Sequence Assembly

Fall 2016

BMI/CS 576

www.biostat.wisc.edu/bmi576/

Irene Ong

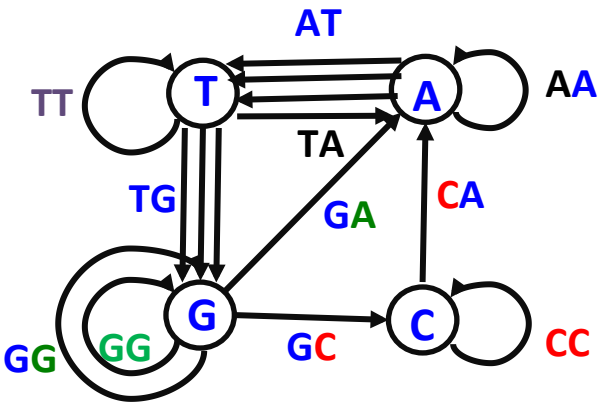
Irene.ong@wisc.edu

Directed multigraph

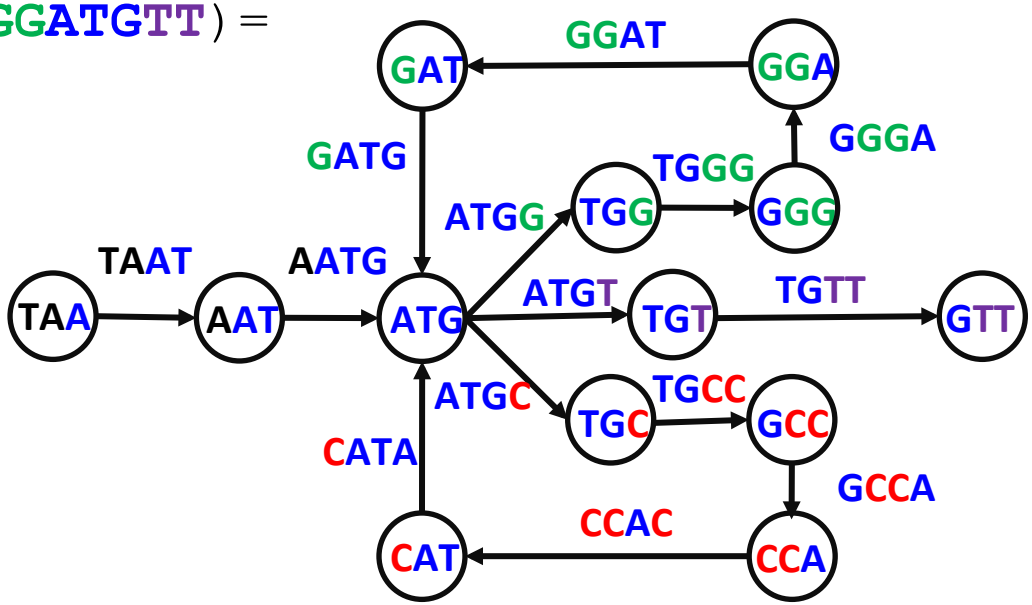
- Directed **multigraph** G(V, E) consists of
  - set of vertices, V and
  - multiset of directed edges, E
- Otherwise, same as directed graph
- Repeated edges
- De Bruijn graph is a directed multigraph

