Chapter 1

UNAFold

Software for Nucleic Acid Folding and Hybridization

Nicholas R. Markham and Michael Zuker

Abstract

The UNAFold software package is an integrated collection of programs that simulate folding, hybridization, and melting pathways for one or two single-stranded nucleic acid sequences. The name is derived from "Unified Nucleic Acid Folding." Folding (secondary structure) prediction for single-stranded RNA or DNA combines free energy minimization, partition function calculations and stochastic sampling. For melting simulations, the package computes entire melting profiles, not just melting temperatures. UV absorbance at 260 nm, heat capacity change (C_p), and mole fractions of different molecular species are computed as a function of temperature. The package installs and runs on all Unix and Linux platforms that we have looked at, including Mac OS X. Images of secondary structures, hybridizations, and dot plots may be computed using common formats. Similarly, a variety of melting profile plots is created when appropriate. These latter plots include experimental results if they are provided. The package is "command line" driven. Underlying compiled programs may be used individually, or in special combinations through the use of a variety of Perl scripts. Users are encouraged to create their own scripts to supplement what comes with the package. This evolving software is available for download at http://www.bioinfo.rpi.edu/applications/hybrid/download.php.

Key words: RNA folding, nucleic acid hybridization, nucleic acid melting profiles (or DNA melting profiles), free energy minimization, partition functions, melting temperature.

1. Introduction

The UNAFold software predicts nucleic acid foldings, hybridizations, and melting profiles using energy-based methods and a general computational technique known as dynamic programming *(1)*. Early software for RNA folding predicted minimum

Jonathan M. Keith (ed.), *Bioinformatics, Volume II: Structure, Function and Applications, vol. 453*
© 2008 Humana Press, a part of Springer Science+Business Media, Totowa, NJ
Book doi: 10.1007/978-1-60327-429-6 Springerprotocols.com

free energy foldings only *(2–6)*. It became clear early on that such methods were unreliable in the sense that many different foldings, with free energies close to the computed minimum, could exist. Although constraints deduced from experiments or phylogenetic analyses could be applied to reduce uncertainty, a method to compute a variety of close to optimal foldings was needed.

The mfold software *(7–10)* computes a collection of optimal and suboptimal foldings as well as a triangular-shaped plot called an energy dot plot (EDP). The EDP contains a dot or other symbol in row *i* and column *j* (*i* < *j*) to indicate that the base pair between the *i*th and *j*th nucleotides can occur in a folding within some user prescribed free energy increment from the minimum. The *Vienna RNA Package* (Vienna RNA) *(11–13)* differs fundamentally from mfold because the underlying algorithm computes partition functions, rather than minimum free energies. This leads naturally to the computation of base pair probabilities *(14)* and what they call "boxplots." We call these triangular shaped plots probability dot plots (PDPs). In this case, all base pairs with probabilities above a certain threshold are plotted as boxes (or other symbols) whose area is proportional to the probability of that base pair. This software can compute all possible foldings close to optimal. The *sfold* package *(15–17)* also computes partition functions, but it uses a simpler algorithm than the Vienna RNA package because base pair probabilities are not computed directly (exactly). Instead, it computes a "statistically valid sample" (Gibbs sample) that permits the estimation of not only base pair probabilities, but probabilities of any desired motif(s). UNAFold encompasses all three methods by computing minimum and suboptimal foldings in the mfold style, and full partition functions that allow both exact base pair computations as well as stochastic sampling. In addition, stochastic sampling may be used to compute both ensemble enthalpy and ensemble heat capacity for single sequence folding.

For practical applications, it is common to use crude methods, often ad hoc, to compute melting temperatures for dimers. It is usual to assume that hybridized strands are perfectly complementary. When mismatches are permitted, they are few and isolated. There is no question about base pairs and no computations are required to determine the hybridization. For example, the simple case:

5′-ACGGtCCAGCAA-3′
3′-TGCCgGGTCGTT-5′

shows the dimerization of two almost complementary 12-mers. Note the single T·G wobble pair. A computer is not needed to compute the hybridization. A calculator would be helpful in adding up nearest neighbor free energies and enthalpies from published tables *(18–20)*, and a computer would be needed to process thousands of these dimers. In any case, free energy at 37°C, Δ*G* (or DG_{37}) and enthalpy, ΔH, are easily computed. From this, the entropy, ΔS, is easily computed as:

$$\Delta S = 1000 \times \frac{\Delta H - \Delta G}{T},$$

where T is the hybridization temperature (K) and the factor of 1000 expresses ΔS in e.u. (entropy units; 1 e.u. = 1 cal/mol/K). The terms "free energy," "enthalpy," and "entropy" are really changes in free energy, enthalpy, and entropy with respect to the "random coil state." In reality, "random coil" is a large ensemble of states. The free energy and enthalpy of this ensemble of states may be (and is) set to zero. Only entropy cannot be arbitrarily assigned. Thus a ΔS term is $R\ln(N/N_{\text{ref}})$, where R is the universal gas constant, 1.9872 e.u.; N is the number of states consistent with hybridization; and N_{ref} is the number of states, always larger, in the unconstrained "random coil."

The simplest model for hybridization assumes that there are two states: hybridized, where the structure is unique and known, and "random coil." It is then a simple matter to deduce a melting temperature, T_{m}, at which half of the dimers have dissociated. The formula is given by:

$$T_m = 1000 \times \frac{\Delta H}{\Delta S + R\ln(\text{C}_{\text{t}}/f)}, \qquad [1]$$

where ΔH, ΔS (expressed in e.u.) and R have already been defined, C_{t} is the total strand concentration, and f = 4 when the two strands are different and f = 1 when self-hybridization takes place. This formula implicitly assumes that both strands are present in equal concentrations. Competition with folding and with other possible hybridizations, such as homodimer formation, is not considered at all.

UNAFold offers a number of increasingly sophisticated ways to compute melting temperatures and also computes entire melting profiles: UV absorbance at 260 nm, heat capacity (C_{p}), and mole fractions of various single and double-stranded molecular species as a function of temperature. Even when Equation [1] is used, the hybridization is computed by minimizing free energy. The various levels of complexity and choice of methods are described in the following.

It is important to issue a warning to those who wish to apply these methods to microarrays. Hybridization on microarrays is complicated by the fact that one of each hybridizing pair is immobilized. It is difficult to compute the effective "solution concentration" for such molecules, and diffusion is probably an important factor in slowing the time necessary to reach equilibrium. However, even for hybridization in solution, kinetic simulations that treat different probe-target pairs independently can lead to totally incorrect results for the time required to reach equilibrium as well as the equilibrium concentrations themselves *(21)*.

It is important to emphasize that the computations used by the UNAFold software are based on a number of assumptions.

1. The simulations are for molecules in solution. For microarrays, computing an effective concentration of the immobilized oligos is necessary. One needs to estimate the "surface phase concentration," c, in units such as "molecules/cm^2," and the solution volume per cm^2 of surface area, v (L/cm^2) *(21)*.
2. The system contains one or two different molecules. If, for example, a true system contains three different oligos, A, B and C, then some common sense must be used. If A and B do not hybridize with each other over the temperature range of interest, then A and C could be simulated separately from B and C if, in addition, the concentration of C is a couple of orders of magnitude greater than that for both A and B.
3. The computations assume that the system is in equilibrium. Thus, the melting profile predictions assume that temperature changes are slow enough so that the system is always at equilibrium. When this is not true, observed melting profiles depend on the rate of temperature change. In particular, a hysteresis effect can be observed. For example, even if the temperature is raised at a uniform rate from T_0 to T_1, and then brought down to T_0 at the same rate, the measured UV absorbance profiles for melting and cooling may differ.

2. Materials: The UNAFold Software

2.1. Supported Platforms

The UNAFold package compiles and runs on many platforms. Table 1.1 lists a few operating system/architecture combinations under which the package is known to work.

In fact, the UNAFold package is not known *not* to run on any platform. The software should function on any system having the following basic tools:

- A Unix-style shell.
- Make.
- A C compiler.
- A Perl interpreter.

2.2. Dependencies

As noted, UNAFold requires only some very basic tools to build and run. However, there are several optional libraries and programs that provide additional functionality when installed:

- gnuplot: If gnuplot is detected, the hybrid2.pl script will use it to produce several plots in Postscript format.
- OpenGL/glut: If OpenGL and the glut library *(22)* are detected, the hybrid-plot program will be built.

Table 1.1
Some platforms supported by the UNAFold package

Operating system	Architecture
Linux	x86
Linux	x86_64
FreeBSD	x86
SunOS	sparc
IRIX	mips
IRIX64	mips
AIX	powerpc
MacOS	powerpc
MS Windows	x86

- gd: If the gd library is detected, the hybrid-plot-ng program will use it to create images in GIF, JPEG, and/or PNG formats directly. (With or without this library, hybrid-plot-ng creates Postscript files which can be converted to virtually any other image format.)

