10

RNA structure analysis

Many interesting RNAs conserve a secondary structure of base-pairing interactions more than they conserve their sequence. This makes RNA sequence analysis more complicated and difficult than protein or DNA sequence analysis. RNA secondary structure problems are a natural application for probabilistic models based on the stochastic context-free grammars introduced in Chapter 9. In this chapter, we will examine two RNA analysis problems of biological interest.

The first problem is RNA secondary structure prediction for a single sequence. We will outline two well-known dynamic programming algorithms for RNA secondary structure prediction, the Nussinov and the Zuker algorithms. Then we will use RNA secondary structure prediction as an introductory example for the use of SCFGs for RNA analysis, by developing a small SCFG that implements a probabilistic version of the Nussinov algorithm.

The second is a related set of problems, having to do with the analysis of multiple alignments of families of related RNAs. Like Chapter 5, where profile HMMs were used for both multiple alignment and for database searching, we develop RNA structure profiles called 'covariance models' (CMs) for dealing with RNA multiple alignments with secondary structure constraints included. Covariance models are used for both RNA multiple alignment and database searches. Consensus structure prediction from RNA multiple alignments, a process called comparative RNA sequence analysis, is also somewhat automated by RNA covariance model training algorithms.

As you read this chapter, bear in mind that SCFG-based RNA analysis methods are not widely known or used. All of the SCFG methods we describe are in their infancy and have considerable problems with computational complexity. Improved SCFG methods for RNA analysis might be around the corner. Here, we try to give the fundamentals of SCFG-based probabilistic methods for RNA analysis without getting mired in details that may soon change. At the least, RNA SCFGs provide us with a pedagogical counterpoint to profile HMMs. We will see how much of the same probabilistic machinery developed for HMMs also applies to a different and more complex class of model.

10.1 RNA

To many people, RNA is merely the passive intermediary messenger between DNA genes and the protein translation machinery. Messenger RNA is often described as a linear, unstructured sequence, uninteresting but for the protein amino acid sequence that it encodes. However, many non-coding RNAs exist which adopt sophisticated three-dimensional structures, and some even catalyse biochemical reactions. Since the startling discovery of catalytic RNAs in the early 1980s [Cech & Bass 1986], a number of interesting new structural and catalytic RNAs have been discovered. More recently, novel RNAs have been invented using *in vitro* evolution technologies to screen repertoires of random RNA sequences for new catalysts and new specific ligands [Gold *et al.* 1995].

The discovery of RNA catalysis revived a notion now widely known as the 'RNA world' hypothesis for the origin of life [Gilbert 1986; Gesteland & Atkins 1993]. The RNA world hypothesis posits a primordial world before DNA genomes and protein catalysts when RNA genomes were replicated by RNA catalysts. It is sometimes argued that many modern structural and catalytic RNAs are 'molecular fossils' that have been handed down in evolutionary time from an extinct RNA world.

Structural and catalytic RNAs are also important in the molecular biology of modern organisms. The peptidyl transferase activity of ribosomes is thought to be catalysed by ribosomal RNA [Noller, Hoffarth & Zimniak 1992]. RNA splicing (removal of introns from eukaryotic pre-mRNA transcripts) is catalysed by a complex RNA/protein machine (the spliceosome) which contains five major species of small nuclear RNAs [Baserga & Steitz 1993]. The signal recognition particle that is involved in translocating proteins across the plasma membrane is an RNA/protein complex [Larsen & Zwieb 1993]. Proper ribosomal RNA processing and modification require a host of small nucleolar RNAs [Maxwell & Fournier 1995]. In messenger RNA transcripts, RNA structure (particularly in 5′ and 3′ untranslated regions) is used in a variety of ways to effect post-transcriptional genetic regulation. Known post-transcriptional regulatory mechanisms include alternative mRNA splicing control [McKeown 1992], modulation of translational efficiency [Melefors & Hentze 1993] and regulation of mRNA stability [Peltz & Jacobson 1992].

Terminology of RNA secondary structure

RNA is a polymer of four different nucleotide subunits. The four nucleotides are abbreviated A, C, G and U, for adenine, cytosine, guanine and uracil. In DNA, thymine (T) replaces uracil.

G-C and A-U form hydrogen bonded base pairs and are said to be complementary. G-C pairs form three hydrogen bonds and tend to be more stable than

