Digital Normalization

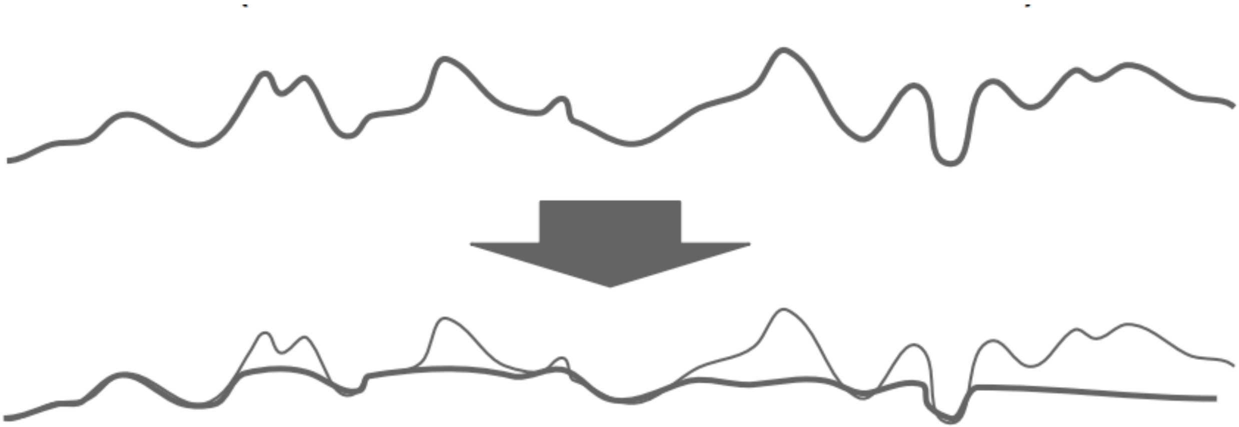
Lecture 16
Oct 10, 2016

Announcements

Perfect Storm of data analysis

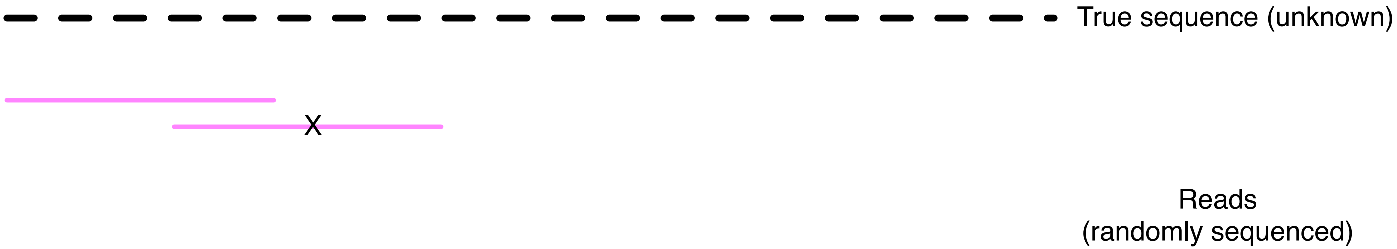
Digital Normalization

Perfect Storm of data analysis – What to do???

