Bioinformatics
(Shotgun Sequencing)

Marc Hellmuth

Lecture 2

DNA sequencing = determine the sequence of nucleotides in DNA.

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

(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
```

```
cing en encing equ gun ing ns otgu seq sequ sh
sho shot tg uenc un
```

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 $s = uv$, $t = vw$, $u, w \neq \emptyset$.

$\mathrm{ov}(s, t) = |v|$.

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

Simple Example 1.

$s_1 = \mathtt{ACCT}$, $s_2 = \mathtt{CCTT}$, $s_3 = \mathtt{TACC}$.

$\mathrm{ov}(\mathrm{s}_1, \mathrm{s}_2) = 3$ $\mathrm{ov}(\mathrm{s}_2, \mathrm{s}_1) = 0$
$\mathrm{ov}(\mathrm{s}_1, \mathrm{s}_3) = 1$ $\mathrm{ov}(\mathrm{s}_3, \mathrm{s}_1) = 3$
$\mathrm{ov}(\mathrm{s}_2, \mathrm{s}_3) = 1$ $\mathrm{ov}(\mathrm{s}_3, \mathrm{s}_2) = 2$

1. $\langle s_1 s_2 \rangle = \mathtt{ACCTT}$
2. $\langle s_3 \langle s_1 s_2 \rangle \rangle = \mathtt{TACCTT}$

Example 2

$P' = \{$`alf ate half lethal alpha alfalfa`$\}$ not substring free.

$P = \{$ `ate half lethal alpha alfalfa`$\}$ substring free.

Trivial superstring $S(P)$ is `atehalethalalphaalfalfa` of length 25.

A shortest common superstring (SCS) $S^*(P)$ is A is `lethalphalfalfate` of length 17.

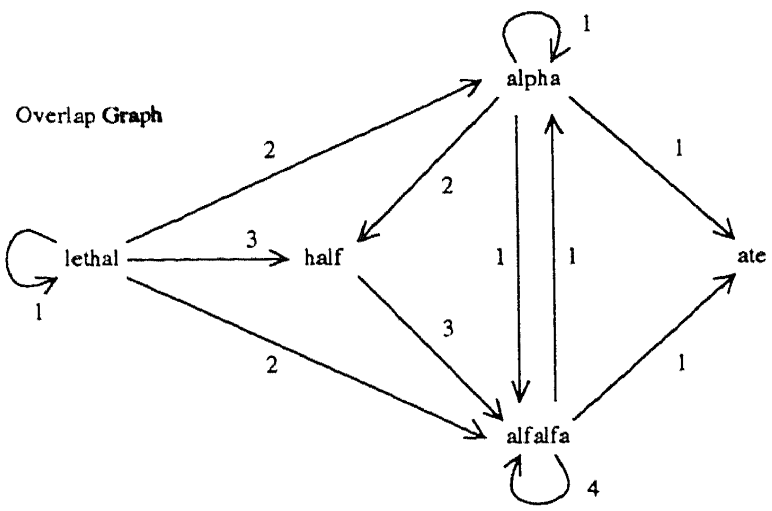
GREEDY:
largest overlaps from `lethal` to `half` to `alfalfa` producing `lethalfalfa`
Then, has 3 choices of single character overlap. One possible solution: `lethalfalfalphate`

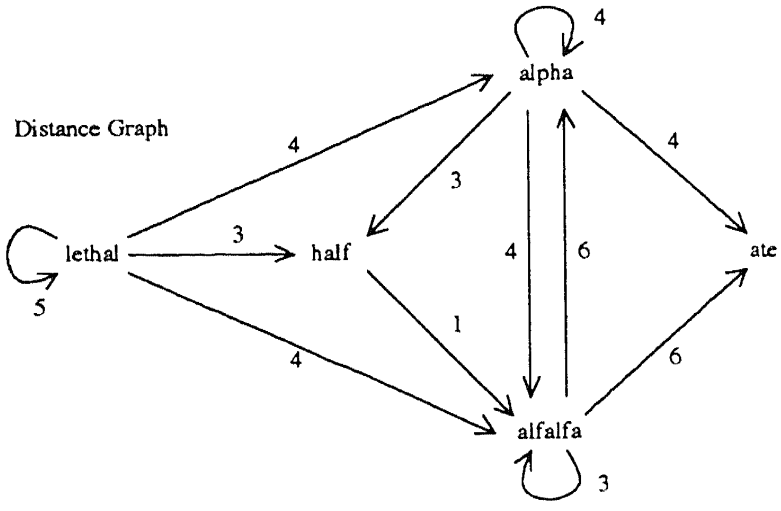
Why does this work and how “good” is the GREEDY solution?
For this:

- Cyclic Strings and Cycle Covers
- Hamiltonian cycles in directed graphs

$\Rightarrow$ 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

$P = \{s_1, \ldots, s_n\}$ 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 Π maximizing

$$\sum_{i=1}^{k-1} \mathrm{ov}(S_{\sigma_i}, S_{\sigma_{i+1}})$$

Summary blackboard

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

$C(P) =$ set of cyclic strings s.t. each $S \in P$ maps to at least one
$\Phi \in C(P)$ is called *cycle cover*.

$C^*(P)$ denotes cycle cover of minimum length and
$\mathrm{OPT(S)} = S^*(P)$ 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 Φ.

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

For a cyclic shift L'_Φ of L_Φ with σ_i as last index in that ordering we call S_{σ_i} *final string*.

Lemma

If S_{σ_i} is final string of L'_Φ, then
$|S(L'_\Phi)| = |\Phi| + \mathrm{ov}((S_{\sigma_i}, S_{\sigma_{i+1}}) \leq |\Phi| + |S_{\sigma_i}|$ *where $t+1$ is taken to be* 1.

Summary blackboard

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_Φ 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 α, α' be any two strings that map to Φ, resp., Φ'. Then, $\mathrm{ov}(\alpha,\alpha') \leq |\Phi| + |\Phi'|$

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 $\mathrm{ov}(s, t)$
   */*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 $(V = P, E = P \times P, \mathrm{ov}(,))$ 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

Summary blackboard

The cycle cover obtained by Algorithm `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: $e \in N \setminus M \Rightarrow$ `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 von 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 coice of e: $\mathrm{ov}(k,j) \geq \max\{\mathrm{ov}(k,l), \mathrm{ov}(i,j)\} \Rightarrow \mathrm{ov}(k,j) + \mathrm{ov}(i,l) \geq \mathrm{ov}(i,j) + \mathrm{ov}(k,l). \Rightarrow$ Replacing in N the edges $(k,l),(i,j)$ by $(k,j),(i,l)$ yield assigmnet 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 $2\frac{11}{23} \simeq 2.48$ (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.