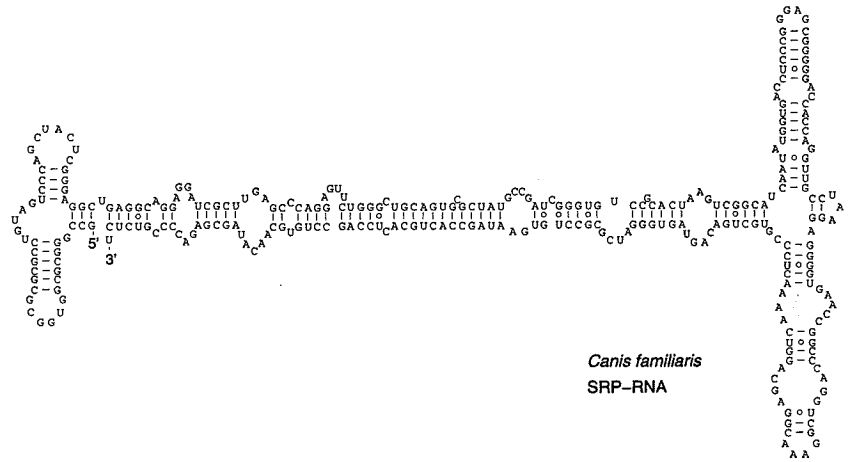

Figure 10.1 *The RNA secondary structure of signal recognition particle (SRP) RNA from the dog,* Canis familiaris.

A-U pairs, which form only two. Base pairs are approximately coplanar and are almost always *stacked* onto other base pairs in an RNA structure. Contiguous stacked base pairs are called *stems*. In three-dimensional space, RNA stems generally form a regular (A-form) double helix. Unlike DNA, RNA is typically produced as a single stranded molecule which then folds intramolecularly to form a number of short base-paired stems. This base-paired structure is called the *secondary structure* of the RNA. RNA secondary structures are typically represented by two-dimensional pictures like the one shown in Figure 10.1.

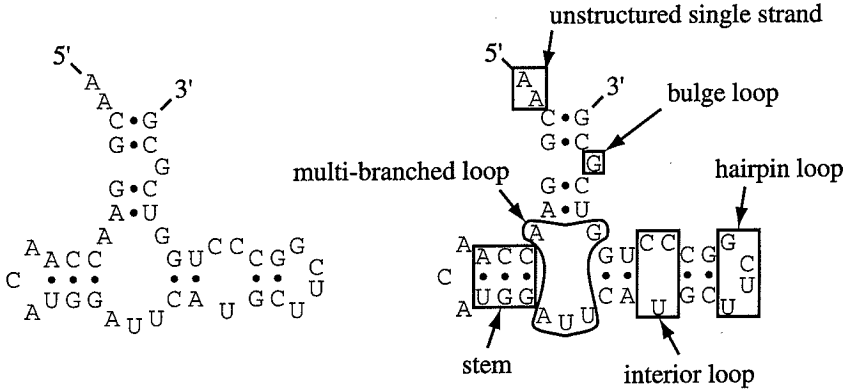

Figure 10.2 *The fundamental elements of RNA secondary structure are indicated for a hypothetical example.*

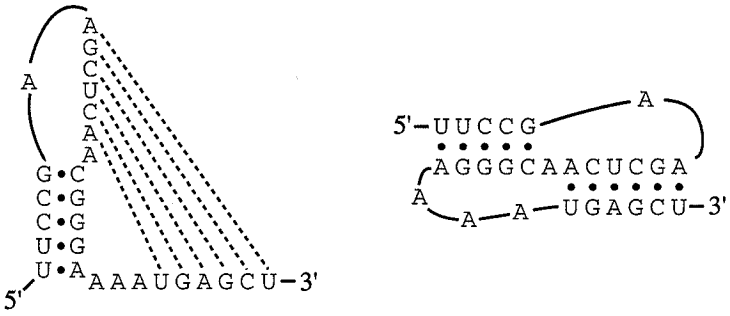

Figure 10.3 *Base pairs between a loop and positions outside the enclosing stem are called a pseudoknot (left). Another representation of the same pseudoknot is shown on the right. In three-dimensional space, the two stems can stack coaxially and mimic a contiguous A-form helix. This particular example is an artificially selected RNA inhibitor of the human immunodeficiency virus reverse transcriptase [Tuerk, MacDougal & Gold 1992].*

The elements of an RNA secondary structure are named as shown in Figure 10.2. Single stranded subsequences bounded by base pairs are called *loops*. A loop at the end of a stem is called a *hairpin loop*. Simple substructures consisting of a simple stem and loop are called *stem loops* or *hairpins* (because the structure resembles a hairpin when drawn). Single stranded bases occurring within a stem are called a *bulge* or *bulge loop* if the single stranded bases are on only one side of the stem, or an *interior loop* if there are single stranded bases interrupting both sides of a stem. Finally, there are *multi-branched loops* from which three or more stems radiate.

In addition to canonical A-U and G-C base pairs, non-canonical pairs also occur in RNA secondary structure. The most common non-canonical pair is the G-U pair, which is almost as thermodynamically favourable as Watson–Crick pairs. Other pairs form as well. Non-canonical pairs distort regular A-form RNA helices. These distortions seem to be a favoured target of proteins specialised for recognising RNA.

Base pairs almost always occur in a nested fashion in RNA secondary structure. Informally, this means that if we draw arcs over an RNA sequence connecting the base pairs, none of the arcs need to cross each other. More formally, a base pair between positions i and j and a base pair between positions i' and j' are nested if and only if $i < i' < j' < j$ or $i' < i < j < j'$. (Recall that this is the condition met by the constraints on palindrome languages in Chapter 9 – this is why context-free grammars apply to RNA secondary structure.) When non-nested base pairs occur, they are called *pseudoknots*. An example of a pseudoknot is given in Figure 10.3.

