Central Dogma of Molecular Biology

by FRANCIS CRICK

The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred from protein to either protein or nucleic acid.

“The central dogma, enunciated by Crick in 1958 and the cornerstone of molecular biology ever since, is likely to prove a considerable oversimplification.”

Fig. 2. The arrows show the situation as it seemed in 1958. Solid arrows represent probable transfers, dotted arrows possible transfers. The absent arrows (compare Fig. 1) represent the impossible transfers postulated by the central dogma. They are the three possible arrows starting from protein.
Non-coding RNA

Messenger RNA - codes for proteins
Non-coding RNA - all the rest
Before, say, mid 1990’s, 1-2 dozen known
(critically important, but narrow roles)
Since mid 90’s dramatic discoveries
  Regulation, transport, stability/degradation
  E.g. “microRNA”: 100s in humans => 50% of genes
  E.g. “riboswitches”: 1000s in bacteria

DNA structure: dull

5’...ACCGCTAGATG...3’
3’...TGGCGATCTAC...5’

RNA Secondary Structure:
RNA makes helices too

Base pairs
A=U
C=G

5’...CAAAUAC...3’

RNA Structure: Rich

• RNA’s fold, and function
• Nature uses what works

Usually single stranded
Why is structure important?

- For protein-coding, similarity in sequence is a powerful tool for finding related sequences
  - e.g. “hemoglobin” is easily recognized in all vertebrates
- For non-coding RNA, many different sequences have the same structure, and structure is most important for function.
  - So, using structure plus sequence, can find related sequences at much greater evolutionary distances

RNA
Secondary Structure:

Not everything, but important, easier than 3d

6S mimics an open promoter

Barrick et al. RNA 2005
Trotochaud et al. NSMB 2005
Willkomm et al. NAR 2005
Chloroflexus aurantiacus
Geobacter metallireducens
Geobacter sulphurreducens
Chloroflexi
β-Proteobacteria

In Bacteria: A typical biosynthetic cycle around a critical metabolite (“SAM”)

Gene Regulation: The MET Repressor

Alberts, et al, 3e.
Epshtein, et al., PNAS 2003
Winkler et al., Nat. Struct. Biol. 2003
And many other examples. Widespread, deeply conserved, structurally sophisticated, functionally diverse, biologically important uses for ncRNA throughout prokaryotic world.
Vertebrates

• Bigger, more complex genomes
• <2% coding
• But >5% conserved in sequence?
• And 50-90% transcribed?
• And structural conservation, if any, invisible (without proper alignments, etc.)
  • What’s going on?

Q: What’s so hard?

A: Structure often more important than sequence

Fastest Human Gene?

Origin of Life?

Life needs
  information carrier: DNA
  molecular machines, like enzymes: Protein
  making proteins needs DNA + RNA + proteins
  making (duplicating) DNA needs proteins

Horrible circularities! How could it have arisen in an abiotic environment?
**Origin of Life?**

RNA can carry information, too
- RNA double helix; RNA-directed RNA polymerase
- RNA can form complex structures
- RNA enzymes exist (ribozymes)
- RNA can control, do logic (riboswitches)

The “RNA world” hypothesis:
1st life was RNA-based

---

**6.5 RNA Secondary Structure**

Nussinov’s Algorithm – core technology for RNA structure prediction

---

**RNA Secondary Structure**

RNA. String $B = b_1 b_2 \ldots b_n$ over alphabet \{ A, C, G, U \}.

Secondary structure. RNA is usually single-stranded, and tends to loop back and form base pairs with itself. This structure is essential for understanding behavior of molecule.

Ex: GUCAUUGACGCCGAAUGUAACACUGGUCAUGCGAGA

complementary base pairs: A-U, C-G

---

**RNA Secondary Structure (somewhat oversimplified)**

Secondary structure. A set of pairs $S = \{ (b_i, b_j) \}$ that satisfy:
- [Watson-Crick.] $S$ is a matching, i.e. each base pairs with at most one other, and each pair in $S$ is a Watson-Crick pair: A-U, U-A, C-G, or G-C.
- [No sharp turns.] The ends of each pair are separated by at least 4 intervening bases. If $(b_i, b_j) \in S$, then $i < j - 4$.
- [Non-crossing.] If $(b_i, b_j)$ and $(b_k, b_l)$ are two pairs in $S$, then we cannot have $i < k < j < l$. (Violation of this is called a pseudoknot.)

Free energy. Usual hypothesis is that an RNA molecule will form the secondary structure with the optimum total free energy. Approximate by number of base pairs.

