Sources of Constraints on the Secondary Structure of RNA and tRNA

John S. Conery, Michael Lynch, Hongwen Deng and Qing Tien
University of Oregon

As part of a larger project that will use genetic information to reconstruct the phylogentic tree for sets of species we are working on a constraint-based approach to secondary structure prediction. In this talk we will describe why we are using a constraint logic programming framework, the types of information we are using to define constraints, and the current status of the project.

We use a logic programming approach to this problem for two reasons: logic programs allow one to define relations, which are more general than functions, and it is easy to represent partial information.

Instead of writing functions, which are one-way mappings from one set (the domain) to another (the range), relations are more general associations between sets. We hope to exploit this feature as we design a system that uses information from three different sources: phylogenetic trees, secondary structure, and sequence alignment. Relations should be well-suited to this problem because information from one area influences the other two. For example, if one knows the secondary structures of a set of tRNA molecules, sequence alignment should be more efficient. Alternatively, if a reliable alignment is known, it might be easier to predict the secondary structures. Rather than defining a function that maps structure to alignment or alignment to structure, we will use a logic programming language to define a relation that will use partial information from both sources to build a more complete description of the relationship between the species.

There are several situations where it will be important to be able to deal with partial information. For example, one might have a partial phylogeny tree, perhaps from the fossil record; a a partial description of the structure of some sequences (e.g. the location of the anticodon loop in two or three sequences); and know the alignment between a few pairs of sequences. The goal of our system is to use all this partial information as input to a relation that uses information from one area to fill in gaps in the other areas and construct a much more complete picture of the evolutionary relationship of the species represented by the inputs.

Currently we are working on a constraint logic programming approach to the secondary structure prediction portion of the system. The structures produced by our program will be used by a multiple alignment program previously developed at ICOT. Eventually we hope to use the alignment methods in the ICOT programs as part of our larger application.

Our constraint-based approach to structure prediction is analogous to a state space search. In operational terms, a "move generator" will propose directions to

move in order to explore the search space, while constraints block moves along paths that lead to dead ends. The object is to define effective constraints that block dead end paths as soon as possible. In more abstract terms, the move generator defines the potential shape of the search space, and the constraints define the actual shape carved out by the search.

The "move generator" in our structure prediction program is a procedure that produces a set of candidate stems. We use a pairwise alignment program to align the sequence with its own tail. For example, the acceptor stem of a tRNA is a sequence of seven base pairs, where the first seven bases on the 5' end line up with the last seven bases on the 3' end. As an example, consider the following sequence:

```
AAGGATTTAGCTTA....AAATCTTGTAATTCTTA
```

The reverse of the sequence is:

```
ATTCTTAATGTTCTAAA....ATTCGATTTAGGAA
```

If we use an alignment algorithm to align the sequence with its reverse we see that by shifting the sequence by one base we find the seven matching pairs in the acceptor stem:

```
 AAGGATTTAGCTTA....AAATCTTGTAATTCTTA
 *******
ATTCTTAATGTTCTAAA....ATTCGATTTAGGAA
```

The first step in our program is to generate all candidate stems through this process of aligning a sequence with itself. In this phase we define a stem to be a sequence of three or more base pairs. We allow at most one mismatch within a base (in which case it must be four or more pairs), and we also allow GT or TG matches.

The second phase of the program is to use constraints to throw out candidates that, for some reason or another, cannot be actual stems. For tRNAs, which have a well-known cloverleaf structure, we look for four stems that can be present simultaneously. We also use several biological constraints to help us order the candidates, so the most likely stems are considered first, and to further prune the set of candidates.

Note that although we described the program in terms of two distinct phases, as is they were functions in which the first phase produces a fully formed set of candidate stems, in the final implementation as a concurrent logic program there will be an interleaving of two concurrent (and eventually parallel) processes that work together to build and prune the search space.

The main part of this talk will be on the sources of information we are using to construct our constraints. Presently we are working on the following:

* using statistical information on the likelihood of different base pairings
* energy rules that favor certain pairs, and sequences of pairs, over others
* a comprehensive energy optimization program that will compute the total free energy in a potential structure
* structural constraints, for example the fact that an anticodon stem contains a seven-base loop with two unattached bases on either side of a three-base anticodon