

Shortest Superstring Problem

- Problem

Given a set of strings (sequences) $s_1, s_2, s_3, \dots, s_n$, find the shortest string s such that for each $i = 1, 2, \dots, n$, s_i is a substring of s .

Example {CATGC, CTAAGT, GCTA, TTCA, ATGCATC} is a set of strings.

Greedy Algorithm: Start with two strings which have the most overlap part. CATGC and ATGCATC $\rightarrow$ CATGCATC (t_1), CTAAGT and GCTA $\rightarrow$ GCTAAGT, TTCA and $t_1 \rightarrow$ TTCATGCATC (t_2), GCTAAGT and $t_2 \rightarrow$ GCTAAGTTCATGCATC (an answer).

(*) The first step is also known as **shotgun sequencing**.

Prefix Graph of Strings

- Consider a set of strings $\{s_1, s_2, s_3, \dots, s_n\}$.

Definition

The **overlap** of two strings s_i and s_j is defined as the common **suffix of s_i and prefix of s_j** , denoted by $\text{overlap}(s_i, s_j)$. The prefix of (s_i, s_j) is defined as the substring after taking off the overlap, denoted by $\text{prefix}(s_i, s_j)$.

For example, $\text{overlap}(\text{ACGGCTAT}, \text{CTATTAG}) = \text{CTAT}$. But, $\text{overlap}(s_j, s_i) = \phi$. $\text{prefix}(\text{ACGGCTAT}, \text{CTATTAG}) = \text{ACGG}$ and $\text{prefix}(\text{CTATTAG}, \text{ACGGCTAT}) = \text{CTATTAG}$.

Continued

- Definition (Prefix graph)

The prefix graph G of a set of strings $\{s_1, s_2, s_3, \dots, s_n\}$ is a **double weighted complete digraph** such that $V(G) = \{s_1, s_2, s_3, \dots, s_n\}$, $A(G) = \{(s_i, s_j) \mid 1 \leq i, j \leq n\}$ and the weight of $(s_i, s_j) = |s_i| - |\text{overlap}(s_i, s_j)|$, denoted by $w(s_i, s_j)$.

Example (A complete digraph of order 5)

$s_1, \dots, s_5$ are the following: CATGC, CTAAGT, GCTA, TTCA, ATGCATC.
Then, $w(s_1, s_5) = 5 - 4 = 1$, $w(s_5, s_1) = 7 - 1 = 6$. (See it?)

Hamilton Path Problem

- In order to find the shortest superstring, it is equivalent to **find a directed Hamilton path of the prefix graph (defined accordingly) with minimum total weight.**
- The problem is similar to the well known **Traveling Salesman Problem (TSP)**. Clearly, it is a very hard problem in general.
- An answer of the previous problem is : $s_3 \rightarrow s_2 \rightarrow s_4 \rightarrow s_1 \rightarrow s_5$ which has total weight 16.
- Note that the answer can only provide a **lower bound**.

Sequencing by Hybridization (SBH Problem)

- Definition (**l-mar composition**)

For a string s of length n , the l -mar composition, or spectrum of s , is the multiset of $n - l + 1$ l -mars (consecutive substrings of length l) in s , denoted by $\text{spec}(s, l)$.

Example

$s = \text{ATGCTGCAAG}$, $|s| = 10$ and $l = 3$. $\text{spec}(s, 3) = \{\text{ATG}, \text{TGC}, \text{GCT}, \text{CTG}, \text{TGC}, \text{GCA}, \text{CAA}, \text{AAG}\}$. Two 3-mars have overlap 2 letters in this example. In general, two l -mars have $l - 1$ overlap letters.

SBH Problem Let S be a set of l -mars obtained from a given s . Reconstruct s from $S = \text{spec}(s, l)$.

Observation of SBH Problem

- We do not know the order of elements in S (spectrum).
- Two consecutive l -mers do have $l - 1$ overlap letters.
- One s gives a unique $\text{spec}(s, l)$ for each $l = 1, 2, \dots, n-1$.
- (*) A given set of l -mers may produce two or more feasible strings, each of the strings has the same spectrum.
- Example ATGCGTGGCA and ATGGCGTGCA have the same spectrum {ATG, TGC, GCG, CGT, GTG, TGG, GGC, GCA}.
- SBH problem is a particular (special) case of the Shortest Superstring Problem. (See it?)

Algorithm for SBH Problem

- **Hamilton Path Idea** (Let S be a set of l -mars obtained from a given s .)

Let $V(G) = S = \{s_1, s_2, s_3, \dots, s_n\}$ and (s_i, s_j) is an arc of G if and only if $\text{overlap}(s_i, s_j) = l - 1$. Then, a directed Hamilton path gives an s .

(*) Clearly, it is a very difficult problem to solve.

- **Eulerian Path Idea**

From the set of l -mars we obtain a set T of $(l - 1)$ -mars, $T = \{t_1, t_2, t_3, \dots, t_m\}$. Then let $\hat{G}$ be the graph obtained from letting $V(\hat{G}) = T$ and t_i is incident to t_j if and only if $\text{overlap}(t_i, t_j) = l - 2$, and $t_i \wedge t_j$ is an l -mar in S . Here, the join of t_i and t_j , $t_i \wedge t_j$, is obtained by combining them together such that the emerging portion (suffix of t_i and prefix of t_j) occurs exactly once. For example, $ATG \wedge TGG = ATGG$.

Example of Eulerian Path

- Let $S = \{\text{ATG, TGC, GCG, CGT, GTG, TGG, GGC, GCA}\}$ be the set of 3-mers.