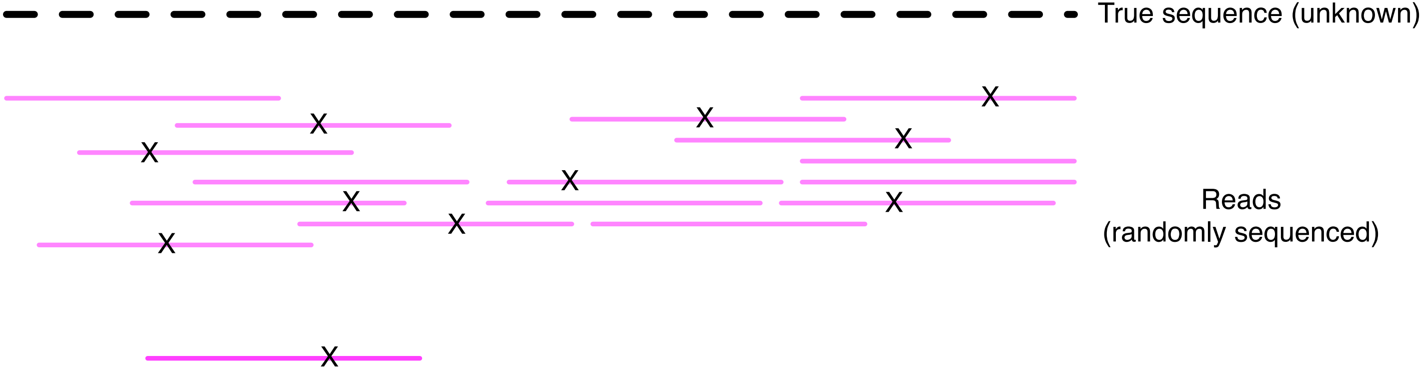


Brown 2012 arXiv:1203.4802v2

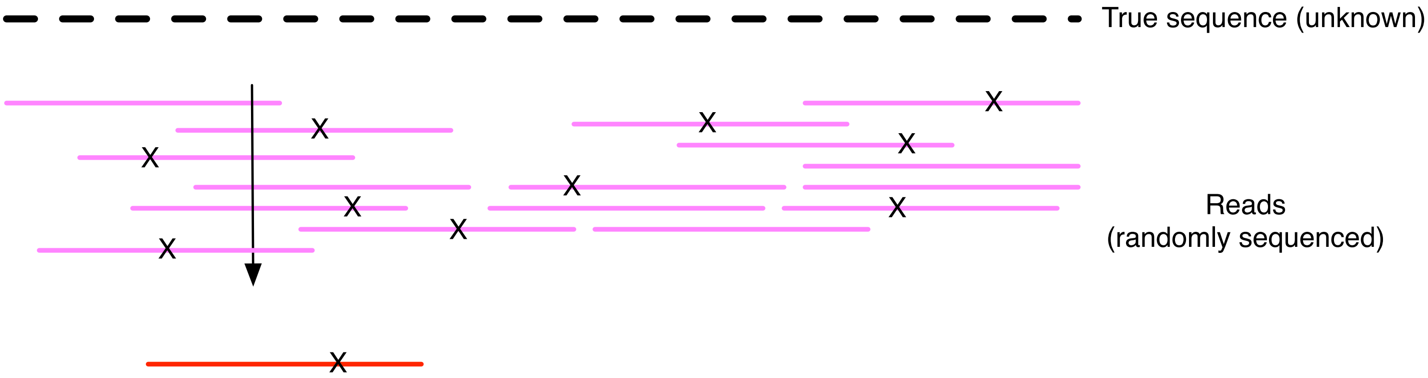
Digital Normalization



Digital Normalization



Digital Normalization

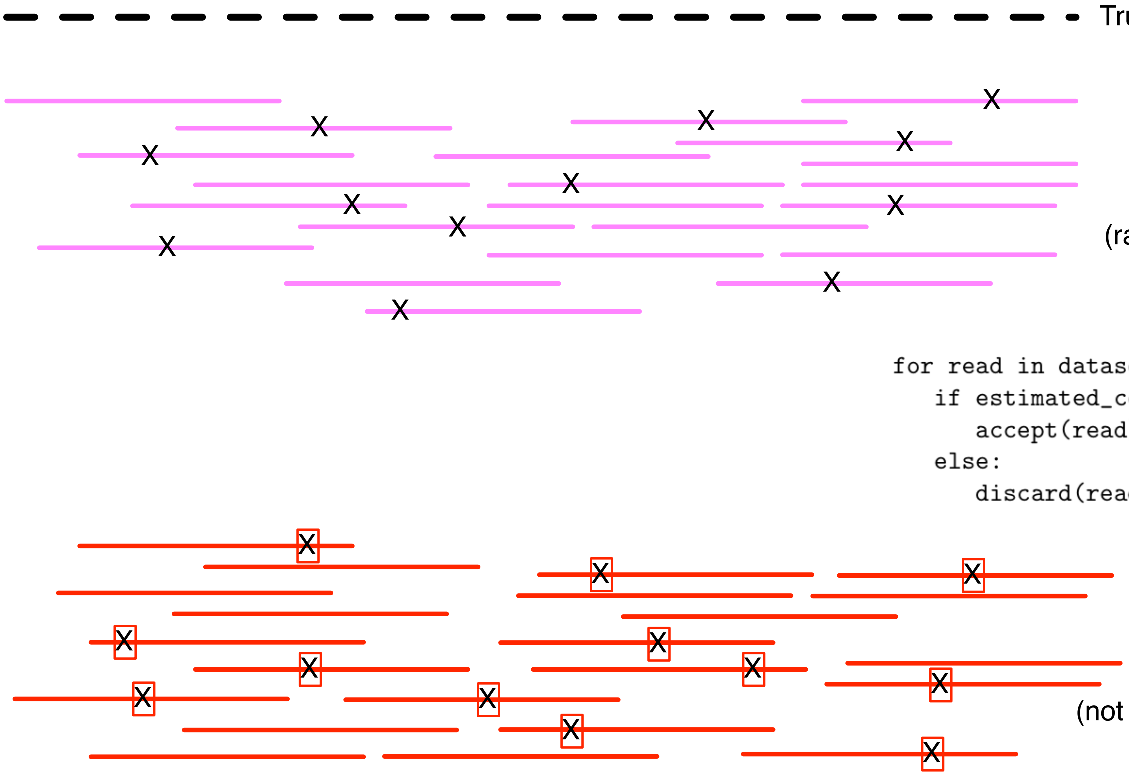


True sequence (unknown)

