Bioinformatics
(Shotgun Sequencing)

Marc Hellmuth

Lecture 2
DNA sequencing = determine the sequence of nucleotides in DNA.

<table>
<thead>
<tr>
<th>Organism type</th>
<th>Organism</th>
<th>Genome Size (bp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virus</td>
<td>Porcine circovirus type 1</td>
<td>1,759</td>
</tr>
<tr>
<td>Virus</td>
<td>Pandoravirus salinus</td>
<td>2,470,000</td>
</tr>
<tr>
<td>Bacterium</td>
<td>Nasuia deltocephalinicola</td>
<td>112,091</td>
</tr>
<tr>
<td>Bacterium</td>
<td>Solibacter usitatus</td>
<td>9,970,000</td>
</tr>
<tr>
<td>Plant</td>
<td>Genlisea tuberosa</td>
<td>61,000,000</td>
</tr>
<tr>
<td>Plant</td>
<td>Paris japonica</td>
<td>150,000,000,000</td>
</tr>
<tr>
<td>Mammal</td>
<td>Mus musculus</td>
<td>2,700,000,000</td>
</tr>
<tr>
<td>Mammal</td>
<td>Homo sapiens</td>
<td>3,200,000,000</td>
</tr>
<tr>
<td>Fish</td>
<td>Tetraodon nigroviridis</td>
<td>385,000,000</td>
</tr>
<tr>
<td>Fish</td>
<td>Protopterus aethiopicus</td>
<td>130,000,000,000</td>
</tr>
<tr>
<td>Amoeboid</td>
<td>Polychaos dubium</td>
<td>670,000,000,000</td>
</tr>
</tbody>
</table>

(Wikipedia)

**Problem:** Current methods allow to read strings of length up to 1100bp
DNA sequencing = determine the sequence of nucleotides in DNA.

One way to do this: Shotgun sequencing

**Idea:**
Break multiple copies of string (DNA) into shorter substrings

**Example:**
shotgunsequencing  shotgunsequencing
shotgunsequencing

**Computing problem:** Assemble string shotgunsequencing

For us: Find a shortest common superstring (SCS)
We will consider a GREEDY strategy and show that GREEDY produces a superstring of length at most $4n$ where $n$ is the length of shortest superstring.

Approximation vs. Heuristics

- Performance guarantee
- Better ratio usually indicates better heuristic
- Approximation provides a good starting point for local-optimization
- Approximation provides good estimation of the optimal solution, which is useful for branch-and-bound
\( P = \{s_1, \ldots, s_n\} \) is a set of strings.

For \( s, t \in P \) let \( v \) be longest string (overlap) such that \( s = uv \), \( t = vw \), \( u, w \neq \emptyset \).

\[ \text{ov}(s, t) = |v|. \]

IDEA: GREEDY takes in each step two strings \( s, t \) that have maximal overlap \( \text{ov}(s, t) \) and merges them to \( \langle st \rangle := uvw \).

Example 1.

\( s_1 = \text{ACCT}, \ s_2 = \text{CCTT}, \ s_3 = \text{TACC}. \)

\[
\begin{align*}
\text{ov}(s_1, s_1) &= 0 \\
\text{ov}(s_2, s_2) &= 0 \\
\text{ov}(s_3, s_3) &= 0 \\
\text{ov}(s_1, s_2) &= 3 \\
\text{ov}(s_2, s_1) &= 0 \\
\text{ov}(s_1, s_3) &= 1 \\
\text{ov}(s_3, s_1) &= 3 \\
\text{ov}(s_2, s_3) &= 1 \\
\text{ov}(s_3, s_2) &= 2 \\
\end{align*}
\]

1. \( \langle s_1 s_2 \rangle = \text{ACCTT} \)

2. \( \langle s_3 s_1 s_2 \rangle = \text{TACCTT} \)
Example 2

\[ P' = \{ \text{alf ate half lethal alpha alfalfa} \} \text{ not substring free.} \]

\[ P = \{ \text{ate half lethal alpha alfalfa} \} \text{ substring free.} \]

Trivial superstring \( S(P) \) is \( \text{atehalethalalphaalfalfa} \) of length 25.

