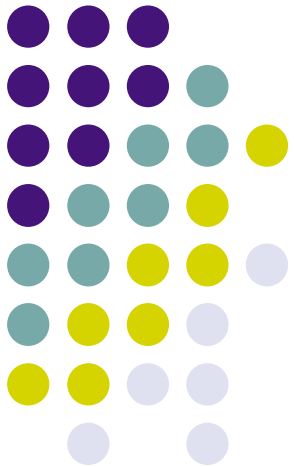
# RNA: secondary structure prediction

**(Institute of Mathematical Sciences, National University of Singapore, 24 July 2007)**

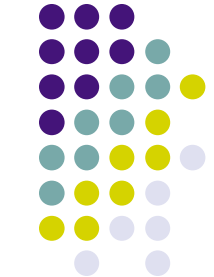


P. Clote

Biology and Computer Science

Boston College

# Overview of Methods



- Covariation using mutual information
  - requires reliable multiple sequence/structure alignment of many RNAs
- Stochastic context free grammars (generalization of HMMs)
- Energy minimization methods

FOCUS of talk on energy minimization, but first briefly mention other two methods.

# Stochastic context free grammars

Context free grammar for

L= {well-balanced parenthesis expressions}

consists of rules

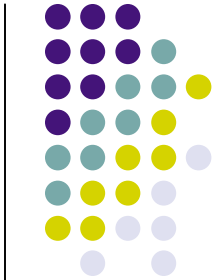
$$S \rightarrow \lambda \mid (S) \mid SS$$

Key notions:

$\lambda$ is empty word

Terminal symbols: ‘(‘, ‘)’

Variables: S

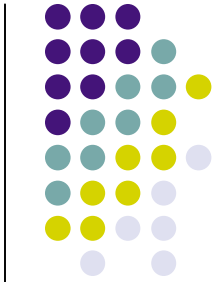


Derivation

$$S \rightarrow (S) \rightarrow (SS) \rightarrow ((S)S) \rightarrow$$
$$(( )S) \rightarrow (( )(S)) \rightarrow (( )( ))$$

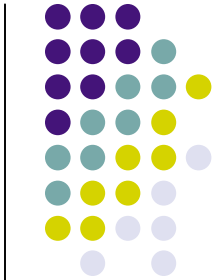
Leftmost derivation (apply rule to leftmost occurring variable in each step)

One-one association between leftmost derivations and parse trees



Context free grammar for RNA consists of following rules (here, balanced parentheses correspond to Watson-Crick or GU base pairs).

$S \rightarrow \lambda \mid AS \mid CS \mid GS \mid US \mid ASU \mid USA \mid CSG \mid GSC \mid GSU \mid USG \mid SS$



- Branching given by rule S→ SS
- Watson-Crick base pairings given
- by rules

$$S \rightarrow ASU \mid GSC \mid \text{etc.}$$

- Unpaired bases (e.g. in hairpin loop) given by rules

$$S \rightarrow AS \mid GS \mid \text{etc.}$$

- Stochastic grammar has probabilities associated with rule applications

Washington University in St.Louis

# tRNAscan-SE Search Server

Search for tRNA genes in genomic sequence

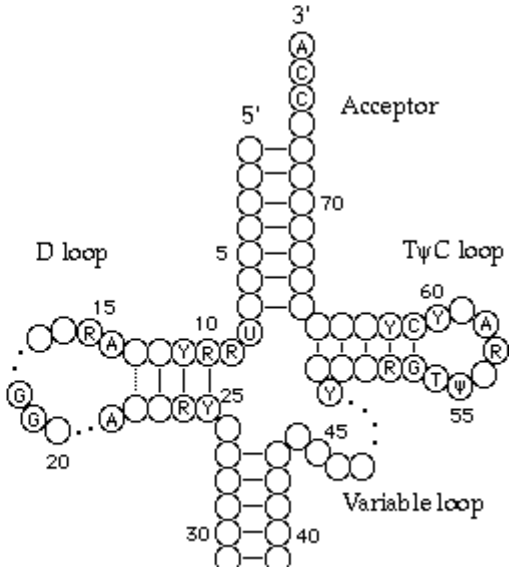
Dept. of Genetics | WashU | Medical School | Sequencing Center
Eddy lab | HMMER | PFAM | tRNAscan-SE | Software | Publications

## tRNAscan-SE 1.21

User Manual for command-line UNIX version of program

If you would like to run tRNAscan-SE locally, you can get the UNIX source code (compressed tar file).

*NEW*: Analyzing tRNAs in a published genome? See our own tRNAscan-SE analyses of completed genomes in the Genomic tRNA Database




http://www.genetics.wustl.edu/eddy/tRNAscan-SE/

## Nussinov-Jacobson dynamic programming

Given RNA sequence $s_1, \ldots, s_n$, define matrix $M = (m_{i,j})$ where $m_{i,j}$ is maximum number of base pairs occurring in an optimal secondary structure for the subsequence $s_i, \ldots, s_j$.

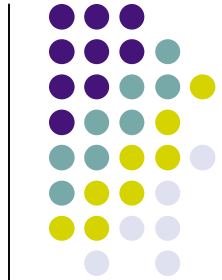
Absence of pseudoknots allows the recurrence relation

$$m_{i,j} = \max \left\{ m_{i,j-1}, \max_{i \leq k < j} \{(1 + m_{i,k-1} + m_{k+1,j-1}) \cdot bp(i,j)\} \right\}$$

First term of $m_{i,j-1}$ if $j$ is not paired with any $k$ in interval $[i, j]$, and second term if there is pairing of $j$ to $k$.

NOTE. Above only defined if $i < j$, so use lower half of matrix to store the index $k$ such that $j$ is base-paired with $k$, else 0.

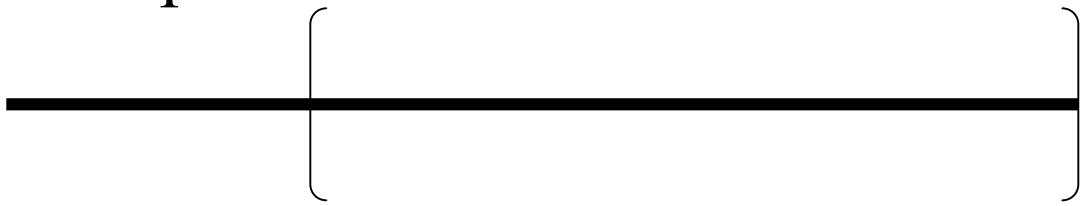
- $M_{i,j}$ equals the maximum number of base pairs in $s_i, , s_j$
- $M_{j,i}$ equals index $k$ such that if $k, j$ base pair then maximum number of base pairs is realized in $s_i, , s_j$
- $M_{j,i}$ used for linear time traceback

