FRAGMENT ASSEMBLY

based on chapter 4 of Setubal, Meidanis:
Introduction to Computational molecular biology
and Blum et.al:
Linear Appriximation of Shortest Superstrings

Motivation

The shotgun method gives a number of fragments from an unknown position of an unknown strand.

Typical situation:

- Target sequence of 30−100 kbp (known within 10%)
- 500−2000 fragments
- Each fragment 200−700 bp long

Example:

input:

```
ACCGT
CGTGC
TTAC
TACCGT
```

Layout:

```
--ACCGT--
----CGTGC
TTAC-----
-TACCGT--
```

Target sequence: (?)

```
TTACCGTGC
```

Complications

- Errors
  - base call errors
  - chimeric fragments
  - contamination
- Unknown orientation
- Repeated regions
- Lack of coverage

Base call errors

- 1–5 errors per 100 bp
- concentrated at one end
- majority voting

```
ACCGT
CGTGC
TTAC
TGCCGT
```

```
--ACCGT--
----CGTGC
TTAC-----
-TGCCGT--
_________

TTACCGTGC
```

Chimeric fragments

- Two fragments from different parts joins
- Can be detected in preprocessing if only one of its kind

Contamination

- Unrelated DNA−fragments in input
- From host used for copying
- Detected at preprocessing since host DNA known

Unknown orientation

Two posibilities:

- As given directed from 5’ to 3’ AACTG
- Other strand, reversed complement CAGTT

Testing all combinations would be exponetiall

Repeated Regions

Not a problem if fragment exists that cover the entire repeat.

Most difficult are *inverted repeats*

```
     --AACTGCCTAGCTCAGTT--
f1:       TGCCTA
f2:             TAGCTCA
```

or

```
     --AACTGAGCTAGGCAGTT--
f2:       TGAGCTA
f1:             TAGGCA
```

Lack of coverage

Definitions:
•*Coverage* of a position is # overlapping fragments at that point
•*Mean coverage*
•*Contigs* is continously covered areas

A high mean coverage can avoid gaps (in practice 8 is "high")

Shotgun method is random but *directed sequencing* to fill gaps is possible but expensive

Desirable to have entire sequence covered with fragments from both strands as one strand can be prone to errors

Lack of coverage

formulas:

Number of expected contigs = $ne^{-n(l-t)/T}$

Expected fraction covered by *exactly* k fragments = $\frac{e^{-c} c^{k}}{k!}$

n	=	#sampeled fragments
l	=	length of each fragment
t	=	needed overlap to be recognized as such
T	=	Length of molecule
c	=	mean coverage (nl/T)

Alternative methods for DNA sequencing

- *Direct sequencing* to fill gaps, can build from end of gap using primer
- *Dual end sequencing* can sequence ends of longer sequences, gets approximate distance
- *Sequencing by hybridization (SBH)* tests target sequence for existence (only) of a k−tuple for all k−tuples ($k \leq 8$)

Models

- Shortest common superstring
- Reconstruction
- Multicontig

Increasingly "better" but more difficult to compute

Shortest common superstring

Given: F = {S_1,S_2,...,S_n}

Find: A shortest string S with strings of F as substrings

Problems: Assumes no errors and known orientations. Shortests may also be wrong, repeats are not well handled (if sequence repeated many times all all fragments from those parts may be concentrated at only one instance − no desire to get "even coverage")

NP−hard

Reconstruction

Given: F = $\{S_1, S_2, ..., S_n\}$ and error tolerance ε

Find: A shortest string S with strings of F or their reverse complement as approximate substrings

approximate = edit distance $\leq \varepsilon |S_i|$ ignoring gaps at ends

Problems: Still problem with repeats and coverage and actual size of target not taken into account
and NP−hard...

Multicontig

Given: F = $\{S_1, S_2, ..., S_n\}$, error tolerance ε and an integer t

Find: A partition of F into $C_1 ... C_k$ such that every C_i admits a t−contig with ε−consensus

t−contig = overlaps by at least t

ε−consensus = every fragment S_i has edit distance $\leq \varepsilon |S_i|$ to consensus string

Problems: Size of target and some repeats and NP−hard...

