Computing RNA Secondary Structure:

1) Partition functions:

Say we have a system that can exist in several possible energy states, Ei (i.e. ground state + a bunch of excited states). In statistical physics we can compute a bunch of physical properties of the system in equilibrium with its environment from the partition function, Z. The partition function is a weighted sum of how each state contributes to the whole ensemble in thermodynamic equilibrium. The weights are determined by the Boltzman factor for each state, exp(-Ei/kT). Explicity,

$$Z = \sum_{i=1}^{N} g_i \exp(-E_i / kT)$$

where the g_i account for the possible degeneracy of each state. From Z, the free energy can be calculated from $F = -kT \ln(Z)$. The probability of a given state can be found from,

$$p_i = \frac{g_i \exp(-E_i / kT)}{Z}$$

2) Minimal RNA secondary structure model:

An RNA polymer folds up by pairing up complementary bases (i.e. G-C (or U) and A-U)). The dominant energy comes not from base-pairing, but from stacking several paired bases in a row. A pair of back-to-back base pairs, a stack, contributes an energy $-E_s$. We'll assume that there are no differences in the stack energies between a stack formed between an AU & AU or AU & GC (or any other combination). Dangling or unbound bases are free to flop around. This adds extra degrees of freedom to each structure. This is entropy.

We now have a particular sequence and we're interested in determining the possible structures that it can adopt and how each contributes to the overall ensemble. The contribution to the partition function from a possible structure, i, in this ensemble is,

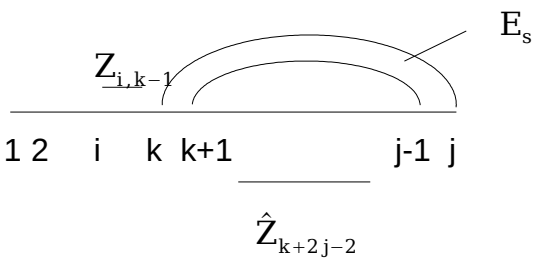
$$Z_i = \exp(n_s^i E_s / kT)$$

where n_s^i are the number of stacks in the given structure. The free energy of this state is $F_i = -kT \ln(Z_i) = -ST - n_s^i E_s$. The T dependent part of the free energy is the entropic contribution which goes to zero when T is zero. At T=0 folding is dominated by the making of stacks. At $T \neq 0$, entropy might win and keep a structure more open than stacked. The total partition function is given by,

$$Z = \sum_i Z_i$$

We could compute Z via 'brute force' by systematically enumerating all possible structures and summing up the contributions from all those structures which are compatible with our chosen sequence. This is OK for small RNA, but for large RNA this becomes a computational nightmare.

The elegant, albeit more abstract way of computing Z is to compute it recursively. We start with a single base and then sequentially add a base at a time, allowing only those structures that are possible. When we add a base, j, it can remain unpaired, or it might also be able to participate in the formation of a stack. This will contribute a weight $\exp(E_s/kT)$. And we need to sum over all possible stacks that might be formed by adding j. We keep adding bases, j, until we get to the last base. By doing this we build up the entire Z.



The recursion relation is:

$$Z_{i,j}=Z_{i,j-1}+\sum_{k=i}^{j-3} Z_{i,k-1}\exp(E_s/kT)\hat{Z}_{k+2,j-2}P_s(k,k+1;j-1,j)$$

where $P_s(k,k+1;j-1,j)=1$ if the bases at (k,k+1;j-1,j) can form a stack and 0 otherwise. The other partition function that enters the recursive formula is $\hat{Z}_{i,j}$. This accounts for the possible states that are interior to an exterior stack. It has slightly different boundary conditions than Z. Because it always resides on the interior of a base pair (i.e. there is an implicit base pair between i-1 and j+1) then if i pairs with j, we must add an extra term that accounts for the stacking energy due to the stack formed with the external pair. So the recursion relation for $\hat{Z}$ is,

$$\hat{Z}_{i,j}=Z_{i,j-1}+\exp(E_s/kT)\hat{Z}_{i+1,j-1}P_b(i,j)+$$
$$\sum_{k=i+1}^{j-3} Z_{i,k-1}\exp(E_s/kT)\hat{Z}_{k+2,j-2}P_s(k,k+1;j-1,j)$$

where $P_b(i,j)=1$ if the base i can pair with j and 0 otherwise. Using these two formula and the boundary conditions $Z_{i,j}=1$ for j < i, the full partition function $Z_{1,N}$ can be computed.