUGCUGUAAUCU------
P_inundat  --A-UCU-GCGGCCAUACCUC-C-A-UG-AAAAUACUGCUUCCCGUCC-GAUCAGCAA-AGUCAAGCAUGGAAGGGAUUC-GU-UAGUACUAUCGUGGGAGACC-AGAUGGGAAUCCG---G---GUUGCUGCAAGUU------
P_carinii  --A-GUU-ACGGCCAUACCUC-A-G-AG-AAUAUACCGUAUCCCGUUC-GAUCUGCGA-AGUUAAGCUCUGAAGGGCGUC-GU-CAGUACUAUAGUGGGUGACC-AUAUGGGAAUACG---A---CGUGCUGUAGCUUU-----
A_chrysoge --A-CAU-ACGACCAUACCUA-G-U-GG-AAAAUACGGGAUCCCGUCC-GCUCUCCCAUAGUCAAGCCGCUAAGGGGCUG-AU-UAGUAGUUGGGUCGGUGACG-ACCAGCGAAUACC---G---GCUGUUGUAUGU-------
A_persici1 --A-CAU-ACGACCAUAGCUC-U-C-AG-AGAACUCGGGAUCCCGUCU-GCUCUCCUGUAGAUAAGCUGACAAGCGGCCG-AU-UAGUAGUUGGGUCGGUGACG-ACCAGCGAAUACC---G---GCUGUUGUAUGU-------
A_persici2 --A-CAU-ACGACCAUAGACG-C-U-AG-AAAAUACGGGAUCCCGUCC-GCUCUCCCAUAGUCAAGCUGGCGAUCGGCGG-AU-UAGUAGUUGGGUCGGUGACG-ACCAGCGAAUACC---U---GCUGUUGUAUGU-------
A_persici3 --A-CAU-ACGACCAUAGGCG-C-U-AG-AAAAUACGGGAUCCCGUCC-GCUCUCCCAUAGUCAAGCUGGCGACCGGCUG-AU-UAGUAGUUGGGUCGGUGACG-ACCAGCGAAUACC---G---GCUGUUGUAUGU-------
A_flavus   --A-CAU-ACGACCAUAGGGU-G-U-GG-AGAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGGAG-GU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCC---U---CCUGUUGUAUGU-------
A_nidula1  --A-CAU-ACGACCAUAGGGU-G-U-GG-AGAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGUAG-GU-UAGUAGUAUGGUGGGUGACC-ACAUGCGAAUCCC---U---ACUGUUGUAUGU-------
A_nidula2  --A-CAU-ACGACCAUAGGGU-G-U-GG-AAAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGCUG-GU-UAGUAGUAUGGUGGGUGACC-ACAUGCGAAUCCC---A---GCUGUUGUAUGU-------
A_nidula3  --A-CAU-ACGACCAUAGGGU-G-U-GG-AGAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGCUG-GU-UAGUAGUAUGGUGGGUGACC-ACAUGCGAAUCCC---A---GCUGUUGUAUGU-------
A_niger    --A-CAU-ACGACCACAGGGU-G-U-GG-AAAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGGAG-GU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCC---U---UCUGUUGUAUGU-------
H_jadinii  --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUCUCCGUCC-GAUCAACUGUAGUUAAGCUGCUAAGAGCCUG-AUCGAGUAGUGUAGUGGGUGACC-AUACGCGAAACUC---A---GGUGCUGCAAUCU------
K_microst  --A-CNU-ACGGCCACAGUCA-G-C-UG-AAAACUGGGCAUCCCGUCC-GCUCUGCCAUACAUAAGCAGUUGAACGGCAG-AU-UAGUACUACGGUGGGUGACC-ACGUGGGAAUCCC---U---GCUGCUGUAUGUU------
S_cerevis  --G-GUU-GCGGCCAUAUCUA-C-C-AG-AAAGCACCGUUUCCCGUCC-GAUCAACUGNAGUUAAGCUGGUAAGAGCCUG-ACCGAGUAGUGUAGUGGGUGACC-AUACGCGAAACUC---A---GGUGCUGCAAUCU------