Problem Formulation

- **Given:** Strings $S_1, S_2, ..., S_n$ over a finite alphabet Σ.
- **Find:** Shortest string S containing each S_i.

```
ATAT          ..ATAT....
TATT          ...TATT...
TTAT          .....TTAT.
TATA          ......TATA
TAAT          TAAT......
AATA          .AATA.....
              TAATATTATA
```

- Strings $S_1, S_2, ..., S_n$ are substringfree.
- NP-hard. Transformation from Vertex Cover for cubic graphs.
- Remains NP-hard if all strings have up to 8 letters and contain no repeated letters.
- Solvable in polynomial time if all strings have at most 2 letters.

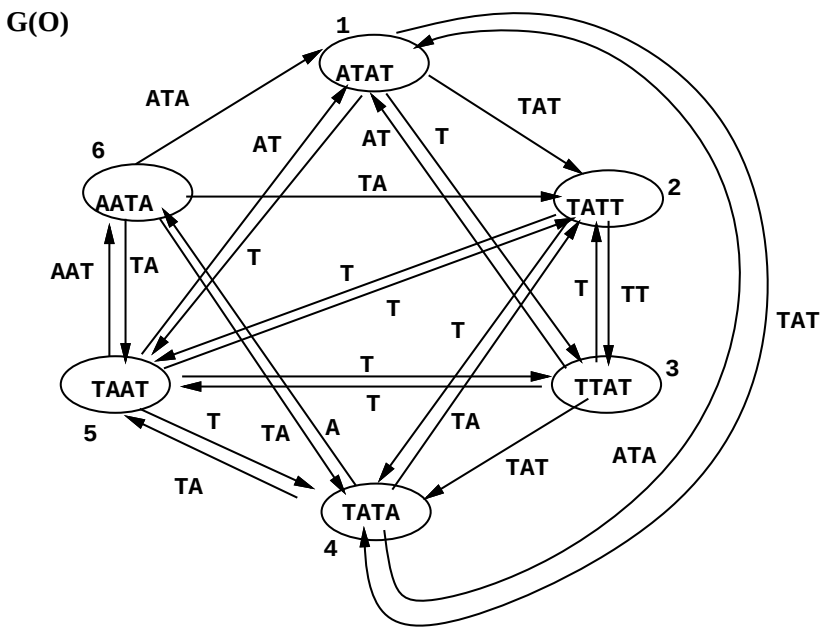
Basic Definitions

- Let s and t be two, not necessarily distinct, strings.
- The *overlap* between s and t is the longest string v such that $s = uv$ and $t = vw$. The length of the overlap is $|v|$ and is also denoted by $o(s, t)$.

```
UNDERGROUND          u=UNDERGRO    v=UND    w=ERSTAND
      UNDERSTAND
```

- Substring u is called the *prefix* of s with respect to t. It is denoted by $p(s, t)$, and $d(s, t) = |p(s, t)|$.
- $o(s, t) + d(s, t) = |s|$.

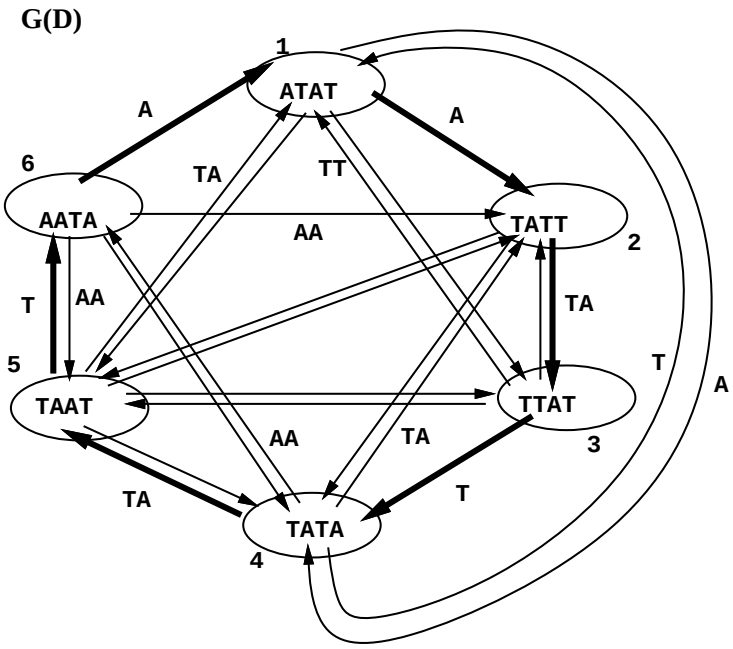
Overlap Graphs [1]



