Fragment assembly of DNA

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Dr. Jaspreet Kaur Dhanjal
Assistant Professor, Center for Computational Biology
Email ID: jaspreet@iiitd.ac.in

March 13, 2024

Complications in the assembly of fragments

The main factors that add to the complexity of the problem of fragment assembly are:

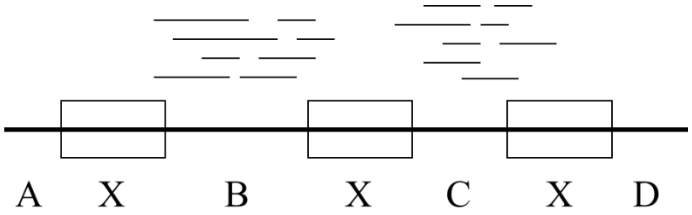
- Error
- Unknown orientation
- Repeated regions
- Lack of coverage

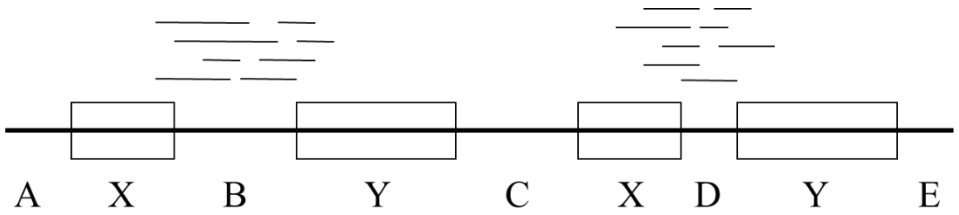
Repeat regions

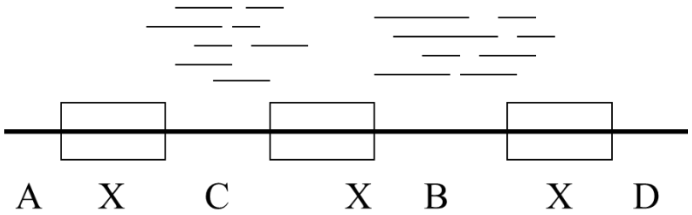
- Repeats are sequences that appear two or more times in the target molecule.
  - Short repeats
  - Longer repeats
- If the level of similarity between two copies of a repeat is high enough, the differences can be mistaken for base call errors.
- If a fragment is totally contained in a repeat, we may have several places to put it in the final alignment. When the copies are not exactly equal, we may weaken the consensus by placing a fragment in the wrong way copy.

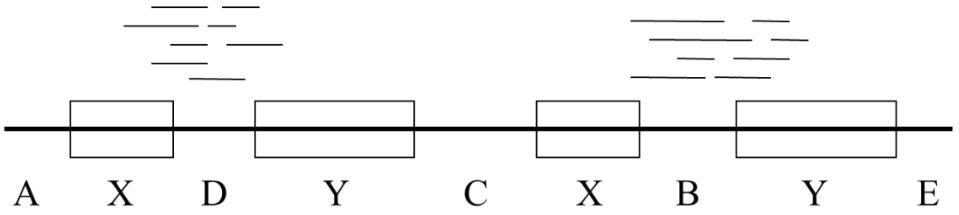
Direct repeats

- Repeats can be positioned in such a way as to render assembly inherently ambiguous.