None of the dynamic programming algorithms that we describe can deal with pseudoknots, including the Zuker and Nussinov RNA folding algorithms as well

as SCFG algorithms. We saw in the previous chapter that describing the crossing interactions of pseudoknots in full generality would require context-sensitive grammars. Since pseudoknots occur in many important RNAs, we are ignoring biologically important information. Fortunately, the total *number* of pseudoknotted base pairs is typically small compared to the number of base pairs in nested secondary structure. For example, one authoritative secondary structure model of *E. coli* SSU rRNA indicates 447 Watson–Crick and G–U base pairs supported by comparative sequence analysis, only eight of which are in non-nested pseudoknot interactions [Gutell 1993]. For many purposes, including database searching for RNA homologues, it is usually acceptable to sacrifice the information in pseudoknots in return for efficient dynamic programming algorithms. For other purposes such as three-dimensional structure prediction, pseudoknots must be considered and the same sacrifice cannot be made.

RNA sequence evolution is constrained by structure

It is relatively common to find examples of homologous RNAs that have a common secondary structure without sharing significant sequence similarity. Drastic changes in sequence can often be tolerated as long as compensatory mutations maintain base-pairing complementarity. It would be advantageous to be able to search for conserved secondary structure in addition to conserved sequence when searching databases for homologous RNAs.

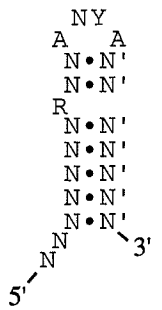

Figure 10.4 *The consensus binding site for R17 phage coat protein.* N, Y *and* R *are standard 'degenerate' symbols for multiple possible nucleotides.* N *indicates {*A,C,G,U*}*, Y *indicates {*C,U*} and* R *indicates {*A,G*}.* N' *indicates a complementary base pairing to* N.

The structure shown in Figure 10.4 is the consensus RNA binding site for the coat protein of the bacterial RNA virus R17 [Witherell, Gott & Uhlenbeck 1991]. R17 coat protein binds this site and represses translation of its replicase as part of the normal timing of an R17 lytic cycle. Only four primary sequence positions are specified in the consensus, and two of them are degenerate. If we were interested

in searching a nucleotide sequence for occurrences of consensus R17 coat prote binding sites, it would be useless to use a standard sequence alignment method

How useless? It is instructive to extract some rules of thumb from Shann information theory. In information theoretic terms, a consensus base pair conve as much information as a conserved base. The information (relative entrop contributed by a completely conserved base ($p_x = 1$) is $\sum_x p_x \log_2 \frac{p_x}{f_x} = 2$ bi (assuming equiprobable initial expected base frequencies, $f_x = \frac{1}{4}$). Similarl the degenerate R and Y in Figure 10.4 each convey 1 bit of information, and tl N is worth 0. The information contributed by a Watson–Crick base pair of a sequence is also 2 bits, since $\sum_x \sum_y p_{xy} \log_2 \frac{p_{xy}}{f_{xy}} = 2$ (again assuming that o initial expectation is equiprobable, $f_{xy} = \frac{1}{16}$, and that the observed Watson–Cri pairs occur equiprobably, $p_{AU} = p_{CG} = p_{GC} = p_{UA} = \frac{1}{4}$).

Considering only primary sequence conservation, the R17 consensus therefo conveys 6 bits of information. We expect to find a match to it by chance eve 64 (2^6) nucleotides. Adding the seven base pairs to the consensus descriptic adds 14 bits of information, bringing the information content up to 20 bits, a reducing the chance of finding a spurious match to once in every million (2^2 nucleotides. If we search for NNN NNN NRN NAN YAN NNN NNN in the genome the related bacteriophage MS2 (GENBANK MS2CG; the R17 genome is not in t database), we find 38 matches in the 3569 bp genome, 37 of which are spuriou If we repeat the search while requiring the seven base pairs, we find just a sing match at the authentic coat protein binding site.

The above search was done with an RNA pattern-matching program simil to the program RNAMOT [Gautheret, Major & Cedergren 1990]. The progra searches for deterministic (non-stochastic) motifs but with secondary structu constraints as extra terms. It works fine for small, well-defined patterns but somewhat insensitive and problematic for finding matches to less well conserve structures. Currently, the prevailing wisdom for more sensitive, more statist cally based RNA database searches is that one must write a carefully customise program for each RNA structure of interest [Dandekar & Hentze 1995]. Sev eral such programs exist for finding transfer RNA genes [Fichant & Burks 1991 Pavesi *et al.* 1994; Lowe & Eddy 1997], and one exists for finding catalytic grou I introns [Lisacek, Diaz & Michel 1994]. However, as the number of different in teresting RNAs grows, this is an increasingly unsatisfactory state of affairs.