Reads
(randomly sequenced)

```
for read in dataset:
    if estimated_coverage(read) < C:
        accept(read)
    else:
        discard(read)
```

Digital Normalization



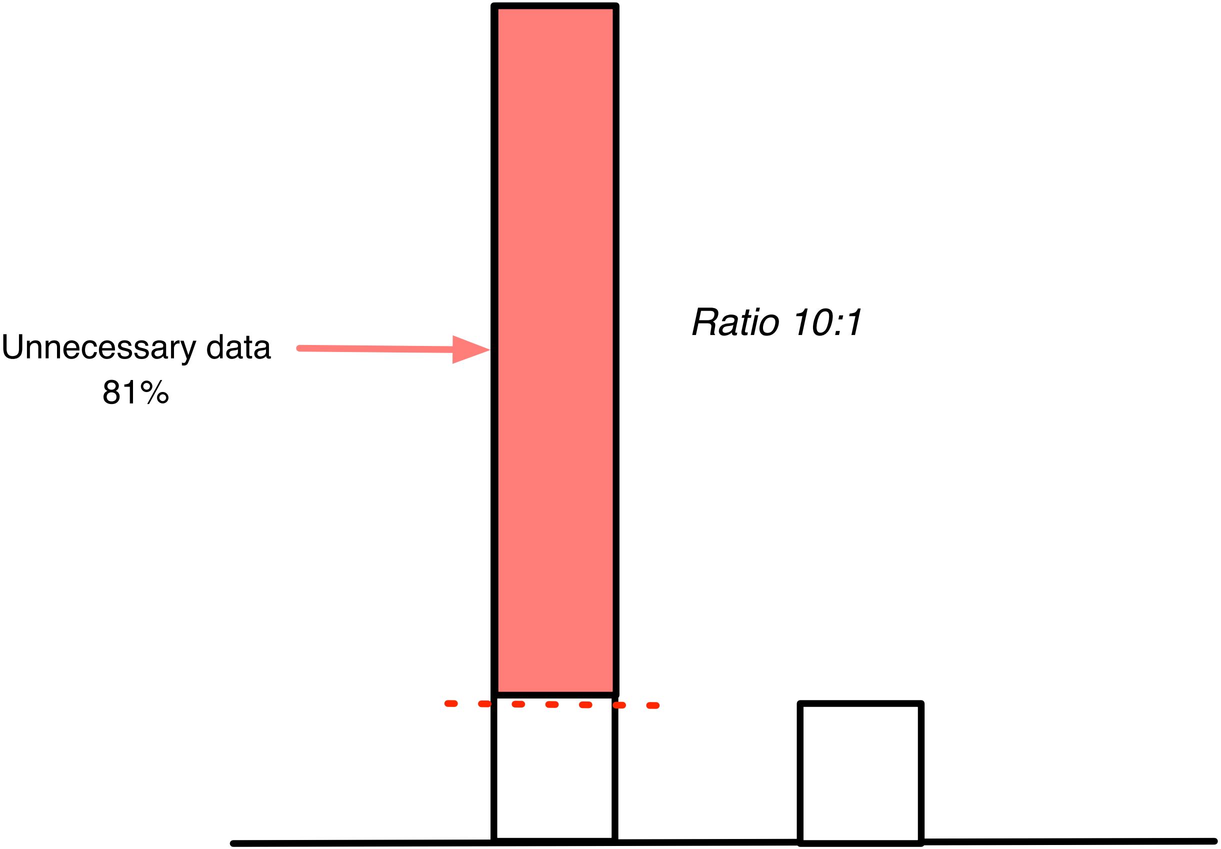
True sequence (unknown)

Reads
(randomly sequenced)

```
for read in dataset:
    if estimated_coverage(read) < C:
        accept(read)
    else:
        discard(read)
```

Redundant reads
(not needed for assembly)