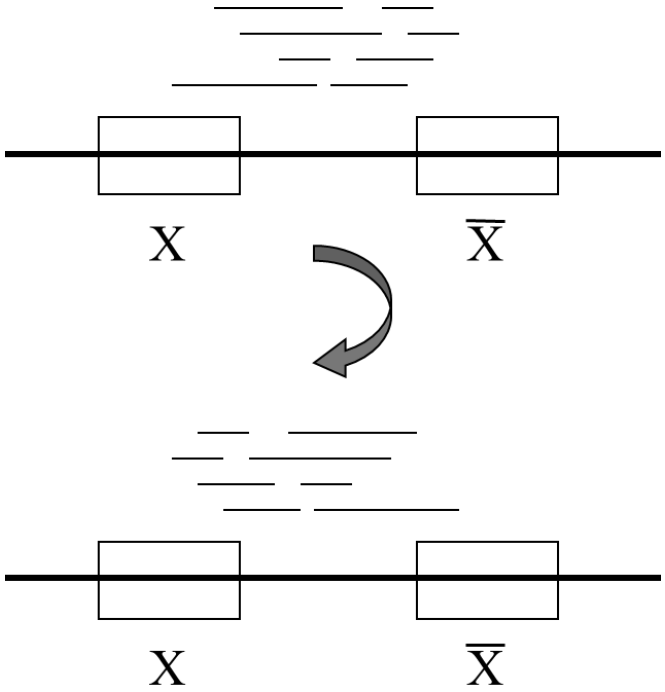




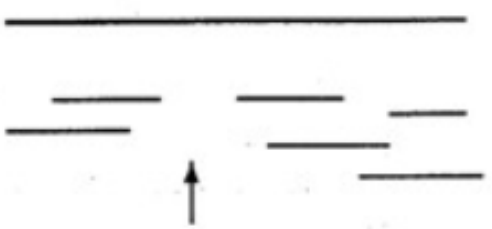


Inverted repeats

- Repeats can be positioned in such a way as to render assembly inherently ambiguous.



Lack of coverage or linkage



Uncovered area

```
------ACTTTT------
TCCGAG------ACGGAC
------ACTTTT------
TCCGAG------ACGGAC
------ACTTTT------
TCCGAG------ACGGAC
TCCGAGACTTTTACGGAG
```

Good coverage but no linkage

- Directed sequencing: a method that can be used to cover small remaining gaps in a shotgun project.
- Problem:
  - It is expensive to build special primers
  - Sequential rather than parallel

Basic models for fragment assembly

- Shortest Common Superstring (SCS)
- RECONSTRUCTION
- MULTICONTIG

All three assume that the fragment collection is free of contamination and chimeras.

Basic models for fragment assembly

- Shortest Common Superstring (SCS)
- RECONSTRUCTION
- MULTICONTIG

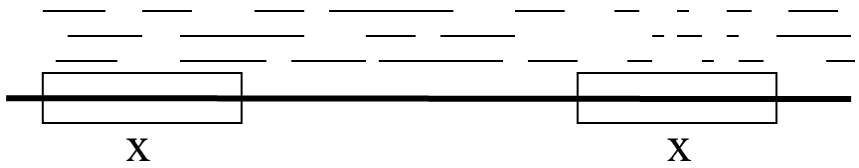
All three assume that the fragment collection is free of contamination and chimeras.

Shortest Common Superstring (SCS)

- Shortest common superstring
- Input: A collection, F, of strings (fragments)
- Output: A shortest possible string S such that for every f ∈ F, S is a superstring of f.
- Example:

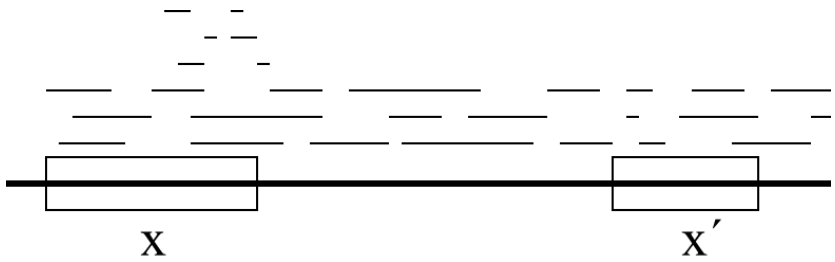
  F = {ACT, CTA, AGT}

  S = ACTAGT



- Problem:

