Supplemental Information

Here we will describe our approach to disentangle incompatible secondary structures, resulting in two clusters of compatible structures, as seen in Figure 1(h).

Most folding algorithms (Zuker, 1989; Mathews *et al.*, 2004; Markham and Zuker, 2008; McCaskill, 1990; Hofacker, 2003; Hofacker *et al*, 1994) use the set of *nested* structures to build their partition functions. In the partition function, it is possible to have two probable pairs P_{ij} and P_{kl} which are non-nested, see Figure 1(g), but non-nested pairs do not co-occur in any of the individual states in the partition function sum.

Non-nestedness of pairs is therefore a hallmark of the incompatibility of the structures.

In our simple and iterative *PF* clustering method, we sum over non-nested (k, l) pairs to find the (i, j) pair that produces the biggest non-nestedness score,

$$\psi_{ij} = P_{ij} \sum_{k<i<l<j} P_{kl} + P_{ij} \sum_{i<k<j<l} P_{kl}. \quad (1)$$

Once the maximum ψ_{ij} has been identified, we then perform a new partition function computation with bases (i, j) forced to pair to create the *Forced F* cluster.

The F cluster is the portion of the original partition function that contains the (i, j) pair. The F cluster's free energy

$$G_F = -RT \ \log \sum_{s \in F} e^{G_s},$$

and base pair probabilities P_{ij}^F are output. The fraction of the original partition function is

$$p_F = \exp\{-(G_F - G_0)/kT\},$$

where G_0 is the original free energy. The original partition function minus the forced partition function leaves all states in which the (i, j) pair is prohibited, the P cluster. The values for the prohibited cluster P can be obtained from the conservation of probability:

$$p_P = p_0 - p_F$$

and

$$p_P P_{ij}^P = p_0 P_{ij}^0 - p_F P_{ij}^F,$$

or by computing the partition function with a prohibit (i, j) constraint.

It is clear from Figure 1(h) that the effect that forcing even one base pair can have in dividing the ensemble of structures is remarkable.

Our *PF* procedure can also be repeated on any previously defined cluster as well at the root; each time separating further incompatibilities. The overall run time is $O(KN^3)$, where K is the number of clusters. [The sums of Eq. (1) appear to require $O(N^4)$ operations if one sums over the indices; however, by iterating over a list of the $O(N)$ base pairs which exceed a probability threshold, $P_{ij} > \theta \sim 10^{-6}$, Eq. (1) requires only $O(N^2)$ operations.] A detailed description of our related NESTOR algorithm, which instead creates clusters from stochastically sampled states, is also forthcoming.