de Bruijn k-mer examples

de Bruijn$_2$(**TAATGCCATGGGATGTT**) =

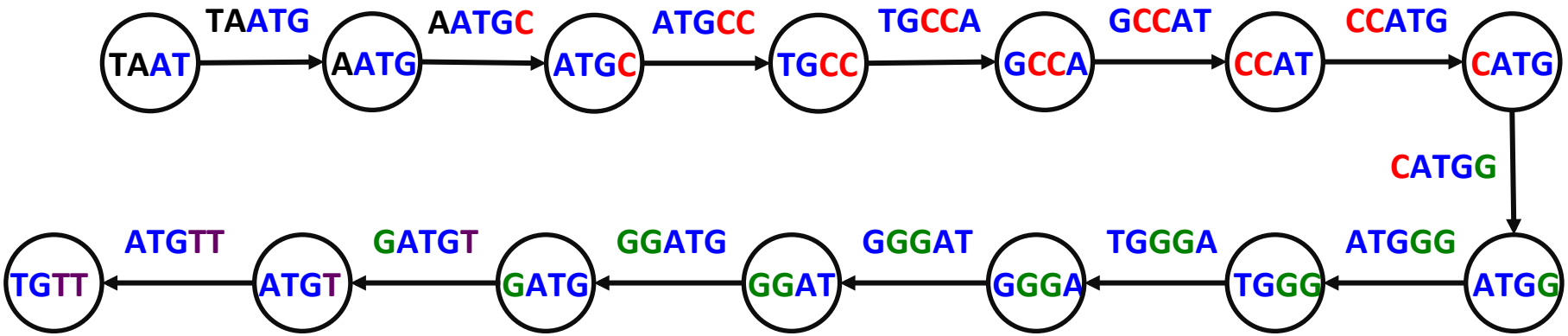


de Bruijn$_4$(**TAATGCCATGGGATGTT**) =



de Bruijn example

de Bruijn$_5$(**TAATGCCATGGGATGTT**) =



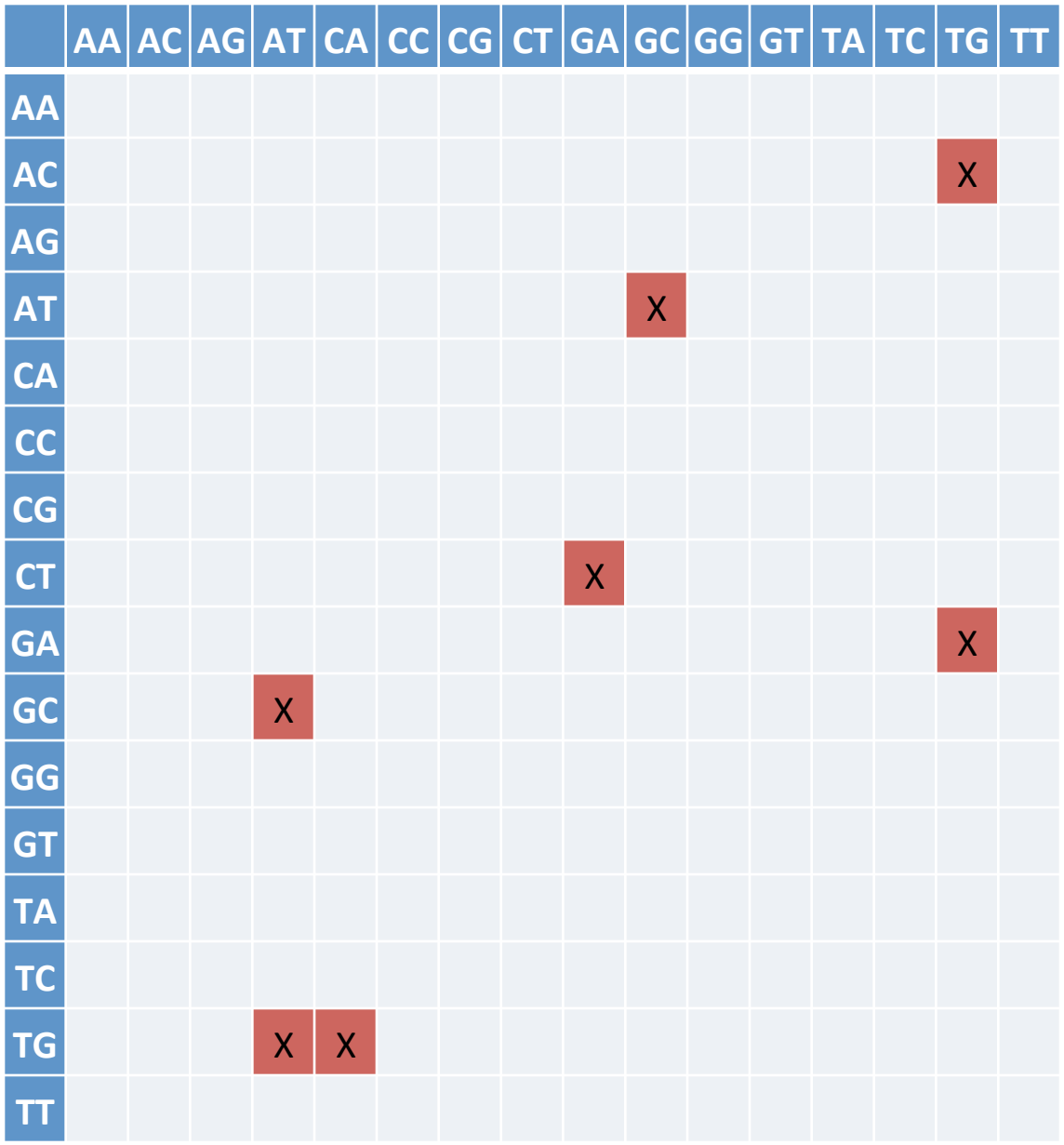
Sequencing by Hybridization (SBH)

- SBH array has probes for all possible *k*-mers
- For a given DNA sample, array tells us whether each *k*-mer is *PRESENT* or *ABSENT* in the sample
- The set of all *k*-mers present in a string *s* is called its *spectrum* (a.k.a. *composition*)
- Example:
  - s = ACTGATGCAT
  - *spectrum(s, 3)* = {ACT, ATG, CAT, CTG, GAT, GCA, TGA, TGC}

Example DNA Array

Sample:
ACTGATGCAT

Spectrum (k=4):
{ACTG, ATGC,
CTGA,GATG,
GCAT,TGAT,
TGCA}



SBH Problem

- Given: A set *S* of *k*-mers
- Do: Find a string *s*, such that *spectrum(s,k) = S*

{ACT, ATG, CAT, CTG, GAT, GCA, TGA, TGC}

?

SBH as Eulerian path

- Could use Hamiltonian path approach, but not useful due to *NP*-completeness
- Instead, use *Eulerian* path approach
- *Eulerian path*: A path through a graph that traverses every edge exactly once
- Construct graph with all *(k-1)-mers* as vertices
- For each *k*-mer in spectrum, add edge from vertex representing *first k-1* characters to vertex representing *last k-1* characters