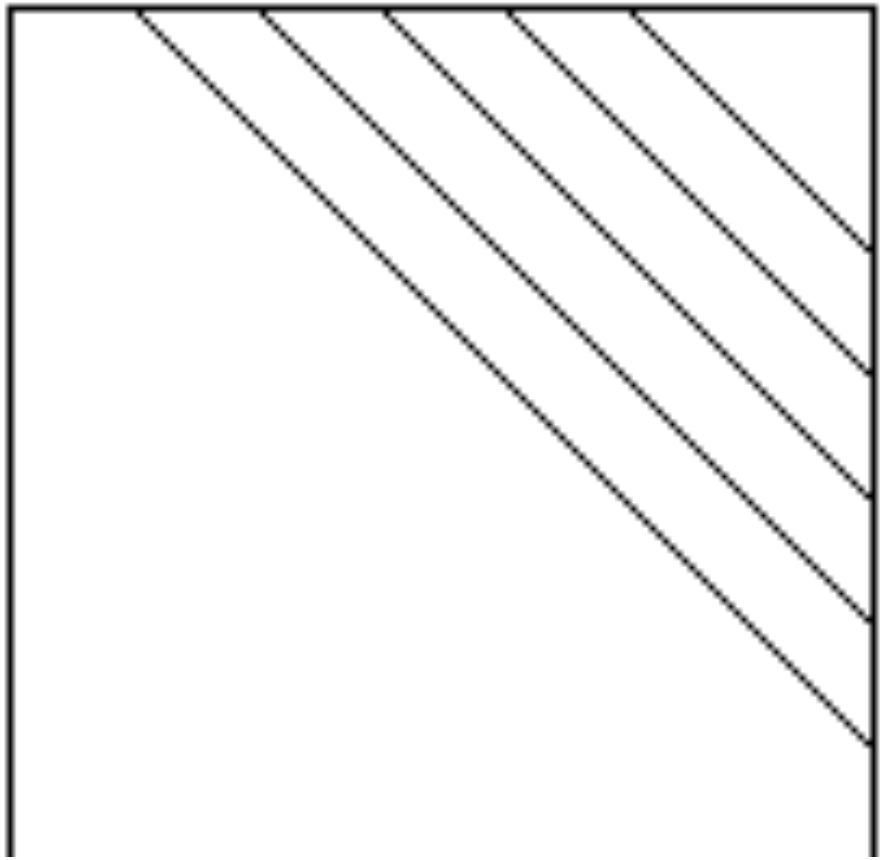


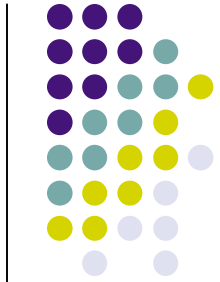
**Case 1**: j not base paired

**Case 2**: j base paired with 1

**Case 3**: j base paired with intermediate k





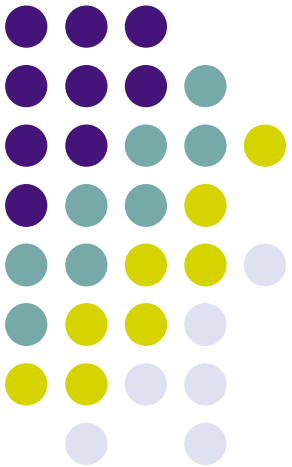


Dynamic programming order of filling in matrix:
left to right along principal diagonal, then successively fill off-diagonals proceeding outwards.

```
for d = μ to n − 1
  for i = 0 to n − 1
    j=i+d; min =0; index=-1
    if ( E(i, j − 1) < min )   // Case 1
      min = E(i, j − 1)
      index = -1; // j is unpaired
    if (a(i, j, S) + E(i + 1, j − 1) < min)   // Case 2
      min = a(i, j) + E(i + 1, j − 1); index=i
    for k = i to j-μ   // Case 3
      if a(k, j) + E(i, k − 1) + E(k + 1, j − 1) < min
        min = val; index=k
    E(i, j) = min
    E(j, i) = index
```

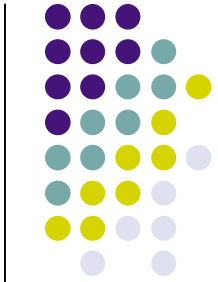
# backtrack(i,j)

Linear time backtracking algorithm after the energy matrix is filled



```
k = E(j, i)
if(k ≠ −1)
  paren[k] = '('
  paren[j] = ')'
  if( μ ≤ (j − 1) − (k + 1) )
     backtrack(k+1,j-1,paren)
  if (μ ≤ k − 1 − i   )
     backtrack(i,k-1,paren)
else { // Here k = −1
  if( μ ≤ j − 1 − i )
     backtrack(i,j-1,paren)
  else
     return 0
```

# Zuker-Stiegler modification of Nussinov-Jacobson



1) An isolated base pair contributes no energy by itself; instead consider experimentally measured stacking energies

5'-XA-3'

3'-YB-5'

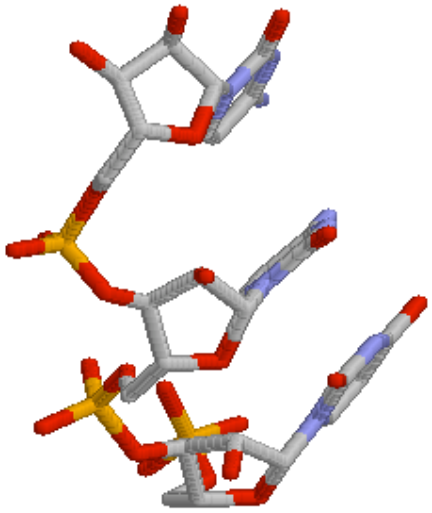
2) Consider experimentally measured loop energies for hairpin,bulge and internal loops (not multiloops).

3) Affine approximation for multiloop energy

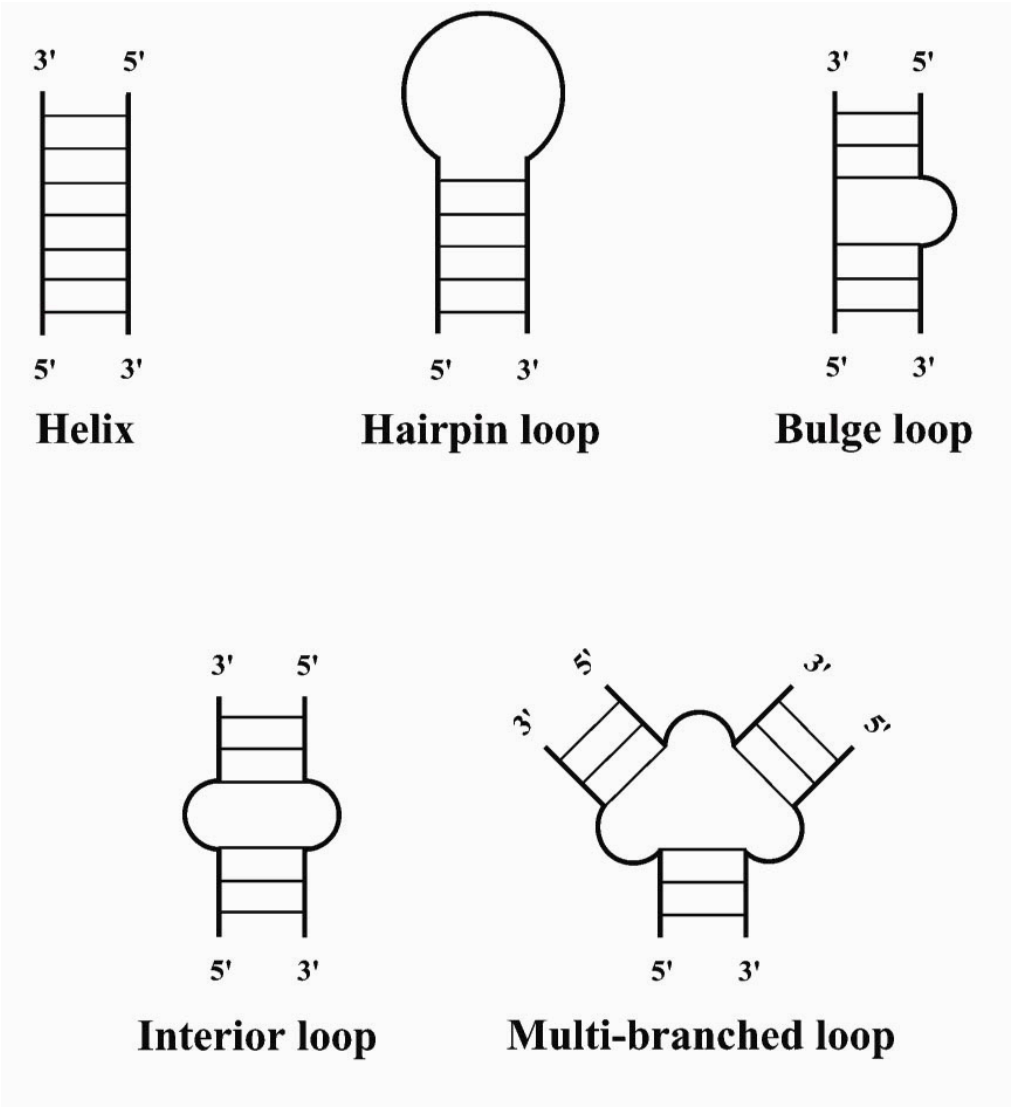
$a+bI+cU$

for I internal base pairs and U unpaired bases in multiloop having external pair (i,j), i.e. within i+1,…,j-1.