2.3. Downloading and Installing

The UNAFold software is available for download at http://www.bioinfo.rpi.edu/applications/hybrid/download.php. Binaries for Linux (in RPM format) and Windows (in EXE format) are available along with the source code.

After downloading and unpacking, building the software consists of three steps:

- Configuration: Usually this is as simple as typing ./configure. Running ./configure --help lists options that may be used to configure installation locations, specify locations of libraries, and set compiler options.
- Compilation: A single command—make—compiles and links all programs.
- Installation: Typing make install copies the programs, scripts, documentation, and data files to the location set with the configure script.

2.4. Core Programs

The core programs in UNAFold are written in C and are optimized when compiled since they are computationally intensive. Man pages exist for all of these programs. In addition, invoking any program with the --help option will generate an abbreviated set of instructions. Some programs have counterparts formed by adding the suffix -same. When simulating a one-sequence ensemble (in which the dominant dimer is a homodimer rather than a heterodimer) the -same version replaces the regular one.

- Folding:
  - hybrid-ss: Computes full partition functions for folding RNA or DNA. It may be run using the --energyOnly option, in which case the probabilities of base pairs and of single-stranded nucleotides and dinucleotides will not be computed, saving significant time and memory. It may also be run with the --tracebacks option to generate a stochastic sample of foldings. UNAFold also includes two simplified versions of hybrid-ss: hybrid-ss-simple and hybrid-ss-noml. hybrid-ss-simple assigns fixed entropic costs to multibranch loops and ignores single-base stacking, while hybrid-ss-noml does not allow multibranch loops at all.
  - hybrid-ss-min: Computes minimum energy foldings. It can predict a single, minimum free energy folding or, using an extended algorithm, generate an mfold-like collection of foldings and an EDP. For fast folding of many sequences with the same parameters, the --stream option reads sequences, one at a time, from standard input and writes free energies to standard output.
- Hybridization:
  - hybrid: Computes full partition functions for hybridizing RNA or DNA (without intramolecular base pairs). The --energyOnly and --tracebacks options function as in hybrid-ss.
  - hybrid-min: Computes minimum energy hybridizations. The --mfold and --stream options function as in hybrid-ss-min.
- Ensemble computations:

  These programs rely on the output of the core programs.
  - concentration(-same): Computes the mole fractions of different molecular species using the already computed free energies for individual monomer (folded) and dimer (hybridized) species.
  - ensemble-dg(-same): Computes ensemble free energies using species free energies and computed mole fractions.
  - ensemble-ext(-same): Computes UV absorbance at 260 nm using computed probabilities for single-strandedness of nucleotides and (optionally) dinucleotides, computed mole fractions, and published extinction coefficients *(23)*.

2.5. Auxiliary Programs

- ct2rnaml: Converts structures in .ct format to RNAML *(24)*. ct2rnaml supports files with multiple structures on a single molecule as well as multiple molecules.
- ct-energy: Evaluates the free energy of structures in .ct format, with detailed free energy information for each loop if desired.
- ct-ext: Evaluates the UV absorbance of structures in .ct format.
- ct-prob: Estimates base pair probabilities from a stochastic sample in .ct format. ct-prob can also compare these probabilities to the "correct" probabilities if they are available.
- ct-uniq: Selects the unique structures from a set in .ct format. ct-uniq is typically used to remove duplicates structures from a stochastic sample.
- dG2dH, dG2dS, dG2Cp: Compute enthalpy, entropy and heat capacity, respectively, by differentiating free energy. Enthalpy and entropy are given by $\Delta H = \Delta G + T\dfrac{\partial \Delta G}{\partial T}$ and $\Delta S = \dfrac{\partial \Delta G}{\partial T}$; heat capacity is given by $\mathrm{C_p} = -T + \dfrac{\partial^2 \Delta G}{\partial T^2}$. dG2Cp also determines the melting temperature(s) $T_\mathrm{m}(\mathrm{C_p})$, the maximum or local maxima of the heat capacity curve.
- hybrid-plot(-ng): Displays PDPs created with hybrid or hybrid-ss. hybrid-plot is an interactive program based on glut, while hybrid-plot-ng produces static images in Postscript and GIF/JPEG/PNG formats.

2.6. Linking Software: Perl Scripts

We describe our scripts that combine a number of basic programs. This set is expected to grow, and we welcome both suggestions from users and user-generated scripts.

- concentrations(-same).pl: Produces normalized mole fractions (between 0 and 1) from the concentrations calculated by concentration(-same). concentrations(-same).pl also computes T_m(Conc), the temperature at which one half of the strands are single-stranded and unfolded.
- ct-energy-det.pl: Converts the verbose output of ct-energy to energy details in plain text or HTML.
- h-num.pl: Computes h-num values from an EDP. h-num is a measure of well-definedness for a helix.
- hybrid(-ss)-2s.pl: Simulates two-state hybridization or folding. hybrid-2s.pl uses hybrid-min to compute a minimum energy hybridization and extrapolates it over a range of temperatures. hybrid-2s.pl can be used in place of hybrid to obtain a two-state model instead of a full ensemble. Likewise, hybrid-ss-2s.pl uses hybrid-ss-min and can replace hybrid-ss.

- hybrid2.pl: Uses hybrid, hybrid-ss, concentration, ensemble-dg, and ensemble-ext to produce a full melting profile. The "flagship" script, hybrid2.pl also employs gnuplot to produce Postscript plots of mole fractions (from concentrations.pl), heat capacity (from dG2Cp) and UV absorbance. hybrid2.pl operates in several different modes depending on how it is invoked:
  - If hybrid2.pl is invoked with only one sequence (or with two copies of the same sequence), it automatically replaces concentration with concentration-same, ensemble-dg with ensemble-dg-same, and so on.
  - If hybrid2.pl is invoked as hybrid2-min.pl, hybrid-min and hybrid-ss-min replace hybrid and hybrid-ss so that the entire computation is performed using only an optimal structure for each species at each temperature, instead of a full ensemble. Likewise, invoking hybrid2-2s.pl replaces hybrid(-ss) with hybrid(-ss)-2s.pl to obtain a two-state model for each species.
  - Finally, if hybrid2.pl is invoked as hybrid2-x.pl, hybrid2-min-x.pl or hybrid22s-x.pl, the homodimers and monomers are excluded from consideration, so that only the heterodimer is simulated. hybrid2-2s-x.pl thus implements the classical two-state model, although it is still enhanced in that strand concentrations may be unequal.
- hybrid-select.pl: Folds or hybridizes sequences using only selected base pairs. hybridselect.pl runs two folding or hybridization programs; the PDP or EDP from the first run is used to select base pairs to be considered in the second run. If the first run is an energy minimization (EM), base pairs that appear in foldings with energies in a user-specified percentage of the minimum energy are selected; if it is a partition function (PF), base pairs with at least a user-specified probability are selected. hybrid-select.pl may be used for foldings (running hybrid-ss and/or hybrid-ss-min) and for hybridizations (running hybrid and/or hybrid-min). Since there are four possible methods—EM followed by EM, EM followed by PF, PF followed by EM, and PF followed by PF—there are a total of eight ways to use hybrid-select.pl. Additionally, the --mfold option may be used if the second stage is an EM, whereas --tracebacks is acceptable when the second stage is a PF.
- ITC-plot.pl: Produces isothermal titration calorimetry (ITC) plots *(25)*. In an ITC plot, the concentration of one strand is varied while the other is held constant, and enthalpy change is plotted as a function of the changing strand concentration.
- melt.pl: Quickly computes two-state melting for monomers, homodimers or heterodimers. Only ΔG, ΔH, ΔS, and

T_m are output, but melt.pl can process many sequences very quickly.

- plot2ann.pl: Creates probability annotations for a structure in .ct format. Paired bases are annotated with the probability of the pair, whereas unpaired bases are annotated with the probability of being single-stranded. The resulting .ann file can be used by sir_ graph (part of the mfold_util package) to display an annotated structure graph.
- ss-count.pl: Estimates the probability of each base being single-stranded from a stochastic sample in .ct format. Like ct-prob, ss-count.pl can also compare these probabilities to the "correct" probabilities if they are available.
- UNAFold.pl: Folds a sequence with detailed textual and graphical output. The package's namesake script, UNAFold.pl, uses hybrid-ss-min to fold a sequence and then uses components of mfold_util to expand on the output. boxplot_ng is used to produce an EDP in Postscript format and, if possible, as a PNG, JPEG, or GIF. sir_graph_ng creates a structure plot for each structure in the same formats; the --ann option allows these structures to be automatically annotated with ss-count or p-num information. Finally, UNAFold.pl can produce an HTML file with hyperlinks to all of the plots and other information, unifying the folding results.
- vantHoff-plot.pl: Produces van't Hoff plots. In a van't Hoff plot, the concentrations of both strands are varied together, and the inverse of the melting temperature is plotted as a function of total strand concentration. A linear van't Hoff plot may be taken as evidence that the transition follows a two-state model.

3. Methods

This section presents a number of examples of using the UNAFold software.