$$O = \begin{pmatrix} * & 3 & 1 & 3 & 1 & 0 \\ 0 & * & 2 & 1 & 1 & 0 \\ 2 & 1 & * & 3 & 1 & 0 \\ 3 & 2 & 0 & * & 2 & 1 \\ 2 & 1 & 1 & 1 & * & 3 \\ 3 & 2 & 0 & 2 & 2 & * \end{pmatrix} \quad O^* = \begin{pmatrix} 2 & 3 & 1 & 3 & 1 & 0 \\ 0 & 1 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 3 & 1 & 0 \\ 3 & 2 & 0 & 2 & 2 & 1 \\ 2 & 1 & 1 & 1 & 1 & 3 \\ 3 & 2 & 0 & 2 & 2 & 1 \end{pmatrix}$$

[1] Not all edges shown.

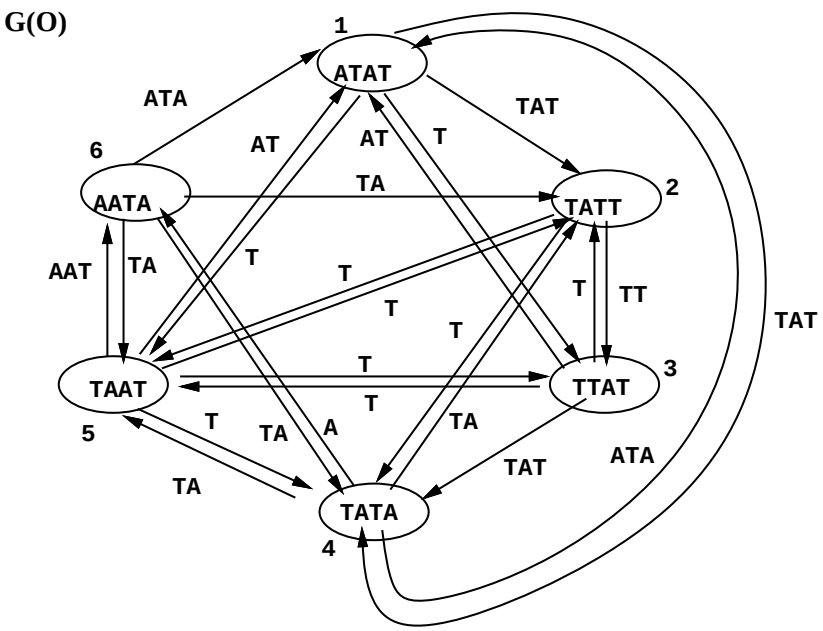
Distance Graphs [2]



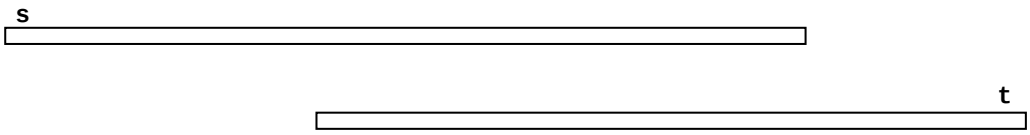
$$D = \begin{pmatrix} * & 1 & 3 & 1 & 3 & 4 \\ 4 & * & 2 & 3 & 3 & 4 \\ 2 & 3 & * & 1 & 3 & 4 \\ 1 & 2 & 4 & * & 2 & 3 \\ 2 & 3 & 3 & 3 & * & 1 \\ 1 & 2 & 4 & 2 & 2 & * \end{pmatrix}$$

[2] Not all edges shown.

Greedy Algorithm



- $S = \{S_1, S_2, ..., S_n\}$.
- While $|S| \geq 1$:
  - Select $s, t \in S$, $s \neq t$, with greatest overlap (ties broken arbitrarily). Let $q = p(s,t)t$
  - Add q to S (replacing s and t).
- Overlap of q with remaining strings in S does not need to be recomputed.