Inferring structure by comparative sequence analysis

The same base-pair induced sequence constraints that make database searchin hard make consensus RNA secondary structure prediction relatively easy – rel ative to protein structure prediction, at least. In a structurally correct multipl alignment of RNAs, conserved base pairs are often revealed by the presence o

frequent correlated compensatory mutations. Despite being a theoretical structure prediction method, RNA secondary structure prediction by this process of *comparative sequence analysis* is considered to be the most reliable means of determining an RNA secondary structure, short of solving a three-dimensional crystal or NMR structure. The accepted consensus structures of most well-studied RNAs have been derived by comparative analysis [Woese & Pace 1993] (Figure 10.5).

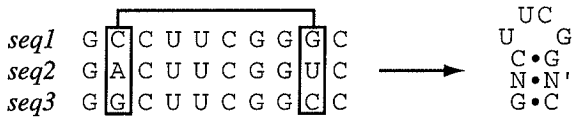

Figure 10.5 *Comparative sequence analysis recognises that the two boxed positions in this example of a multiple alignment (left) are covarying to maintain Watson–Crick complementarity. This covariation implies a base pair, leading to a consensus secondary structure prediction (right).*

Comparative analysis is a painstaking art. Inferring the correct structure by comparative analysis requires knowing a structurally correct multiple alignment, but inferring a structurally correct multiple alignment requires knowing the correct structure. A structure is 'solved' by an iterative refinement process of guessing the structure based on the current best guess of the multiple alignment, then realigning based on the new guess at the structure. The sequences to be compared must be sufficiently similar that they can be initially aligned by primary sequence identity alone to start the process, but they must be sufficiently dissimilar that a number of covarying substitutions can be detected.

A quantitative measure of pairwise sequence covariation comes from information theory [Chiu & Kolodziejczak 1991; Gutell *et al.* 1992]. The mutual information M_{ij} between two aligned columns i and j is given by

$$M_{ij} = \sum_{x_i,x_j} f_{x_i x_j} \log_2 \frac{f_{x_i x_j}}{f_{x_i} f_{x_j}}. \qquad (10.1)$$

f_{x_i} is the frequency of one of the four bases (A, C, G, U) observed in column i. $f_{x_i x_j}$ is the joint (pairwise) frequency of one of the sixteen possible base pairs observed in columns i and j. M_{ij} measures how much the joint frequency distribution deviates from the distribution that is expected if the two columns vary independently. For the four-letter RNA alphabet, M_{ij} varies between 0 and 2 bits. M_{ij} is maximal if i and j individually appear completely random ($f_i = f_j = 0.25$), but i and j are perfectly correlated, for instance in a Watson–Crick base pair.

Intuitively, M_{ij} tells us how much information we get about the identity of the residue in one position if we are told the identity of the residue in the other position. In the case of a base pair with no sequence constraints, we get 2 bits

of information: for instance, if we are told that i is a G, our uncertainty about j collapses from four possibilities to just one (C) so we gain 2 bits of information. If i and j are uncorrelated, the mutual information is zero. If either i or j are highly conserved positions, we also get little or no mutual information: if a position does not vary, we do not learn anything more about it by knowing the identity of its partner.

Figure 10.6 shows a contour plot of M_{ij} values calculated from a multiple alignment of 1415 tRNA sequences. The four base-paired stems of the cloverleaf structure are readily apparent. The D and TψCG stems, which are relatively highly conserved in primary sequence, are somewhat less apparent than the anticodon and acceptor stems which are extremely variable in primary sequence.

Exercise

10.1 The mutual information calculation in (10.1) requires counting frequencies of all sixteen different base pairs. This has the advantage that it makes no assumptions about Watson–Crick base pairing, so mutual information can be detected between covarying non-canonical pairs like A-A and G-G pairs. On the other hand, the calculation requires a large number of aligned sequences to obtain reasonable frequencies for sixteen possibilities. Write down an alternative information theoretic measure of base-pairing correlation that considers only two classes of i, j identities instead of all sixteen: Watson–Crick and G-U pairs grouped in one class, and all other pairs grouped in the other. Compare the properties of this calculation to the M_{ij} calculation both for small numbers of sequences and in the limit of infinite data.

10.2 RNA secondary structure prediction

Suppose we wish to predict the secondary structure of a single RNA. Many plausible secondary structures can be drawn for a sequence. The number increases exponentially with sequence length. An RNA only 200 bases long has over 10^{50} possible base-paired structures. We must distinguish the biologically correct structure from all the incorrect structures. We need both a function that assigns the correct structure the highest score, and an algorithm for evaluating the scores of all possible structures.