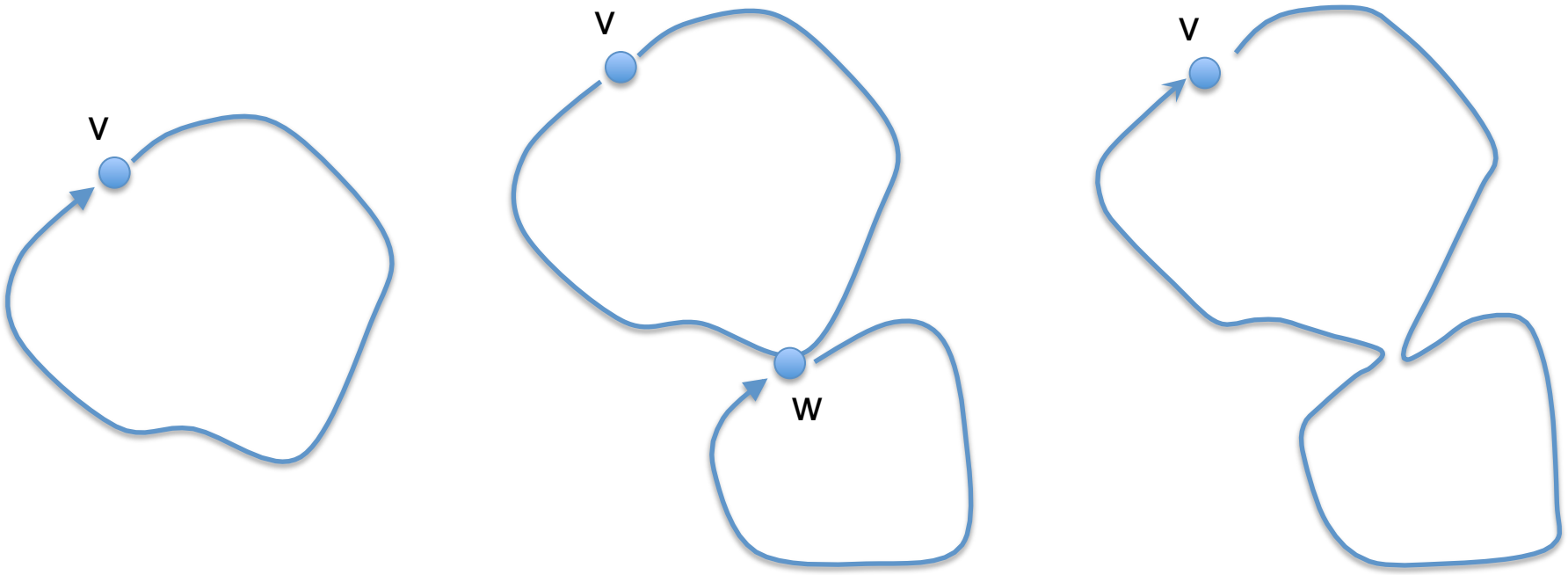
Properties of Eulerian graphs

- It will be easier to consider *Eulerian cycles*: Eulerian paths that form a cycle
- Graphs that have an *Eulerian cycle* are simply called *Eulerian*
- ***Theorem:*** A connected directed graph is *Eulerian* if and only if each of its vertices are *balanced*
- A vertex *v* is *balanced* if *indegree(v) = outdegree(v)*
- There is a polynomial-time algorithm for finding Eulerian cycles!

Eulerian cycle algorithm

- Convert graph into Eulerian (if not one) by adding an edge to make all nodes balanced, then recursively find cycles while not Eulerian
- Start at any vertex *v*, traverse unused edges until returning to *v*
- While the cycle is not Eulerian
  - Pick a vertex *w* along the cycle for which there are untraversed outgoing edges
  - Traverse unused edges until ending up back at *w*
  - Join two cycles into one cycle

Joining cycles

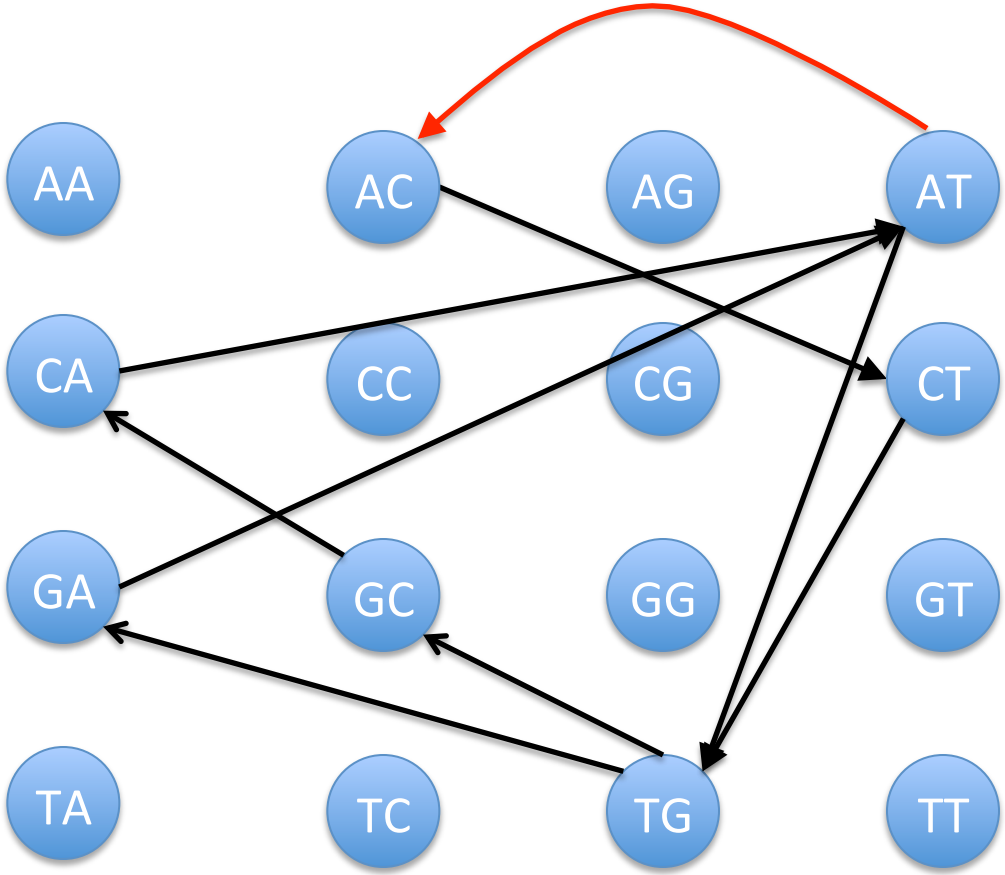


Eulerian Path -> Eulerian Cycle

- If a graph has an Eulerian Path starting at *s* and ending at *t* then
  - All vertices must be balanced, except for *s* and *t* which may have $|indegree(v) - outdegree(v)| = 1$
  - If *s* and *t* are not balanced, add an edge between them to balance
    - Graph now has an Eulerian cycle which can be converted to an Eulerian path by removal of the added edge