3.1. Energy Data

The UNAFold software has two built in sets of energy files for DNA and RNA, respectively. For DNA, we use free energies at 37°C and enthalpies from the SantaLucia laboratory at Wayne State University in Detroit, MI *(19)*. For RNA, we use equivalent parameters from the Turner laboratory at the University of Rochester in Rochester, NY *(26)*. In both cases, it is possible to generate free energies at different temperatures in order to compute melting profiles. These DNA and RNA energy parameters are referred to as versions 3.1 and 2.3, respectively. More up-to-date free energies for RNA are available, also from the Turner laboratory, but they are for 37°C only *(27)*. These are denoted as version 3.0 energies. The built in energies are the default. A suffix flag, --suffix, is required to use any other set of (free) energies. Thus, the flag, --suffix DAT, tells any folding or hybridization program that version 3.0 RNA free energy parameters should be used.

The transition from minimum free energy (mfe) folding to partition function computations forced us to examine the energy rules with great care to avoid over-counting various states. For example, for version 3.0 RNA parameters, single wobble pairs (G·T or T·G) are allowed, but tandem wobble pairs are prohibited. Such motifs may still occur, but only as tandem mismatches in 2 × 2 symmetric interior loops. The result is that the number of valid base pairs varies slightly when using the RNA version 3.0 rules. In this context, three sets of zero energies are available and may be used in conjunction with partition function programs to count the number of valid hybridizations or foldings. They are NULD, NUL, and NULDAT, and refer to DNA version 3.2, RNA version 2.3 and RNA version 3.0 parameters, respectively. The first parameter sets are currently identical. Only RNA version 3.0 rules alter permitted base pairs, 1 × 1 loops and 2 × 2 loops; and these changes always involve wobble pairs.

3.2. Extended ct File Format

A ct file is a text file that defines a nucleic acid sequence together with its secondary structure. It may contain multiple foldings of a single sequence or even multiple foldings of different sequences. The original format was described by Zuker *(8)*, although it originated years earlier.

The first line (record) of a ct file contains an integer, *N*, which is the length of the nucleic acid sequence, followed by a "title." The title comprises all characters that occur on this line after *N*. For example, suppose that:

dG = –244.9 AvI5 Group 2 intron

is the first line of a ct file. This intron contains 565 bases. The title is:

dG = –244.9 AvI5 Group 2 intron

In this example, the title contains a predicted free energy and a description of the RNA. However, any text is valid in the title. Unfortunately, this format is not well understood or else it is simply ignored. For example, the RnaViz program *(28, 29)*, requires "ENERGY =" in the title line. Certain programs in the mfold package expect "dG =".

The following *N* lines (records) contain eight columns. The last two columns were introduced for UNAFold. The *i*th record following the title contains:

1. *i*, the index of the *i*th base.

2. r_i, the *i*th base. It is usually A, C, G, T, or U (upper or lower case), although other letters are permitted.
3. $5'(i)$. The 5′, or upstream connecting base. It is usually i–1, and is 0 when r_i is the 5′ base in a strand. If $5'(1) = N$, then the nucleic acid is circular.
4. $3'(i)$. The 3′, or downstream connecting base. It is usually i+1, and is 0 when r_i is the 3′ base in a strand. If $3'(N) = 1$, then the nucleic acid is circular.
5. bp(i). The base pair, $r_i \cdot r_{\mathrm{bp}(i)}$, exists. It is 0 if r_i is single-stranded.
6. The "historical numbering" of r_i. It may differ from i when a fragment of a larger sequence is folded or to indicate genomic DNA numbering in spliced mRNAs. The UNAFold software currently folds entire sequences only, but this feature is used when two sequences hybridize.
7. 5′stack(i). If this is 0, then r_i does not stack on another base. If it is $k \neq 0$, then r_i and r_k stack. In our applications, $k = i - 1$ when not zero, indicating that the base 5′ to r_i stacks on r_i. When bp(i) = $j > 0$, then r_k stacks 5′ on the base pair $r_i \cdot r_j$.
8. 3′stack(i). If this is 0, then r_i does not stack on another base. If it is $k \neq 0$, then r_i and r_k stack. In our applications, $k = i + 1$ when not zero, indicating that the base 3′ to r_i stacks on r_i. When bp(i) = $j > 0$, then r_k stacks 3′ on the base pair $r_i \cdot r_j$.

The UNAFold software uses the last two columns of the ct file only to indicate stacking on adjacent base pairs. Such stacking is assumed in base pair stacks. That is, if two adjacent base pairs, $r_i \cdot r_j$ and $r_{i+1} \cdot r_{j-1}$, both exist, then 5′stack(i + 1) = i, 5′stack(j) = j – 1, 3′stack(i) = i + 1 and 3′stack(j – 1) = j.

When 5′stack(i) = i – 1 and bp(i – 1) = 0, then r_{i-1} is single-stranded and stacks 5′ on the adjacent base pair, $r_i \cdot r_j$. When 3′stack(i) = i + 1 and bp(i + 1) = 0, then r_{i+1} is single-stranded and stacks 3′ on the adjacent base pair, $r_i \cdot r_j$.

At the present time, the last two columns of ct files contain redundant information and were introduced solely so that the stacking of single-stranded bases on adjacent base pairs could be specified unambiguously. They could be used, for example, to indicate coaxial stacking or adjacent or almost adjacent helices, but UNAFold neither predicts foldings with coaxial stacking, nor does it accept ct files that specify such stacking.

3.3. Sequence File Format

The UNAFold software accepts either raw sequence files or the FASTA format. The former term denotes files that contain only nucleotides. There is no line (record) length limit and blank spaces are ignored, including tab and control characters. The letters "A," "C," "G," "U," and "T" are recognized. Upper and lowercase are accepted and case is conserved so that both can be used to distinguish, for example, one region from another. Note that "U" is treated as "T" if DNA energies are used, and "T" is recognized as "U" for RNA energies. All other characters are treated as nucleotides that cannot pair. In terms of energies, they are treated as neutral. The UNAFold software does not recognize the IUPAC ambiguous base nomenclature. So, for example, "R" and "Y" are not recognized as purine and pyrimidine, respectively, and are not permitted to form base pairs. In particular, "N" does not mean "can take on any value." Thus, in UNAFold, the base pair, G·N, cannot form. In other contexts, it could form as either G·C or G·T. The semicolon character ";" is recognized as a sequence divider. That is, multiple sequences may be stored in a single file, as long at they are separated by semicolons. They do not have to be on separate lines.

UNAFold also accepts FASTA format. The first line of a sequence file must begin with >. The remainder of the line is interpreted as the sequence name, and is read and used by the software. Subsequent lines contain pure sequence. The FASTA format may be used for files containing multiple sequences. The semicolon is no longer required to separate sequences. Only the minimum energy prediction programs use the multiple sequence file (msf) format. The partition function programs read the first sequence only.

If a sequence file name ends with .seq, which we recommend, then the "prefix" is formed by stripping the final four characters (.seq). In all other cases, the prefix is the same as the sequence file name. Output file names are standard and based on the prefix (or prefixes). Suppose then, that two sequence files are named probe.seq and target.seq. Then output files will begin with either probe or target and signify folding output. File names beginning with probe-target, probe-probe, or target-target indicate hybridization output and refer to the heterodimer and two possible homodimers. Because UNAFold automatically uses the hyphen character "-" to form output file names, sequence file names should not contain hyphens.

When raw sequence files are used, the prefix of the file name is used as the sequence name. The sequence name appears only in "ct file" output and in secondary structure or hybridization plots. Only raw sequence format should be used when melting profiles are being computed.

3.4. Folding a Sequence

We selected a Group II intron, *A.v.*I5, from the Eubacterium, *Azotobacter vinelandii (30)*. The sequence was placed in the file A_v_I5.seq using FASTA format.

The phylogenetically derived secondary structure is from the "Mobile group II introns" web site.* It is classified as type B, or "group IIB-like" *(31, 32)*. It was downloaded from this web site as a ct file named a_v_i5_r.ct, renamed here as A_v_I5phylo.ct.

* Zimmerly Lab Web site http://www.fp.ucalgary.ca/group2introns/

In the sequel, this folding is called the "correct" folding, or the reference folding.

A variety of programs were used to predict foldings of the intron, and results were compared to the correct folding. Our purpose is to illustrate how the software may be used rather than how to model particular ribozymes.

When a variety of UNAFold programs are run separately by a user, instead of together by a Perl script, it is easy to overwrite output previously predicted. A ct file from a stochastic sample, for example, could be overwritten by a ct file from mfe computations. In such cases, it is prudent to create one or more copies of the sequence file using different names. In this case, we used the file name A_v_I5min.seq for mfe computations. In addition, programs that write to "standard output" often require the user to direct the results into files whose names should be chosen with some care.