A shortest common superstring (SCS) \( S^*(P) \) is \( \text{A is lethalhalfalfal} \) of length 17.

**GREEDY:**

largest overlaps from \( \text{lethal to half to alfalfa} \) producing \( \text{lethalfalfalfa} \)

Then, has 3 choices of single character overlap. One possible solution: \( \text{lethalfalfalphe} \)
Why does this work and how “good” is the GREEDY solution?
For this:
  • Cyclic Strings and Cycle Covers
  • Hamiltonian cycles in directed graphs
⇒ blackboard
Overlap and Distance Graph (from Example 2)

All edges not shown have overlap 0.
Note, the sum of the distance and overlap weights on an edge \((S_i, S_j)\) is the length of the string \(S_i\).
Taken from: Avrim Blum, Tao Jiang, Ming Li, John Tromp, and Mihalis Yannakakis. 1994. Linear approximation of shortest superstrings. J. ACM 41, 4, 630-647.
Summary blackboard

\[ P = \{s_1, \ldots, s_n\} \text{ is a set of strings.} \]

Find permutation \( \Pi = \sigma_1 \ldots \sigma_k \) minimizing

\[ |S(\Pi)| = \sum_{i=1}^{k-1} p(S_{\sigma_i}, S_{\sigma_{i+1}}) + |S_{\sigma_k}| \]

is equivalent to find \( \Pi \) maximizing

\[ \sum_{i=1}^{k-1} \text{ov}(S_{\sigma_i}, S_{\sigma_{i+1}}) \]
Summary blackboard

\[ P = \{s_1, \ldots, s_n\} \] is a set of strings.

\[ C(P) = \text{set of cyclic strings s.t. each } S \in P \text{ maps to at least one } \Phi \in C(P) \text{ is called cycle cover.} \]

\[ C^*(P) \] denotes cycle cover of minimum length and \[ \text{OPT}(S) = S^*(P) \] denotes SCS for \( P \).

**Lemma**

\[ \|C^*(P)\| \leq |S^*(P)|. \]
Summary blackboard

\( P = \{s_1, \ldots, s_n\} \) is a set of strings.

Associate each string \( S \in P \) with exactly one \( \Phi \in C^*(P) \) that \( S \) maps to and denote with \( P_\Phi \subseteq P \) the set of strings associated with \( \Phi \).

For \( \Phi \in C^*(P) \) let \( L_\Phi = \sigma_1 \ldots \sigma_t \) be indices of strings in \( P_\Phi \) in order of starting positions in \( \Phi \).

For a cyclic shift \( L'_\Phi \) of \( L_\Phi \) with \( \sigma_i \) as last index in that ordering we call \( S_{\sigma_i} \) final string.

Lemma

If \( S_{\sigma_i} \) is final string of \( L'_\Phi \), then
\[
|S(L'_\Phi)| = |\Phi| + \text{ov}((S_{\sigma_i}, S_{\sigma_{i+1}}) \leq |\Phi| + |S_{\sigma_i}| \text{ where } t + 1 \text{ is taken to be } 1.
\]
Algorithm **ConcatCycle**

1. Find minimum length cycle cover $C^*(P)$ of $P$ and associate each string $S \in P$ with exactly one $\Phi \in C^*(P)$ that $S$ maps to.
2. For every cyclic string $\Phi \in C^*(P)$ form ordered list $L_\Phi$ and create $S(L_\Phi)$. Let $P'$ be set of superstrings obtained in this step.
3. Concatenate the strings in $P'$ in any order to obtain superstring $H$.

Let $P_f$ be the set of final strings of the strings contained in $P'$.