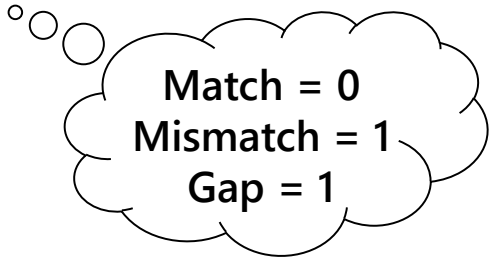
  A superstring may contain only one copy, which will absorb all fragments totally contained in any of the copies



RECONSRUCTION

- Deals with errors and unknown orientation
- Uses dynamic programming
- Uses distance rather than similarity
- Definitions: f is an approximate substring of S at error level ε when $d_s(f, S) \leq \varepsilon \mid f \mid$

  d_s = substring edit distance



One unit of distance is charged for every insertion, deletion or substitution, except for deletion in the extremities of the second sequence

Input: A collection, F, of strings, and a tolerance level, ε

Output: Shortest possible string, S, such that for every $f \in F$

$\min(d_s(f, S), d_s(\bar{f}, S)) \leq \varepsilon \mid f \mid$,

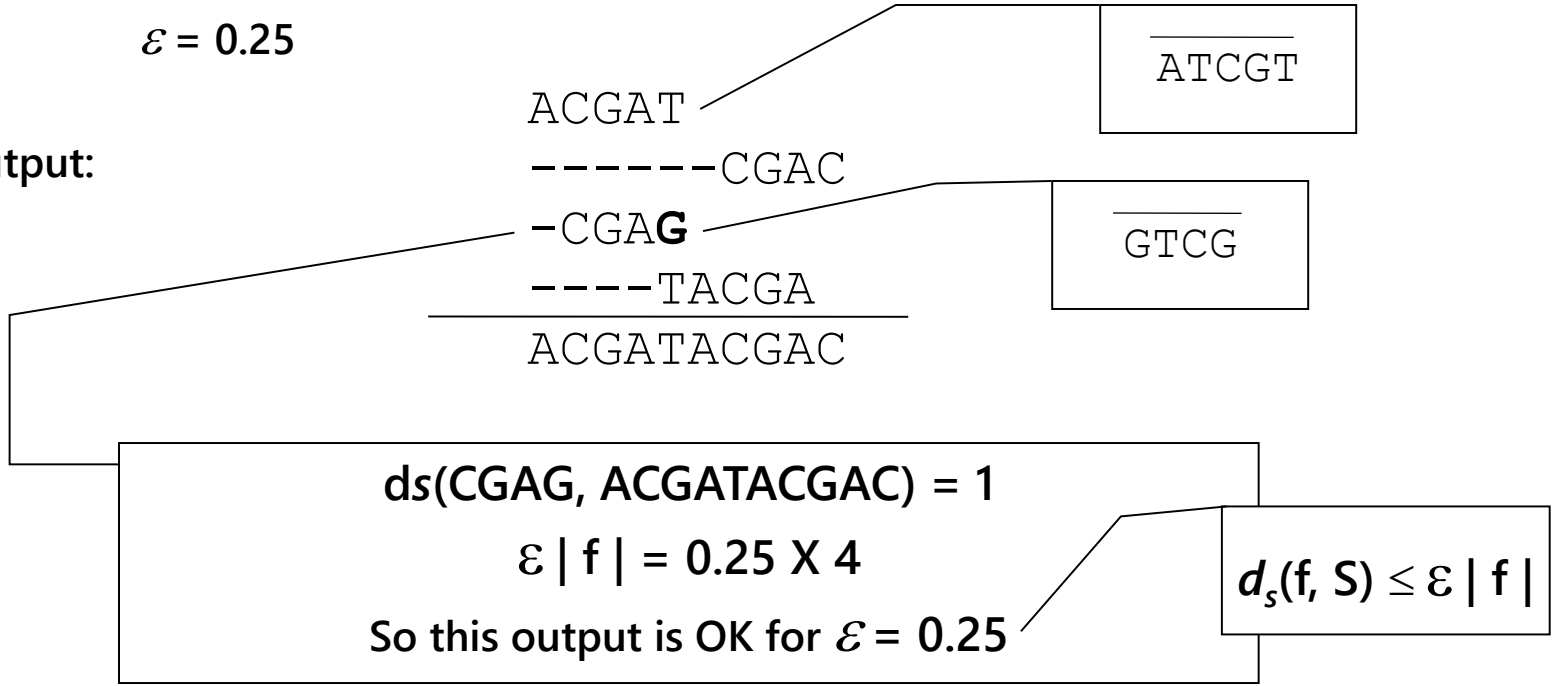
Where $\bar{f}$ is the reverse compliment of f.