3.4.1. Partition Function and Stochastic Sampling

The hybrid-ss program was run using the command:

```
hybrid-ss --suffix DAT --tracebacks 100 A_v_
    15.seq
```

The --suffix flag specifies version 3.0 RNA parameters, which are recommended in situations in which melting profiles are not being computed. The --tracebacks 100 flag requests a stochastic sample of 100 secondary structures. The output comprises five new files, `A_v_I5.run, A_v_I5.dG, A_v_I5.37.ct, A_v_I5.37.ext, and A_v_I5.37.plot`. The contents of these text files are listed according to suffix.

run A simple description of what program was run. Some, but not all, of the flags appear.

dG The Gibbs free energy, -$RT\ln(Z)$ and the partition function value, Z. Free energy units are kcal/mol.

ct A concatenation of all 1,000 secondary structures. This is a stochastic sample, so the ordering is not relevant. The free energies are randomly distributed.

ext As indicated in the top record, this file lists, for every i between 1 and N: (the sequence size), i, the probability that r_i is single-stranded, and the probability that both r_i and r_j are single-stranded. These files are used to compute UV absorbance, as previously described.

plot This file contains a list of all possible base pairs and their probabilities. The simple format for each record is i, j and the probability of the base pair $r_i \cdot r_j$ in the Boltzmann ensemble. Unlike the ct files, these files always assume that $i < j$ for folding. (This is not the case in plot files for hybridization.) UNAFold does not output base pairs with probabilities $<10^{-6}$, and these files are easily filtered to select only higher probability base pairs.

Note that a double suffix is employed in naming output files. When appropriate, the temperature is inserted in °C. Even when the temperature or range of temperatures is not specified on the command line, it is specified in the last line of the miscloop energy file.

3.4.2. Simple mfe Computation

The command:

```
hybrid-ss-min --suffix DAT A_v_I5min.seq
```

produced a single mfe folding. The resulting ext and plot files contain zeros or ones for probabilities. A value of one means that, as the case may be, r_i is single-stranded, both r_i and r_{i+1} are single-stranded, or that the base pair $r_i \cdot r_j$ exists in the predicted folding. Zero probabilities indicate the negation of the above. The resulting dG file contains the mfe and the Boltzmann factor for that free energy alone. It is impossible to distinguish between dG files created by partition functions and those created by energy minimization. The formats are identical and the suffix dG is added to the sequence file name prefix in both cases. However, the single mfe structure is placed in a ct file named by adding only a ct suffix. The folding temperature is not added.

3.4.3. Energy Computations from ct Files

The free energies of the 100 stochastically sampled foldings were evaluated using the ct-energy program as follows:

```
ct-energy --suffix DAT A_v_I5.37.ct
```

Used in this manner, the output is a stream of free energies, one per line, containing the evaluated free energies of the successive foldings. Note that neither hybrid-ss nor hybrid computes free energies for sampled foldings, so that running ct-energy is the only way to evaluate their free energies. Using the --suffix DAT flag evaluates the free energies using the same free energies that generated them. Substituting, for example, --suffix DH, would compute the enthalpies of these structures. The output stream from ct-energy was sorted by energy and directed to a file named A_v_I5sample.dG. These random free energies have a Γ-distribution, but this file was used only to compute a mean and standard deviation of the sample free energies.

The command:

```
ct-energy --suffix DAT A_v_I5phylo.ct
```

was invoked to evaluate the free energy of the correct folding. The result was "+inf", which means $+\infty$! This situation required some "detective work," and so ct-energy was rerun with the --verbose flag. This produced a stream of output containing the energy details of all stacks and loops in the structure. The line:

```
Interior: 88-C 109-G, 90-U 107-G: +inf
```

was detected. This turns out to be a 1 × 1 interior loop closed by a U·G wobble pair. The version 3.0 RNA rules do not allow this. To be specific, this motif must be treated as a 2 × 2

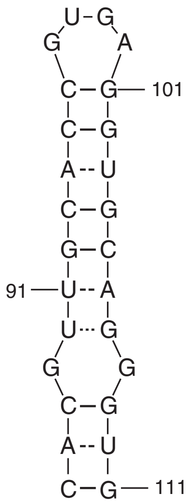

Fig. 1.1. A portion of the phylogenetically determined secondary structure for the group IIB intron, I5 from the Eubacterium, *Azotobacter vinelandii*. The UNAFold software with version 3.0 RNA energies requires that the G.G mismatch be treated as a 2×2 interior loop by deleting the $U^{90} \cdot G^{107}$ base pair.

Table 1.2
Free energy data on folding the group II intron, *A.v.*I5

ΔG_{ens}	ΔG_{min}	ΔG_{phylo}	Sample ΔG_{min}	Sample mean	Sample std
–273.2	–244.9	–235.2	–235.9	–223.6	5.6

A sample of 100 foldings was generated using stochastic traceback. From left to right, the Gibbs free energy, mfe, the free energy of the correct structure, the sample mfe, and the sample mean and standard deviation. The units are kcal/mol.

interior loop. The "offending" base pair is shown in **Fig. 1.1**. The ct-energy program does not currently flag such peculiarities and users should be aware that slight adjustments may be required when dealing with secondary structures derived in other ways. The base pair was removed from the ct file using a text editor, after which the free energy of the reference structure was evaluated with ease.

Table 1.2 gives information on the various free energies associated with these predictions. Note that the correct folding is almost 10 kcal/mol less stable than the mfe folding. Its free energy is, however, almost identical to that of the most stable folding in a sample of 100, just over two standard deviations above the mean. The message is that one should not expect the free energy of the correct folding to be close to the mfe. However, it should be within the range of energies from a stochastic sample, and in this case the result is as good as one could hope for.

3.4.4 Running Hybrid-ss-min in mfold Mode

To create an mfold-like sample of foldings the UNAFold.pl script was run in default mode:

```
UNAFold.pl A_v_I5.seq
```

It produced 18 different foldings together with structure plots and EDPs. As with mfold, the free energies are re-evaluated when version 3.0 RNA energies are used. Free energies of multi-branch loops are (re)assigned energies using a more realistic function that grows as the logarithm of the number of single-stranded bases. The same set of foldings, minus plots and energy re-evaluation, would be predicted using the more primitive command:

```
hybrid-ss-min --suffix DAT --mfold A_v_I5.seq
```

The –mfold flag indicates mfold mode. It accepts three parameters; P, W, and MAX. These are the energy increment for suboptimal foldings expressed as a Percent of the mfe, the Window parameter and the MAXimum number of foldings that may be computed, respectively. The default values are 5%, 3%, and 100%, respectively.

Note that the default is for 2.3 free energies and that the --suffix DAT flag is required to ensure the use of version 3.0 energies. The UNAFold.pl script behaves like mfold, in which the default is version 3.0 free energies.

3.4.5. Comparisons and Plots

The ct_compare program from the mfold package and the ct_boxplot program from the mfold_util package were used to compare results. The ct_compare program reads a reference structure, in this case the correct structure for the intron, and compares it with a number of other foldings on the same sequence from a second ct file. ct_compare was run on both the stochastic sample and mfold-like samples. Table 1.3 shows the results. There are 166 base pairs in the correct structure. False-positive refers to predicted base pairs that are not correct, and false-negative refers to base pairs that are correct but not predicted.

Thus, none of the foldings in the stochastic sample do well, whereas the fourth of 18 foldings in an mfold-like sample scores very well. In fact, the free energy re-evaluation scores this folding as the most stable in the sample. This latter fact is a matter of luck. However, it is reasonable to expect that folding close to the mfe can be a fairly good approximation to the correct fold.

Secondary structure plots of several structures were generated from ct files using the interactive sir_graph program from the mfold_util package. They may be "annotated" in a variety of ways *(33–35)*. We used the command:

```
plot2ann.pl A_v_I5.37.plot A_v_I5phylo.ct
```

to create a stream of output (standard output) that was directed into a file named A_v_I5phylo.ann. This file was used to annotate the correct folding of the intron, as shown in Fig. 1.2.

It is not surprising that the correct folding has low probability base pairs. However, it might surprise some that mfe and close to mfe foldings may also contain low probability base pairs. The

Table 1.3
Numbers of correctly and incorrectly predicted base pairs

Structure	PF min	PF best	ΔG_{min}	mfold best
Correct	101	113	101	145
False-positive	68	51	67	23
False-negative	65	53	65	21

The reference folding contains 166 base pairs. "PF" denotes partition function, so that "PF min" is the mfe structure from the stochastic sample. "PF best" is the structure from the stochastic sample with the largest number of correctly predicted base pairs. ΔG_{min} is the overall mfe folding, always the first in an "mfold-like" sample, and mfold best refers to the structure in the "mfold-like" sample that scores best.

