Genome Assembly

19 Oct 15

Announcements

Recap

- Form Hypothesis — other class
- Collect Seq Data — What type, how much
- QC — FastQC - SolexaQA
- Error Correct — yes
- Trimming — not too harsh
- Normalize — its cool!
- QC — Once again to eval changes
- Assemble — Starting today
    - Genome v. transcriptome
- QC
- mapping
- Post-assembly
    - BLAST/HMM
- Biology — other class

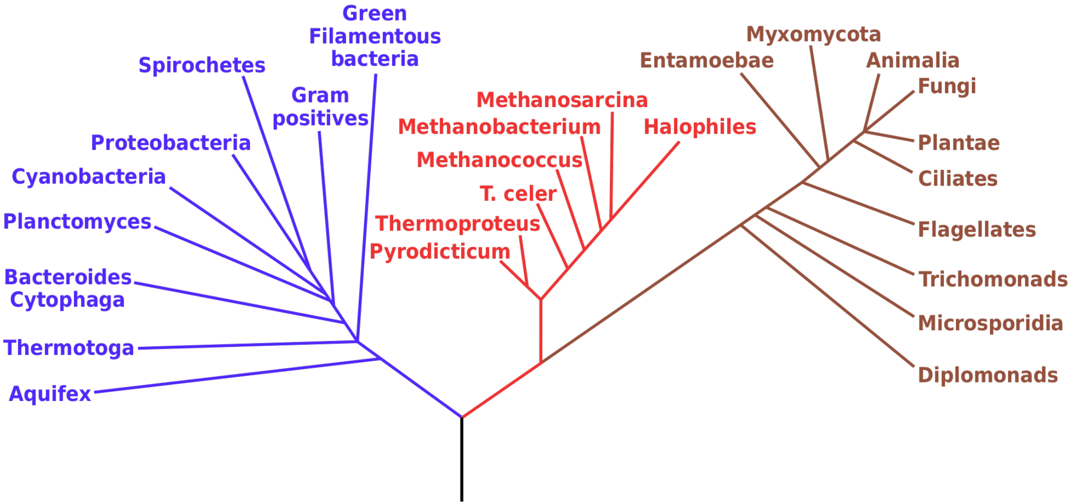
Why do you want to Assemble a genome?

What do you need to Assemble a genome?

Assemble a genome? General Strategies

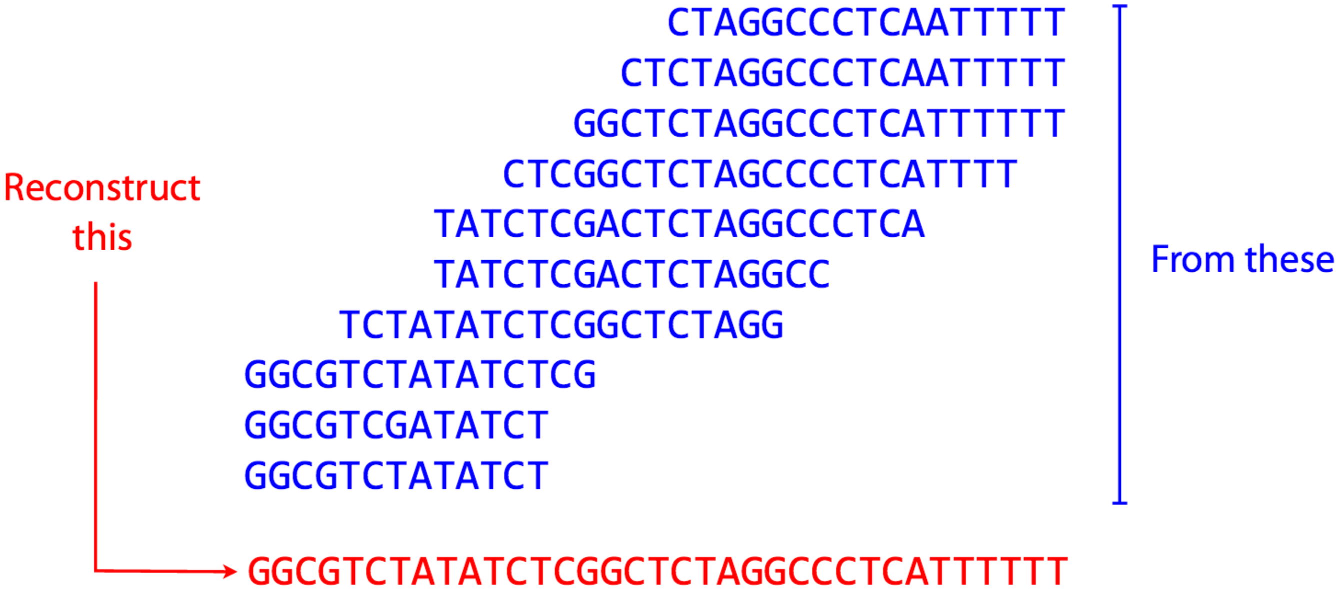
Genome size	Unlimited $$	Typical
>10Mb		
10Mb - 100Mb		
> 100 Mb		