RECONSRUCTION

Input: F={ATCGT, GTCG, CGAG, TACGA}

ε = 0.25

Output:



Reconstruction allows gaps in fragments.

Practice question

Which of the fragments is wrongly placed in the following alignment layout for finding the target sequence, provided ε = 0.3?

```
          TATAGCATCAT
    CGTC      CATGATCA
  ACGGATAG     GTCCA
-----------------------
  ACGTATAGCATGATCA
```

Solution:

d_s (TATAGCATCAT, ACGTATAGCATGATCA) = 1
ε| f | = 0.3 X 11 = 3.3
Here, d_s (f, S) ≤ | f |, therefore correct placement

d_s (CGTC, ACGTATAGCATGATCA) = 1
ε| f | = 0.3 X 4 = 1.2
Here, d_s (f, S) ≤ | f |, therefore correct placement

d_s (ACGGATAG, ACGTATAGCATGATCA) = 1
ε| f | = 0.3 X 8 = 2.4
Here, d_s (f, S) ≤ | f |, therefore correct placement

d_s (GTCCA, ACGTATAGCATGATCA) = 2
ε| f | = 0.3 X 5 = 1.5
Here, d_s (f, S) > | f |, therefore incorrect placement

Practice question

What is the smallest value of ε such that the layout below is valid under the Reconstruction model?

F= (ACCGT, CGTGC, TTAC, TGCCGT)

```
--ACCGT--
----CGTGC
TTAC-----
-TGCCGT--
--------
TTACCGTGC
```

Solution:

There exists one error between the last fragment and the consensus sequence.
So, d_s(TGCCGT, TTACCGTGC) = 1
Now, we know that ds(TGCCGT, TTACCGTGC) ≤ ε |TGCCGT|
Therefore, 1 ≤ ε 6.
So, the smallest value for ε= 1/6

Limitations of RECONSRUCTION

- Doesn’t model repeats
- Doesn’t handle lack of coverage
- Only handles linkage in a very simple way
- Always produces a single contig or shortest possible string, therefore modelled target might be different in size.

MULTICONTIG

- Adds a notion of good linkage to the answer
- *Definitions*

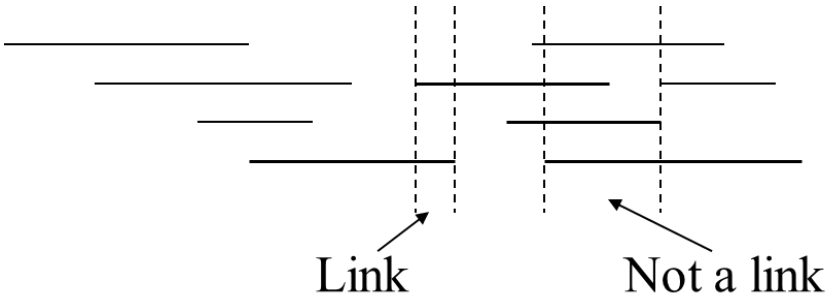
  A layout, *L*, is a multiple alignment of the fragments

  Columns numbered from 1 to |*L* |

  Endpoints of a fragment: l(f) and r(f)

  Overlap for fragment f and g will be [l(f)...r(f)]∩[l(g)...r(g)]