SBH graph example

{ACT, ATG, CAT, CTG, GAT, GCA, TGA, TGC}



de Bruijn graphs

- Assume *perfect sequencing* where each length-k substring is sequenced exactly once with no errors
- With perfect sequencing, this procedure always yields an Eulerian graph. Why?
- Node is semi-balanced if indegree differs from outdegree by 1
- Node for k-1-mer from left end is semi-balanced with one more outgoing edge than incoming *
- Node for k-1-mer at right end is semi-balanced with one more incoming than outgoing *
- Other nodes are balanced since # times k-1-mer occurs as a left k-1-mer = # times it occurs as a right k-1-mer

* Unless genome is circular

SBH difficulties

- In practice, sequencing by hybridization is hard
  - Arrays are often inaccurate -> incorrect spectra
    - False positives/negatives
  - Need long probes to deal with repetitive sequence
    - But the number of probes needed is exponential in the length of the probes!
    - There is a limit to the number of probes per array (currently between 1-10 million probes / array)

Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

Some Unrealistic Assumptions

- Perfect coverage of genome by reads (every *k*-mer from the genome is represented by a read)
- Reads are error-free.
- Multiplicities of *k*-mers are known
- Distances between reads within read-pairs are exact.

Some Unrealistic Assumptions

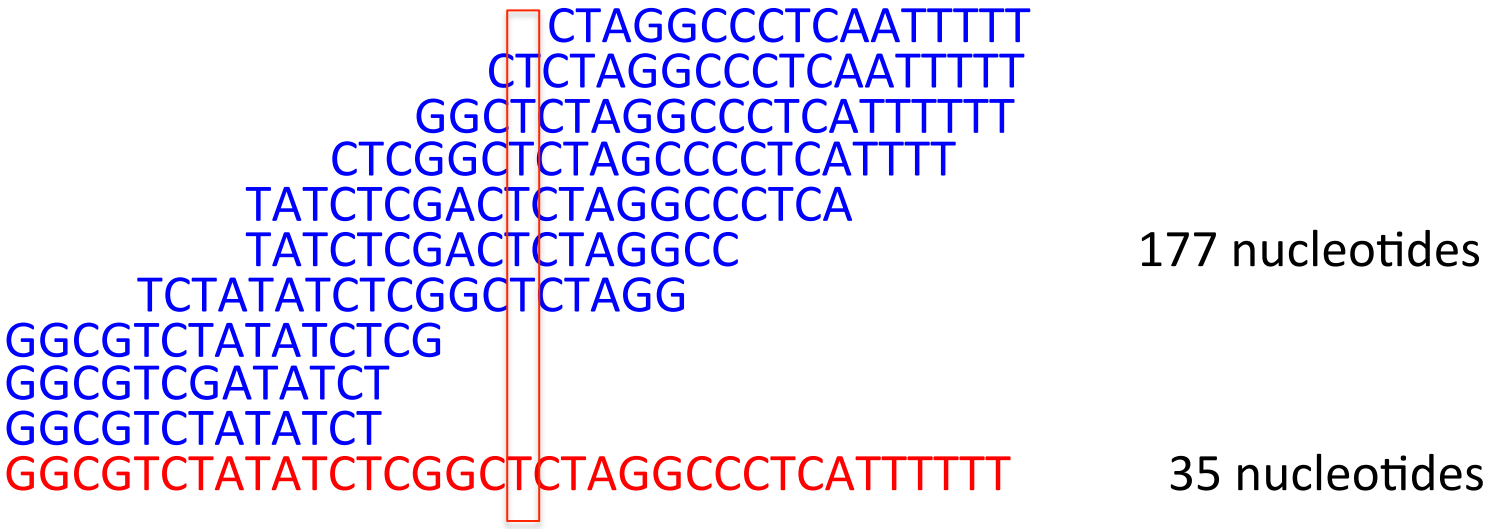
- **Imperfect** coverage of genome by reads (every *k*-mer from the genome is represented by a read)
- Reads are **error-prone**.
- Multiplicities of *k*-mers are **unknown.**
- Distances between reads within read-pairs are **inexact.**

Coverage

- Coverage is defined as the number of reads to which the k-mer belongs.
- In typical assembly projects, average coverage is ~ 30 − 50
- Same edge might appear in dozens of copies; we can use edge weights instead
- Weight = # times k-mer occurs
- Using weights, there's one weighted edge for each distinct k-mer

Average Coverage

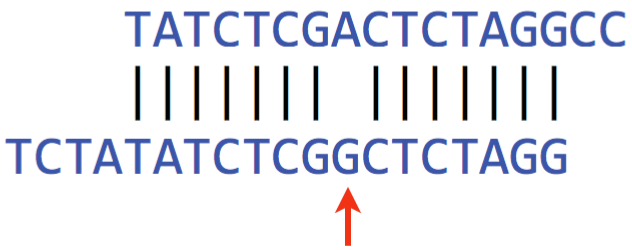
- Average # reads covering a genome position



