

Sequencing by Hybridization (SBH Problem)

(雜交繁殖)

Definition (l -mer composition)

For a string \vec{s} of length n , the l -mer composition, or spectrum, of \vec{s} , is the multiset of $n-l+1$ l -mers (consecutive l letters, substring of length l) in \vec{s} , denoted by $\text{Spectrum}(\vec{s}, l)$.

For example

$$\vec{s} = \text{ATGCTGCAAG}, \quad |\vec{s}| = 10$$

$$l = 3$$

$$\Rightarrow \text{Spectrum}(\vec{s}, l) = \{ \text{ATG}, \text{TGC}, \text{CTG}, \text{TGC}, \text{GCA}, \text{CAA}, \text{AAG} \}$$

SBH problem

Let S be a set of l -mers obtained from \vec{s} .

Reconstruct \vec{s} from $S = \text{Spectrum}(\vec{s}, l)$.

Note: Clearly, we did not know the "order" of elements in S .

Idea: Two consecutive l -mers, \vec{p} and \vec{q} has overlap $(\vec{p}, \vec{q}) = l-1$.

(**) One \vec{s} gives a unique $\text{Spectrum}(\vec{s}, l)$ for each $l=1, 2, \dots, n-1$.

(***) A given set of l -mers may produce two or more feasible strings, i.e., each of the strings has the same spectrum.

(2)

For example: $ATG \boxed{CGT} GCA = \vec{s}_1$
 $ATG \boxed{GCGT} GCA = \vec{s}_2$

We may obtain the above two distinct strings from the

set $\text{Spectrum}(?, 3) = \{ ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT \}$
 $v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7 \quad v_8$

$$\vec{s}_1 : v_1 - v_3 - v_7 - v_8 - v_4 - v_2 - v_5 - v_6 .$$

$$\vec{s}_2 : v_1 - v_2 - v_5 - v_7 - v_8 - v_4 - v_3 - v_6 .$$

Observation

SBH problem is a particular (special) case of the shortest Superstring problem, see it? However, in contrast to SSP, there exists a simple linear-time algorithm for the SBH problem.

(*) Construct one \vec{s} ; not all \vec{s} 's.

Approach

① Hamiltonian Path Problem

↙ digraph

Let $V(G) = \{ \vec{s}_1, \vec{s}_2, \dots, \vec{s}_n \}$ and $(\vec{s}_i, \vec{s}_j) \in A(G)$ iff $\text{overlap}(\vec{s}_i, \vec{s}_j) = l - 1$. Then, a Hamiltonian path gives an \vec{s} .
(directed)

② Eulerian Path Problem

From the set of l -mers we obtain a set T of $(l-1)$ -mers

$\{\vec{k}_1, \vec{k}_2, \dots, \vec{k}_m\}$. Then, let \tilde{G} be the graph obtained from

letting $V(\tilde{G}) = T$ and \vec{k}_i is incident to \vec{k}_j if and only

if $\text{overlap}(\vec{k}_i, \vec{k}_j) = l-2$ and $\vec{k}_i \wedge \vec{k}_j$ is an l -mer in S ,

$\vec{k}_i \wedge \vec{k}_j$ is obtained by combining \vec{k}_i and \vec{k}_j together such that overlapping portion

the (suffix of \vec{k}_i and prefix of \vec{k}_j) occurs exactly once. For example

$$ATG \wedge TGG = ATGG.$$

(*) Then, an eulerian path (directed) of \tilde{G} gives an \vec{s} .

(Note) The number of ^{directed} eulerian circuits in a ^{connected} balanced

digraph can be enumerated.
(directed eulerian graph)

Problem The number of eulerian circuits in an eulerian graph can also be enumerated, but how?

(**) After the short 500-700 bp DNA reads are sequenced, biologists need to assemble them together to reconstruct the entire genomic DNA sequence, (Fragment Assembly Problem) $\hookrightarrow l?$

(Notable Facts)

1. The error rate in DNA reads produced by modern machines varies from 1% to 3%.

2. One never knows whether a read came from a target strand DNA sequence or from its Watson-Crick complement (DNA is double-stranded).

3. Repeats in DNA cause a lot of trouble in sequencing DNA

For example, roughly 300 nucleotide Alu sequence is repeated more than a million times throughout the genome, with only ^{5%} 15% sequence variation.

(Repeats can occur at several different scales.)

4. Fragment assembly algorithms: (Three steps)

(1) Overlap: Finding potentially overlapping reads

(2) Layout: Finding the order of reads along DNA

(3) Consensus: Deriving the DNA sequence from the layout

Review of Graph Theory

1. Any tournament contains a Hamiltonian path.
2. Any even connected graph contains an eulerian circuit.
3. A graph contains an eulerian path if the graph is connected and has at most two odd vertices.
4. (Kotzig, 1968) Let G be a colored connected graph with even degree of vertices. Then, there is an alternating eulerian circuit in G if and only if every vertex is balanced, i.e. \forall color c , the number of edges incident to v and colored

$$d_c(v) \leq d(v)/2. \quad (d(v): \text{the degree of } v \text{ in } G)$$

Problem
Proof. (?) 把每一点的相接边分成 pairs of edges with distinct colors.

Corollary G is bicolored $\Rightarrow d_1(v) = d_2(v)$ for each $v \in V(G)$.

