The Conway Equation

Review of Correlation Polynomial

Finding Signals in DNA (continued)

12/17~18

A, B : ℓ-letterwords

Correlation of A, B : $AB = (c_0, c_1, \cdots c_{\ell-1})$ where the ith bit of AB is defined to be <u>one</u> "1" if the $(n-i)$-prefix (the first $n-i$ letters) of B coincides with the $(n-i)$-suffix of A, and "0" otherwise.

e.g. $A = 1\underline{00100}$

$B = \underline{00100}1$

$AB = (0,1,0,0,1,1)$

$AA = (1,0,0,1,0,0)$

$BB = (1,0,0,1,0,0)$

$BA = (0,0,1,0,0,1)$

(*) Let $AB = (c_0, c_1, \cdots, c_{\ell-1})$ and $c_{m_1}, c_{m_2}, \cdots, c_{m_k}$ be the bits of AB equal to 1.

In AB, $c_1 = 1$, $c_4 = 1$, $c_5 = 1$ and thus $m_1 = 1$, $m_2 = 4$, $m_3 = 5$.

(*) Denote as $\mathcal{A}_{AB}$ the set of k prefixes of $A = a_1 a_2 \cdots a_\ell$ of length $m_1, m_2, \cdots, m_k$:

$(a_1, \cdots, a_{m_1}), (a_1, a_2 \cdots a_{m_1}, \cdots, a_{m_2}), \cdots, (a_1, a_2, \cdots, a_{m_{k-1}}, \cdots, a_{m_k})$,

e.g. If $A = XYYXYYY$, then

$\mathcal{A}_{AB} = \{X, XYYYX, XYYYXY\}$

Definition Let $A = a_1 a_2 \cdots a_\ell$ and $B = b_1 b_2 \cdots b_\ell$, then the concatenation of A and B, $A * B = a_1 a_2 \cdots a_\ell b_1 b_2 \cdots b_\ell$. If $\mathcal{A} = \{A\}$ and $\mathcal{B} = \{B\}$, then $\mathcal{A} * \mathcal{B} = \{A * B\}$ which has possibly $|\mathcal{A}||\mathcal{B}|$ words (perhaps with repeats).

Definition If W is an ℓ-letter word, then let $P(W) = \frac{1}{2^\ell}$. For a set $\mathcal{W} = \{W\}$, let $P(\mathcal{W}) = \sum_{W \in \mathcal{W}} P(W)$.

e.g. $P(\mathcal{A}_{AB}) = \frac{1}{2} + \frac{1}{2^4} + \frac{1}{2^5}$.

Lemma $K_{AB}\left(\frac{1}{2}\right) = P(\mathcal{A}_{AB})$.

Proof. (Explain with an example)

In the above example $K_{AB}(t) = t^1 + t^4 + t^5 = \frac{1}{2} + \left(\frac{1}{2}\right)^4 + \left(\frac{1}{2}\right)^5$.

A word W is an A-victory if it contains A in the end and does not contain B.

(B) (B) (A)

A word W is an A-previctory if W * A is an A-victory.

(B) (B) (B)

Let $S_A = \{A\text{-previctories}\}$, $S_B = \{B\text{-previctories}\}$ and

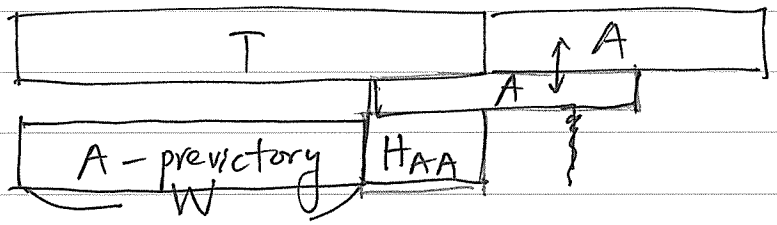
$\mathcal{T} = \{T : T \text{ is neither A-victory nor B-victory}\}$.