L_lipofer  --A-GUU-GCGGCCAUACCAA-G-A-AG-AAANCACAGUUCCCCGUCC-GAUCAACACUAGUUAAGCUUCUUUGGGCCUA-CU-UAGUACUGCGGUGGGAGACC-ACGCUGGAAUCGU---A---GGUGCUGCAAUU-------
M_jugland  --A-UCC-ACGGCCAUAGGAC-A-C-AG-AAAACAUCGCAUCCCGUCC-GAUCUGCGC-AAUCAAGCUGUGUACCGCCCA-GU-CAGUACCGGAGUGGGGGACC-AUCCGGGAAUCCU---GCCAGGUGCUGUGGUU-------
M_fructico --A-CAU-ACGACCAUAGACU-G-A-NG-AGAAUUGGGCAUCCCGUCC-GCUCUGCCAUACACAAGCUUCAGAUCGGUGG-AU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCC---U---NCUGUUGUAUGU-------
N_fulvesc  --G-UCU-GCGGCCAUAUCCA-C-U-GG-AAAGCACGAUUUCCCGUCC-GAUCAAUCAUAGUUAAGCCAGUGAGAGCCUUCAU-AAGUACUACGGUGGGAGACC-ACGUGGGAACAUA---A---GGUGCUGCAGUCU------
P_patulum  --A-CAU-ACGACCAUAGGGU-G-U-GG-AAAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGUGA-GU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCU---C---ACUGUUGUAUGU-------
P_membran  --G-GUU-GCGGCCAUAUCUA-G-C-AG-AAAGCACCGUUUCCCGUCC-GAUCAACUGNAGUUAAGCUGCUAAGAGCCUG-ACCGAGUAGUGUAGAGGGCGACC-AUACGCGAAACUC---A---GGUGCUGCAAUC-------
P_graminea --A-GCU-ACGGCCAUACAAU-G-U-UG-AAAACACCGGAUCCCGUCC-GAUCUCCGC-AGUUAAAGCAACAUCUGGACC-AGUCAGUACUAUGGUGGGGGACC-ACAUGGGAAUACU---G---GUUGCUGUAGUU-------
S_pombe    --G-UCU-ACGGCCAUACCUA-G-G-CG-AAAACACCAGUUCCCGUCC-GAUCACUGC-AGUUAAGCGUCUGAGGGCCUC-GU-UAGUACUAUGGUUGGAGACA-ACAUGGGAAUCCG---G---GGUGCUGUAGGCU------
S_carestia --A-UCU-GGGGCCAUACCAC-A-G-UG-AACACACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCACUGUAGGGCCGA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAAUACU---CU--GGUGCCCCAGGU-------
T_encephal --A-UUU-UCGGCCACAGGAU-U-G-UG-AAAACGCUGCAUCCCGUCC-GAUCUGCGC-AGCCAAGCACAACACCGCUCA-GU-CAGUAGGAAGGUGGGGGACC-AUUUCCGAAUCCU---G---GGUGCCGAAGUU-------
T_abundans --A-GCU-ACGGUCAUAGAAC-A-C-UG-AAAACACGGCUUCCCGUCC-GCUCAGCCCUAGUUAAGCANNGUAUCGCGAG-GU-UAGUACUAUGGUGGGUGACC-ACAUGGGAAUCCC---U---CGUACUGUAGUU-------
T_deforma  --A-UCU-GCGGCCAUACCUC-C-A-UG-AAAAUACUGCUUCCCGUCC-GAUCAGCAA-AGUCAAGCAUGGAAGGGAUUG-GGUUAGUACUAUCGUGGGAGACC-AGAUGGGAAUCCG---G---GUUGCUGCAGUU-------
Y_lipolyt  --G-GUU-NCGGCCAUAUCCU-G-G-UG-AAAAUACGGCUUCCCGUCC-GAUCAGCCAUAGUCAAGCACCAGAGAGCCUA-GU-UAGUAUUGUAGUGGGAGACC-AUACGAGAAUCCU---G---GGUGCUGUAAUCU------
N_crassa   --A-CAU-ACGACCAUACCCA-C-U-GG-AAAACUCGGGAUCCCGUCC-GCUCUCCCAUAGAUAAGCCAGUGAGGGCCAG-AC-UAGUAGUUGGGUCGGUGACG-ACCAGCGAAUCCC---U---GGUGUUGUAUGUU------
T_lanugin2 --A-CAU-GCGACCAUAGGGU-G-U-GG-AAAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCACACGCCGGCUG-GU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCC---A---GCUGUUGCAUGU-------