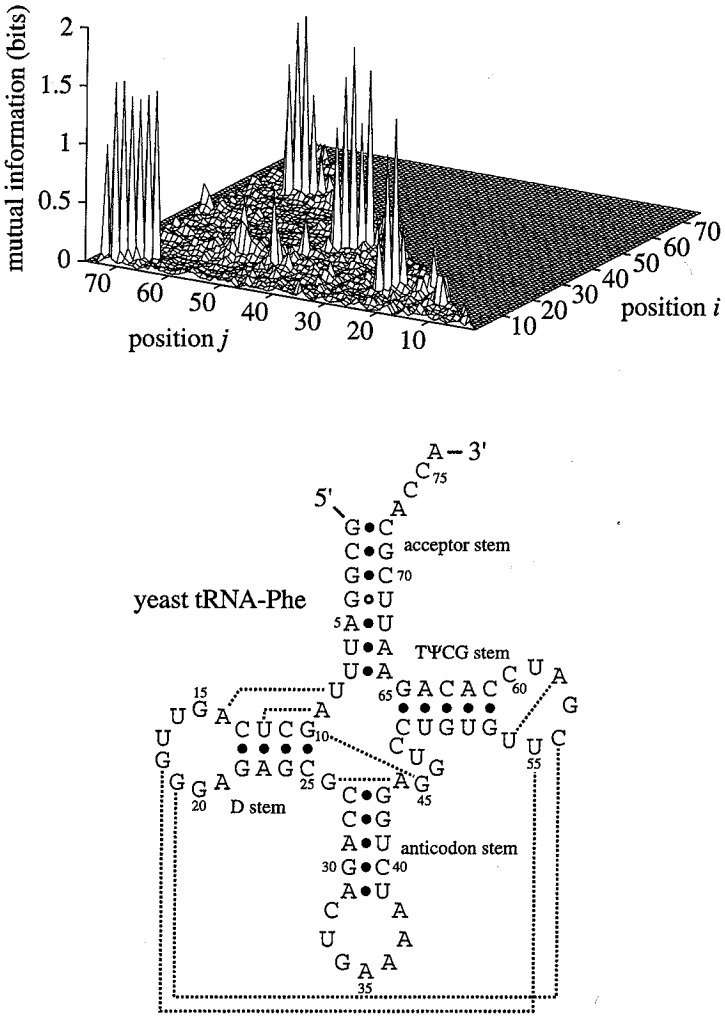

Figure 10.6 *A mutual information plot of a tRNA alignment (top) shows four strong diagonals of covarying positions, corresponding to the four stems of the tRNA cloverleaf structure (bottom; the secondary structure of yeast phenylalanine tRNA is shown). Dashed lines indicate some of the additional tertiary contacts observed in the yeast tRNA-Phe crystal structure. Some of these tertiary contacts produce correlated pairs which can be seen weakly in the mutual information plot.*

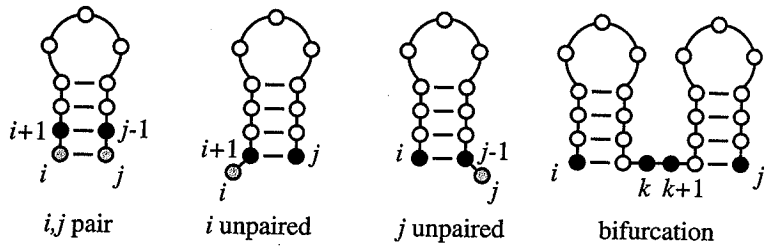

Figure 10.7 *The Nussinov algorithm looks at four ways in which the best RNA structure for a subsequence i,j can be made by adding i and/or j onto already calculated optimal structures for smaller subsequences. Pseudoknots are not considered.*

Base pair maximisation and the Nussinov folding algorithm

One approach might be to find the structure with the most base pairs. Nussinov introduced an efficient dynamic programming algorithm for this problem [Nussinov *et al.* 1978]. Although this criterion is too simplistic to give accurate structure predictions, the example is instructive because the mechanics of the Nussinov algorithm are the same as those of the more sophisticated energy minimisation folding algorithms and of probabilistic SCFG-based algorithms.

The Nussinov calculation is recursive. It calculates the best structure for small subsequences, and works its way outwards to larger and larger subsequences. The key idea of the recursive calculation is that there are only four possible ways of getting the best structure for i,j from the best structures of the smaller subsequences (Figure 10.7):

(1) add unpaired position i onto best structure for subsequence $i+1,j$;
(2) add unpaired position j onto best structure for subsequence $i,j-1$;
(3) add i,j pair onto best structure found for subsequence $i+1,j-1$;
(4) combine two optimal substructures i,k and $k+1,j$.

More formally, the Nussinov RNA folding algorithm is as follows. We are given a sequence x of length L with symbols $x_1,\ldots,x_L$. Let $\delta(i,j)=1$ if x_i and x_j are a complementary base pair; else $\delta(i,j)=0$. We will recursively calculate scores $\gamma(i,j)$ which are the maximal number of base pairs that can be formed for subsequence $x_i,\ldots,x_j$.