Then, the set of 2-mers obtained from S are $\{\text{AT, TG, GC, CG, GT, GG, CA}\}$. Now, from AT to TG we have 3-mar **ATG**, TG to GC we have **TGC**, From GC to CG we have **GCG**, ..., GC to CA we have **GCA**.

- An Eulerian path from AT to CA covers all 3-mers (exactly once for each 3-mar).
- It is possible to have two or more different Eulerian paths constructed from given set of l-mers. From above S , we have **ATGGCGTGCA** and **ATGCGTGGCA** at least.

Enumerating Eulerian circuits

- In real world, the I–mars we consider is around 500 to 700 letters.
- Finding Eulerian circuits is easier than find Hamilton cycles of a graph.

Review of Graph Structures

Theorem (Graph version)

A connected graph has an Eulerian circuit if and only if each vertex is of even degree.

Theorem (Digraph version)

A connected digraph has a directed Eulerian circuit if and only if the outdegree and indegree of each vertex are equal.

Enumeration Problem

- Two Eulerian circuits (or trails) are said to be the same if they can be obtained from each other by using **cyclic shift**.
- For example, in 4-cycle (u, v, w, z) , (u, v, w, z) and (v, w, z, u) are two Eulerian circuits which are the same.

Problem (Open)

Given an Eulerian graph, find the number of different Eulerian circuits.

Exercise

Find existent results and **possible new (?)** results of the above problem.

Fragment Assembly Algorithm

- There are three steps:
 1. **Overlap**: Finding potentially overlapping reads
 2. **Layout**: Finding the order of the reads along DNA
 3. **Consensus**: Deriving the DNA sequence from the layout
- Note. The error rate in DNA reads produce by modern machines varies from 1% to 3%.
- There are repeats, for example, roughly 300 nucleotide “**Alu sequence**” (GAA-Alu) is repeated more than a million times throughout the genome.

Set Cover Problem

- Set Cover Problem

Let $S_1, S_2, \dots, S_k$ be subsets of S with $\text{wt}(S_i) = w_i$, $i = 1, 2, \dots, k$, and the union of these k sets contains S . Find a subset of $\{1, 2, \dots, k\}$, I , such that (1) the union of all sets S_j , $j \in I$, contains S , and (2) the sum of their weights is minimum.

Example $S = [1, 6]$, $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}$, $S_3 = \{2, 3\}$, $S_4 = \{2, 4, 6\}$, $S_5 = \{3, 5, 6\}$, $S_6 = \{4, 5, 6\}$, $S_7 = \{4, 5\}$, $\text{wt}(S_i) = |S_i|$.

Answer: $I = \{2, 3, 6\}$ gives an answer with total weight 7. Can we find a better answer?

Hitting Set Problem

- Hitting set problem

Given a collection of sets $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$, find a set with minimum size T (a subset of the union of sets in $\mathbf{S}$) such that T intersects each set in $\mathbf{S}$.

- Set cover problem mentioned in previous page can be converted into a hitting set problem of a bipartite graph (A, B) where A is the union of sets in $\mathbf{S}$, and $B = \mathbf{S}$. Now, we are looking to find a subset of B , B' , such that the neighborhood of B' , $N(B')$ contains A and the degree sum of vertices in B' is minimized.

Real World Problem

- IBM finds around 5,000 computer viruses. Over there, 9,000 substrings of 20 or more consecutive bytes from viruses are not found in **good** codes.
- A set cover of 180 substrings was found to cover the viruses. It suffices to search for 180 substrings to verify the existence of known viruses.
- **Exercise** Find an algorithm of the set cover problem.

Note. How good can a **greedy algorithm** do?

Greedy Algorithm for Set Cover Problem

- C : the set of elements covered so far.
- $\sigma(T)$: average cost per newly covered node T (new input set of $\mathbf{S}$).
- S is the union of all sets in $\mathbf{S}$.

Algorithm (For reference)

1. $C \leftarrow \phi$
2. While $C \neq S$ or C is a proper subset of S do
Find the set T whose cost effectiveness $\sigma(T) = \text{wt}(T) / |T \setminus C|$ is smallest.
3. $C \leftarrow C \cup T$
4. Output picked sets

Linear Programming

- Definition

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

- Example

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

- If maximization is considered, then the constraints are of “ $\leq$ ” sign.

Continued

- Use Linear Programming to find the solution shortest superstring of the following collection of strings: $S = [1, 6]$, $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}$, $S_3 = \{2, 3\}$, $S_4 = \{2, 4, 6\}$, $S_5 = \{3, 5, 6\}$, $S_6 = \{4, 5, 6\}$, $S_7 = \{4, 5\}$, $\text{wt}(S_i) = |S_i|$.
- Consider the occurrence of 1, 2, ..., 6.

$$x_1 + x_2 \geq 1 \text{ (1 has to be included)}$$

$$x_1 + x_3 + x_4 \geq 1 \text{ (2 has to be included)}$$

$$x_2 + x_3 + x_5 \geq 1$$

$$x_4 + x_6 + x_7 \geq 1 \quad \dots\dots\dots$$

$$x_5 + x_6 + x_7 \geq 1$$

$$x_4 + x_5 + x_6 \geq 1$$

$$\text{Minimize } \sum_{j=1 \text{ to } 7} c_j x_j \text{ where } c_i = |S_i|.$$

Exercise (1 – 5)

1. Find an algorithm for **alignment**. How about **multiple sequences alignment**?
2. Find an algorithm for finding **longest common subsequences** of two given sequences.
3. Given a permutation π of S_n , find an algorithm to represent this permutation by a pair tableaux P and Q. Also, prove that the length of the longest increasing subsequence of π is the length of the first row of P.
4. Find existent results and **possible new (?)** results of counting the number of distinct Eulerian circuits is an Eulerian graph.
5. Find an algorithm of **shortest superstring problem**.