Genome Sizes



Assembly

Assume sequencing produces such a large # fragments that almost all genome positions are *covered* by many fragments...



```
                  CTAGGCCCTCAATTTTT
                CTCTAGGCCCTCAATTTTT
              GGCTCTAGGCCCTCATTTTTT
           CTCGGCTCTAGCCCTCATTTT
        TATCTCGACTCTAGGCCCTCA
        TATCTCGACTCTAGGCC
    TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
```

From these

Reconstruct this

```
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
```

Assembly

...but we don’t know what came from where

CTAGGCCCTCAATTTTT
GGCGTCTATATCT
CTCTAGGCCCTCAATTTTT
TCTATATCTCGGCTCTAGG
GGCTCTAGGCCCTCATTTTTT
CTCGGCTCTAGCCCCTCATTTT
TATCTCGACTCTAGGCCCTCA
GGCGTCGATATCT
TATCTCGACTCTAGGCC
GGCGTCTATATCTCG

From these

Reconstruct this

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

Assembly

Key term: *coverage*. Usually it's short for *average coverage*: the average number of reads covering a position in the genome.

```
                  CTAGGCCCTCAATTTTT
                CTCTAGGCCCTCAATTTTT
              GGCTCTAGGCCCTCATTTTTT
           CTCGGCTCTAGCCCCTCATTTT
        TATCTCGACTCTAGGCCCTCA                  177 nucleotides
        TATCTCGACTCTAGGCC
    TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT           35 nucleotides
```