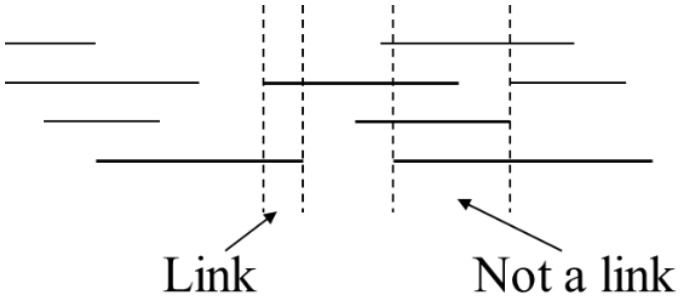
  An overlap is a link if no other fragment completely covers the overlap



MULTICONTIG

Definitions

- The size of a link is the number of overlapping positions
- The weakest link is the smallest link in the layout
- A *t*-contig has a weakest link of size *t*
- A collection, *F*, admits a *t*-contig if a *t*-contig can be constructed from the fragments in *F*




```
ACGTATAGCATGA
  GTA   CATGATCA
ACGTATAG   GATCA
```

A link of size 5

MULTICONTIG

Input: F, and t

Output: a minimum number of collections, C_i, such that every C_i admits a t-contig

Let F = {GTAC, TAATG, TGTAA}

$t = 3$

```
GTAC   --TAATG
       TGTAA--
       -------
       TGTAATG
```

$t = 1$

```
TGTAA-----
--TAATG---
------GTAC
----------
TGTAATGTAC
```

Handling errors:

- The *image* of a fragment is the portion of the consensus sequence, S, corresponding to the fragment in the layout
- S is an ε-*consensus* for a collection of fragments when the edit distance from each fragment, f, and its image is at most $\varepsilon \mid f \mid$

```
   TATAGCATCAT
CGTC    CATGATCA
ACGGATAG   GTCCA
-----------------
ACGTATAGCATGATCA
```

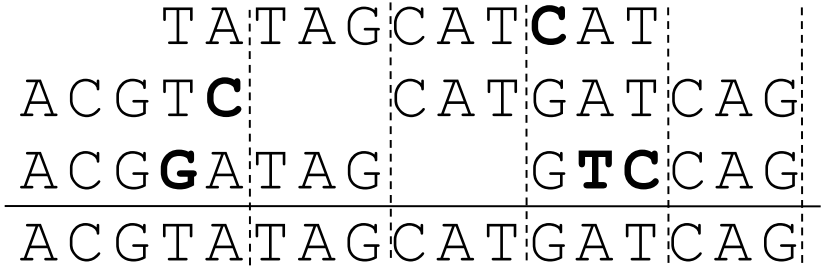
An ε-*consensus* for $\varepsilon = 0.4$

MULTICONTIG

Input: A collection, F , of strings, an integer t ≥ 0, and an error tolerance ε between 0 and 1

Output: A partition of F into the minimum number of collections C_i such that every C_i admits a *t*-contig with an ε-consensus