↑ the set of no-victor words.

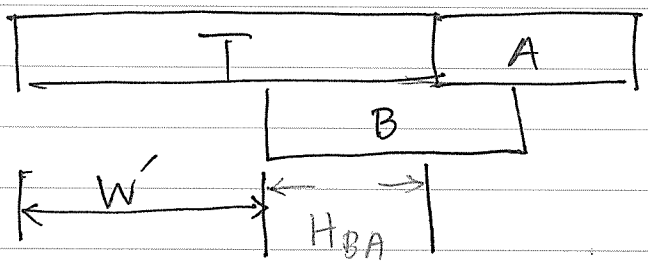
$\forall\, T \in \mathcal{T}$,

Fact 1 $T * A$ corresponds to either an A-victory or a B-victor

Fact 2 If $T * A$ corresponds to an A-victory, then T can be represented as $W * H_{AA}$ where W is an A-previctory.



Fact 3 If $T * A$ corresponds to a B-victory, then $T = W' * H_{BA}$ where W' is a B-previctory.

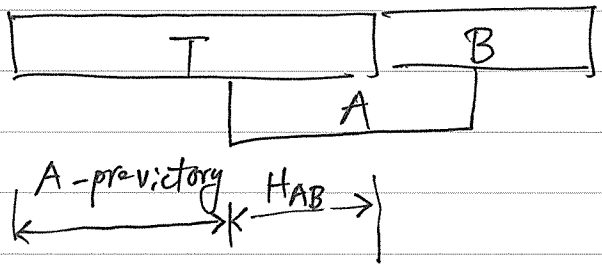


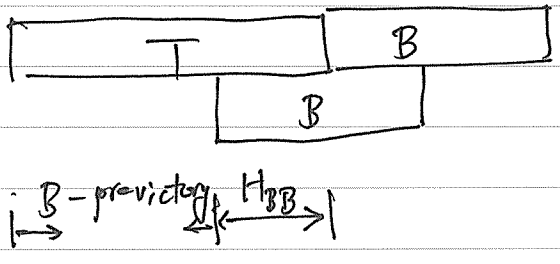
Fact 4 (Combine Fact 2 and Fact 3.)

$$\mathcal{J} = \mathcal{J}_1 = (S_A * H_{AA}) \cup (S_B * H_{BA})$$

Fact 5 Similar to Fact 4,

$$\mathcal{J} = \mathcal{J}_2 = (S_A * H_{AB}) \cup (S_B \cup H_{BB})$$





Theorem The odds that B wins over A is

$$\frac{K_{AA} - K_{AB}}{K_{BB} - K_{BA}}. \qquad (K_{AB} \doteq K_{AB}(\tfrac{1}{2})).$$

Proof.

$$p(\mathcal{T}_2) = p(S_A * \mathcal{A}_{AB}) + p(S_B * \mathcal{A}_{BB})$$

$$= p(S_A)\,p(\mathcal{A}_{AB}) + p(S_B) \cdot p(\mathcal{A}_{BB})$$

$$= p(S_A)\,K_{AB} + p(S_B) \cdot K_{BB}$$

$$p(\mathcal{T}_1) = p(S_B)\,K_{BA} + p(S_A)\,K_{AA}$$

Since $p(\mathcal{T}_1) = p(\mathcal{T}_2)$,

$$p(S_A)\,K_{AB} + p(S_B)\,K_{BB} = p(S_B)\,K_{BA} + p(S_A)\,K_{AA}.$$

$$\frac{p(S_B)}{p(S_A)} = \frac{K_{AA} - K_{AB}}{K_{BB} - K_{BA}}.$$

(Note that the odds that B wins over A is equal to the ratio of $p(S_B)$ over $p(S_A)$, i.e., the probability of B-previctories over the probability of A-previctories.)

∎

Signals are everywhere!