Average coverage = 177 / 35 ≈ 7x

Other Assembly Terms

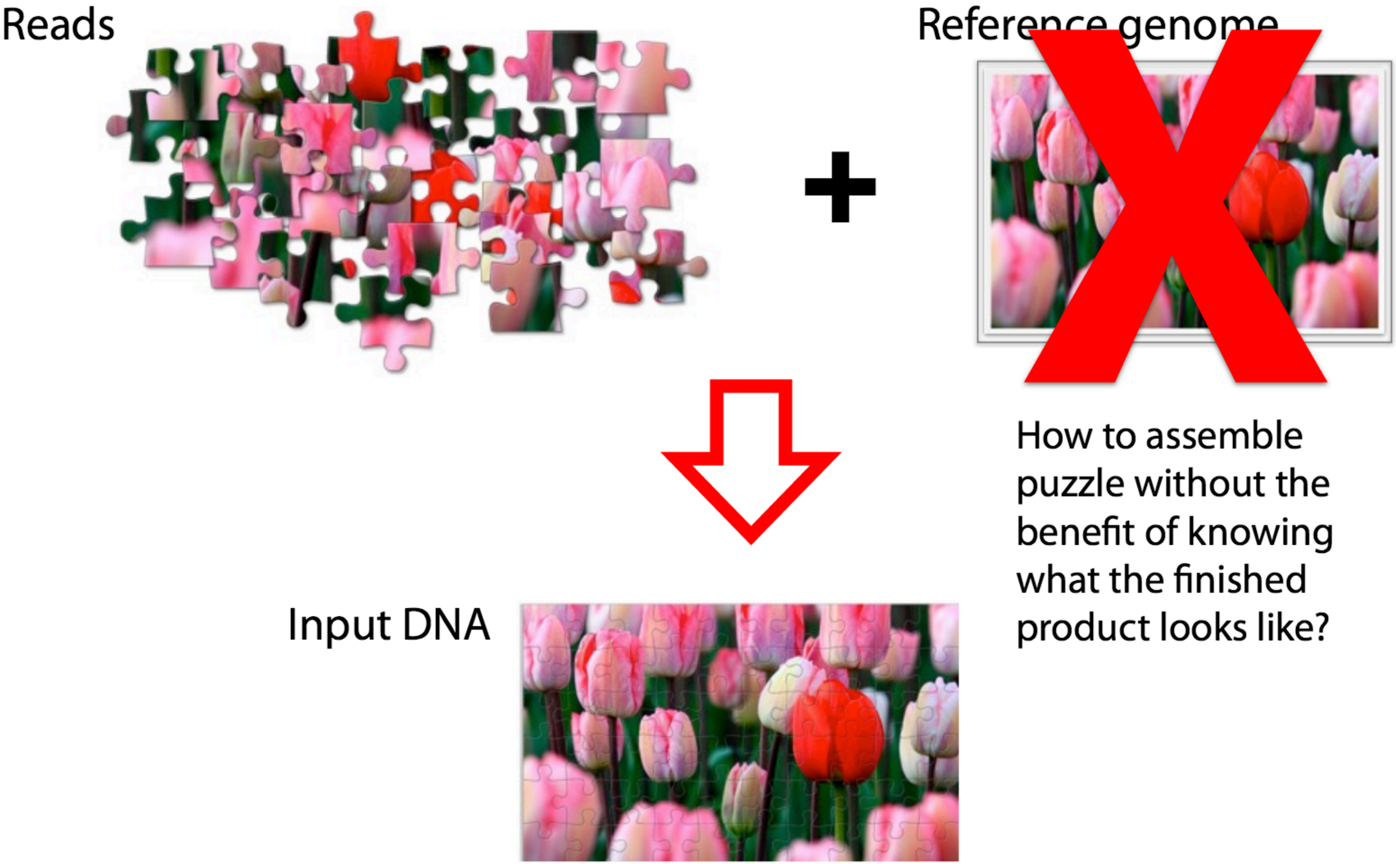
Unitig

Contig

scaffold

Assembly

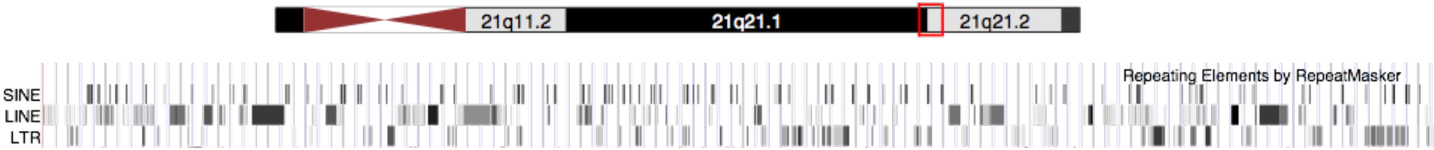
- Complicated by:



How to assemble puzzle without the benefit of knowing what the finished product looks like?

Assembly

- Complicated by:



ASSEMBLY

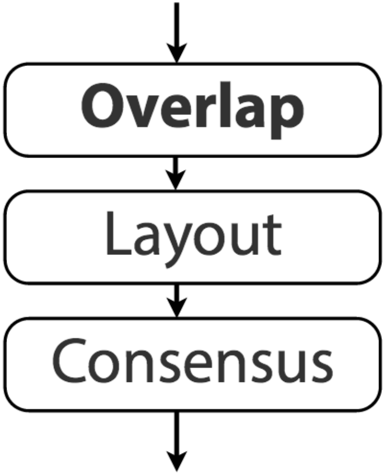
- Workflow:

Assembly

- 3 assembly strategies:

ASSEMBLY

- OLC Assembly



Build overlap graph

Bundle stretches of the overlap graph into *contigs*

Pick most likely nucleotide sequence for each contig

ASSEMBLY

- OLC Assembly: Characteristics

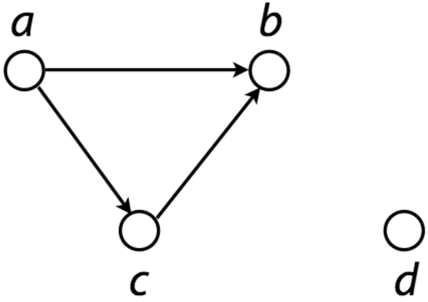
Assembly

Directed graph *G*(*V*, *E*) consists of set of *vertices, V* and set of *directed edges, E*

Directed edge is an *ordered pair* of vertices. First is the *source*, second is the *sink*.