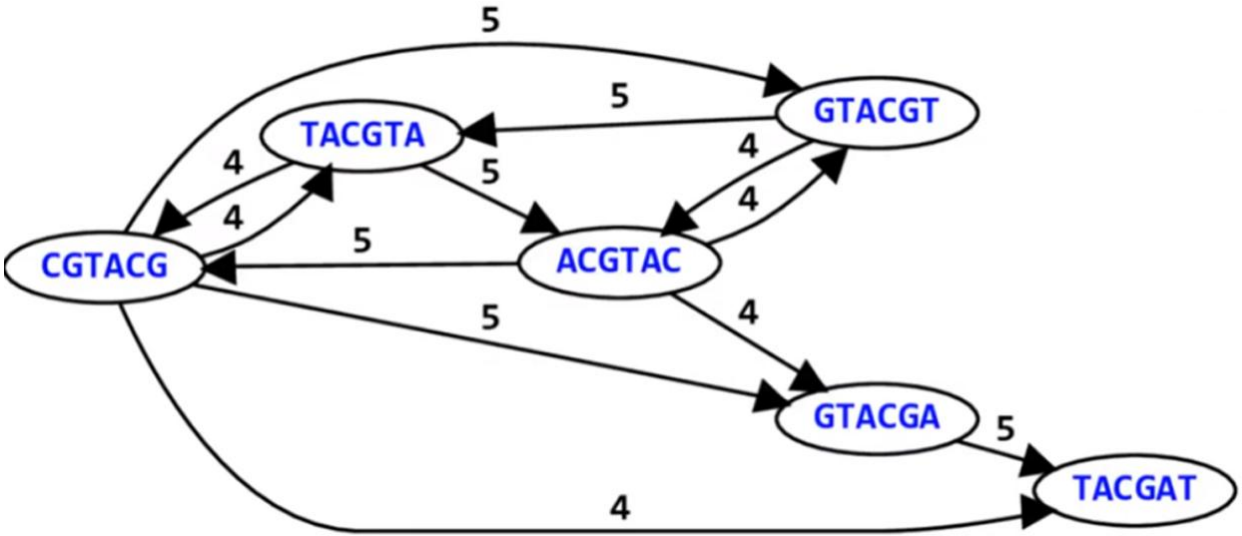
Let ε = 0.4, *t* = 3



Graph based method for fragment assembly

Overlap Graph

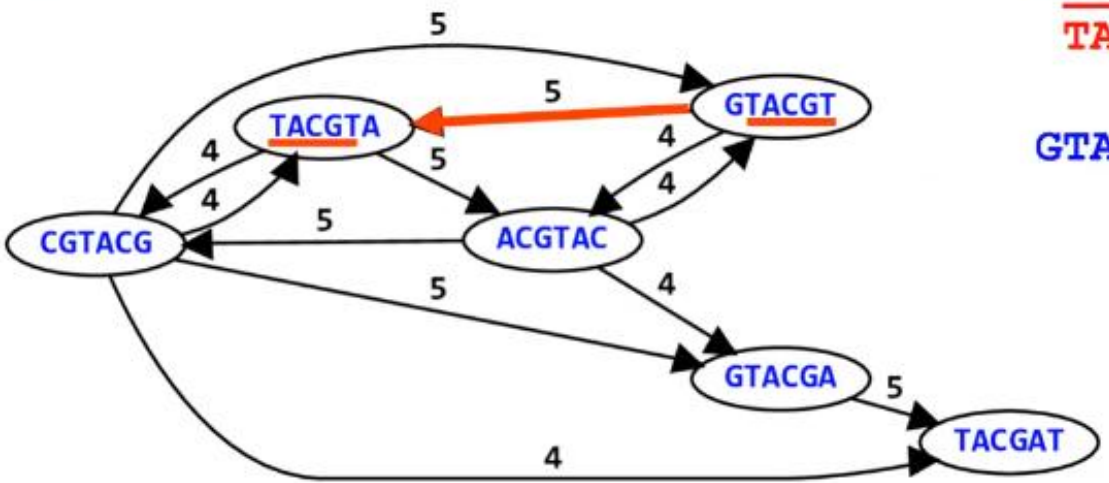
Reads: CGTACG, TACGTA, GTACGT, ACGTAC, GTACGA, TACGAT



One node for each read/fragment, an edge shows the overlap between two reads.