- Average coverage = 177 / 35 ≈ 7x

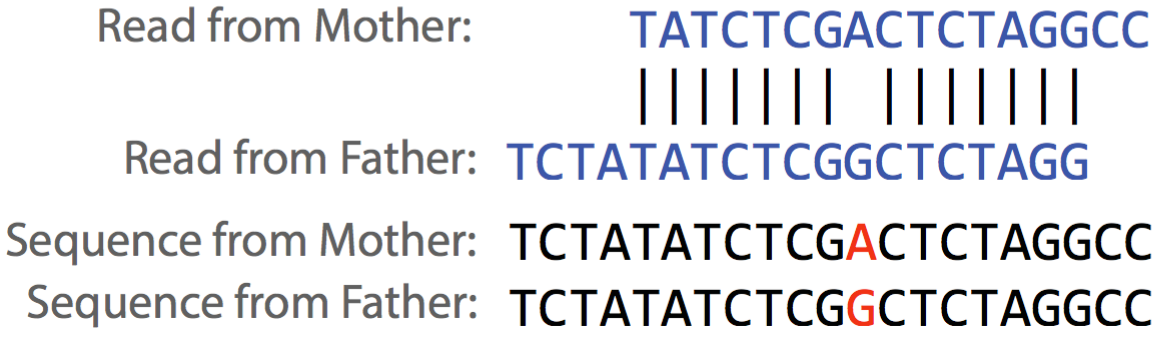
Assembly

Say two reads truly originate from overlapping stretches of the genome. Why might there be differences?



1. Sequencing error
2. Difference between inhereted *copies* of a chromosome

   E.g. humans are diploid; we have two copies of each chromosome, one from mother, one from father. The copies can differ:



We'll mostly ignore ploidy, but real tools must consider it

1st Unrealistic Assumption: Perfect Coverage

```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
     gtatggacaa
          gacaacgact
```

250-nucleotide reads generated by Illumina technology capture only a small fraction of 250-mers from the genome, thus violating the key assumption of the de Bruijn graphs.

Breaking Reads into Shorter *k*-mers

```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
     gtatggacaa
          gacaacgact
```

```
atgccgtatggacaacgact
atgcc
 tgccg
  gccgt
   ccgta
    cgtat
     gtatg
      tatgg
       atgga
        tggac
         ggaca
          gacaa
           acaac
            caacg
             aacga
              acgac
               cgact
```

2nd Unrealistic Assumption: Error-free Reads

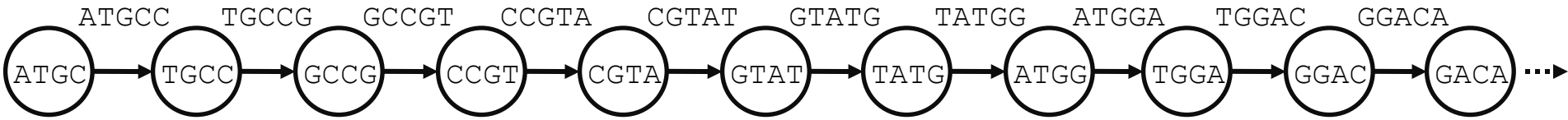
```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
     gtatggacaa
          gacaacgact
    cgtaCggaca
```

Erroneous read
(change of t to C)

```
atgccgtatggacaacgact
atgcc
 tgccg
  gccgt
   ccgta
    cgtat
     gtatg
      tatgg
       atgga
        tggac
         ggaca
          gacaa
           acaac
            caacg
             aacga
              acgac
               cgact
  cgtaC
   gtaCg
    taCgg
     aCgga
      Cggac
```

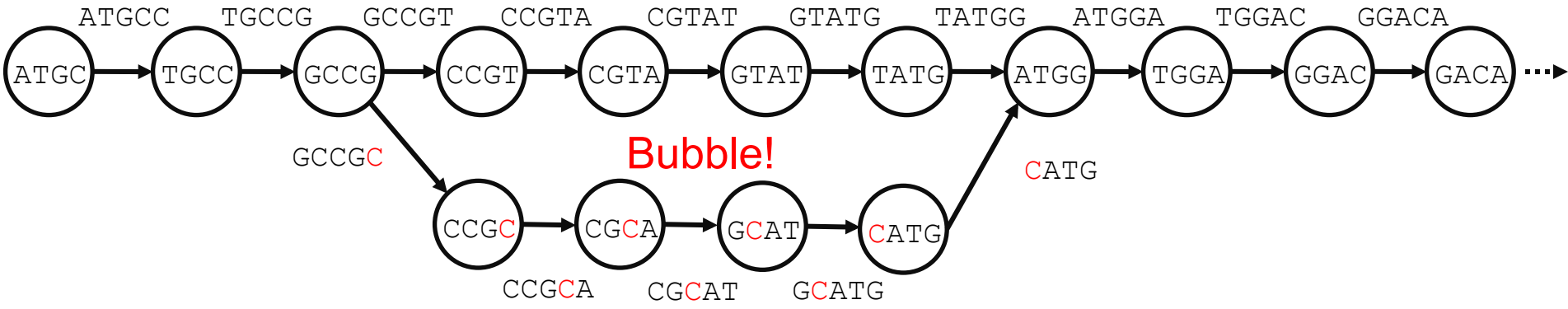
De Bruijn Graph of ATGGCGTGCAATG… Constructed from Error-Free Reads

.