Lemma

$|H| \leq ||C^*|| + \sum_{S \in P_f} |S|$
Summary blackboard

Theorem (GCD Theorem)
If string $S$ has two periods of length $p$ and $q$ and $|S| \geq p + q$, then $S$ has a period of length $\gcd(p, q)$.

Lemma (Overlap Lemma)
Let $\Phi, \Phi' \in C^*(P)$ and $\alpha, \alpha'$ be any two strings that map to $\Phi$, resp., $\Phi'$. Then, $ov(\alpha, \alpha') \leq |\Phi| + |\Phi'|$

Theorem
Let $H$ be the superstring for string set $P$ obtained by Algorithm ConcatCycle. Then, $|H| \leq 4|S^*(P)|$. 
Summary blackboard

Algorithm **MGreedy**

**INPUT**: \( P \) and \( T = \emptyset \)

1. **WHILE** \( P \neq \emptyset \) **DO**
   
   Choose \( s, t \in P \) (not nec. distinct) with maximum \( ov(s, t) \)
   
   /*breaking ties arbitrarily*/
   
   IF \( s \neq t \) THEN \( P \leftarrow P \setminus \{s, t\} \cup \langle s, t \rangle \)
   
   ELSE \( P \leftarrow P \setminus \{s\} \) and \( T \leftarrow T \cup \{s\} \)

2. **OUTPUT**: Concatenation of strings in \( T \).

Algorithm **MGreedy** can be considered as method that stepwisely takes edges from the overlap graph \( (V = P, E = P \times P, ov(,)) \) with maximum weight and thus creates/joins paths and connects them to cycles. Thus, we get a cycle cover (with possibly none min. weight)

**Theorem**

*The cycle cover obtained by Algorithm **MGreedy** is optimal.*

**MGreedy** runs in \( O(|P|^3) \) time
The cycle cover obtained by Algorithm $\text{MGreedy}$ is optimal.

**Proof-sketch:**
Let $N$ be optimal having max.nr. of edges in common with $M$.
Need to show $N = M$.
Let $e$ be an edge with max.overlap in $M \Delta N$.
*Ties are broken in the same way.*

1st case: $e \in N \setminus M \Rightarrow \text{MGreedy}$ has not chosen $e$, and thus has taken another edge $f$ that dominates $e$. Note $f \notin N$ (since each vertex contained in exactly one cycle)
$\Rightarrow f \in M \setminus N$ contradicting our choice of $e$.

2nd case: $e \in M \setminus N$. Let $e = (k, j)$. Thus $(k, l), (i, j) \in N \setminus M$ and by choice of $e$:
$\text{ov}(k, j) \geq \max\{\text{ov}(k, l), \text{ov}(i, j)\} \Rightarrow \text{ov}(k, j) + \text{ov}(i, l) \geq \text{ov}(i, j) + \text{ov}(k, l)$. $\Rightarrow$
Replacing in $N$ the edges $(k, l), (i, j)$ by $(k, j), (i, l)$ yield assignment $N'$ that has more edges in common with $M$ and not less overlap, contradicting our choice of $N$. 
Conjecture: The Greedy Algorithm has approximation factor 2. The best know approximation ratio is $2\frac{11}{23} \approx 2.48$ (Mucha, 2013)

The problems in practice:

- *Repeated regions*. Repeats are difficult to separate and often cause the fragment assembly program to assemble reads that come from different locations.

- *Base-calling errors or sequencing errors*. The limitation in current sequencing technology results in varying quality of the sequence data between reads and within each read.

- *Contamination*.

- *Unknown orientation*. It is not known from which strand each fragment originates. This increases the complexity of the assembly task. Hence, a read may represent one strand or the reverse complement sequence on the other strand.

- *Incomplete coverage*. Coverage varies in different target sequence locations due to the nature of random sampling. The coverage has theoretically a certain probability to be zero depending on the average sampling coverage of the target genome.