# UCC, 1ASZ:R:601-603

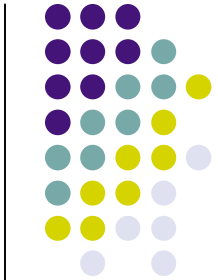


**Note the base stacking.**



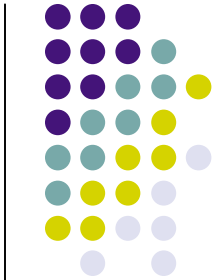

Ding and Lawrence, *A statistical sampling algorithm for {RNA} secondary structure prediction*, **Nucleic Acids Res.**, 31(24):7280--7301 (2003)

# mfold base stacking energies at 37 degrees Celsius



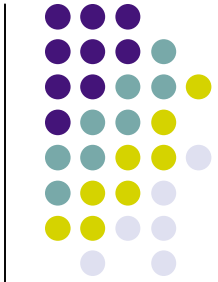
| | Y | | | | Y | | | | Y | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | C | G | U | A | C | G | U | A | C | G | U |
| 5' | | | | | 5' | | | | 5' | | | |
| G | | | | | G | | | | G | | | |
| C | | | | | G | | | | U | | | |
| 3' | | | | | 3' | | | | 3' | | | |
| . | . | . | -2.3 | | . | . | . | . | . | . | . | -0.5 |
| . | . | -3.4 | . | | . | . | . | . | . | . | -1.9 | . |
| . | -2.9 | . | -1.3 | | . | . | . | . | . | -1.5 | . | -0.5 |
| -2.1 | . | -1.9 | . | | . | . | . | . | -0.7 | . | -0.5 | . |

# Loop energies from Vienna RNA package v1.6



```
PUBLIC int hairpin37[31] = {
  INF, INF, INF, 570, 560, 560, 540, 590, 560, 640, 650,
       660, 670, 678, 686, 694, 701, 707, 713, 719, 725,
       730, 735, 740, 744, 749, 753, 757, 761, 765, 769};

PUBLIC int bulge37[31] = {
  INF, 380, 280, 320, 360, 400, 440, 459, 470, 480, 490,
       500, 510, 519, 527, 534, 541, 548, 554, 560, 565,
  571, 576, 580, 585, 589, 594, 598, 602, 605, 609};

PUBLIC int internal_loop37[31] = {
  INF, INF, 410, 510, 170, 180, 200, 220, 230, 240, 250,
       260, 270, 278, 286, 294, 301, 307, 313, 319, 325,
       330, 335, 340, 345, 349, 353, 357, 361, 365, 369};
```

# stacked base pair



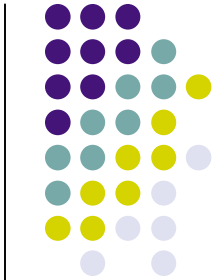
| CG | GC | GU | UG | AU | UA |
|---|---|---|---|---|---|
| -240, | -330, | -210, | -140, | -210, | -210 |
| -330, | -340, | -250, | -150, | -220, | -240 |
| -210, | -250, | 130, | -50, | -140, | -130 |
| -140, | -150, | -50, | 30, | -60, | -100 |
| -210, | -220, | -140, | -60, | -110, | -90 |
| -210, | -240, | -130, | -100, | -90, | -130 |

# 5’ dangle

**5' dangling ends (unpaired base stacks on first paired base)**

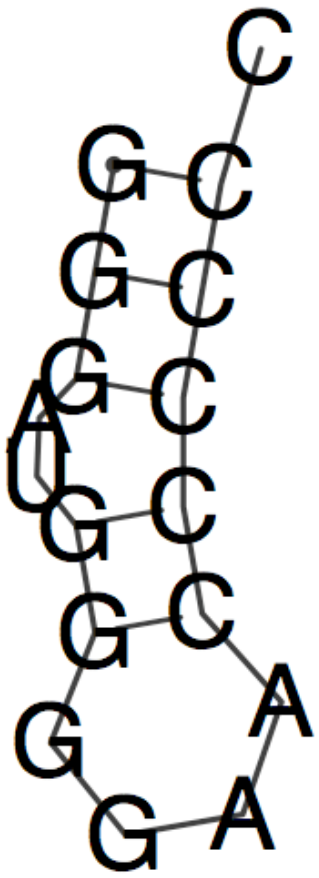
```
  A      C      G      U
-50,   -30,   -20,   -10 CG   (stacks on C)
-20,   -30,    -0,    -0 GC   (stacks on G)
-30,   -30,   -40,   -20   GU
-30,   -10,   -20,   -20   UG
-30,   -30,   -40,   -20   AU
-30,   -10,   -20,   -20   UA
```

# 3’ dangle



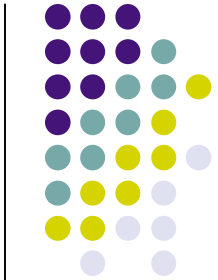
**3' dangling ends (unpaired base stacks on second paired base**

```
   A      C      G      U
-110,  -40, -130,  -60 CG  (stacks on G)
-170,  -80, -170, -120 GC
- 70,  -10,  -70,  -10 GU
- 80,  -50,  -80,  -60 UG
- 70,  -10,  -70,  -10 AU
- 80,  -50,  -80,  -60 UA
```

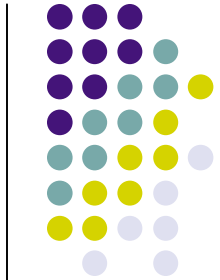



**tetraloop: +5.6**
**GNRA tetraloop bonus: -1.5**
**terminal mismatch: -2.4**
**GC stacked bp: -3.3**
**2-bulge: +2.8**
**2xGC stacked bp: -6.8**
**3’ dangle: -0.8**

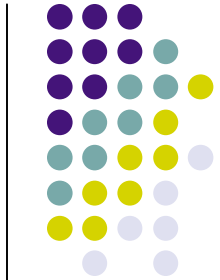
**MFE of GGGAUGGGGAACCCCCC is -6.20 kcal/mol**



```
GGGAAC
(....)
energy =  +1.70
tetraloop:  +5.60
GNRA bonus: -1.50
terminal mismatch: -2.40

GGGGAACC
((....))
energy = -1.60
previous:   +1.70
stacked bp: -3.30
```



```
UGGGGAACC
.((....))
energy = -1.60
previous:          -1.60
5' dangle on GC: 0

AUGGGGAACC
..((....))
energy = -1.60
previous:          -1.60
unpaired nondangle: 0

GAUGGGGAACCC
(..((....)))
energy =  +1.20
previous: -1.60
2-bulge:  +2.80
```

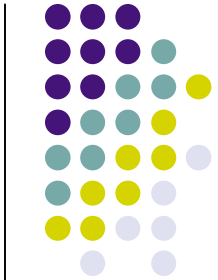


```
GGAUGGGGAACCCC
((..((....))))
energy =  -2.10
previous: +1.20
stacked bp: -3.30

GGGAUGGGGAACCCCC
(((..((....)))))
energy =  -5.40
previous: -2.10
stacked bp: -3.30


GGGAUGGGGAACCCCCC
(((..((....))))).
energy =  -6.20
previous: -5.40
3' dangle of C on GC: -0.80
```

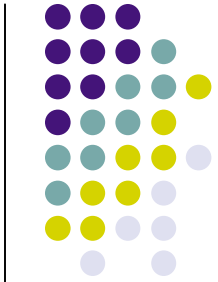
```
AGUGUACGUACGUACAGUGUACGUACGUACG
.((..((....)))).((..((....)))).
12345678901234567890123456789 01
```

- Position 1 is a $5'$ dangle
- Position 16 is indicates *coaxial stacking*
- Position 31 is a $3'$ dangle

# Free energy measurements



- Free energy determination using absorption measurements in spectrophotometer (optical melting experiments)
- Two state equilibrium: all or none, folded or unfolded
- Pioneering work by Tinoco, followed by Turner.
- Energy model often called Turner energy model.



Consider $n$ nucleotide strands forming a product complex $nS \rightarrow P$ with equilibrium constant

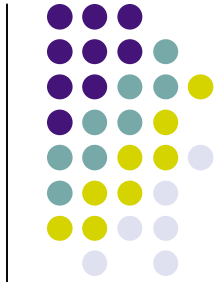
$$K = \frac{[P]}{[S]^n}$$

Mass balance equation yields

$$[S] + n[P] = c_T$$

where $c_T$ is the total strand concentration in moles per liter.