T_lanugin3 --A-CAU-GCGACCAUAGGGU-G-C-GG-AAAACAGGGCUUCCCGUCC-GCUCAGCCGUACUUAAGCCGCACGCCGGCUG-GU-UAGUAGUUGGGUGGGUGACC-ACCAGCGAAUCCC---A---GCUGUUGCAUGU-------
A_pulcher1 --A-AUU-ACGGCCAUAGCAA-C-C-CU-AAAACAAUCUUUCCCGUUC-GAUCAAGAA-AUUUAAGGGGGUUAGCGAGCA-CU-CAGUACUAUGGUCGGGGACG-ACAUGGGAAUCGU---GA--CUUGCCGUAAUU-------
A_pulcher2 --A-AUU-ACGGCCACAGCAA-C-C-CC-AAAACACUCUUUCCCGUUC-GAUCAAGAA-AGUUAAGGGGGUUAGCGAGCA-CU-CAGUACCAUGGUCGGGGACG-ACAUGGGAAUCGU---GA--CUUGCCGUAAUU-------
A_solani   --A-GGU-GCGACCAUACCGU-G-U-UG-AAAAUUCUGCAUCCCGUCC-GAUCUGCAA-AGACAAGCAACACAGGGCCCA-GU-CAGUAGUGCGGUGGGUGACC-ACGUGCGAAUACU---GU--GGUGUUGCACUUU------
B_alba     --A-UCC-ACGGCCAUAGGAC-U-C-AG-AAAGUACCGCAUCCCGUCC-GAUCUGCGA-AGUCAAGCUGAGUACCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGUGGGAAUCCU---A---GGUGCUGUGGUU-------
U_fusispor --A-UCC-ACGGCCAUAGGAC-U-U-CG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCGGAGUGCCGCCUA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUUU------
D_stillat  --A-UCC-ACGGCCAUAGGAC-A-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGN-AGUUAACCAGUGUGCCGCCCG-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
E_albesc   --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACCGCAUCCCGUCC-GAUCUGCGC-AGUUAACCAGAGUGCCGCCCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
E_glandul  --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACUGCAUCCCGUCC-GAUCUGCAA-AGUUAACCAGAGUACCGCUCA-GU-UAGUACCAUGGUGGGGGACC-AUGCGGGAAUCCU---G---GGUGCUGUGGUU-------
H_auricula --A-UCC-ACGGCCAUAGGAC-U-C-UG-AAAGCACUGCAUCCCGUCC-GAUCUGCAA-AGUUAACCAGAGUACCGCCCA-GU-UAGUACCACGGUGGGGGACC-ACGCGGGAAUCCU---G---GGUGCUGUGGUU-------
M_violace  --A-UCU-GCGGCCAUACCGC-G-C-UG-AACGUUCCGCGUCUCGUCC-GAUCCGCGC-AGACAAGCAUCGCAGGGGCCA-GA-GAGUAUUGACGUGGGUGACC-AGUCGAGAACACU---GU--GCUGCCGCAGGU-------
P_ferrugin --A-UCU-GGGGCCAUACCAC-A-G-UG-GAAUUACCGCAUCCCGUCU-GAUCUGCGC-AGUCAANCACUGUAGGGCCGA-GU-CAGUAGUGCGGUGGGGGACC-ACGCGCGAAUACU---CU--GGUGCCCCAGGU-------
P_faginea  --A-UGU-GCGACCAUACCAA-G-C-UG-AAAAUACUGCAUCCCGUCU-GAUCUGCAC-AGUCAAGCAGCUUAGGGCCCA-GU-CAGUAGUGCGGUGGGGGACC-AUGCGCGAACAUU---GU--GGUGUUGCACUU-------
P_peniopho --A-UCU-GCGGCCAUACCGU-G-A-UG-AACAUUCCGCGUCCCGUCC-GAUCCGCGC-AGACAAGCAUCACAGGGGCCA-GA-GAGUAUUGACGUGGGUGACC-AGUCGAGAACACU---GU--GCUGCCGCAGGU-------
```

Figure 51: The complete alignment of 316 known Eukaryotic 5S rRNA sequences.