Goal. Given an RNA molecule $B = b_1 b_2 \ldots b_n$, find a secondary structure $S$ that maximizes the number of base pairs.
**RNA Secondary Structure: Examples**

Examples:

- Base pair
- Sharp turn
- Crossing

**RNA Secondary Structure: Subproblems**

First attempt. \( \text{OPT}[j] \) = maximum number of base pairs in a secondary structure of the substring \( b_ib_{i+1}...b_j \).

Difficulty. Results in two sub-problems:
- Finding secondary structure in: \( b_1b_2...b_{t-1} \).
- Finding secondary structure in: \( b_{t+1}b_{t+2}...b_{j-1} \).

**Dynamic Programming Over Intervals: (R. Nussinov’s algorithm)**

Notation. \( \text{OPT}[i, j] \) = maximum number of base pairs in a secondary structure of the substring \( b_ib_{i+1}...b_j \).

- Case 1. If \( i \geq j - 4 \).
  - \( \text{OPT}[i, j] = 0 \) by no-sharp turns condition.

- Case 2. Base \( b_j \) is not involved in a pair.
  - \( \text{OPT}[i, j] = \text{OPT}[i, j-1] \)

- Case 3. Base \( b_j \) pairs with \( b_t \) for some \( i \leq t < j - 4 \).
  - Non-crossing constraint decouples resulting sub-problems
  - \( \text{OPT}[i, j] = 1 + \max_t \{ \text{OPT}[i, t-1] + \text{OPT}[t+1, j-1] \} \)
    - Take max over \( t \) such that \( i \leq t < j-4 \) and \( b_i \) and \( b_t \) are Watson-Crick complements

**Bottom Up Dynamic Programming Over Intervals**

Q. What order to solve the sub-problems?
A. Do shortest intervals first.

Running time. \( O(n^2) \).

Remark. Same core idea in CKY algorithm to parse context-free grammars.
Computing one cell: $\text{OPT}[2,18] = ?$

The figure shows a dynamic programming approach for computing the score of alignments. The figure is divided into several cases:

1. **Case 1:** $t = 2$: No pair
   - OPT[2,18] = 0
   - Unpaired position

2. **Case 2:** $t = 3$: No pair
   - OPT[2,18] = 0
   - Unpaired position

3. **Case 3, $2 \leq t < 18-4$:**
   - OPT[2,18] = 1
   - Pair at position 2

4. **Case 4:** $t = 18$:
   - OPT[2,18] = 2
   - Pair at position 18

The figure uses matrices to illustrate the computation of scores for different alignment positions. The matrices represent the score matrix for the alignment problem, with each cell representing the maximum score achievable up to that point.

The formulas for computing the values in the matrix are:

$$\text{OPT}(i,j) = \max \begin{cases} 0 & \text{if } i = j = 4 \\ \text{OPT}(i,j-1) + \text{score} & \text{if } i < j \\ \text{OPT}(i+1,j-1) + \text{score} & \text{if } i = j \\ \text{OPT}(i-1,j+1) + \text{score} & \text{if } i = j + 1 \\ \text{OPT}(i,j) + \text{score} & \text{otherwise} \end{cases}$$

Where:
- $\text{score}$ is the score of matching or mismatching characters.
- $\text{OPT}(i,j)$ is the optimal score for aligning the first $i$ characters of the first sequence and the first $j$ characters of the second sequence.

The figure also includes examples of alignments with specific scores, such as $\text{OPT}[1,6] = 1$ and $\text{OPT}[6,16] = 2$, illustrating how the scores are computed for different segments of the sequences.
Computing one cell: \( \text{OPT}[2,18] = ? \)

\[
\text{OPT}(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \left[0, \text{OPT}[i,j-1] \right] & \text{otherwise} \\
1 + \max \left( \text{OPT}[i,t-1] + \text{OPT}[t+1,j-1] \right) & \text{if } i \geq j - 4 \\
1 + \max \left( \text{OPT}[i,t-1] + \text{OPT}[t+1,j-1] \right) & \text{otherwise}
\end{cases}
\]
Computing one cell: \( \text{OPT}[2,18] = ? \)

\[
\text{OPT}(i,j) = \begin{cases} 
0 & \text{if } i \geq j - 4 \\
\max \{ \text{OPT}[i,j-1], 1 + \max(\text{OPT}[i,t-1] + \text{OPT}[t+1,j-1]) \} & \text{otherwise}
\end{cases}
\]

Case 3, 2 \( \leq t < 18 - 4 \): 
\( t = 8: \) no pair

Overall, Max = 4
several ways, e.g.:
GGAAAACCC GGGGU

(tree shows trace back:
square = case 3
octagon = case 1)

Another Trace Back Example

E.g.: \( \text{OPT}[1,16] = 3: \)
CUCGGGUCGACAGUC
\((\ldots)((\ldots))\ldots\)