- Conjecture:

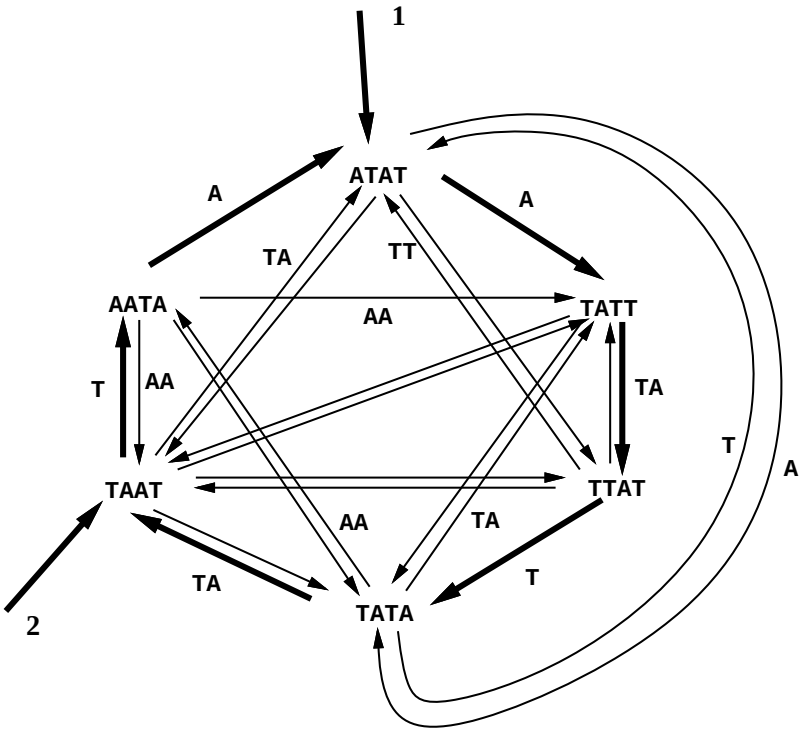
$$\frac{GRD(S)}{OPT(S)} \leq 2$$

Shortest Superstring and TSP

- Solve TSP in the distance graph.
- $TSP(D)$: string associated with the solution.
- A superstring $TSP(S)$ is obtained by breaking $TSP(D)$ after any prefix and adding the rest of its string.
- Assume that strings are ordered $S_1, S_2, ..., S_n$ in $OPT(S)$.

$$|TSP(D)| \leq |OPT(S)| - o(S_n, S_1) \leq$$

$$|OPT(S)| \leq |TSP(D)| + \min_i\{|S_i|\}$$



START1:

A+TA+T+TA+T+A + ATA

START2:

T+A+A+TA+T+TA + TA

- TSP is NP-hard.
- Since $G(D)$ is directed and asymmetric, no approximation algorithm with constant error bound is available.

Minimum Cycle Cover Algorithm

- Assume that $G(D)$ is given.
- Determine minimum cycle cover $MMC(D)$ of $G(D)$ (can be done in $O(n^3)$ when $G(D)$ is given).
- Open up cycles and concatenate to obtain a solution $MCC(S)$.

$$|MCC(D)| \leq |TSP(D)| \leq |OPT(S)|$$

$$D = \begin{pmatrix} * & 1 & 3 & 1 & 3 & 4 \\ 4 & * & 2 & 3 & 3 & 4 \\ 2 & 3 & * & 1 & 3 & 4 \\ 1 & 2 & 4 & * & 2 & 3 \\ 2 & 3 & 3 & 3 & * & 1 \\ 1 & 2 & 4 & 2 & 2 & * \end{pmatrix} 1--2--3--4--1,5--6--5$$

```
1.    ATAT              5.    TAAT
2.     TATT             6.     AATA
3.       TTAT           5.       TAAT
4.        TATA
1.         ATAT

      ATATT                   TAA
      ATATTATA                TAATA
      ATATTATATAATA
```

Periodicity of Strings

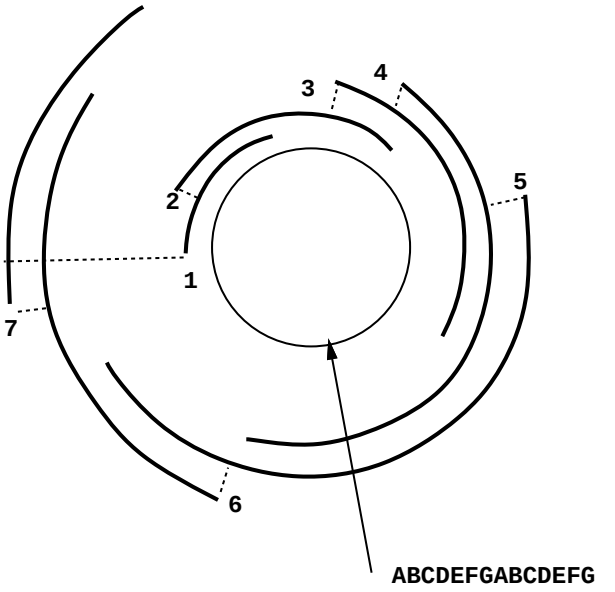
- A string t is *irreducible* if all cycle shifts of t yield different strings.
- Every string s has a unique prefix t such that t is irreducible and $s = t^k$ for some $k \geq 1$.