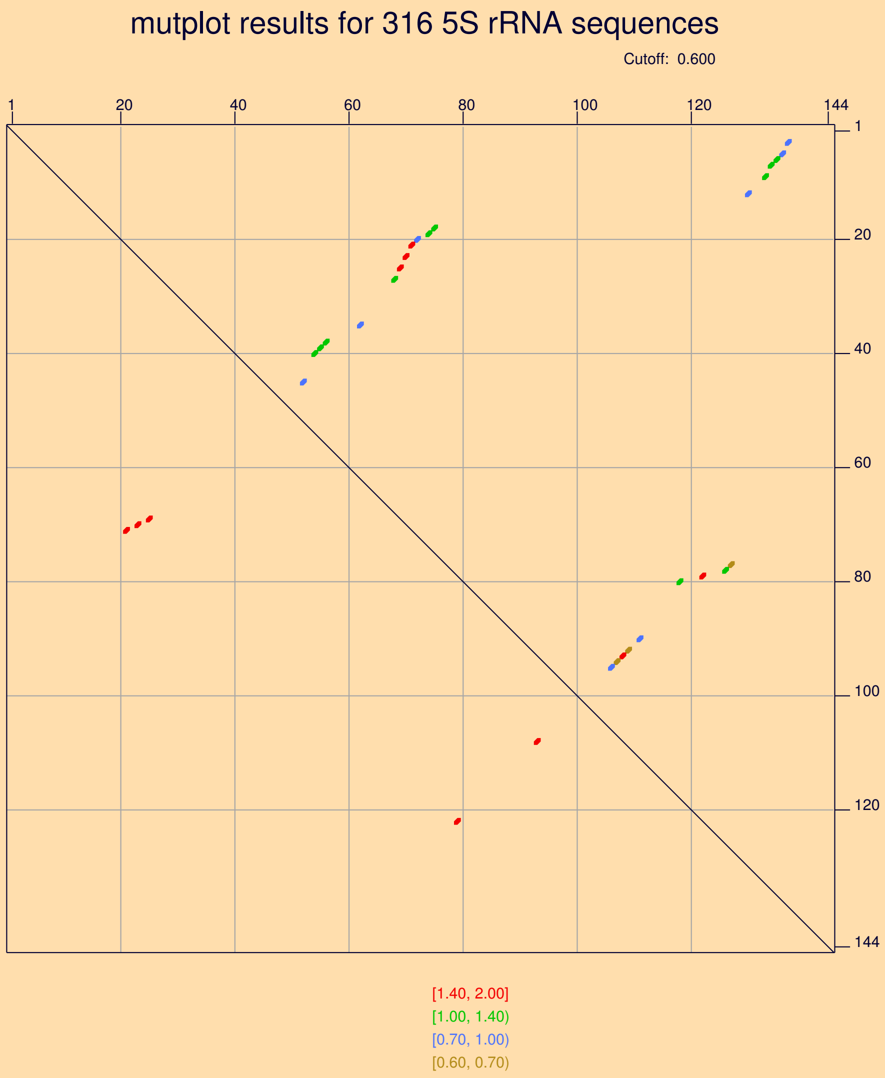


Figure 52: Mutual information plot for the 316 5S rRNA sequences from Figure 51
.