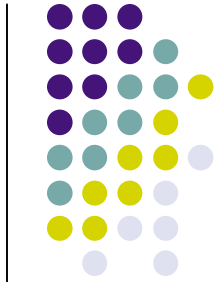
**See "Thermodynamics of formation of secondary structure in nucleic acids", Tinoco, Schmitz, in Thermodynamics in Biology, ed E. di Cera (2000)**



Let $f = n[P]/c_T$ be fraction of strands in $n$-stranded product species, so $1 - f = [S]/c_T$. Then

$$\begin{aligned} K &= \frac{[P]}{[S]^n} \\ &= \frac{c_T f/n}{(c_T(1-f))^n} \\ &= \frac{f}{nc_T^{n-1}(1-f)^n} \end{aligned}$$

This equation holds when $n = 1$, as when an unfolded strand forms a hairpin loop, when $n = 2$, as when two self-complementary strands form helix, etc.



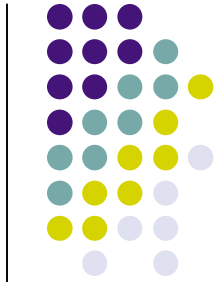
Beer-Lambert equation

$$A = \epsilon c I$$

where $A$ is absorbance, $\epsilon$ is extinction coefficient, $c$ is strand concentration, and $I$ is path length in spectrophotometer, taken as 1 cm. Now

$$\begin{aligned} A(T) &= \epsilon_S[S] + \epsilon_P[P] \\ &= (1-f)A_S(T) + f A_P(T) \end{aligned}$$

where $\epsilon_S$ is extinction coefficient of single-stranded species and $\epsilon_P$ is extinction coefficient of $n$-stranded product species.



$A_P(T) = \epsilon_P(T) \cdot c_T/n$, hence

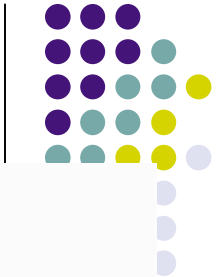
$$f = \frac{A(T) - A_S(T)}{A_P(T) - A_S(T)}.$$

# Nearest neighbor model (stacked base pairs)

Let $\epsilon(T)$ denote teh measured extinction coefficient at temperature $T$, and $\epsilon_{ss}(T)$, $\epsilon_{dd}(T)$ denote the coefficient for the single-stranded resp. double-stranded state. Let $\alpha$ denote the fraction of molecules i double-stranded state:

$$\alpha = \frac{\epsilon_{ss}(T) - \epsilon(T)}{\epsilon_{ss}(T) - \epsilon_{ds}(T)}$$

**See "Thermodynamic parameters for an expanded nearest-neighbor model for Formation of RNA duplexes with Watson-Crick base pairs", Xia et al. Biochemistry 37:14719-14735 (1998)**



Letting $K$ denote the equilibrium constant,

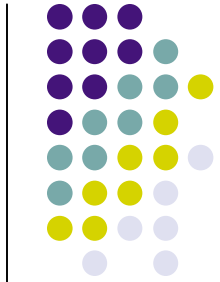
$$\begin{aligned} K &= exp(-\Delta G(T)/RT) = exp(-\frac{\Delta H}{RT} + \frac{\Delta S}{R}) \\ &= \frac{\alpha}{2(C_T/a)(1-\alpha)^2} \end{aligned}$$

Here $a = 1$ for self complementary molecules, and 4 for non self-complementary molecules, while $C_T$ is the concentration of single strands. Determine $\Delta H$, $\Delta S$ by least squares fit from melting curves.

Free energy: $\Delta G = -RT \log(K)$

Enthalpy: $\Delta H = RT^2 \frac{\partial}{\partial T} \log K$ (van't Hoff equation)

Entropy: $\Delta S$ obtained from $\Delta G = \Delta H - T\Delta S$.

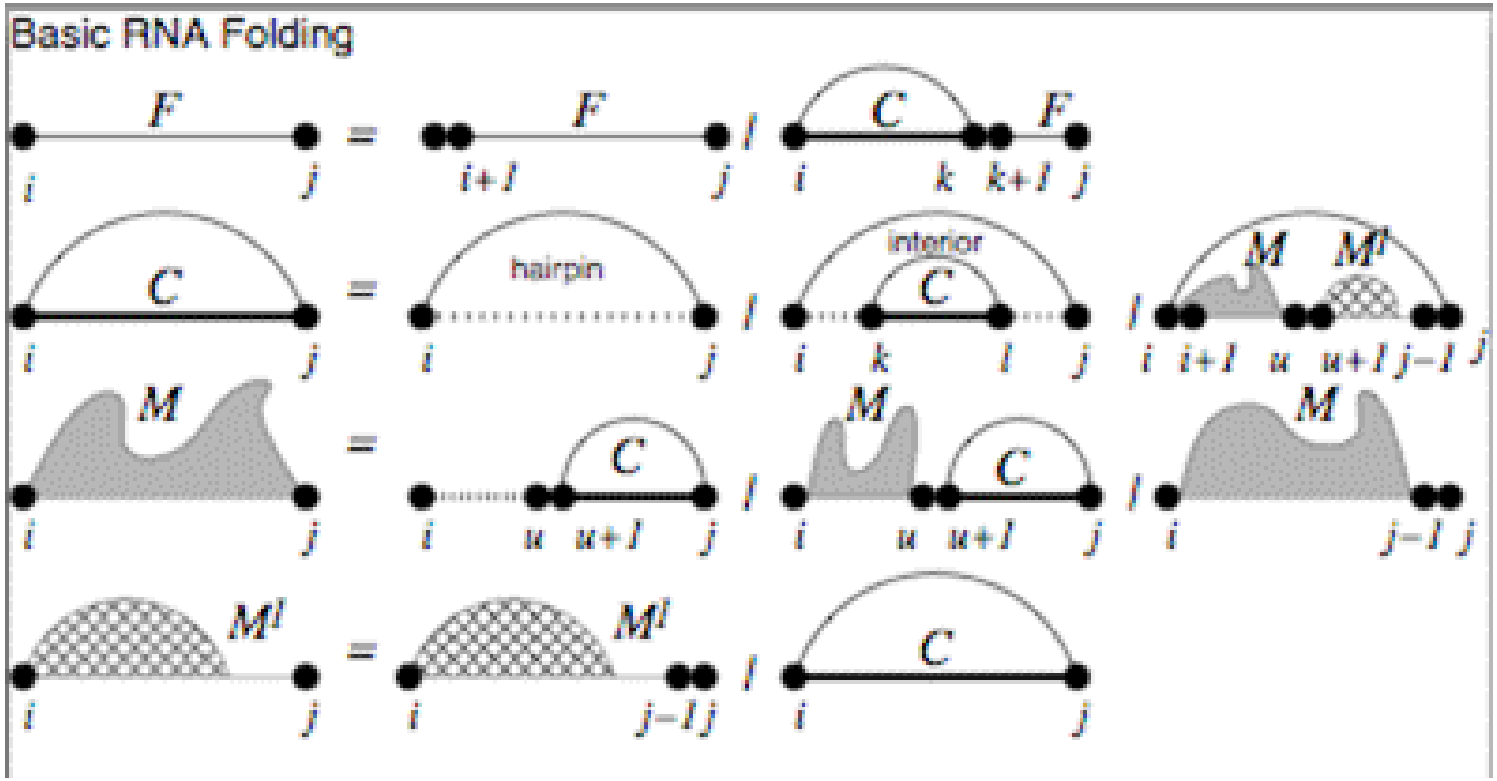


Letting $T_M$ denote the melting temperature,

$$T_M^{-1} = \frac{R}{\Delta H} \ln(C_T/a) + \frac{\Delta S}{\Delta H}$$

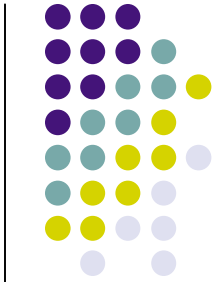
All measurements taken at 1 M NaCl, so should designate by $H^\circ$, etc.

# “Feynman diagram of recursions for Zuker’s algorithm”




Variations on RNA Folding and Alignment,
Athanasius F. Bompfuenewer, J Math Biol (2007) in press.

# Types of loop in pseudocode



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

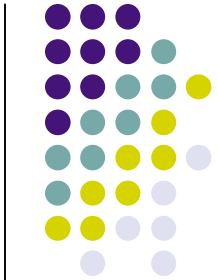
$C_{ij}$ free energy of the optimal substructure on the subsequence $x[i, j]$ subject to the constraint that $i$ and $j$ form a basepair.

$M_{ij}$ free energy of the optimal substructure on the subsequence $x[i, j]$ subject to the constraint that $x[i, j]$ is part of a multiloop and has at least one component, i.e., a sub-sequence that is enclosed by a basepair.