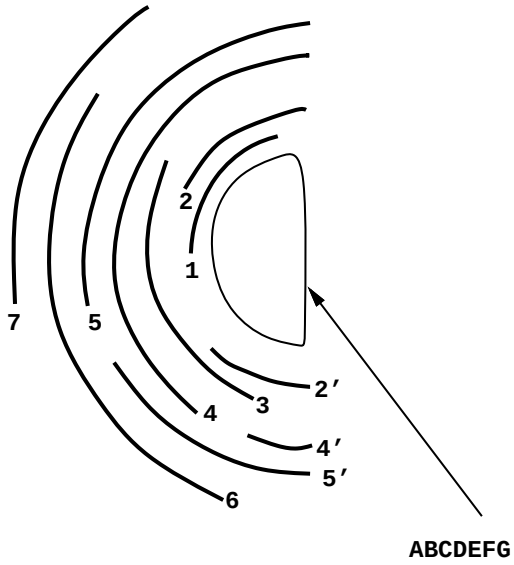
$$TATA = (TA)^2$$

- t is called the *period* of s.

Cycles of MCC are Irreducible

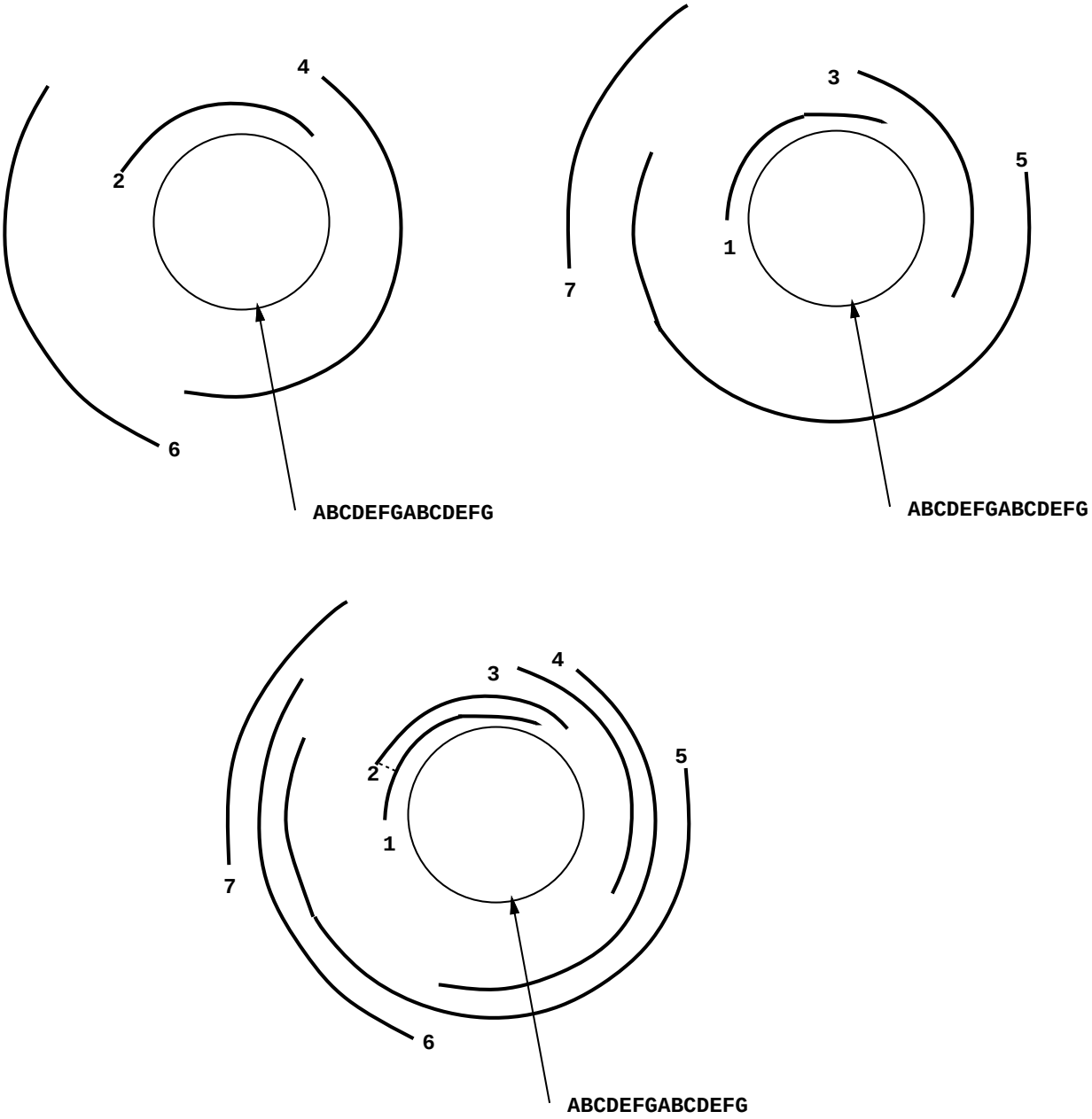
- Strings associated with cycles in $MCC(D)$ are irreducible.