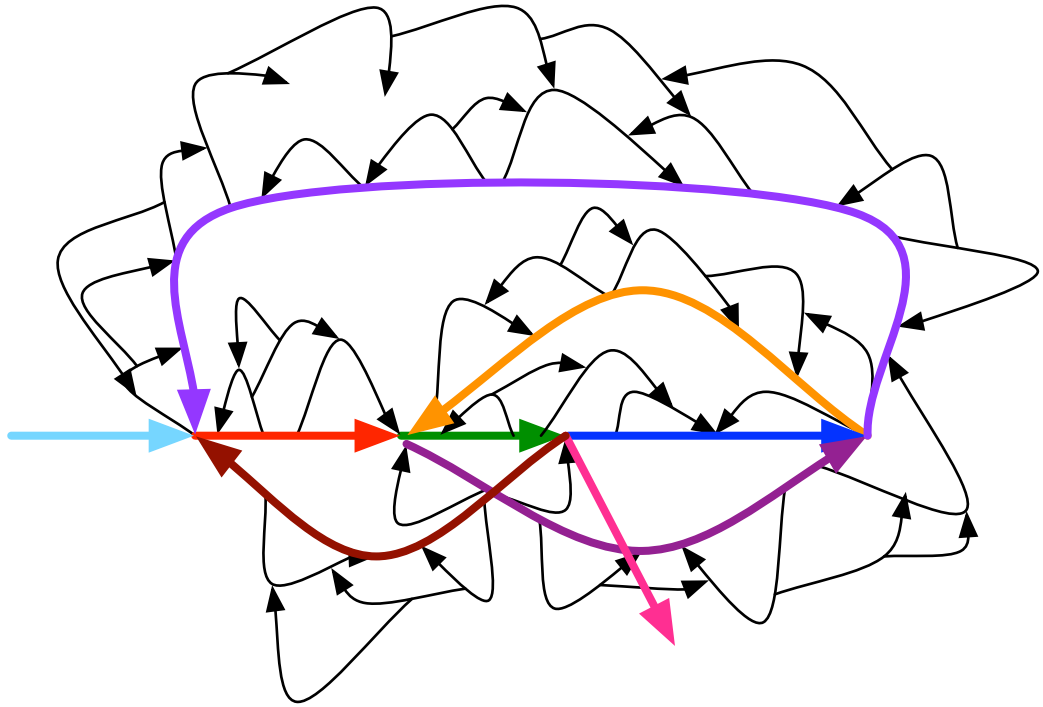


Errors in Reads Lead to **Bubbles** in the De Bruijn Graph

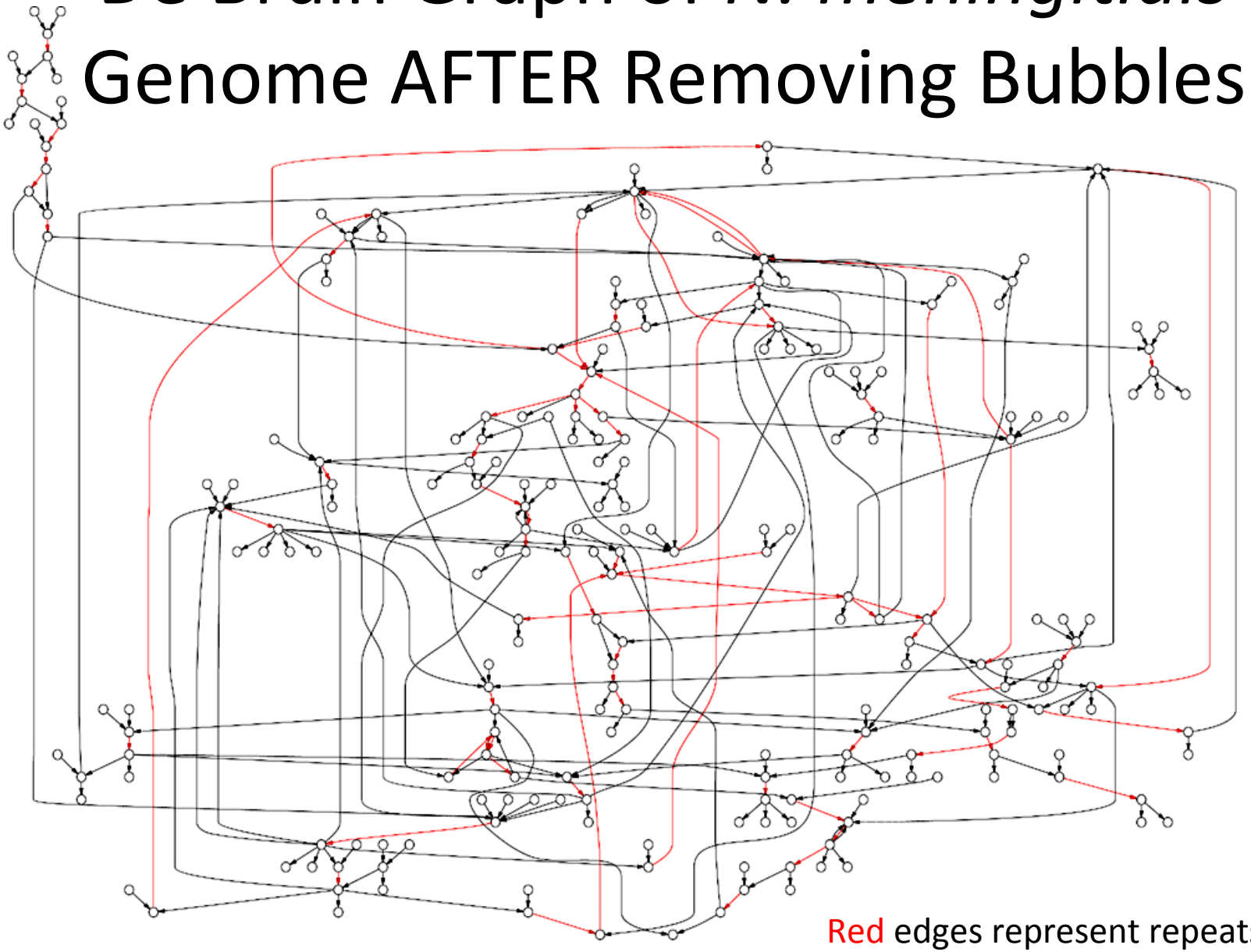
.



Bubble Explosion…Where Are the Correct Edges of the de Bruijn Graph?



