Exploring Life through Logic Programming

Alessandro Dal Palú    Agostino Dovier
Andrea Formisano    Enrico Pontelli

Dept. Computer Science, New Mexico State University, USA

University of Udine, Italy
December 2015
RNA and Central Dogma
RNA and Central Dogma

RNA is a sequence of nucleotides (A,C,G,U) that (often) is just an intermediary between DNA and proteins.

- DNA strands are transcribed to mRNA, in order to exit the cell’s nucleus.
- Nucleotides replacement: DNA T → RNA U.

Transcription

Translation
Central Dogma
RNA Secondary Structure

- RNA folds according to favorable matchings (A-U, C-G, ~ U-G)
- The **secondary structure** is the set of its base pairings
- Secondary structure determines the 3D properties
RNA Secondary Structure

- RNA folds according to favorable matchings (A-U, C-G, \(~\) U-G)
- The **secondary structure** is the set of its base pairings
- Secondary structure determines the 3D properties
Mathematically

- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in $\{A, C, G, U\}^*$
- Structure described by set of pairs of interacting bases
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that
  - $(i, j) \in P \leftrightarrow (j, i) \in P$
  - $(i, j) \in P$ only if $(s_i, s_j) \in \{(A, U), (U, A), (C, G), (G, C), (U, G), (G, U)\}$
Mathematically

- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in \{A, C, G, U\}*
- Structure described by set of pairs of interacting bases
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that
  - $(i, j) \in P \iff (j, i) \in P$
  - $(i, j) \in P$ only if
    - $(s_i, s_j) \in \{(A, U), (U, A), (C, G), (G, C), (U, G), (G, U)\}$
- We are interested in a solution with maximal pairings (and/or minimizing a more complex energy function)

\[\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\
\end{array}\]
Hypothetical Arrangement
The general problem is NP-complete [Lyngsø and Pedersen 2000].

A large sub-class has *polynomial time* complexity:
- the absence of pseudo-knots.
Pseudo-knots

Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop

CGUUGUGUGUACACGUAUAGUACAU
Pseudo-knots

- Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop

CGUUGUGUACACG AUAGUACAU
Pseudo-knots

Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop.

CGUUGUGUACACGAUAGUACAU

Inversion
Pseudo-knots

- Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop.

![RNA Secondary Structure Diagram](attachment:RNASecondaryStructureDiagram.png)
Pseudo-knots

Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop
Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop.
Pseudo-knots

- Pseudo-knot: secondary structure where a loop is paired with a region outside of the stem flanking the loop.

```
CGU
`UGU
GUACACG`AU
AUAGUACAU
```
To avoid pseudo-knots, we impose a constraint:
If \( i < \ell < j \) and \((i, j) \in P\), and \(((\ell, k) \in P \lor (k, \ell) \in P)\), then \( i < k < j \).
ASP Encoding

RNA sequence

Each sequence is encoded with $n$ facts, e.g.:

```
seq(1,a).  seq(2,g).  seq(3,u).  
seq(4,c).  seq(5,c).  seq(6,a).
```

The main predicate is `pairing/2` which is a partial function.
ASP Encoding

Pairings

(1) sequence_index(X) :- seq(X, _).
(2) sequence_base(B) :- seq(_, B).
(3) 0 {pairing(X, Y) : sequence_index(Y)} 1 :-
    sequence_index(X).

- (1-2) collect domains for indexes and bases
- (3) defines the pairing: a partial function (at most one association for X)
ASP Encoding

Pairings

1. `sequence_index(X) :- seq(X,_)`.
2. `sequence_base(B) :- seq(_,B)`.
3. \(0 \{\text{pairing}(X,Y) : \text{sequence_index}(Y)\} 1 :-\text{sequence_index}(X)\).
4. \(\neg \text{sequence_index}(X_1), \text{sequence_index}(X_2), \text{sequence_index}(Y), X_1 < X_2, \text{pairing}(X_1,Y), \text{pairing}(X_2,Y)\).
5. `pairing(B,A) :- \text{sequence_index}(A), \text{sequence_index}(B), \text{pairing}(A,B)`.

- (4) Injective constraint: the pairing can’t cover a base two times
- (5) The pairs are symmetric
**ASP Encoding**

**Pairings**

1. `sequence_index(X) :- seq(X,_)`.  
2. `sequence_base(B) :- seq(_,B)`.  
3. `0 {pairing(X,Y):sequence_index(Y)} 1 :- sequence_index(X)`.  
4. `:-sequence_index(X1), sequence_index(X2), sequence_index(Y), X1<X2, pairing(X1,Y), pairing(X2,Y)`.  
5. `pairing(B,A):- sequence_index(A), sequence_index(B), pairing(A,B)`.  
6. `wrong(X,X):- sequence_base(X)`.  
7. `wrong(a,c). wrong(a,g). wrong(c,u)`.  
8. `:-wrong(B1,B2), seq(X1,B1), seq(X2,B2), pairing(X1,X2)`.  

- Discarded associations  
- Can’t pair two discarded associations
**ASP Encoding**

**Pairings**

(9) :- sequence_index(X1), sequence_index(X2), X1=X2+1, pairing(X1,X2).

- Cannot pair consecutive basis—chemical constraint
ASP Encoding

Pairings

(9) :- sequence_index(X1), sequence_index(X2), X1=X2+1, pairing(X1,X2).

(10) :- sequence_index(X1), sequence_index(X2), sequence_index(X3), sequence_index(X4), X1<X3,X3<X2,X2<X4, pairing(X1,X2),pairing(X3,X4).

- No Pseudo-knots (optional)
ASP Encoding

Pairings

(9) :- sequence_index(X1), sequence_index(X2), X1=X2+1, pairing(X1,X2).

(10) :- sequence_index(X1), sequence_index(X2), sequence_index(X3), sequence_index(X4),
      X1<X3,X3<X2,X2<X4,
      pairing(X1,X2),pairing(X3,X4).

(11) contacts(C):- C = #count{ (A,B):pairing(A,B) }.
(12) #maximize { C:contacts(C) }.

Maximize the number of pairings (Nussinov Energy Function)
ASP Encoding

- Alternative energy functions are possible
- Statistics: 35% AU, 53% CG, 12% GU
- $NC = n - \#(contacts)$
- minimize:
  $$c_1 \frac{NC}{n} + c_2 \frac{\#(AU) - 0.35(n - NC)}{n} + c_3 \frac{\#(CG) - 0.53(n - NC)}{n}$$
ASP Encoding

\[
\text{energy}(E) :- \text{total}(N), \text{contacts}(C), \text{au}(AU), \\
\text{cg}(CG), \\
E = c_1 \times (N-C) + c_2 \times (100 \times AU - 35 \times C) + \\
c_3 \times (100 \times CG - 53 \times C). \\
\#\text{minimize}\{E:\text{energy}(E)\}.
\]
(Some) References


- A. Dal Palù, M. Möhl, S. Will. A Propagator for Maximum Weight String Alignment with Arbitrary Pairwise Dependencies. CP 2010: 167-175 (also in WCB 10)