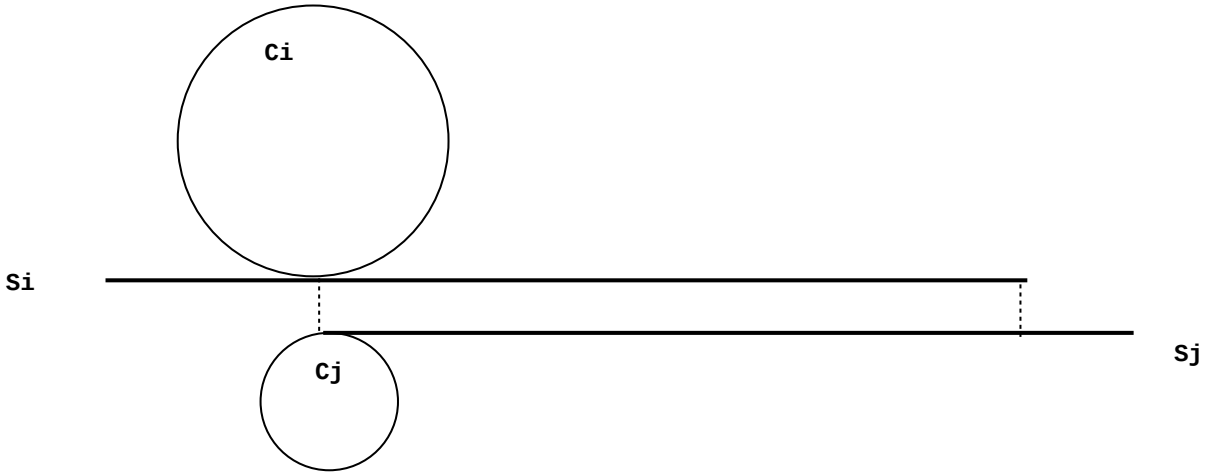
Cycles of MCC are Distinct

- No pair of cycles in $MCC(D)$ has the same strings.



Bounding Length of Cycle Concatenation

- Let C_i and C_j denote two cycles of $MCC(D)$.
- Let S_i be a string in C_i.
- Let S_j be a string in C_j.
- Claim: $o(S_i, S_j) < |C_i| + |C_j|$.
- Proof by contradiction. Assume that $o(S_i, S_j) \geq |C_i| + |C_j|$.
- Case 1: $|C_i| = |C_j|$. Then $C_i = C_j$. This is impossible since C_i and C_j were taken from $MCC(D)$.
- Case 2: $|C_i| > |C_j|$.



- $|C_i|$ divisble by $|C_j|$. Then C_i is reducible, a contradiction.
- Otherwise C_j is reducible.

Error Ratio

$$MCC(D) = \sum_{i+1}^{p} |C_i| \leq |TSP(D)| \leq |OPT(S)|$$

- Let L_i denote the longest string in C_i.
- The overlap between L_i and L_j is less than $|C_i| + |C_j|$.
- Let $L = \{L_1, L_2, ..., L_p\}$ and assume w.l.o.g that $L_1, L_2, ..., L_p$ appear in that order in $OPT(L)$.

$$|OPT(S)| \geq |OPT(L)| \geq \sum_{i=1}^{p}(|L_i| - 2|C_i|) + |C_1| + |C_p| \geq \sum_{i=1}^{p}(|L_i| - 2|C_i|)$$