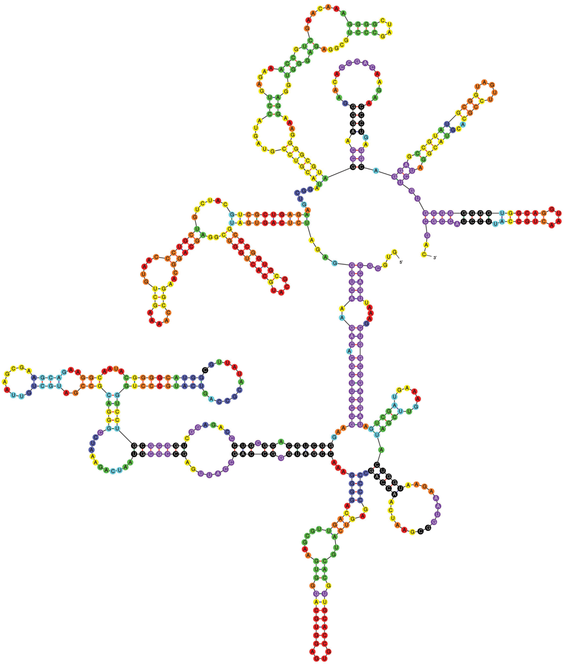

Fig. 1.2. Probability annotation of the secondary structure of the group II intron from *Azotobacter vinelandii*. Probabilities of correct base pairing correspond to colors as follows: black: <0.01, magenta: 0.01–0.10, blue: 0.10–0.35, cyan: 0.35–0.65, green: 0.65–0.90, yellow: 0.90–0.99, orange: 0.99–0.999, and red: > 0.999. Twenty-eight (17%) of the base pairs have probabilities <1%, and 31 more (19%) have probabilities <10%.

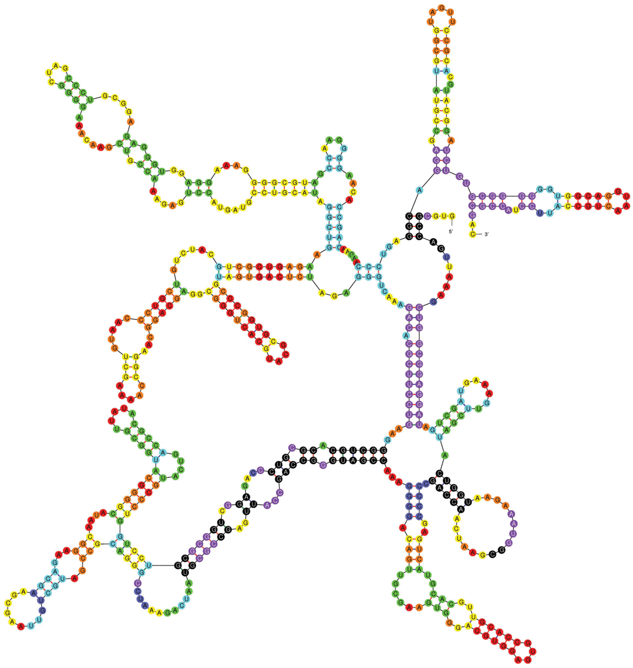

Fig. 1.3. Probability annotation of a computed folding of the group II intron from *Azotobacter vinelandii*. The color scheme is the same as in Fig. 1.2. Its free energy is only 1.9 kcal/mol above the mfe (non-revised), and yet it contains 25 base pairs (15%) with probabilities <1%. Another 24 base pairs (14%) have probabilities <10%.

structure closest to the correct folding in the mfold-like folding was the fourth folding, placed in A_v_I5_4.ct by UNAfold.pl. Proceeding as with the correct folding, an annotation file for this structure was created, resulting in **Fig. 1.3**.

We generated probability annotation files using a plot file computed by hybrid-ss. Plot files may also be computed from stochastic samples using ct-prob, as described in **Section 2.5**. However, there is no reason to restrict the use of ct-prob to stochastic samples. By concatenating the ct files for the correct folding and for the "best mfold-like" folding (A_v_I5_4.ct) and storing the result in temp.ct, the command:

```
ct-prob temp.ct
```

produces an output stream that we directed to the file phylo-mfold_best.plot. It contains a list of all the base pairs that appear in either folding. The probabilities are 1 (occur in both structures) or 0.5 (occur in just one structure). Running plot2ann.pl using this "plot" file allows us to annotate one structure with respect to another. **Figure 1.4** presents the correct intron folding and shows clearly which base pairs occur in the A_v_I5_ct structure.

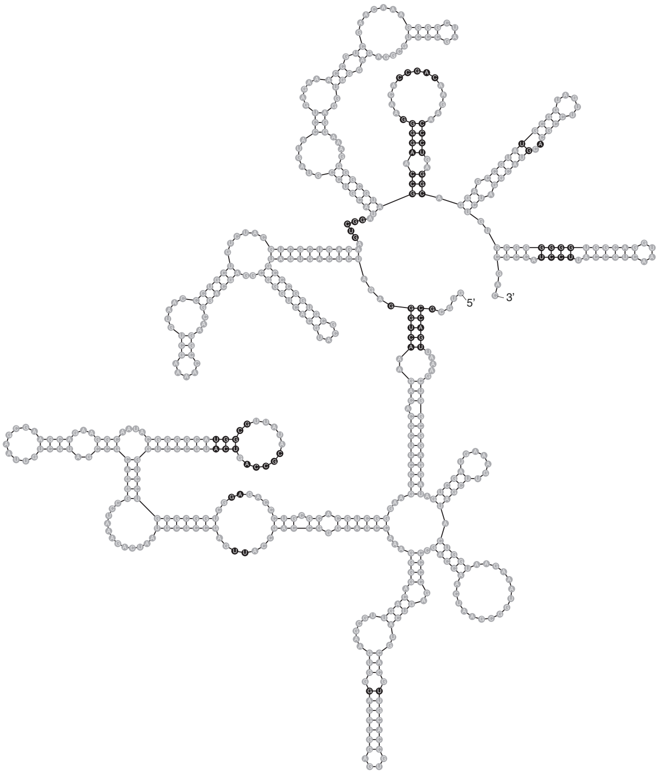

Fig. 1.4. The secondary structure of the group II intron from *Azotobacter vinelandii*. Paired gray encircled bases occur in the predicted folding, A_v_I4_4.ct, shown in Fig. 1.3. Single-stranded gray encircled bases are also single-stranded in the predicted folding. Paired black encircled bases do not occur in the predicted folding, and black encircled single-stranded bases are paired in the predicted folding.

3.5. Hybridizing Two Sequences

Both hybrid and hybrid-min predict intermolecular base pairs that link one strand with the other. They do not predict intramolecular base pairs. Software to do this will be released in a later version of UNAFold. We currently recommend that hybridization simulations be limited to "short" sequences of lengths up to at most 100. It is also appropriate to allow a small sequence (probe) to hybridize to a much larger one (target) to assess the likelihood of binding at various sites.

No assumption is made about two sequences that are hybridized. They might be perfectly complementary, there may be some mismatches and or small bulges, or they may be totally unrelated.

3.5.1. Self-Hybridization of a Short Oligo

We selected the DNA oligomer, 5′-CGCAATTCGCG-3′, and stored in A.seq, a generic name for the first of up to two sequences. The command:

```
hybrid-min --NA DNA --tmin 20 --tmax 60 A.seq
    A.seq
```

computed single, mfe self-hybridizations at temperatures from 20°C to 60°C, inclusive. Only two distinct hybridizations were computed over the entire temperature range. They are displayed in Fig. 1.5.

3.5.2. Hybridization PDPs (Probability Dot Plots)

The apparent sudden switch from one structure to another is artificial. In reality, many conformations would exist simultaneously at each temperature. This phenomenon is easily modeled using the hybrid program. Stochastic samples at 20°C contain the two main hybridizations depicted in Fig. 1.5 and many more, most of which are slight variants. The same is true for a stochastic sample at 40°C, for example. What changes are the frequencies of the various structures. The ct-uniq program is a simple "standard input" "standard output" program that selects unique structures

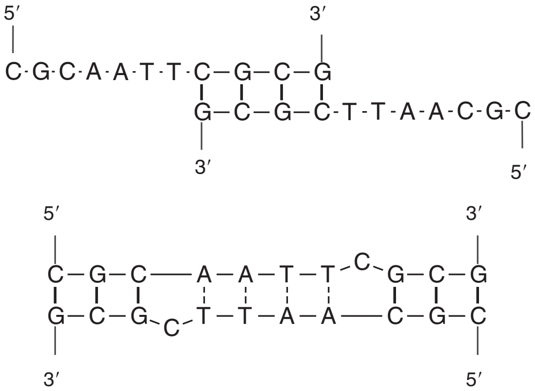

Fig. 1.5. Only two distinct mfe self hybridizations occur over a temperature range from 20°C to 60°C, inclusive. Standard DNA conditions were used (Na^+ = 1M, Mg^{2+} = 0). The bottom hybridization has the lowest mfe from 20°C to 28°C, inclusive. For temperatures 29°C to 60°C, the top hybridization has the lowest mfe. At 28°C, the free energies of the bottom and top hybridization are −6.43 and −6.32 kcal/mol, respectively. At 29°C, they are −6.18 and −6.22 kcal/mol, respectively.