Vertex is drawn as a circle

Edge is drawn as a line with an arrow connecting two circles



$V = \{ a, b, c, d \}$

$E = \{ (a, b), (a, c), (c, b) \}$

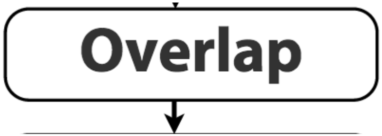
Source Sink

Vertex also called *node* or *point*

Edge also called *arc* or *line*

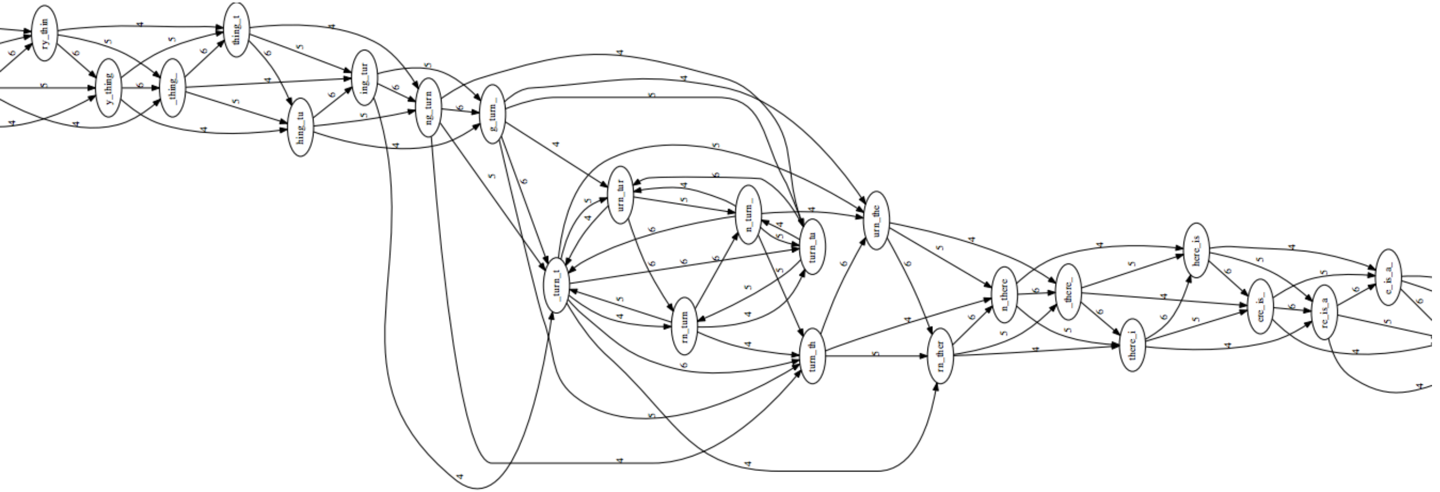
Directed graph also called *digraph*

ASSEMBLY

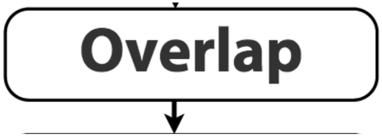


Build overlap graph

to_every_thing_turn_turn_turn_there_is_a_season
L=4, k=7



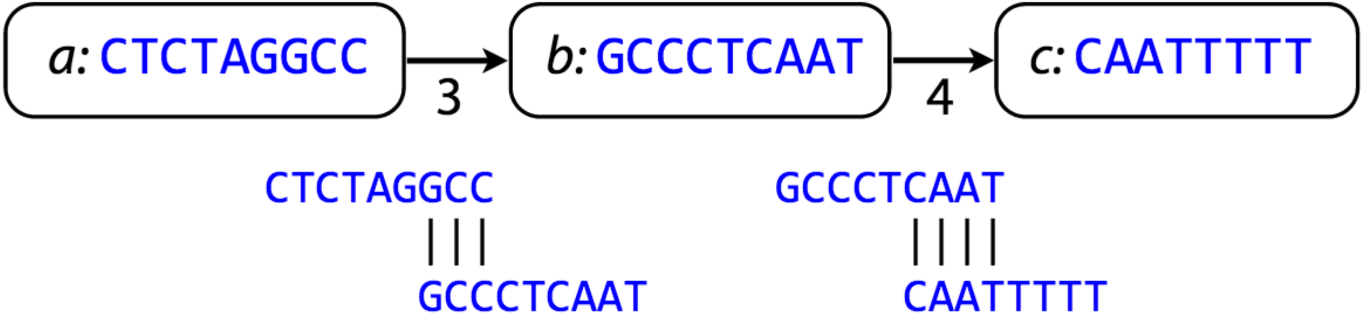
Assembly



Build overlap graph

Vertices (reads): { *a*: CTCTAGGCC, *b*: GCCCTCAAT, *c*: CAATTTTT }

Edges (overlaps): { (*a*, *b*), (*b*, *c*) }


a: CTCTAGGCC → 3 → *b*: GCCCTCAAT → 4 → *c*: CAATTTTT

```
CTCTAGGCC
      |||
      GCCCTCAAT
```

```
GCCCTCAAT
     ||||
     CAATTTTT
```