De Bruin Graph of *N. meningitidis* Genome AFTER Removing Bubbles



Red edges represent repeats

K-mer spectrum approach with read data (de Bruijn approach)

- Generate spectrum from set of all *k*-mers contained within reads
- Choose *k* to be small enough such that the majority of the genome's *k*-mers will be found within the reads
- Particularly useful for short-read data, such as that produced by Illumina
- Made popular by methods such as Euler and Velvet

Difficulties with de Bruijn approach

- Not all *k*-mers may be contained within the reads even if reads completely cover the genome
- DNA repeats result in *k*-mers that are present in multiple copies across the genome
- Reads often have sequencing errors!

Fragment assembly challenges

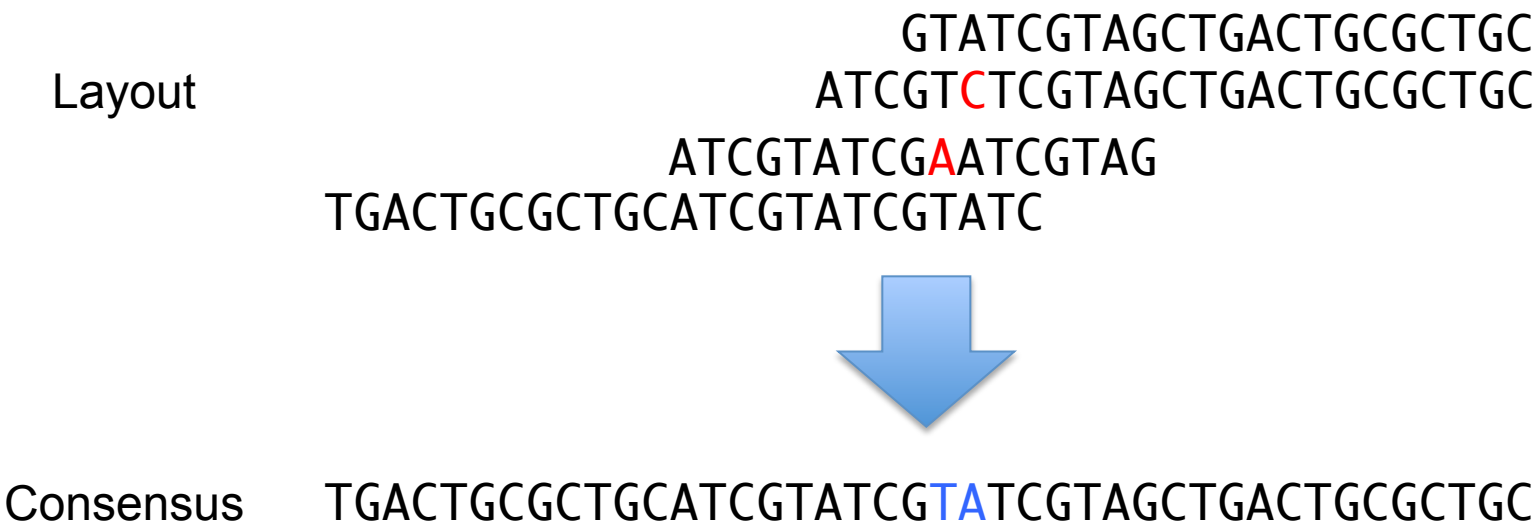
- Read errors
  - Complicates computing read overlaps
- Repeats
  - Roughly half of the human genome is composed of repetitive elements
  - Repetitive elements can be long (1000s of bp)
  - Human genome
    - 1 million Alu repeats (~300 bp)
    - 200,000 *LINE* repeats (~1000 bp)

Overlap-Layout-Consensus

- Most common assembler strategy for long reads

1. *Overlap*: Find all significant overlaps between reads, allowing for errors
2. *Layout*: Determine path through overlapping reads representing assembled sequence
3. *Consensus*: Correct for errors in reads using layout

Consensus


GTATCGTAGCTGACTGCGCTGC
Layout
ATCGTCTCGTAGCTGACTGCGCTGC
ATCGTATCGAATCGTAG
TGACTGCGCTGCATCGTATCGTATC
Consensus
TGACTGCGCTGCATCGTATCGTATCGTAGCTGACTGCGCTGC