Open problem Find the number of alternating eulerian circuits in a bicolored eulerian graph.

Set Cover Problem

Let S_1, S_2, \dots, S_k be subsets of S with $wt(S_i) = w_i, i=1,2,\dots$

and $\bigcup_{i=1}^k S_i \supseteq S$. Find a set of indices $I \subseteq \{1,2,\dots,k\}$ such

that (1) $\bigcup_{j \in I} S_j \supseteq S$, and (2) $\sum_{j \in I} w_j$ is minimized.

Example $S = [1,6], S_1 = \{1,2\}, S_2 = \{1,3\}, S_3 = \{2,3\}, S_4 = \{2,4,6\},$

$S_5 = \{3,5,6\}, S_6 = \{4,5,6\}, S_7 = \{4,5\}, wt(S_i) = |S_i|.$
Natural weight

| | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 |
|---|-------|-------|-------|-------|-------|-------|-------|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 5 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

(*) In general S_7 is allowed. ~~not allowed~~

(**) So we are looking for some rows $S_{i_1}, S_{i_2}, \dots, S_{i_k}$ such

that $\bigcup_{j=1}^k S_{i_j} \supseteq S$ and $\sum_{j=1}^k |S_{i_j}|$ is minimized.

Natural weight

(***) $S_1 \cup S_2 \cup S_3 \cup S_6 \supseteq S$ and weight = 9. (Can you find a better one yes!)

7

Example (Real)

IBM finds computer viruses (Wikipedia)

S : 5000 known viruses

Subsets (strings) 9000 substrings of 20 or more consecutive bytes from viruses, not found in "good" code.

(*)

A set cover of 180 was found. It suffices to search for these 180 substrings to verify the existence of known computer

viruses.

Greedy Algorithm

C : represent the set of elements covered so far

α : average cost per newly covered node

Algorithm

1. $C \leftarrow \emptyset$

2. While $C \neq S$ or $C \not\subseteq S$ do

Find the set whose cost effectiveness is smallest, say \tilde{S}

$$\text{Let } \alpha_{\tilde{S}} = \frac{\text{wt}(\tilde{S})}{|\tilde{S} \setminus C|}$$

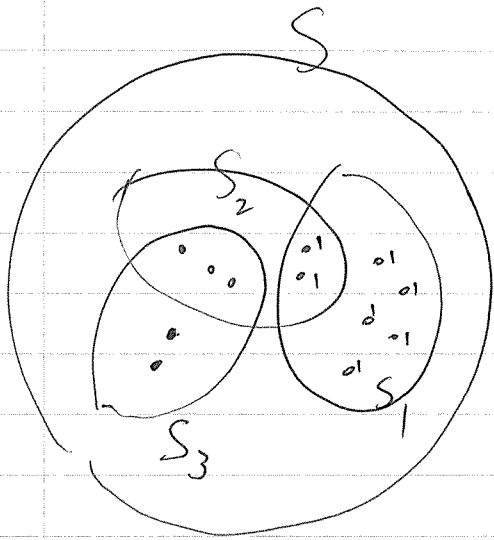
$\forall e \in S \setminus C$, set $\text{price}(e) = \alpha$

$C \leftarrow C \cup S$

3. Output picked sets

==

Example



$\text{wt}(S_1) = 7$

$\text{wt}(S_2) = 6$

$\text{wt}(S_3) = 15$

==

By looking, optimal solution

is $S_1 \cup S_3$ of weight 22.

By greedy algorithm

1. Choose S_1 ,

$\alpha_{S_1} = \frac{\text{wt}(S_1)}{|S_1 \setminus \emptyset|} = 1$, $C \leftarrow S_1$

2. Choose S_2 :

$\alpha_{S_2} = \frac{\text{wt}(S_2)}{|S_2 \setminus S_1|} = \frac{6}{3} = 2$, $C \leftarrow S_1 \cup S_2$

3. choose S_3 :

$\alpha_{S_3} = \frac{\text{wt}(S_3)}{|S_3 \setminus (S_1 \cup S_2)|} = \frac{15}{2} = 7.5$.

Total : $7 + 6 + 15 = 28$

How good is a greedy algorithm?

(Explain to you in next lecture.)

We can solve the Set Cover Problem by using Linear Programming.

Definition (Linear Programming)

A "Linear Programming" is the problem of optimizing (minimizing or maximizing) a linear function subject to linear inequality constraints.

Example

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^m c_j x_j \\ \text{Subject to} \quad & \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i=1, 2, \dots, m \\ & x_j \geq 0 \end{aligned}$$

(Note) 由於答案不一定是 Integer, 有 polynomial time algorithm

Set Cover Problem

variable x_j for set $S_j = \begin{cases} 1 & \text{if } S_j \text{ is selected;} \\ 0 & \text{if not.} \end{cases}$

上面例子

$$S = [1, 6], \dots \quad \text{Min. } \sum_{j=1}^k c_j x_j = \text{OPT}$$

subject to

$$\left\{ \begin{array}{l} x_1 + x_2 \geq 1 \\ x_1 + x_3 + x_4 \geq 1 \\ x_2 + x_3 + x_5 \geq 1 \\ x_4 + x_6 + x_7 \geq 1 \\ x_5 + x_6 + x_7 \geq 1 \\ x_4 + x_5 + x_6 \geq 1 \end{array} \right.$$

How to apply?