from a multiple structure ct file. A stochastic sample of 100 structures computed at 30°C was reduced to 27 in this way, and only 24 in terms of distinct sets of base pairs, since structures with the same base pairs may differ by the addition or deletion of single-base stacking. Many of the structures are merely sub-structures of others. An effective way to view ensembles of hybridizations is to run the hybrid-plot program:

```
hybrid-plot A-A
```

and to view base pair probabilities as the temperature varies. It is easy to create PDP images at various temperatures by running the non-interactive version of the same program, hybrid-plot-ng. The command:

```
hybrid-plot-ng --temperature $T --colors linear\*
--grid 20 --cutoff 0.001 A-A
```

was run a number of times, with "$T" replaced by temperatures 15, 20, …. Because the sequence is so small, the grid display was turned off by setting the spacing to a number larger than the sequence length (--grid 20). The --colors linear flag chooses colors for the base pair dots that vary linearly with the probability. Using an "rgb" scale, the colors vary linearly with probability from (1,0,0) to (1,1,0) (red to yellow), from (1,1,0) to (0,1,0) (yellow to green), (0,1,0) to (0,1,1) (green to cyan), (0,1,1) to (0,0,1) (cyan to blue), and finally from (0,0,1) to (1,0,1) (blue to magenta). The --cutoff 0.001 flag is the cutoff. Base pairs with probabilities below the cutoff are not plotted. By default, UNAFold programs do not output probabilities $<10^{-6}$ into plot or ext files. Six of these PDP files were combined into **Fig. 1.6**.

3.6. Melting Profiles

The original motivation for creating the UNAFold software was to simulate entire melting profiles. In the most general case, two DNA or RNA molecules are interacting. In general, there may be competition between dimer formation and folding of the individual molecules. Self-hybridization may occur. Thus, competition among a total of five molecular species is computed. There is the heterodimer, two homodimers, and two folded species. When only one molecule is present, only dimer formation (a single homodimer) and folding are considered. Partition function computations are performed for each molecular species over a user-specified range of temperatures. Finally, results for individual species are combined to compute overall ensemble quantities: free energy, enthalpy, and entropy change (ΔG, ΔH, and ΔS, respectively), UV absorbance at 260 nm, heat capacity (C_p), and equilibrium molar concentrations of each of the species.

* Long command lines may be broken into multiple lines if a backslash, \, is the last character of all but the final line.

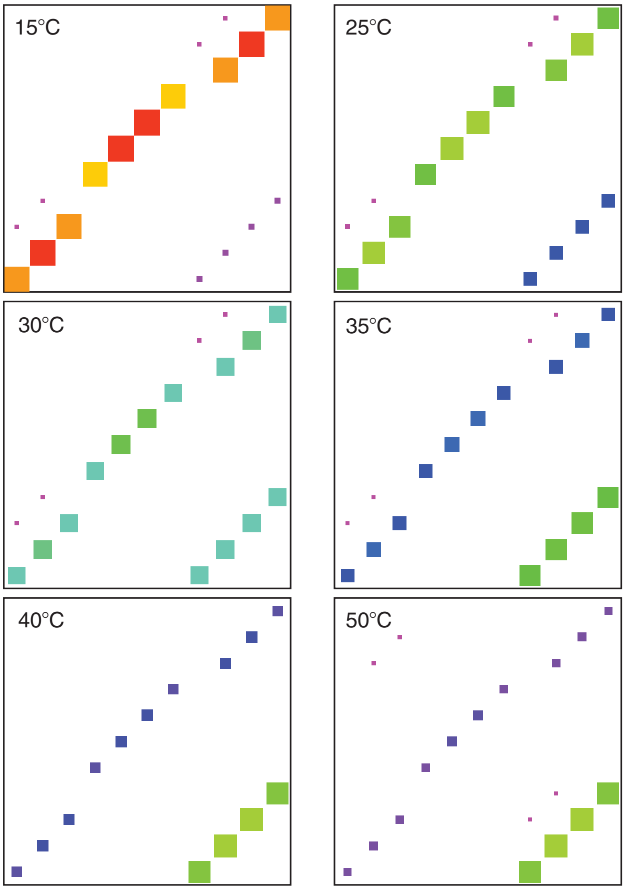

Fig. 1.6. Self-hybridization probability dot plots (PDPs) of the DNA oligomer, 5′-CGCAATTCGCG-3′ computed for [Na$^+$] = 1M, [Mg^{2+}] = 0 and for a variety of temperatures as shown in the plots. At 15°C, the large red and orange dots along the diagonal comprise the base pairs of the bottom structure from Fig. 1.5, the "low temperature" hybridization. These base pairs have probabilities that vary from 96.5% to 85%. The four small consecutive magenta dots *(lower right)* comprise the upper, high temperature hybridization from Fig. 1.5. These base pairs all have probabilities of 3%. At 25°C, the base pair probabilities of the low temperature hybridization have fallen (63–73%), whereas those of the high temperature hybridization have risen to about 24%. The base pair probabilities of the two main hybridizations are roughly equal at 30°C, and the trend continues. At 50°C, the base pair probabilities of the low temperature hybridization have fallen to the 7–9% range, whereas the four base pairs of the high temperature hybridization have probabilities in the 68–73% range.

During the development of the UNAFold software, our analyses of measured melting profiles indicated that an additional, internal energy term is required for each molecule in order to avoid a systematic underestimation of overall enthalpy change. An ad hoc rule was adopted based on observations. For each strand, the enthalpy change from a perfect hypothetical hybridization to its reverse complement to total dissociation is computed using the published enthalpies. By default, 5% of this enthalpy change is assigned to that strand as an internal energy term independent of folding or hybridization. It is useful to attribute this extra energy to base stacking in unfolded, single strands. Assuming a cooperative transition (melting) from "stacked single-stranded" to "unstacked random coil" that takes place at about 50°C, an entropy change can be computed. The default behavior of the UNAFold Perl scripts that compute ensemble quantities and melting profiles is to include this additional energy term. It is also used in the DINAMelt web server *(36)*.

Melting profiles were computed for the short oligomer, 5′-CGCAATTCGCG-3′ considered above. The command:

```
hybrid2.pl --NA DNA --tmin 10 --tmax 90 --sodium\
0.05 --A0 10e-6 A.seq A.seq
```

was used. The temperature range is from 10°C (--tmin 0) to 90°C (--tmax 90). The temperature increment is 1°C unless specified otherwise using the --tinc flag. --NA DNA defines the nucleic acid(s) as DNA. The default is RNA. --sodium 0.05 defines 50 mM Na$^+$ (or equivalent). The concentration of "A" (first sequence) is 10 µmol, as specified by --A0 10e-6. It is not necessary to specify another concentration in this case, but two sequences must be specified, even if they are the same. The molar concentration plot, shown in **Fig. 1.7**, reveals that no hybridization is taking place! The predicted melting profiles (not shown) are simulating the melting of a single-stranded, folded 11-mer. "What about the PDPs computed for A-A self hybridization?" The answer is that the probabilities in any of these plots are conditional. For hybridization, they are conditional on hybridization; for folding, they are conditional on

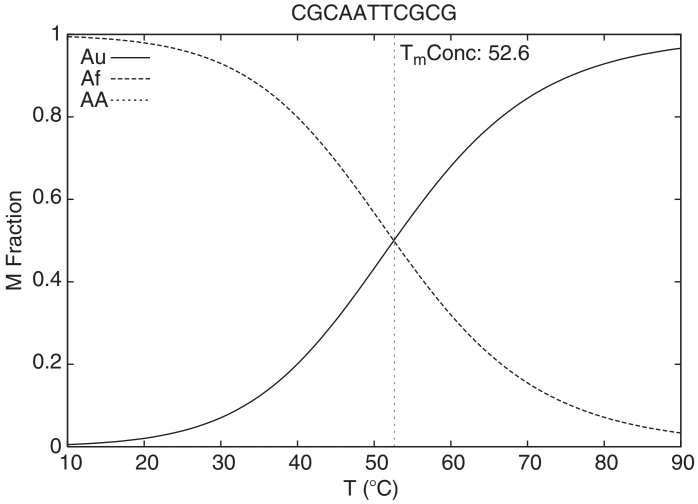

Fig. 1.7. No dimer is present in this plot. The only transition is from folded to unfolded.

folding. Thus, if a base pair has a 50% probability in an A-A PDP, but the equilibrium constant for dimer formation is 10^{-6}, the effective probability is $5 \times 10{-}7$. In fact, by increasing Na^+ to 1M and [A] to 500 µM, simulation predicts a mole fraction of almost 0.2 for the dimer at 10°C.