$M_{ij}^1$ free energy of the optimal substructure on the subsequence $x[i, j]$ subject to the constraint that $x[i, j]$ is part of a multiloop and has exactly one component, which has the closing pair $(i, h)$ for some $h$ satisfying $i < h \leq j$.

Variations on RNA Folding and Alignment,
Athanasius F. Bompfuenewer, J Math Biol (2007) in press.

# Pseudocode of Vienna Package implementation of Zuker's algorithm



$$
\begin{aligned}
F_{ij} &= \min\left\{F_{i+1,j},\ \min_{i<k\le j} C_{ik} + F_{k+1,j}\right\} \\
C_{ij} &= \min\left\{\mathcal{H}(i,j),\ \min_{i<k<l<j} C_{kl} + \mathcal{I}(i,j;k,l),\ \min_{i<u<j} M_{i+1,u} + M^1_{u+1,j-1} + a\right\} \\
M_{ij} &= \min\left\{\min_{i<u<j}(u-i+1)c + C_{u+1,j} + b,\ \min_{i<u<j} M_{i,u} + C_{u+1,j} + b,\ M_{i,j-1} + c\right\} \\
M^1_{ij} &= \min\left\{M^1_{i,j-1} + c,\ C_{ij} + b\right\}
\end{aligned}
$$

Variations on RNA Folding and Alignment,
Athanasius F. Bompfuenewer, J Math Biol (2007) in press.

# Vienna Package implementation of McCaskill's algorithm

$$
\begin{aligned}
Z_{ij} &= Z_{i+1,j} + \textstyle\sum_{i<k\leq j} Z^B_{ik} Z_{k+1,j} \\
Z^B_{ij} &= e^{-\beta \mathcal{H}(i,j)} + \textstyle\sum_{i<k<l<j} Z^B_{kl} e^{-\beta \mathcal{I}(i,j;k,l)} + \textstyle\sum_{i<u<j} Z^M_{i+1,u} Z^{M1}_{u+1,j-1} e^{-\beta a} \\
Z^M_{ij} &= \textstyle\sum_{i<u<j} e^{-\beta(u-i+1)c} Z^M_{u+1,j} + \textstyle\sum_{i<u<j} Z^M_{i,u} Z^B_{u+1,j} e^{-\beta b} + Z^M_{i,j-1} e^{-\beta c} \\
Z^{M1}_{ij} &= Z^{M1}_{i,j-1} e^{-\beta c} + Z^B_{ij} e^{-\beta b} \\
& Z_{ii} = 1,\ Z^B_{ii} = Z^M_{ii} = Z^{M1}_{ii} = 0
\end{aligned}
$$

Variations on RNA Folding and Alignment,
Athanasius F. Bompfuenewer, J Math Biol (2007) in press.

## Pseudocode for Zuker's algorithm

```
for d = 1 to n
  for i = 1 to d
    j=i+d;
    compute V(i,j)
    compute W(i,j)
    compute WM(i,j)
return W(1,n)  // free energy of sequence s_1,...,s_n
```

**Notation change: $F_{i,j}$ = W(i,j), $C_{i,j}$ = V(i,j), $M_{i,j}$ = WM(i,j).**

Here `V(i,j)` is

$$\min \begin{cases} Hairpin(i,j), \\ \min\{Interior(i,j;p,q) + V(p,q) : i < p < q < j\}, \\ \min\{WM(i+1,k) + WM(k+1,j-1) + a : i < k < j\} \end{cases}$$

and `W(i,j)` is

$$\min \begin{cases} V(i,j), \\ \min\{WM(i,k) + WM(k+1,j) : i < k < j\} \end{cases}$$

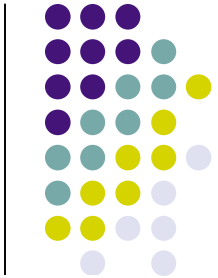
and `WM(i,j)` is

$$\min \begin{cases} b + V(i,j), \\ c + WM(i+1,j), \\ c + WM(i,j-1), \\ \min\{WM(i,k) + WM(k+1,j) : i < k < j\} \end{cases}$$

# bmatch. i-match

- "An RNA folding method capable of identifying pseudoknots and base triples", by Tabaska, Cary, Gabow and Stormo *Bioinformatics* 14(8) 1998.

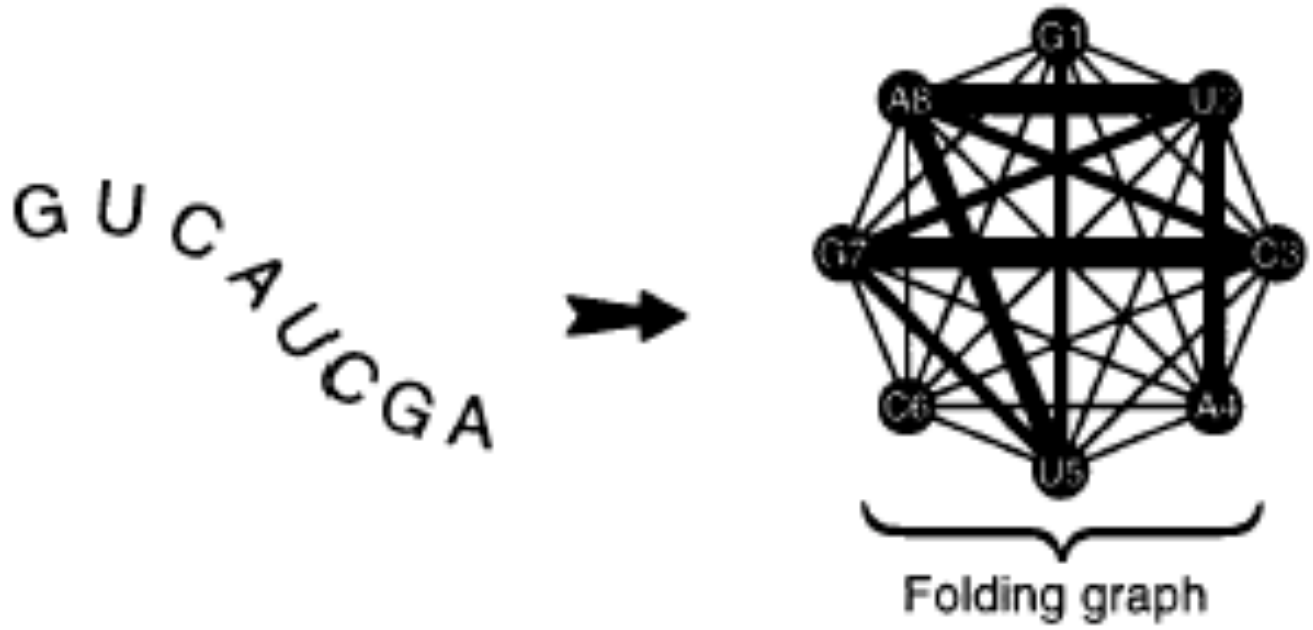
- BASIC IDEA: Given RNA sequence of length n, apply maximum weight matching to complete, weighted (undirected) graph G = (V,E)
  - V = {1,2,…,n}
  - E = all possible Watson-Crick and GU pair positions {i,j}
  - weight edges by mutual information using reliable sequence-structure alignment of many orthologous RNAs.



$$M_{i,\,j} = \sum_{x_i, x_j} f_{x_i,\,x_j} \log 2(f_{x_i,\,x_j} / f_{x_i} f_{x_j})$$

Cary-Stormo in ISMB'95 weighted edges {i,j} by mutual information, and applied Ed Rothberg's implementation wmatch of Gabow's $O(n^3)$ maximum weight matching algorithm.