- Error ratio follows now immediately:

$$|MCC(S)| \leq \sum_{i=1}^{p}(|L_i| + |C_i|) = \sum_{i=1}^{p}(|L_i| - 2|C_i|) + \sum_{i=1}^{p} 3|C_i| \leq$$

$$|OPT(S)| + 3|OPT(S)| = 4|OPT(S)|$$

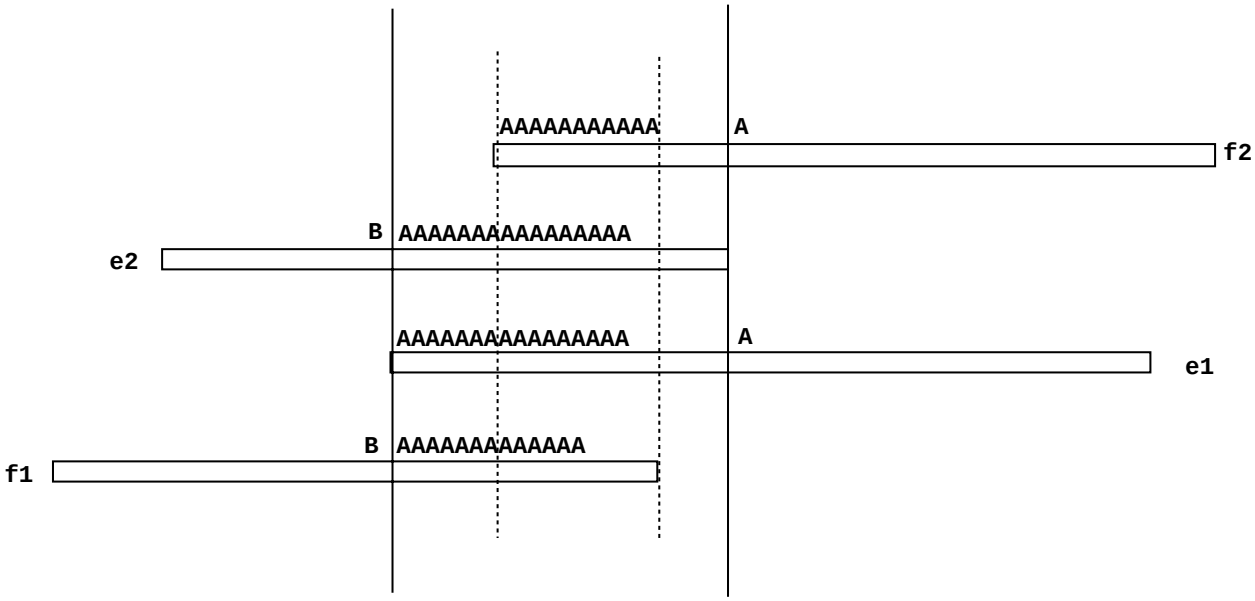
Greedy Cycle Cover Algorithm

- $S = \{S_1, S_2, ..., S_n\}$, $T = \emptyset$.
- While $S \neq \emptyset$:
  - Select $s, t \in S$ ($s = t$ not excluded) with greatest overlap.
  - Remove s and t from S. Let $q = p(s, t)t$.
  - If $s \neq t$, then add q to S.
  - If $s = t$, then add q to T.
- Concatenate strings in T.

Greedy Cycle Cover Algorithm - Example

```
1.     ATAT        2 3 1 3 1 0
2.     TATT        0 1 2 1 1 0     Delete row 1
3.     TTAT        2 3 1 3 1 0     Delete clumn 2
4.     TATA        3 2 0 2 2 1
5.     TAAT        2 1 1 1 1 3
6.     AATA        3 2 0 2 2 1


12.    ATATT       0 2 1 1 0
3.     TTAT        2 1 3 1 0
4.     TATA        3 0 2 2 1
5.     TAAT        2 1 1 1 3
6.     AATA        3 0 2 2 1


12.    ATATT       0 2 1 0
34.    TTATA       3 0 2 1
5.     TAAT        2 1 1 3
6.     AATA        3 0 2 1


3412.  TTATATT     2 1 0
5.     TAAT        1 1 3
6.     AATA        0 2 1


3412.  TTATATT     2 1
56.    TAATA       0 2


341256. TTATATT+TAATA
```