Reconstruction of genome from overlap graph

Reads: CGTACG, TACGTA, GTACGT, ACGTAC, GTACGA, TACGAT

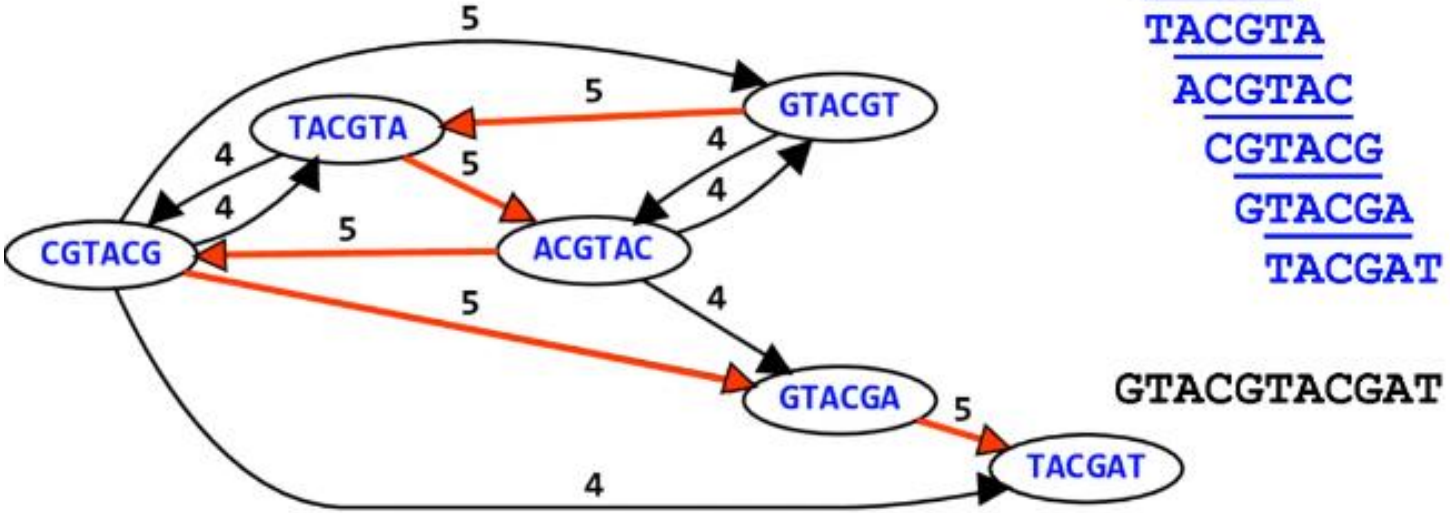


GTACGT
TACGTA

GTACGTA

Reconstruction of genome from overlap graph

Next is to find a walk that visits every node once (Hamiltonian path)

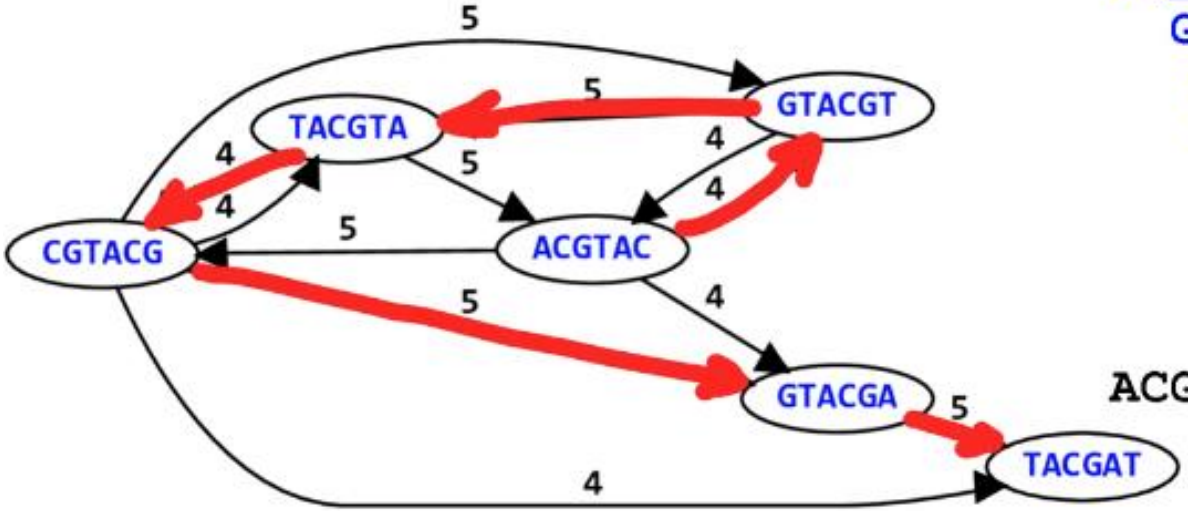


GTACGT
TACGTA
ACGTAC
CGTACG
GTACGA
TACGAT

GTACGTACGAT

Reconstruction of genome from overlap graph

Next is to find a walk that visits every node once (Hamiltonian path)

