# Sequence Assembly

## An introduction to fragment assembly

# Outline

- The two sequence assembly paradigms
- The fragment assembly paradigm
  - Problem statement
  - Computational complexity
  - Greedy algorithm

# Two sequencing paradigms

1. Fragment assembly
   - For technologies that produce "reads"
     - Sanger, Illumina, Pacific Biosciences, etc.
2. Spectral assembly
   - For technologies that produce "spectra"
     - Universal DNA arrays
   - Read data can also be "converted" to spectra

The two paradigms are actually closely related

# The fragment assembly problem

- Given: A set of reads (strings) $\{s_1, s_2, \ldots , s_n\}$
- Do: Determine a large string *s* that "best explains" the reads

- What do we mean by "*best explains*"?
- What *assumptions* might we require?

# Shortest superstring problem

- Objective: Find a string *s* such that
  - all reads $s_1$, $s_2$, … , $s_n$ are substrings of *s*
  - *s* is as short as possible
- Assumptions:
  - Reads are 100% accurate
  - “best” = “simplest”
  - Identical reads must come from the same location on the genome
  - Reads come from a single, single-stranded DNA molecule

# Shortest superstring example

- Reads:

```
{ACG, CGA, CGC, CGT, GAC, GCG, GTA, TCG}
```

# Shortest superstring example

- Reads:

  `{ACG, CGA, CGC, CGT, GAC, GCG, GTA, TCG}`

- Shortest superstring (length 10)

```
TCGACGCGTA
TCG
 CGA
  GAC
   ACG
    CGC
     GCG
      CGT
       GTA
```

# Complexity of the shortest substring problem

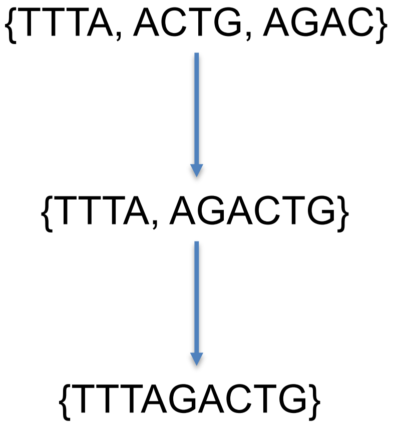
- This problem turns out to be *NP*-complete
- This means that
  - there is no known efficient (polynomial time) algorithm for solving this problem
  - it is unlikely that an efficient algorithm exists

# Algorithms for shortest substring problem

- Simple *greedy* strategy:

```
while # strings > 1 do
   merge two strings with maximum overlap
loop
```

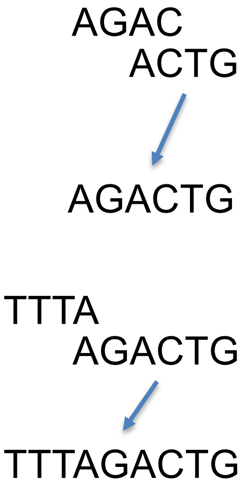
- For example:



{TTTA, ACTG, AGAC}

{TTTA, AGACTG}

{TTTAGACTG}



```
AGAC
  ACTG
AGACTG

TTTA
   AGACTG
TTTAGACTG
```

# Properties of the greedy algorithm for shortest superstring

- Conjectured to give string with length ≤ 2 × minimum length
- "2-approximation"
- Can be cast as a *graph* algorithm

# Summary

- 2 assembly paradigms: spectral and fragment
- Fragment assembly problem is often cast as the “shortest superstring problem”
- No known efficient algorithm for finding shortest superstring
- Greedy algorithm is intuitive but is not guaranteed to find the optimal solution
- The greedy algorithm and others can be described in terms of graph theory (next lecture)