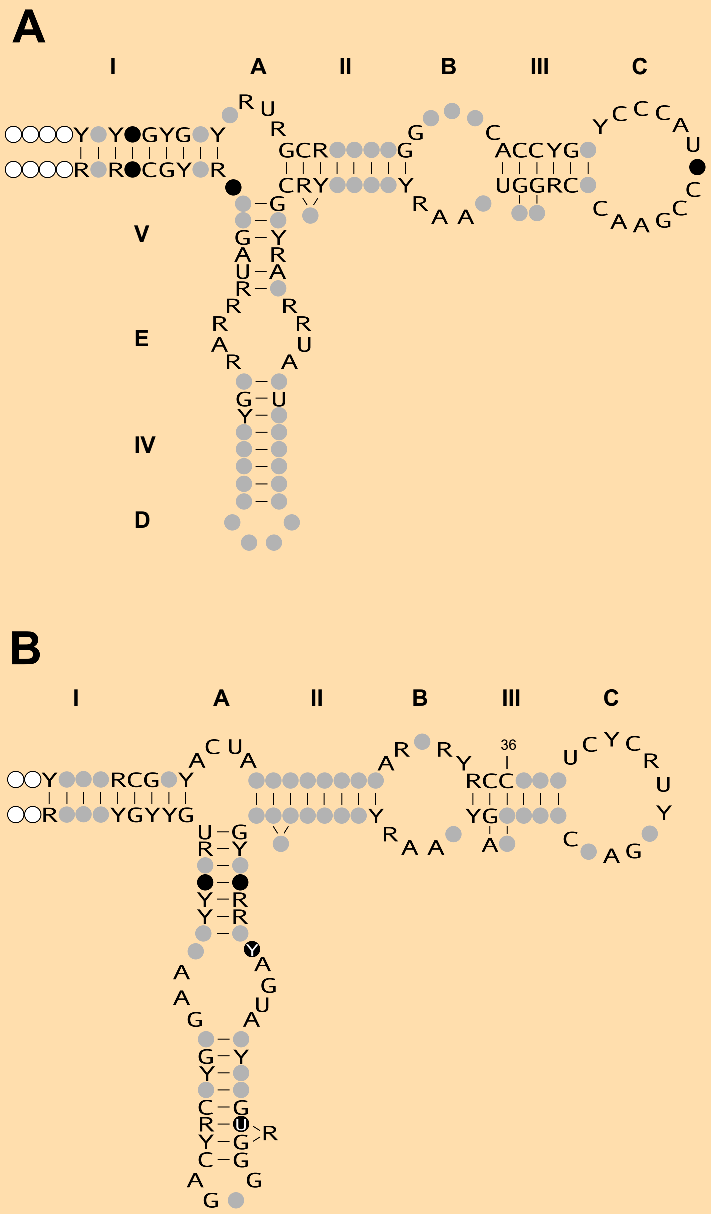


Figure 53: Consensus folding models for 5S rRNA.

References

[1] D. Sankoff, J.B. Kruskal, S. Mainville, and R.J. Cedergren. *Fast algorithms to determine RNA secondary structures containing multiple loops.*, chapter 3, pages 93–120. Time warps, string edits, and macromolecules: the theory and practice of sequence comparison, Sankoff D., Kruskal J.B., Eds. Addison-Wesley, Reading, MA, 1983.

[2] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.

[3] M. Zuker. RNA folding prediction: The continued need for interaction between biologists and mathematicians. *Lectures on Mathematics in the Life Sciences*, 17:86–123, 1986.

[4] H. Jacobson and W.H. Stockmayer. Intramolecular reaction in polycondensations. i. the theory of linear systems. *J. Chem. Phys.*, 18:1600–1606, 1950.

[5] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, 9:133–148, 1981.

[6] M. Zuker. *The use of dynamic programming algorithms in RNA secondary structure prediction.*, chapter 7, pages 159–184. Mathematical methods for DNA sequences, Waterman M.S., Ed. CRC Press, Inc., Boca Raton, Florida, 1989.

[7] M. Vingron and P. Argos. Determination of reliable regions in protein sequence alignments. *Protein Eng.*, 3:565–569, 1990.

[8] M. Zuker. Suboptimal sequence alignment in molecular biology. alignment with error analysis. *J. Mol. Biol.*, 221:403–420, 1991.

[9] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–19, 1990.

[10] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhöffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125(2):167–188, 1994.

[11] S. Winker, R. Overbeek, C.R. Woese, G.J. Olsen, and N. Pfluger. Structure detection through automated covariance search. *Comput. Appl. Biosci.*, 6:365–371, 1990.

[12] L. Chan, M. Zuker, and A.B. Jacobson. A computer method for finding common base paired helices in aligned sequences: application to the analysis of random sequences. *Nucleic Acids Res.*, 19:353–358, 1991.

[13] K. Han and H.-J. Kim. Prediction of common folding structures of homologous RNA's.;. *Nucleic Acids Res.*, 21:1251–1257, 1993.