Algorithm: Nussinov RNA folding, fill stage

Initialisation:

$$\gamma(i,i-1) = 0 \qquad \text{for } i = 2 \text{ to } L;$$
$$\gamma(i,i) = 0 \qquad \text{for } i = 1 \text{ to } L.$$

Recursion: starting with all subsequences of length 2, to length L:

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j), \\ \gamma(i,j-1), \\ \gamma(i+1,j-1)+\delta(i,j), \\ \max_{i<k<j}\left[\gamma(i,k)+\gamma(k+1,j)\right]. \end{cases}$$

◁

Figure 10.8 shows an example of a Nussinov matrix fill in operation.

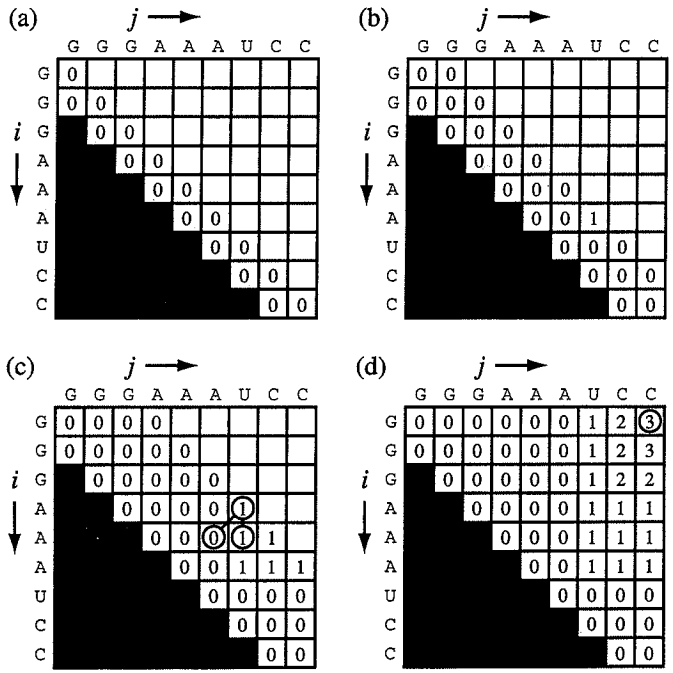

Figure 10.8 *The matrix fill stage of the Nussinov folding algorithm is shown for an example sequence* GGGAAAUCC. *(a) The initialised half-diagonal matrix. (b) The matrix after scores for subsequences of length two have been calculated. (c) An example of two different optimal substructures for the same subsequence. For the subsequence* AAAU, *either the* A *at i and the* U *at j can be paired (diagonal path) or i can be added to a substructure that already pairs the* A *at* $i+1$ *to the* U *at j (vertical path). (d) The final matrix. The value in the upper right indicates that the maximally paired structure has three base pairs.*

The value of $\gamma(1,L)$ is the number of base pairs in the maximally base-paired structure. There are often a number of alternative structures with the same number of base pairs. To find one of these maximally base-paired structures, we trace back through the values we calculated in the dynamic programming matrix, beginning from $\gamma(1,L)$. In pseudocode, the traceback algorithm is:

Algorithm: Nussinov RNA folding, traceback stage

Initialisation: Push $(1,L)$ onto stack.
Recursion: Repeat until stack is empty:
- pop (i,j).
- if $i >= j$ continue;
  else if $\gamma(i+1,j) = \gamma(i,j)$ push $(i+1,j)$;
  else if $\gamma(i,j-1) = \gamma(i,j)$ push $(i,j-1)$;
  else if $\gamma(i+1,j-1)+\delta_{i,j} = \gamma(i,j)$:
    - record i,j base pair.
    - push $(i+1,j-1)$.
  else for $k = i+1$ to $j-1$: if $\gamma(i,k)+\gamma(k+1,j) = \gamma(i,j)$:
    - push $(k+1,j)$.
    - push (i,k).
    - break.

◁

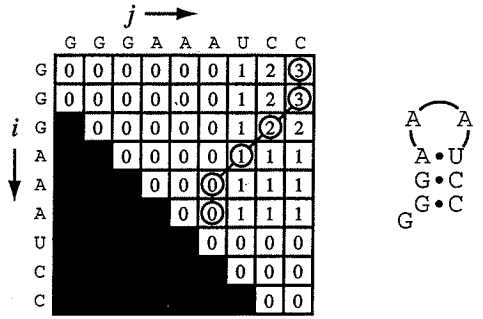

Figure 10.9 *The traceback stage of the Nussinov folding algorithm is shown for the filled matrix from Figure 10.8. An optimal traceback path is indicated with circles. The optimal structure corresponding to this path is shown at right.*

The traceback is linear in time and memory. The fill step is the limiting step as it is $O(L^2)$ in memory and $O(L^3)$ in time. An example traceback is shown in Figure 10.9. The traceback in Figure 10.9 is unbranched, so the need for the pushdown stack in the traceback algorithm is not apparent. The pushdown stack becomes important when bifurcated structures are traced back. The stack

remembers one side of the the bifurcation while the other side is traced back, reminiscent of the push-down automata in Chapter 9.