A naïve user might be tempted to suppress single-stranded folding in this simulation. Re-running the hybridization script with the folding of A excluded:

```
hybrid2.pl --NA DNA --tmin 10 --tmax 90 --sodium\
0.05 --A0 10e-6 --exclude A --reuse A.seq A.seq
```

yields results that predict a dimer mole fraction of just over 60% at 10°C. It is rapidly decreasing with heat capacity and UV absorbance melting temperatures of about 13°C and concentration melting temperature of 16.5°C. These results are nonsense. Even more dangerous would be the unthinking use of the simple melt.pl script that computes a mfe structure at a specified temperature and uses the simple formula given in Equation [1]. The command:

```
melt.pl --NA DNA --sodium 0.05 --temperature $T\
--Ct 10e-6 A.seq A.seq
```

gives the same predicted T_m, 14.0°C, for \$T = 20°C, 50°C, and for the default value of 37°C. This, too, is nonsense. The added "danger" is that folding and homodimer formation are automatically excluded when this script is used. When hybrid2.pl is used, any excluded species must be specified by the user. In general, the melt.pl script is valid only when it may be safely assumed that the heterodimer (when two distinct sequences hybridize) or the homodimer (single sequence) is the only molecular species that forms, that it melts cooperatively, and that strand concentrations are equal when two different molecules hybridize.

3.6.1. Simulation vs. Measurements

When measured data are available, the hybrid2.pl script includes measured profiles with predicted ones. Heat capacity measurements using DSC (differential scanning calorimetry) require relatively high concentrations of nucleic acid. Total strand concentration of 100 to 200 µM is usual. For UV absorbance measurements, total strand concentration is reduced by a factor of 10 (roughly). For this reason, if heat capacity measurements are available, then UV absorbance data will not be available, and vice versa. hybrid2.pl expects observed (measured) heat capacities to be in a .obs.Cp file. Likewise, observed absorbance data should be in a .obs.ext file. The format is simple. Each line (record) contains a temperature and a heat capacity (or absorbance) separated by spaces and/or a tab character. Additional columns are ignored. hybrid2.pl expects a .obs.Tm file containing the measured T_m and heat capacity (or absorbance) at that temperature.

Plots created by hybrid2.pl use the prefix formed from the two sequence file names, as explained above. The PostScript plot files have double suffixes. Below is a list of these suffixes and a description of the plots.

conc.ps: The concentrations of all non-excluded molecular species are plotted *versus* temperature. The concentrations are expressed as mole fractions. The values come from the .conc file, where the units are moles.

Cp.ps A plot of ensemble heat capacity *versus* temperature. Values are from the .ens.Cp file and the plotted *Tm* is from the .ens.TmCp file.

Cp2.ps Not for general use. The contributions from the different species are included in this additional heat capacity plot. Negative values indicate energy transfer from a species to others. The net sum must be positive.

ext.ps A plot of ensemble absorbance of UV radiation at 260 nm vs. temperature. ("ext" stands for "extinction.") Values are from the .ens.ext file. The .ens. MaxExt contains the maximum possible absorbance assuming that no base pairs exist. The plotted T_{m} is from the .ens.TmExt2 file. This file contains T_{m} computed as the absorbance midpoint. It is the temperature at which absorbance is midway between the minimum value plotted and the theoretical maximum contained in the .ens.MaxExt file. It is the UNAFold default. The .ens.TmExt1 file contains an absorbance T_m computed as the inflection point in the absorbance curve. Multiple values might exist.

ext2.ps Not for general use. The contributions from the different species are included in this additional absorbance plot. (The "ext2" in the suffix has nothing to do with the definition of absorbance. It is unrelated to the .ens.TmExt1 and .ens.TmExt2 files.

obs.ps This plot contains both the computed heat capacity (left vertical axis) and the computed absorbance (right vertical axis). In addition, measured heat capacity or absorbance is plotted, together with the measured T_{m}.

Figure 1.8 displays the extra plot that is generated for the melting of two complementary DNA 25-mers when a measured heat capacity file is also present. The command that generated this plot is:

```
hybrid2.pl --NA DNA --tmin 25 --tinc 0.2 --tmax\
100 --A0 9e-05 --B0 9e-05 --sodium 0.069
R948.seq R949.seq
```

Total strand concentration is 180 μM; with both strands having the same molar concentration. $[\mathrm{Na}^+]$ = 69 mM. The --tinc 0.2 means that computations are performed for all temperatures between 25°C and 100°C in increments of 0.2°C. It quintuples the

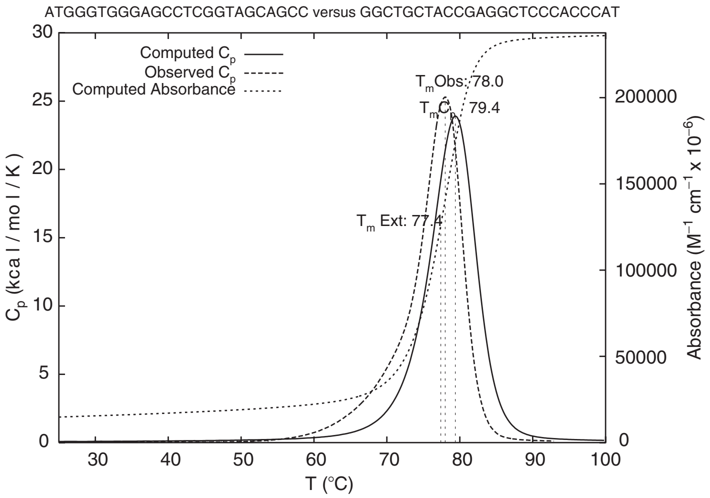

Fig. 1.8. A plot of computed heat capacity, measured heat capacity and computed absorbance vs. temperature for two complementary DNA 25-mers with [Na$^+$] = 69 mM. Both strand concentrations are 90 μM. The computed T_m (C_p) is 1.4°C greater than the measured one. The absorbance plot is "extra." There is no reason to expect T_m (Ext) to match T_m Obs, although the two are very close in this case.

computation time with respect to the default, but it creates a superior computed C_p curve, which is generated by taking the second derivative of the Gibbs free energy profile with respect to temperature. The default C_p plots are "fatter" and have lower peak values. This is not a serious problem, since the computed values for ΔH, ΔS, and T_m are robust with respect to the temperature increment. In these computations, the homodimers may be safely excluded in reruns, since initial computations reveal that homodimer concentrations are negligible over the entire temperature range.

Measured absorbances are available for these complementary 25-mers, but at [Na$^+$] = 1 M and with a total strand concentration of 2.25 μM, which is down by a factor of 80 from that used in the heat capacity measurements.

3.6.2. A Molecular Beacon Example

Molecular beacons *(37)* take advantage of the competition between folding and hybridization to distinguish between a gene and a mutant containing a single nucleotide polymorphism (SNP). A molecular beacon is a DNA oligomer with a fluorophore attached at the 5′ end and with "DABCYL" (a quencher) attached at the 3′ end. The first 5–7 bases are complementary to the last 5–7, so that a stem-loop structure can form that brings the fluorophore close to the quencher. In this conformation, any fluorescence generated by the beacon is immediately quenched. When the hairpin does not form, quenching does not occur and a signal may be detected. The middle portion of a beacon varies from 10 to 40 nt, and is complementary to a region of interest, called the Target. In the simulation we have chosen, the sequence of the molecular beacon, stored in Beacon.seq is:

5′-gcgagcTAGGAAACACCAAAGATGATATTTgctcgc-3′

The complementary bases that form a stem-loop structure are underlined and in lower case. The center 24 bases are complementary to a stretch of 24 nucleotides within a (much) larger DNA molecule, such as a gene. In the simulation, only the complementary region is considered. This defines the Target, stored as Target.seq. It is:

5′-AAATATCATCTtTGGTGTTTCCTA-3′

The reason for selecting this target is that a mutant gene contains a SNP within it. This defines a mutant Target, or TargetSNP, stored as TargetSNP.seq. It is:

5′-AAATATCATCTcTGGTGTTTCCTA-3′

The SNP, a T to C transition, is easy to spot, since both are lowercase. The whole point of molecular beacons is that at some temperature, ionic conditions, and strand concentrations, all beacons should hybridize to the target, whereas hybridization to the SNP should be significantly lower, and the single-stranded beacons should be folded. Preliminary runs indicated that folding and self-hybridization of the targets could be excluded. We did not bother to exclude self-hybridization of beacons. Beacon concentration was set at 50 nM and the target concentrations were twice that; 100 nM. This two-fold excess of target ensures maximum hybridization of the beacons.