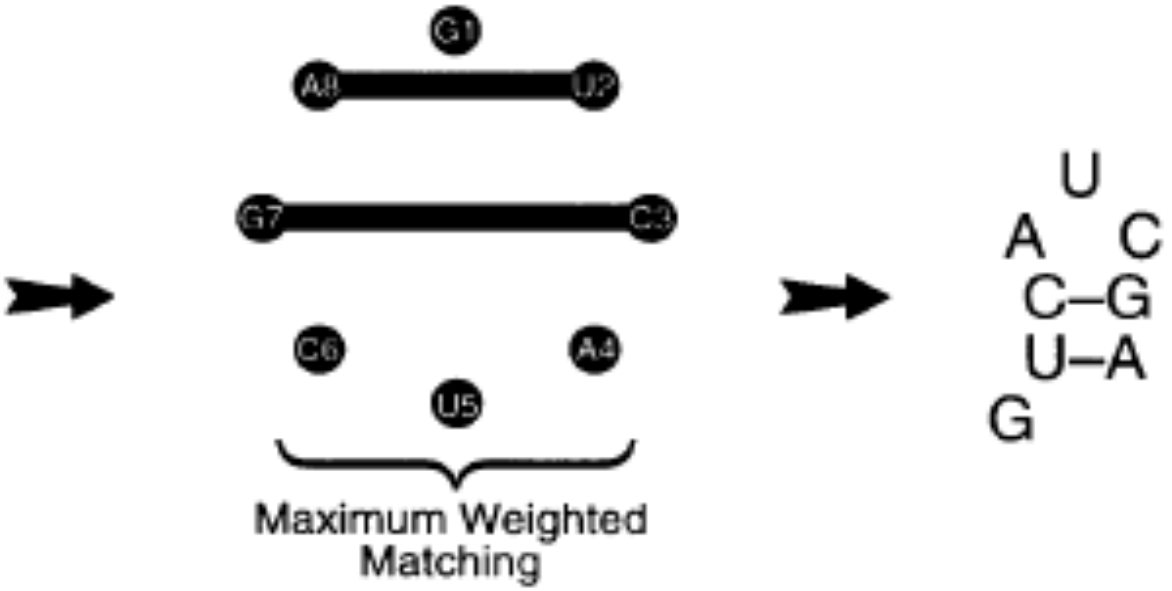
# i-match algorithm




**Pseudoknots can be detected using Cary-Stormo algorithm.**

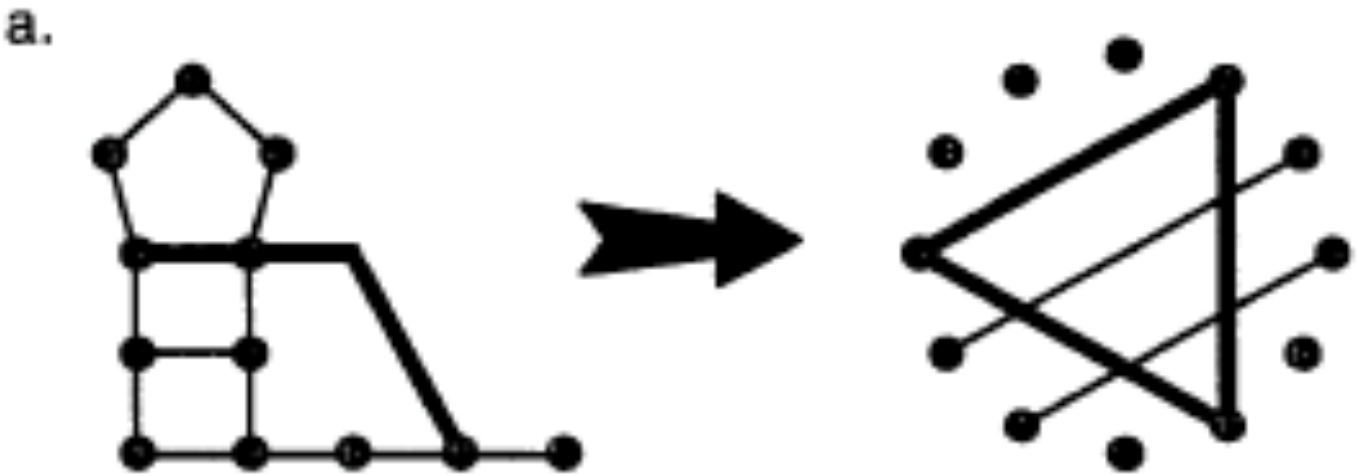
"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# i-match algorithm contd.



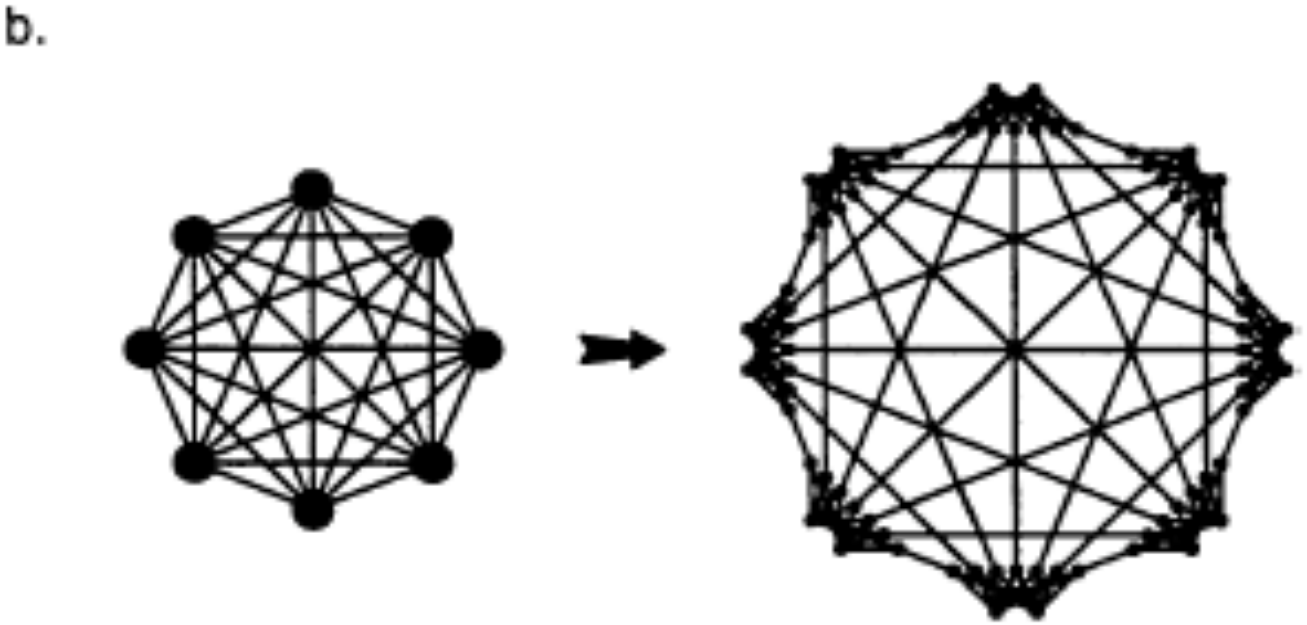

"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# Handling base triples with bmatch



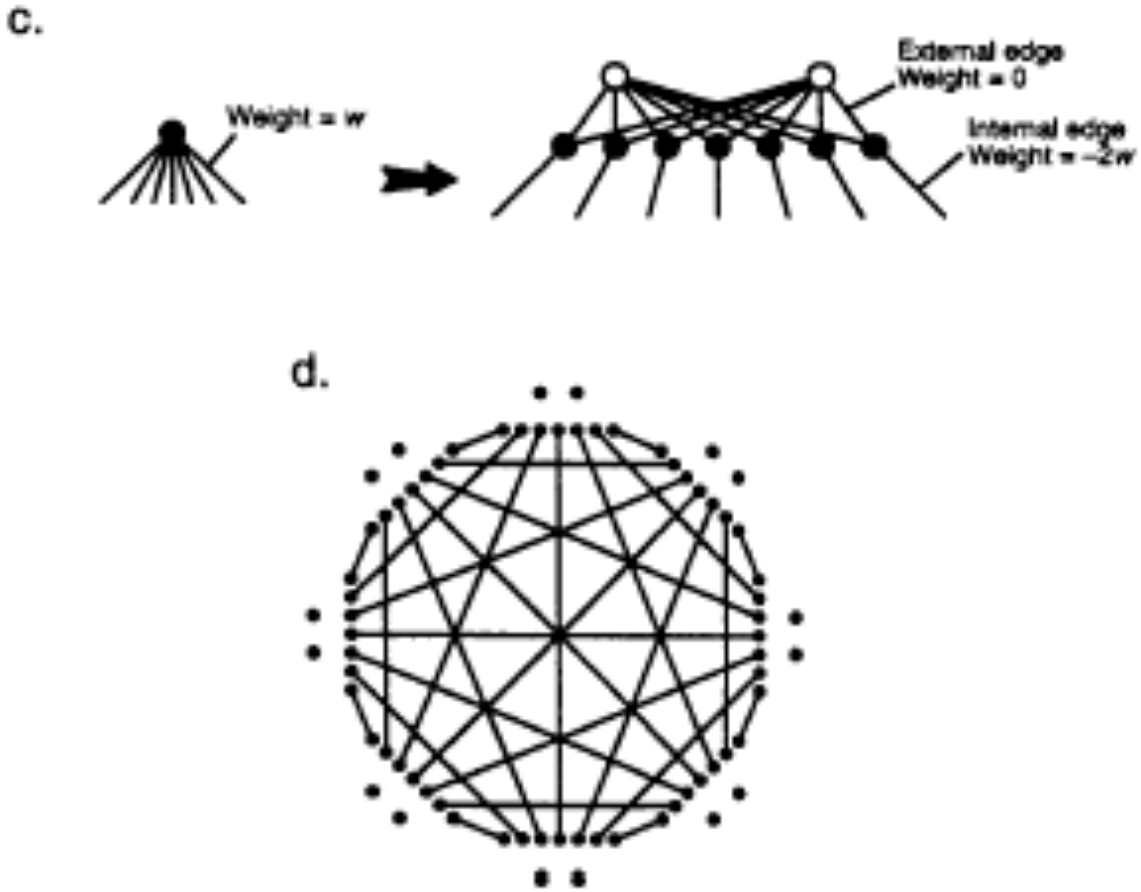

"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# Handling base triples, contd.