Exercises

10.2 Find two more optimal structures with three base pairs besides the one in Figure 10.9. Modify the traceback algorithm so it finds one of your structures instead of the one obtained in Figure 10.9.

10.3 As we have given it, the Nussinov algorithm can produce nonsensical 'base pairs' between adjacent complementary residues (for example, one of the possible structures in the preceding exercise contains such an AU base pair). Modify the Nussinov folding algorithm so that hairpin loops must have a minimum length of h. Give the new recursion equations for the fill and traceback.

10.4 Show that the Nussinov folding algorithm can be trivially extended to find a maximally *scoring* structure where a base pair between residues a and b gets a score $s(a,b)$. (For instance, we might set $s(\mathtt{G},\mathtt{C}) = 3$ and $s(\mathtt{A},\mathtt{U}) = 2$ to better reflect the increased thermodynamic stability of GC pairs.)

An SCFG version of the Nussinov algorithm

The Nussinov algorithm is fundamentally similar to the SCFG algorithms in Chapter 9. As an example of how SCFGs apply to RNA secondary structure analysis, consider the following production rules of a simple RNA folding SCFG:

$$\begin{array}{lcll} S & \rightarrow & aS \mid cS \mid gS \mid uS & (i \text{ unpaired}), \\ S & \rightarrow & Sa \mid Sc \mid Sg \mid Su & (j \text{ unpaired}), \\ S & \rightarrow & aSu \mid cSg \mid gSc \mid uSa & (i,j \text{ pair}), \\ S & \rightarrow & SS & \text{bifurcation}. \end{array} \qquad (10.2)$$

The SCFG has a single nonterminal S and 13 production rules with associated probability parameters. For now, assume that the probability parameters are known. The maximum probability parse of a sequence with this SCFG is an assignment of sequence positions to productions. Because the productions correspond to secondary structure elements (base pairs and single-stranded bases), the maximum probability parse is equivalent to the maximum probability secondary structure. If base pair productions have relatively high probability, the SCFG will favour parses which tend to maximise the number of base pairs in the structure.

Although the production rules for the SCFG are not in Chomsky normal form, a CYK parsing algorithm is readily written that finds the maximum probability secondary structure. Alternatively, we could convert the SCFG to Chomsky normal form and apply the algorithms in Chapter 9. Although the Chomsky normal form approach is attractive in its generality, specific algorithms for specific SCFGs are typically more efficient. The adapted CYK algorithm is as follows. Let the probability parameters of the SCFG productions be denoted by $p(aS)$, $p(aSu)$, etc.

Algorithm: CYK for Nussinov-style RNA SCFG

Initialisation:

$$\begin{aligned} \gamma(i,i-1) &= -\infty \qquad \text{for } i = 2 \text{ to } L; \\ \gamma(i,i) &= \max \begin{cases} \log p(x_i S) \\ \log p(S x_i) \end{cases} \qquad \text{for } i = 1 \text{ to } L. \end{aligned}$$

Recursion: for $i = 1$ to $L-1$, $j = i+1$ to L:

$$\gamma(i,j) = \max \begin{cases} \gamma(i+1,j) + \log p(x_i S); \\ \gamma(i,j-1) + \log p(S x_j); \\ \gamma(i+1,j-1) + \log p(x_i S x_j); \\ \max_{i<k<j} \gamma(i,k) + \gamma(k+1,j) + \log p(SS). \end{cases}$$ ◁

When this is done, $\gamma(1,L)$ is the log likelihood $\log P(x,\hat{\pi}|\theta)$ of the optimal structure $\hat{\pi}$ given the SCFG model θ. The traceback to find the structure corresponding to that best score is either performed analogously to the traceback in the Nussinov algorithm, or by keeping additional traceback pointers in the fill stage analogous to the CYK algorithm description in Chapter 9.

The principal difference between this and the original Nussinov algorithm is that the SCFG description is a probabilistic model. We gain access to several well-principled options for optimising the parameters of the model. We can set the SCFG's parameters by subjective estimation of the relevant probabilities, or by estimating parameters by counting state transitions in known RNA structures and converting the counts to probabilities. We can even learn probabilities from example RNAs of *unknown* structure using expectation maximisation (EM) and inside–outside training to iteratively infer both the structures and the parameters (i.e. the structures are the hidden data in the EM algorithm). Once we have written down the SCFG as a full probabilistic model of the RNA folding problem, we can 'turn the crank', applying all the probabilistic machinery we have learned in previous chapters almost by rote.

Like the Nussinov algorithm, this small SCFG is a good starting example but it is too simple to be an accurate RNA folder. It does not consider important structural features like preferences for certain loop lengths nor preferences for certain nearest neighbours in the structure caused by stacking interactions between neighbouring base pairs in a stem.

Exercises

10.5 Write down a traceback algorithm for determining the best RNA secondary structure after the above algorithm has completed.

10.6 Devise an SCFG which uses different nonterminals to model bulge loops, hairpin loops, multifurcation loops and single strands.