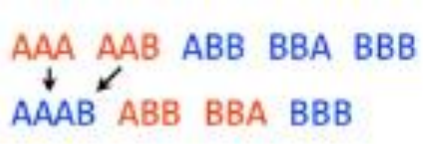


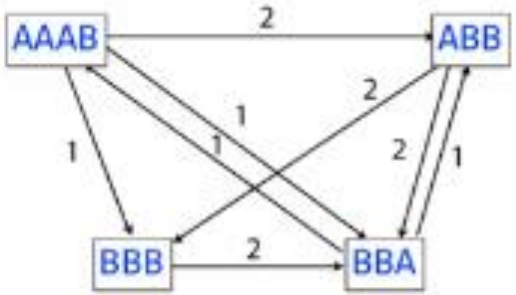
ACGTAC
GTACGT
TACGTA
CGTACG
GTACGA
TACGAT

ACGTACGTACGAT

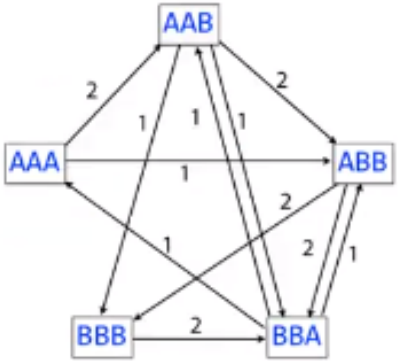
Shortest common superstring using overlap graph

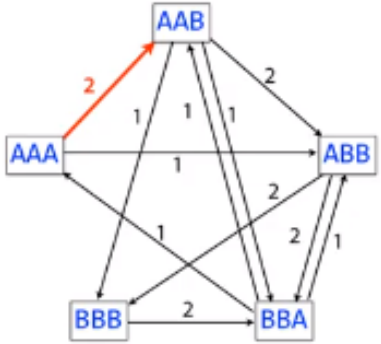
Greedy algorithm to find the shortest common string

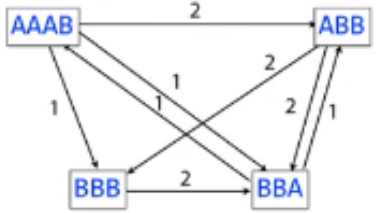


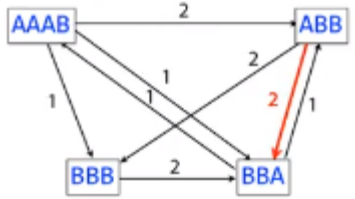


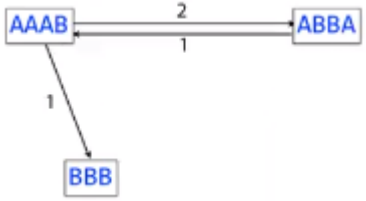
Shortest common superstring using overlap graph

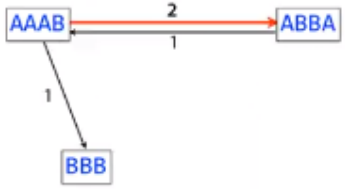












AAA AAB ABB BBA BBB

AAAB ABB BBA BBB

AAAB ABBA BBB

AAABBA BBB

AAABBABBB ← superstring, length=9

AAABBA

AAABBABBB ← superstring, length=9

BBB

Shorter Superstring: AAABBBA ← superstring, length=7

Assembly in practice

Practical implementations often divide the whole problem in three phase:

- Finding overlaps
- Building a layout
- Computing the consensus