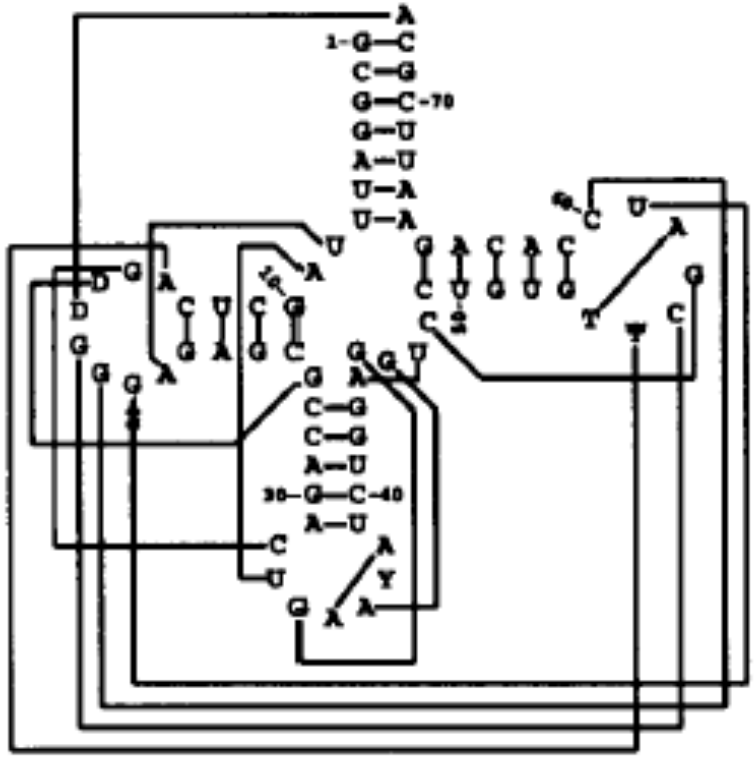

"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# Handling base triples, contd.



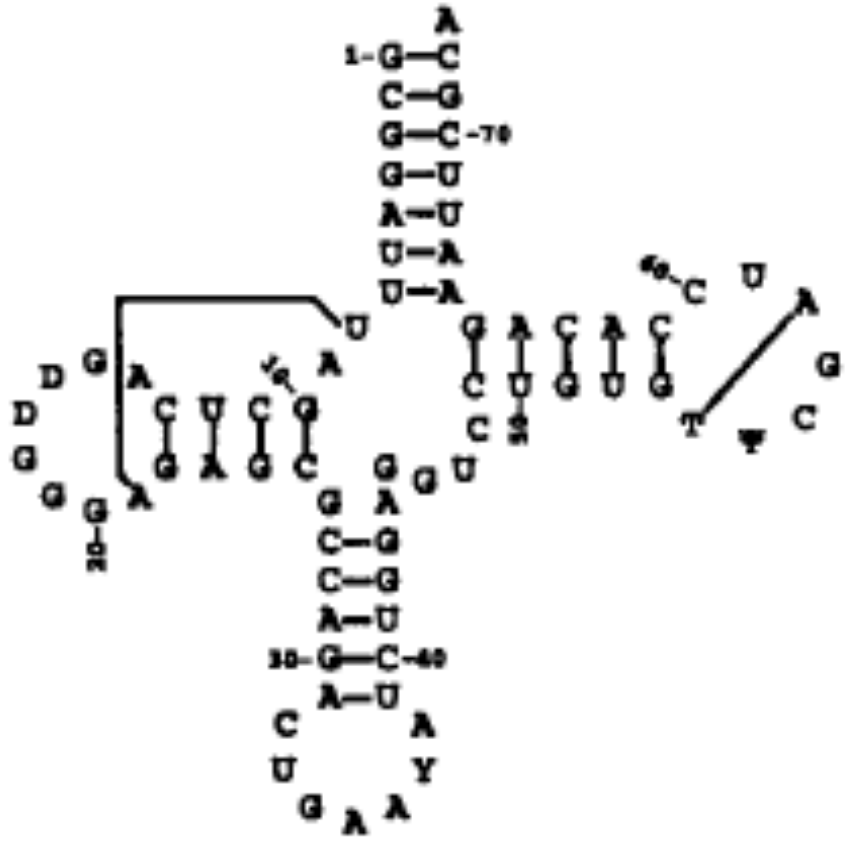

"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# Raw output of i-match on tRNA



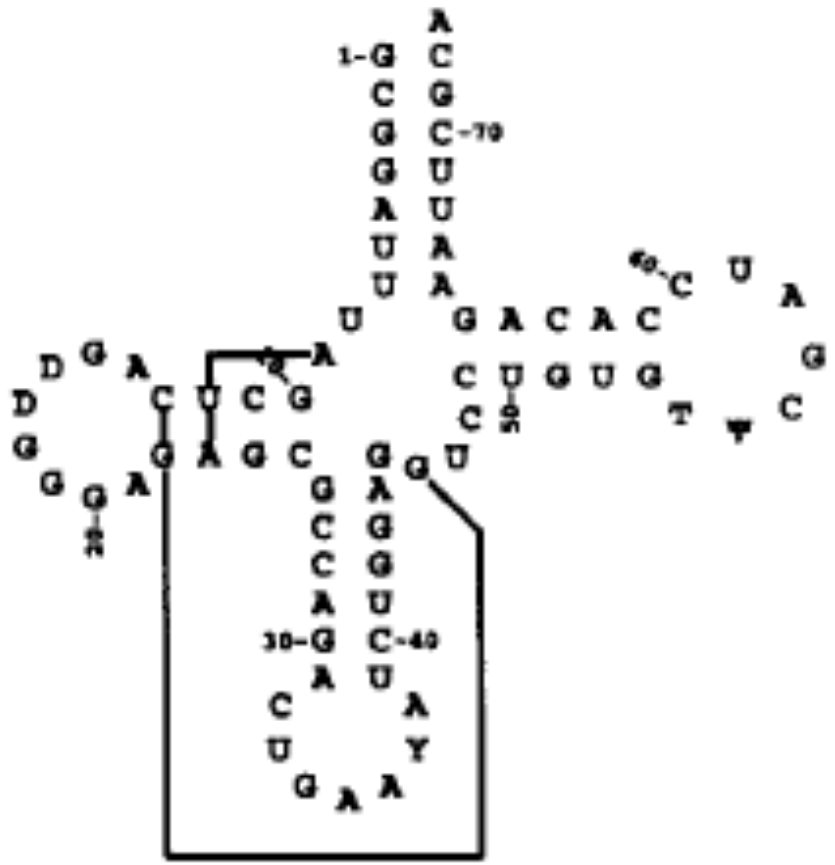
"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# tRMA after i-match output filtration



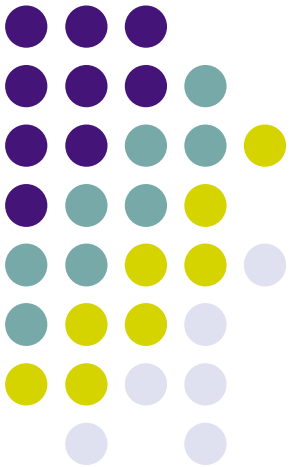
"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# bmatch