(***)

Gene Prediction

Charles Yanofsky, ~~and~~ Sydney Brenner et. al. (1960年代)

Showed that a gene and its protein product are colinear structures with direct correlation between triplets of nucleotides in the gene and amino acids in the protein.

Later (1960年代)

Overlapping genes and genes-within-genes were discovered.

(*) 如此，註定要預測基因是一件極困難的工作。

其實是在1977, Split human genes 的發現，終於造成

"computational gene prediction" puzzle."

1977 (phillip Sharp and independently Richard Roberts).

⑥ Most human genes are interrupted by junk DNA and are broken into pieces call exons.

1. Statistical Approach

2. Similarity-Based Approach

Uses spliced alignment algorithm

⋮

The Twenty Questions Game with Genes

Given an (unknown) set I of integers in the interval $[1, n]$, reconstruct the set I by asking the minimum number of queries of the form "does a given interval contain an integer from I?"

[Note: In this formulation, interval $[1, n]$ corresponds to cDNA, I corresponds to exon boundaries in cDNA, and the queries correspond to PCR reactions defined by a pair of primers.]

↓

original gene with cut-out introns (complementary copy of an mRNA)

If the number of exon boundaries k is known. Then we can do something on finding the lower bounds.

Lemma The number of queries is at least $\log_2 \binom{n}{k}$.

Proof. By decision tree model, the decision assumes sequential queries using one query at a time. Since there are $\binom{n}{k}$ possible outcomes and the answer is either yes or no, we conclude that we need at least h queries to obtain the answer where $2^h \geq \binom{n}{k}$. This concludes the proof.

(Note) $\log_2 \binom{n}{k} \approx k \log_2 n - k \log_2 k$.

$k = 1$ ✓

$k = 2$? $\log_2 \binom{n}{k} \leq h \leq 2 \log_2 n$ Gap

(*) If a biologist tolerates an error Δ in the positions of exon boundaries, then the lower bound on the number of queries is $\approx k \log_2 \frac{n}{\Delta} - k \log_2 k$.

In pratical situation, we use about 30 primers and (typical cDNA)
3 rounds to find exon boundaries.

Hidden Graph Problem

A hidden graph G is known belonging to a given family $\mathcal{G}$ of labeled graphs on the set $N = [1, n]$. The problem is to reconstruct G by asking queries as few as possible, where a query is of the form

(*) "Does $S \subseteq N$ induce one edge of G?" That is "$G[S]$ contains at least an edge or $G[S]$ is not an empty graph (with no edges.

(*) Of course, we may ask $\binom{n}{2}$ pairs of vertices in $[1, n]$

Four models

(1) No restriction on the size of S and the answer ~~are~~ is either "Yes" or "No". (not both)

(2) No restriction on the size of S and the answer is the number of edges in $G[S]$.

(3) $|S| \le k$ and the answer is either Yes or No but not both.

(4) $|S| \le k$ and the answer is the size of $G[S]$.

(1) General model

(2) Quantity model

(3) k-vertex general model

(4) k-vertex quantity model

Algorithms

Step 1. Find a vertex of G.

Step 2. Find an edge of G.

Step 3. Find G. (The best idea so far is via finding a maximal matching of

Algorithm of Step 1 (Find-one-vertex)

$$Q(S) = \begin{cases} 0, & G[S] \text{ contains no edges,} \\ 1, & \text{otherwise.} \end{cases}$$

1. $S \leftarrow N$
2. if $Q(S) = 0$ then
3. Return $\emptyset$.
4. end if
5. $A \leftarrow N$
6. while $|A| > 1$ do
7. Partition (arbitrarily) A into two balanced sets A_0 and A_1
8. if $Q(S \setminus A_0) = 1$ then
9. $S \leftarrow S \setminus A_0$, $A \leftarrow A_1$.
10. else
11. $A \leftarrow A_0$
12. end if
13. end while
14. Return the element in A.