Energy minimisation and the Zuker folding algorithm

RNA folding is dictated by biophysics rather than by counting and maximising the number of base pairs. The most sophisticated secondary structure prediction method for single RNAs is the Zuker algorithm, an energy minimisation algorithm which assumes that the correct structure is the one with the lowest equilibrium free energy (ΔG) [Zuker & Stiegler 1981; Zuker 1989a].

The ΔG of an RNA secondary structure is approximated as the sum of individual contributions from loops, base pairs and other secondary structure elements. An important difference from the simpler Nussinov calculation is that the energies of stems are calculated by adding *stacking* contributions for the interface between neighbouring base pairs instead of individual contributions for each pair. In other words, the energy of a stem of n base pairs is the sum of $n-1$ base stacking terms instead of n base pair terms. This produces a better fit to experimentally observed ΔG values for RNA structures but it complicates the dynamic programming algorithm. Tables of ΔG parameters for RNA structure prediction have been fitted to the results of experimental thermodynamic studies of small model RNAs [Freier *et al.* 1986; Turner *et al.* 1987]. They include parameters for stacking, hairpin loop lengths, bulge loop lengths, interior loop lengths, multi-branch loop lengths, single dangling nucleotides and terminal mismatches on stems.

An example of the prediction of the ΔG of an RNA structure is given in Figure 10.10. Single base bulges are assumed not to disrupt stacking in the stem, so a stacking term is included in the example in the figure. Longer bulges, which are assumed to disrupt stacking, get no added stacking term. The hairpin loop energy is the sum of two terms: a loop destabilisation energy dependent only on the loop length, and a terminal mismatch energy dependent on the closing base pair and the first and last bases of the stem. The energies used in Figure 10.10 are from the older 'Freier rules' [Freier *et al.* 1986] at 37°C.[1]

The minimum energy structure can be calculated recursively by a dynamic programming algorithm (assuming no pseudoknots), very similar to how the maximum base-paired structure was calculated above. The principal difference is that because of the stacking parameters, two matrices (called V and W) are kept instead of one. $W(i,j)$ is the energy of the best structure on i,j. $V(i,j)$ is the energy of the best structure on i,j given that i,j are paired. The algorithm can then keep track of stacking interactions by adding new base pairs only onto the V matrix. Conceptually this two-state calculation is very similar to the use of extra insert states in pairwise dynamic programming alignment with affine gap costs (Chapter 2) to keep track of insert extensions. For a complete description of the Zuker algorithm, see Zuker & Stiegler [1981].

We could write down a SCFG that followed similar rules. The simplest stacking production rule would be, for instance, $cVg \rightarrow cgVcg$ for producing a GC pair in a stem after (stacked on) a CG, using V as a base pair generating nonterminal (as in the Zuker V matrix). With the CG terminals on the left as context for the production of the GC, this is technically a context-sensitive production, so we can't use such rules as the basis for a SCFG. However, we can convert to context-free productions by using four different nonterminals $V^{au}, V^{cg}, V^{gc}, V^{ua}$, and using right-hand sides of the form $\rightarrow gV^{gc}c$ to produce a G-C pair, for instance – the nonterminal identity V^{gc} 'remembers' that a G-C pair was just generated. (In other words, all we are doing is making the model a higher order Markov process.) The probability of a production $V^{cg} \rightarrow gV^{gc}c$, for instance, would be the probability of a C-G pair stacked on a G-C pair.[2] Other details of the Zuker algorithm and its two matrices V and W could be incorporated similarly into an analogous full probabilistic model with two nonterminals V and W (expanded for nearest neighbour context). CYK and inside–outside algorithms for an SCFG version of the Zuker algorithm have the same algorithmic complexity as the Zuker algorithm itself.

[1] Currently the most up-to-date parameters are available on the Web from `http://www.ibc.wustl.edu/~zuker/rna/energy/`.

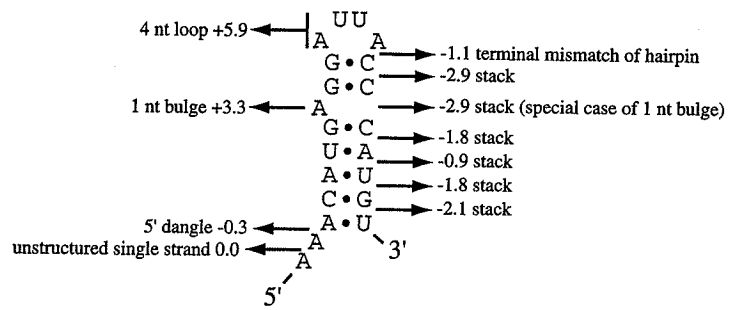

Figure 10.10 *An example ΔG calculation for an RNA stem loop (the wild type R17 coat protein binding site).*

[2] Since only one nonterminal is possible for a given x_i, x_j pair and the other three have zero probability, the four nonterminals behave as one for the purposes of memory and time complexity in parsing algorithms.