"An RNA folding method capable of identifying pseudoknots and base triples", Tabaska et al., Bioinformatics 14 (8) 1998

# Detecting ncRNA genes



**Washietl, Hofacker, Stadler**

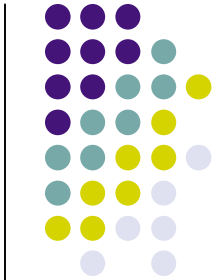
**“Fast and reliable prediction of noncoding RNAs”, Washietl, Hofacker and Stadler,**

**PNAS 102(7): 2454-2459 (2005)**

# Algorithm RNAZ

- INPUT: alignment A of RNAs, as produced by ClustalW or from Comparative Regulatory Genomics (CORG) database for functional RNAs
- OUTPUT: Score between 0 and 1, measuring likelihood that alignment contains structural RNAs

- IDEA: Combine average Z-scores of RNAs in alignment, computed using shuffling, together with a measure of commonality of secondary structure and covariation of base pairs



- Using RNAALIFOLD, compute energy $E_A$ of alignment A using dynamic programming secondary structure prediction together with covariance term for compensatory and consistent mutations
- Using RNAfold, compute average minimum free energy <E> of RNAs in alignment A
- SCI = $E_A$ /<E>
- Z = average Z-scores of RNAs in alignment A

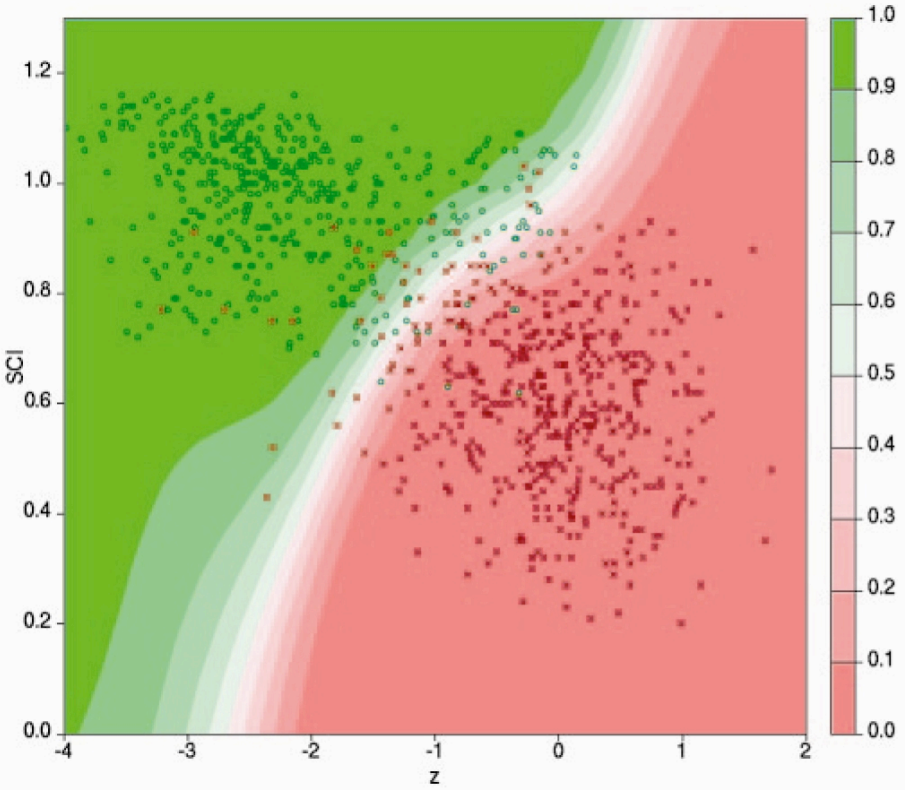



**Fig. 2.** Classification based on *z* scores and SCI by using a SVM. Alignments of tRNAs and 5S rRNAs with two to four sequences per alignment and mean pairwise identities between 60% and 90% are shown. Green circles represent native alignments, and red crosses represent shuffled random controls. The background color ranging from red to green indicates the RNA class probability for different regions of the *z*–SCI plane.

"Fast and reliable prediction of noncoding RNAs", Washietl, Hofacker and Stadler, PNAS 102(7): 2454-2459 (2005)

- Using LIBSVM, train support vector machine (SVM) as binary classifier for structural RNA, given feature vectors:
  - SCI
  - Z
  - number of RNAs in alignment A
  - mean pairwise sequence identity of RNAs in A
- In training set, positive (+1) training examples are alignments of structural RNA from same class in Rfam, negative (-1) training examples obtained by shuffling columns and applying ClustalW
- SVM outputs score P. Compute sensitivity and specificity using threshold P

# Novel procedure to compute Z-scores

- Generate 10,648 random sequences of lengths ranging from 50 to 400 nt. in increments of 50 nt. with base composition GC/AU, A/U, G/C ratios from 0.25 to 0.75 in increments of 0.05
- Using RNAfold, compute Z-scores of random sequences
- Using LIBSVM, train support vector machine (SVM) to determine 2 regression models: model for $\mu$ MFE, model for $\sigma$ MFE.
- SVM uses $\nu$ variant of regression and uses radial basis kernel.

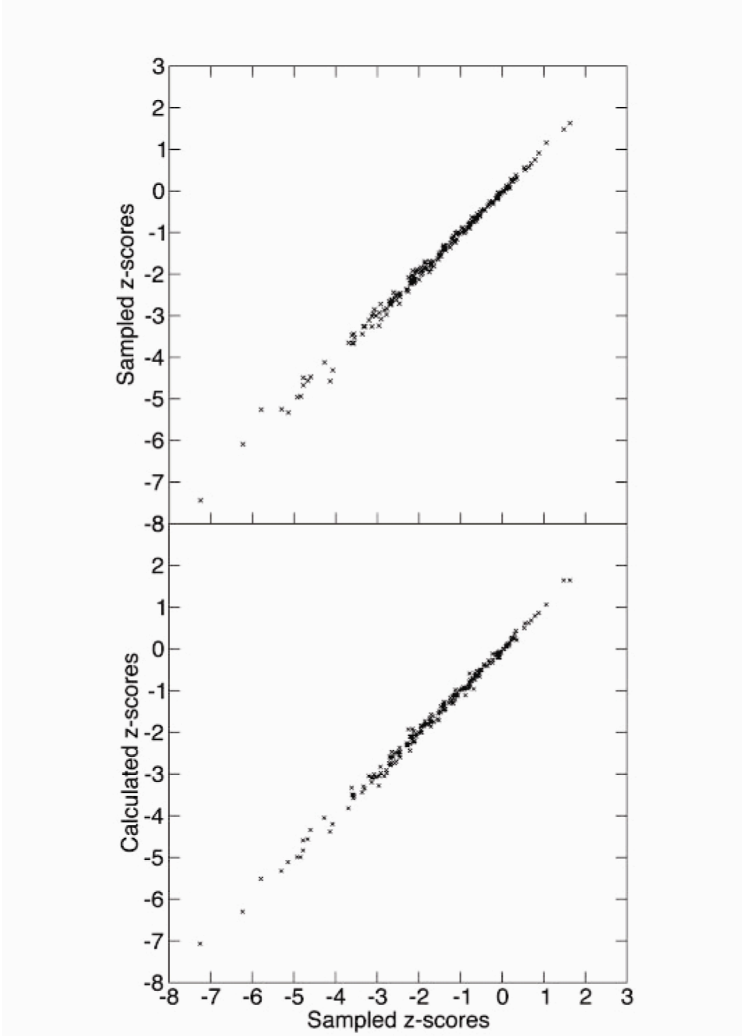
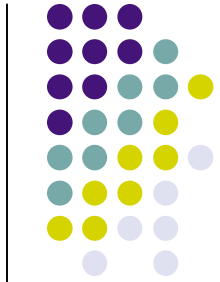



**Fig. 1.** *z* scores calculated by SVM regression in comparison with *z* scores determined from 1,000 random samples for each data point. As test sequences we chose 100 sequences from random locations in the human genome and 100 known ncRNAs from the Rfam database (31). (*Upper*) Correlation of *z* scores from two independent samplings (mean squared error: 0.00990). (*Lower*) Correlation of calculated *z* scores and sampled *z* scores (mean squared error: 0.00998)

"Fast and reliable prediction of noncoding RNAs", Washietl, Hofacker and Stadler, PNAS 102(7): 2454-2459 (2005)



# THANKS!