The commands:

```
hybrid2.pl --tmin 10 --tmax 90 --NA DNA --sodium\
0.05 magnesium 0.0035 --exclude B --exclude BB --\
A0 5e-8 --B0 1e-7 Beacon.seq Target.seq
```

```
hybrid2.pl --tmin 10 --tmax 90 --NA DNA --sodium\
0.05 --magnesium 0.0035 --exclude B --exclude BB\
--A0 5e-8 --B0 1e-7 Beacon.seq TargetSNP.seq
```

produced the usual large number of files. Only the species concentration plots, and especially the conc files, are of interest. Figure 1.9 shows the relevant concentrations for the two simulations. It contains parts of the two concentration plot files produced automatically by hybrid2.pl. The melting temperature of the dimer falls by 5.4°C. It is desirable that single-stranded beacons be folded. Unfolded beacons give a false signal. The optimal experimental temperature is predicted to be 58°C. At this temperature, the Beacon-Target concentration is 83% of its maximum. Single-stranded beacons are mostly folded, but the unfolded ones contribute a small false signal, bringing the total to 86% of maximum. For the Beacon-TargetSNP hybridization, the signal

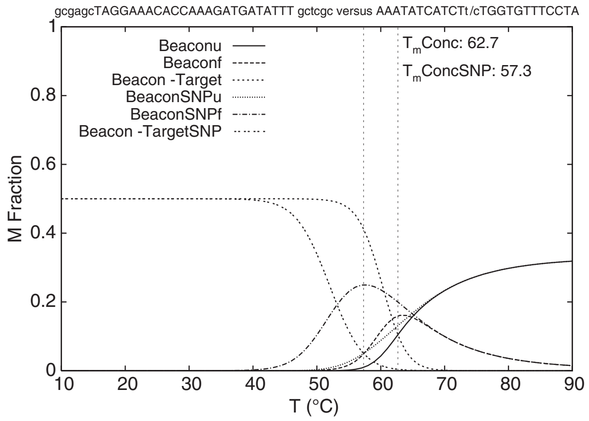

Fig. 1.9. Predicted Beacon-Target and Beacon-TargetSNP concentration plots. The suffix "u" denotes "unfolded" and "f" denotes "folded". BeaconSNP refers to Beacon concentrations in the presence of TargetSNP. As the Beacon-Target dimer melts, single-stranded Beacons are mostly folded vs. unfolded. The Beacon-TargetSNP dimer melts at a lower temperature. As it does, the released Beacons are overwhelmingly folded. However, by 58°C, the false signal from unfolded beacons, although small, has just begun to exceed the rapidly decreasing signal from the dimer.

is down to just 11% of its maximum, but enough of the released beacons unfold to bring the signal up to 28% of maximum. The bottom line is that at 58°C, the fluorescence in the SNP case is one-third of what it would be otherwise.

4. Notes

The examples presented for how some of the UNAFold programs and scripts might be used are just a small sample of possible applications and ways to combine various programs. Users are encouraged to write their own scripts or modify existing ones to better suit their needs. We believe that users without programming skills will find the package easy to use. As with mfold, feedback from users is expected to influence what new features might be added.

Some directions for further development are already clear.

- The current hybrid2.pl script is too inflexible when plots are created. Initial strand concentrations that differ by orders of magnitude can make concentration plots useless unless logarithmic scaling is used for plotting mole fractions.
- The current UNAFold.pl script behaves like the mfold script in the mfold package. It acts on a single sequence, and predicts a sample of foldings together with EDP (energy dot plot). When auxiliary software is installed (mfold_util), text files of structures and EDPs may be used to create plots. The

UNAFold.pl script will be extended to predict (stochastic) samples of structures together with PDPs.

- The internal energy option for single-stranded, unfolded sequences adds enthalpy using a simple ad hoc rule derived from a group of measured DSC profiles. This model needs to be studied and improved so that the added enthalpies can be assigned more objectively.
- The intra-molecular hybridization software requires testing to determine its usefulness when larger oligos are hybridized.
- We now realize that for individual molecular species, enthalpy and heat capacity can be computed directly from stochastic samples. How to extend these species predictions to ensemble predictions remains a bit unclear, but we are confident that it can be done in a numerically stable way. Such calculations would make it unnecessary to compute derivatives numerically.

We believe that continuing feedback from users will have a positive effect on further development.

Acknowledgments

This work was supported, in part, by grants GM54250 and GM068564 from the National Institutes of Health and by a graduate fellowship to N.R.M. from RPI.

References

1. Bellman, R. E. (1957) *Dynamic Programming*. Princeton University Press, Princeton, NJ.
2. Waterman, M. S., Smith, T. F. (1978) RNA secondary structure: a complete mathematical analysis. *Math Biosci* 42, 257–266.
3. Waterman, M. S. (1978) Secondary structure of single-stranded nucleic acids, in (Rota, G.-C., ed.), *Studies in Foundations and Combinatorics*, Academic Press, New York.
4. Nussinov, R. (1980) Some rules for ordering nucleotides in DNA. *Nucleic Acids Res* 8, 4545–4562.
5. Zuker, M., Stiegler, P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res* 9, 133–148.
6. Sankoff, D., Kruskal, J. B., eds. (1983) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA.
7. Zuker, M. (1989) The use of dynamic programming algorithms in RNA secondary structure prediction, in (Waterman, M. S., ed.), *Mathematical Methods for DNA Sequences*. CRC Press, Boca Raton, FL.
8. Zuker, M. (1994) Prediction of RNA secondary structure by energy minimization, in (Griffin, A. M., Griffin, H. G., eds.), *Computer Analysis of Sequence Data*,. Humana Press, Totowa, NJ.
9. Zuker, M., Mathews, D. H., Turner, D. H. (1999) Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide, in (Barciszewski J., Clark, B. F. C., eds.), *RNA Biochemistry and Biotechnology*. Kluwer, Dordrecht.
10. Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res* 31, 3406–3415.
11. Hofacker, I. L., Fontana, W., Stadler, P. F., et al. (1994) Fast folding and compari-

son of RNA secondary structures. *Monatsh Chem* 125, 167–188.

12. Wuchty, S., Fontana, W., Hofacker, I. L., et al. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers* 49, 145–165.
13. Hofacker, I. L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res* 31, 3429–3431.
14. McCaskill, J. S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* 29, 1105–1119.
15. Ding, Y., Lawrence, C. E. (2001) Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. *Nucleic Acids Res* 29, 1034–1046.
16. Ding, Y., Lawrence, C. E. (2003) A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res* 31, 7280–7301.
17. Ding, Y., Lawrence, C. E. (2004) Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Res* 32, W135–W141.
18. Allawi, H. A., SantaLucia, J. Jr. (1997) Thermodynamics and NMR of internal G·T mismatches in DNA. *Biochemistry* 36, 10581–10594.
19. SantaLucia, J. Jr. (1998) A unified view of polymer, dumbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc Natl Acad Sci U S A* 95, 1460–1465.
20. SantaLucia, J. Jr., Hicks, D. (2004) The thermodynamics of DNA structural motifs. *Annu Rev Biophys Biom* 33, 415–440.
21. Zhang, Y., Hammer, D. A., Graves, D. J. (2005) Competitive hybridization kinetics reveals unexpected behavior patterns. *Biophys J* 89, 2950–2959.
22. Kilgard, M. J. (1996) *OpenGL Programming for the X Window System.* Addison-Wesley, Boston.
23. Puglisi, J. D., Tinoco, I. Jr. (1989) Absorbance melting curves of RNA, in (Dahlberg, J. E., Abelson, J. N. eds.), *RNA Processing Part A: General Methods.* Academic Press, New York.
24. Waugh, A., Gendron, P., Altman, R., et al. (2002) RNAML: a standard syntax for exchanging RNA information. *RNA* 8, 707–717.
25. Jelesarov, I., Bosshard, H. R. (1999) Isothermal titration calorimetry and differential scanning calorimetry as complementary tools to investigate the energetics of biomolecular recognition. *J Mol Recog* 12, 3–18.
26. Walter, A. E., Turner, D. H., Kim, J., et al. (1994) Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA. *Proc Natl Acad Sci U S A* 91, 9218–9222.
27. Mathews, D. H., Sabina, J., Zuker, M., et al. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J Mol Biol* 288:911–940.
28. De Rijk, P., De Wachter, R. (1997) RnaViz2, a program for the visualisation of RNA secondary structure. *Nucleic Acids Res* 25, 4679–4684.
29. De Rijk, P., De Wachter, R. (2003) RnaViz2: an improved representation of RNA secondary structure. *Bioinformatics* 19, 299–300.
30. Dai, L. Zimmerly, S. (2002) Compilation and analysis of group II intron insertions in bacterial genomes: evidence for retroelement behavior. *Nucleic Acids Res* 30, 1091–1102.
31. Toor, N., Hausner, G., Zimmerly, S. (2001) Coevolution of group II intron RNA structures with their intron-encoded reverse transcriptases. *RNA* 7, 1142–1152.
32. Michel, F., Umesono, K., Ozeki, H. (1989) Comparative and functional anatomy of group II catalytic introns: a review. *Gene* 82, 5–30.
33. Zuker, M., Jacobson, A. B. (1998) Using reliability information to annotate RNA secondary structures. *RNA* 4, 669–679.
34. Jacobson, A. B., Arora, R., Zuker, M., et al. (1998) Structural plasticity in RNA and its role in the regulation of protein translation in coliphage qβ. *J Mol Biol* 275, 589–600.
35. Mathews, D. H. (2004) Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA* 10, 1174–1177.
36. Markham, N. R., Zuker, M. (2005) DINAMelt web server for nucleic acid melting prediction. *Nucleic Acids Res* 33, W577–W581.
37. Tyagi, S., Kramer, F. R. (1996) Molecular beacons: probes that fluoresce upon hybridization. *Nat Biotechnol* 4, 303–308.