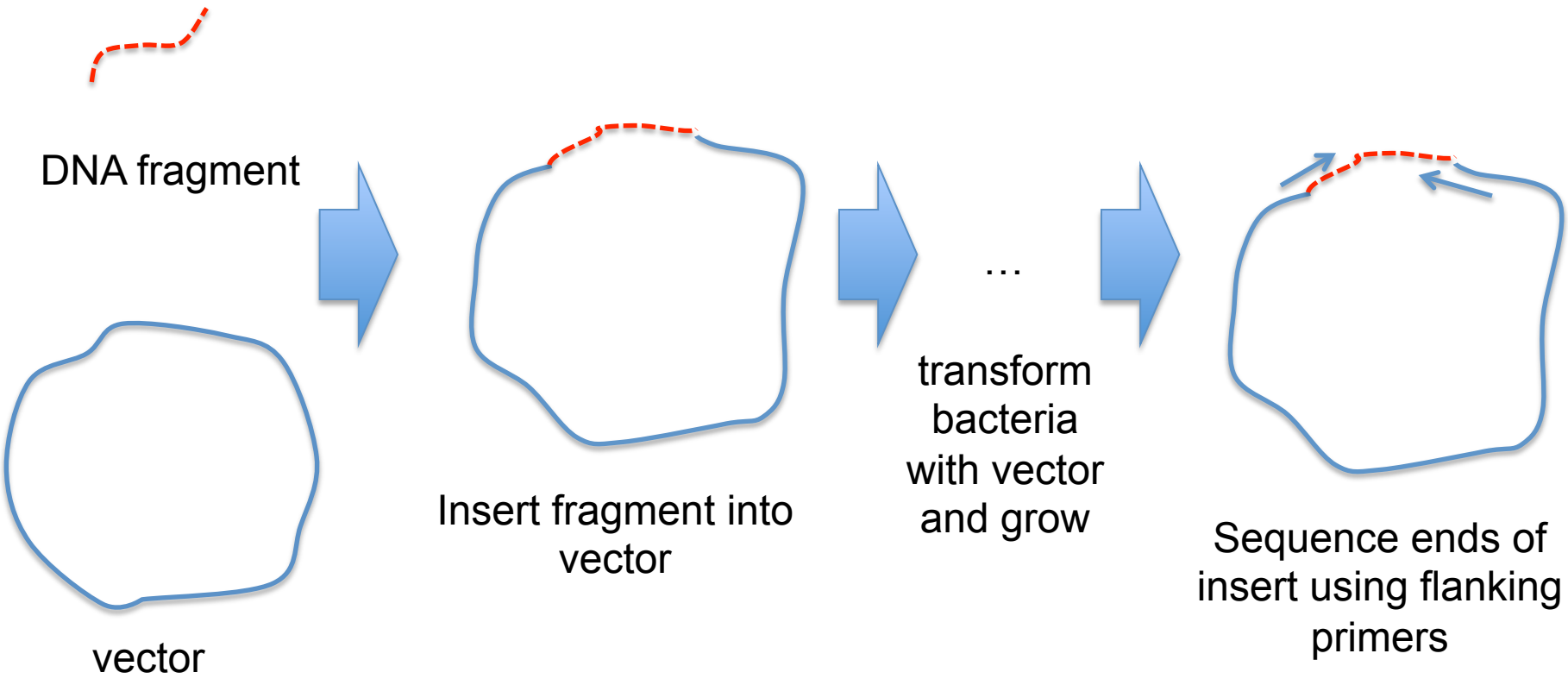
Whole Genome Sequencing

- Two main strategies:
  1. Clone-by-clone mapping
     - Fragment genome into large pieces, insert into BACs (Bacterial Artificial Chromosomes)
     - Choose *tiling set* of BACs: overlapping set that covers entire genome
     - Shotgun sequence the BACs
  2. Whole-genome shotgun
     - Shotgun sequence the entire genome at once

Assembly in practice

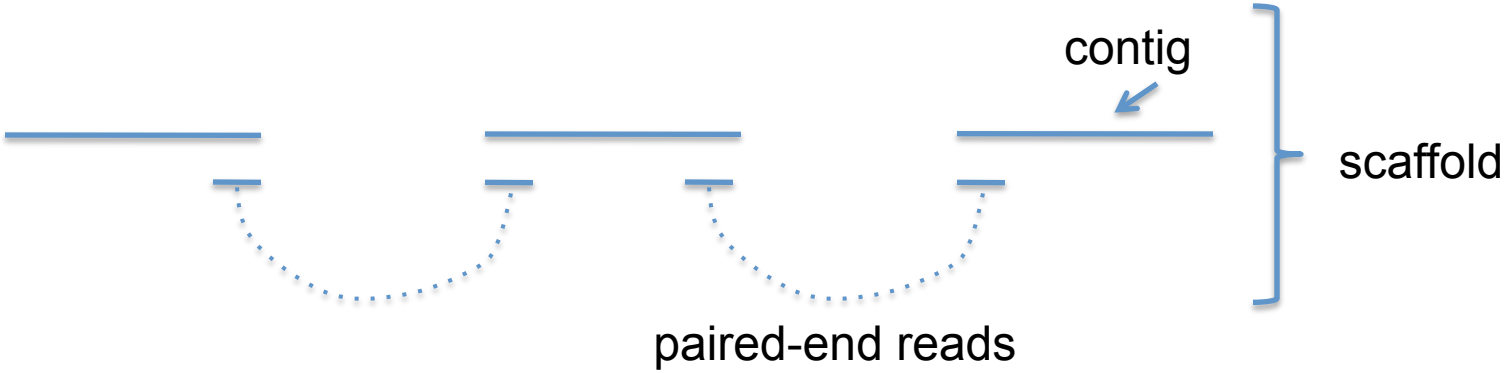
- Assembly methods used in practice are complex
  - But generally follow one of the two approaches
    - Reads as *vertices*
    - Reads as *edges (*or *paths* of edges)
- Assemblies do not typically give whole chromosomes
  - Instead gives a set of "contigs"
  - *contig*: contiguous piece of sequence from overlapping reads
  - contigs can be ordered into *scaffolds* with extra information (e.g., paired end reads)

Cloning and Paired-end reads

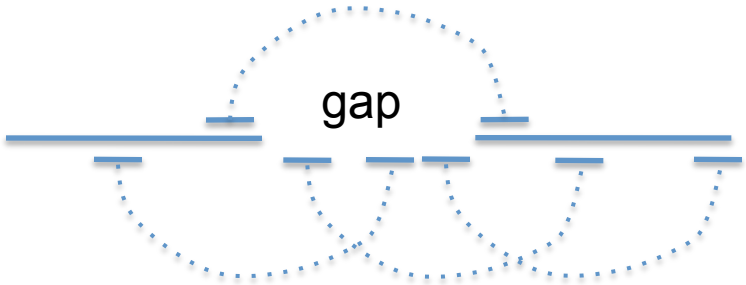


Paired-end read advantages

- *Scaffolding*: layout of adjacent, but not overlapping, *contigs*



- *Gap filling*:



Sequence assembly summary

- Two general algorithmic strategies
  - Overlap graph hamiltonian paths
  - Eulerian paths in k-mer graphs
- Biggest challenge
  - Repeats!
    - Large genomes have a lot of repetitive sequence
- Sequencing strategies
  - Clone-by-clone: break the problem into smaller pieces which have fewer repeats
  - Whole-genome shotgun: use paired-end reads to assemble around and inside repeats