Greedy Cycle Cover = Minimum Cycle Cover



- Consider strings e_1, e_2, f_1, f_2 such that

$$o(e_1, e_2) \geq \max\{o(e_1, f_1), o(e_2, f_2)\}$$

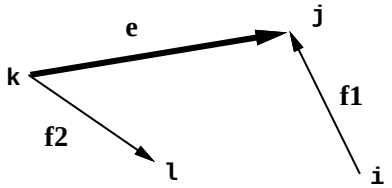
- Then

$$o(e_1, e_2) + o(f_1, f_2) \geq o(e_1, f_1) + o(e_2, f_2)$$

Greedy Cycle Cover = Minimum Cycle Cover

- $N = GCC(D)$, $M = MCC(D)$.
- Assume that $N \neq M$.
- Let e be an edge in the symmetric difference between N and M. Assume that e is chosen such that it has maximum overlap.
- Suppose that $e \in M \setminus N$.
  - There is an edge $f \in N$ sharing either tail or head with e.
  - $f \notin M$ since M is a cycle cover containing e.
  - Selection of f into N implies that f has greater overlap than e. But this contradicts our choice of e.

- Suppose that $e = e_1 = (k, j) \in N \setminus M$.
  - Let $f_1 = (i, j)$ and $f_2 = (k, l)$ be the edges in M.
  - They are not in N, and by the choice of e_1, they are both dominated by e_1.
  - Replacing f_1 and f_2 by e_1 and $e_2 = (i, l)$ yields a cycle cover with no less overlap and with more edges in common with N, a contradiction.



Improving Error Bound

- Consider an algorithm which works as the MCC-algorithm except that in the last step strings are merged using GRD-algorithm.
- It can be shown that this modification leads to an approximation algorithm with error ratio at most 3.
- It can be shown that GRD-algorithm has error ratio at most 4. Complicated proof.
- It has been conjectured that GRD-algorithm has error ratio 2.
- Several approximation algorithms with error ratio below 3 have been suggested. $2\frac{2}{3}$-algorithm is currently the best.
- Interesting generalization:
  - **Given:** A set of positive strings $S = \{S_1, ..., S_n\}$ and a set of negative strings $T = \{T_1, ..., T_m\}$.
  - **Find:** A shortest superstring containing every string from S but no string from T.
- No algorithm with constant error ratio is available.

Sequencing by Hybridization

- 2-dimensional grid of all k-tuples.
- Cloned single-stranded DNA chains are labeled with a radioactive or fluorescent material.
- Each k-tuple present in the sample is hybridized with its reverse complement in the matrix.

Acyclic Graphs

The NP−hardness can be avoided is a *good sampling* is available.

Definitions:

- A sampling of S is a collection A of intervals of S
- Two intervals are linked at level t if overlap ≥ t
- Entire sampling connected at level t if there is a path of intervals linked at level t between every pair in A
- A good sampling is connected at level t and covers S
- Sample is subinterval free if no interval is included in another

Acyclic Graphs

OG(F,t) is a directed graph with the set of fragments F as vertex set and an edge from S_1 to S_2 if the maximum overlap between S_1 and S_2 is $\geq t$.

The weight of the edge is the size of the maximum overlap.

Acyclic Graphs

A false positive at level t is a pair of intervals α and β such that there is a $w \geq t$ and the contents of α and β overlap by w but the interval themselves does not

Lemma: The existence of a false positive of level t implies the existence of a repeat of size $\geq t$

t=3

ATT**GCC**A**GCC**TA

   ---- ----

Acyclic Graphs

Theorem: Let F be a collection generated by a sampling A of S. If OG(F,t) has a directed cycle then there is a repeat in S of size at least t.

Proof idea: there must be at least one false positive at level t

Conclusion: if the sample is "good enough" (at a level higher than the repeats) the graph of overlaps is acyclic and a layout can be easily found by topological sort.

Acyclic Graphs

Example (t=3):

SCS

```
AGTATTGGCAATC---AATCGATG------------
--------------------ATGCAAACCT-----
----TTGGCAATCACT------------CCTTTTGG
```

Length 36

```
ACTATTGGCAATCACTAATCGATGCAAACCTTTTGG
```

Hamiltonian path in OG(F,t)

```
AGTATTGGCAATC--------CCTTTTGG--------
----------AATCGATG---------TTGGCAATCACT
--------------ATGCAAACCT-------------
```

Length 37

```
AGTATTGGCAATCGATGCAAACCTTTTGGCAATCACT
```

Longer string but more likely correct