Digital Normalization

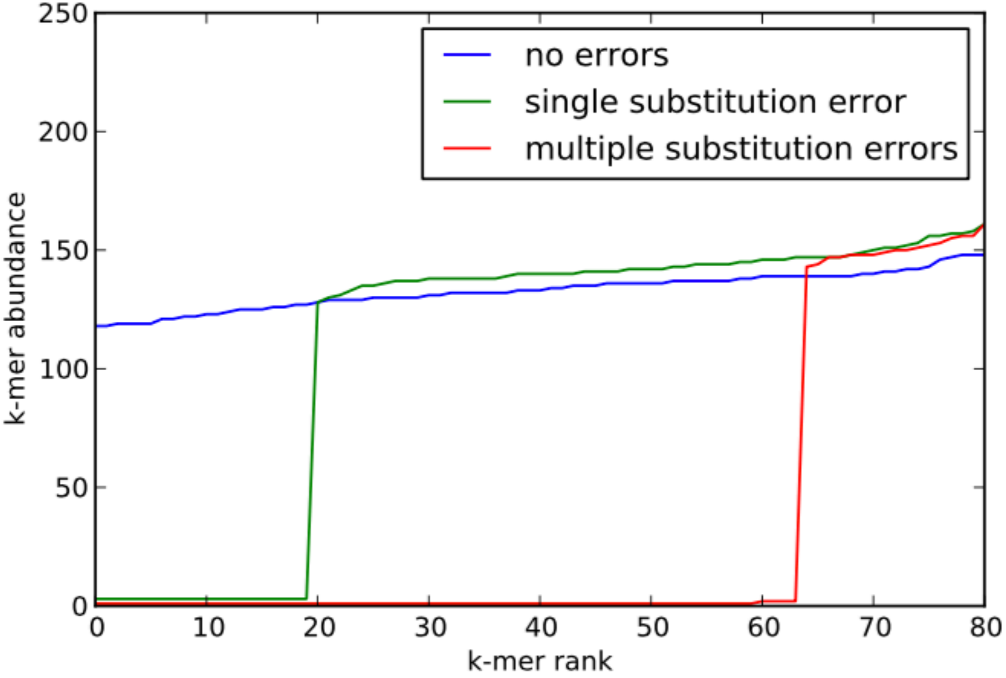


Unnecessary data
81%

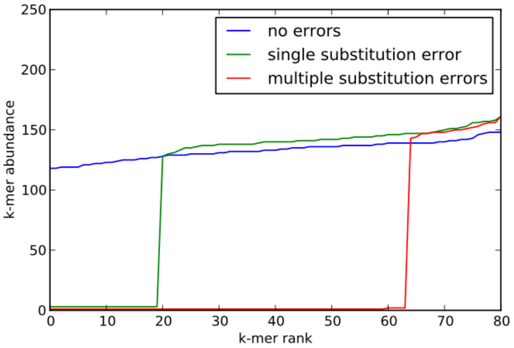
Ratio 10:1

Digital Normalization

```
for read in dataset:
   if estimated_coverage(read) < C:
      accept(read)
   else:
      discard(read)
```



DIGITAL NORMALIZATION



No error

3mer freq.

CAT=32

ATG=34

TGC=36

GCA=35

CAT=33

ATT=34

TTG=40

CATGCATTG

CAT

ATG

TGC

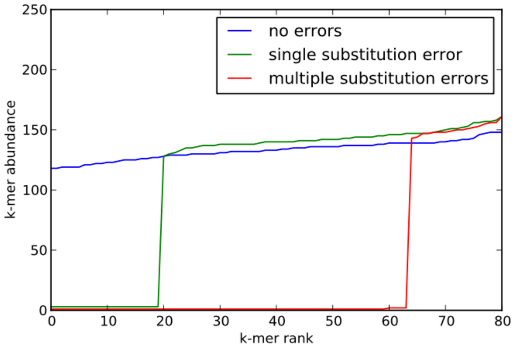
GCA

CAT

ATT

TTG

Digital Normalization



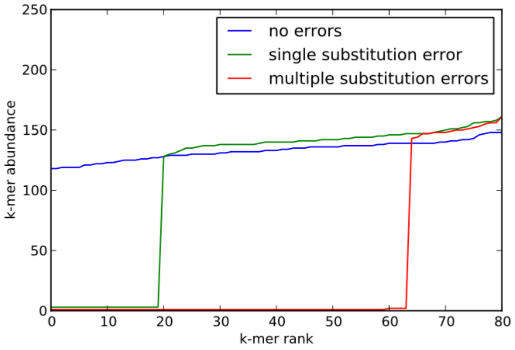
1error

3mer freq.

CAT=32
ATG=34
TG**A**=1
G**A**A=1
AAT=1
ATT=34
TTG=40

CATG**A**ATTG
CAT
ATG
TG**A**
G**A**A
AAT
ATT
TTG

DIGITAL NORMALIZATION



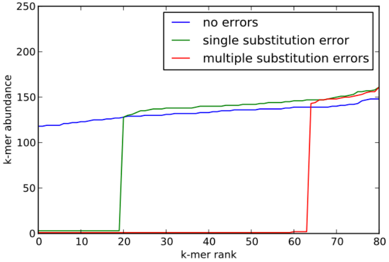
>1 error

3mer freq.

CAT=32
ATG=34
TG**A**=1
G**A**A=1
AAT=1
AT**C**=1
T**C**G=1

CATG**A**AT**C**G
CAT
ATG
TG**A**
G**A**A
AAT
AT**C**
T**C**G

Digital Normalization



Median kmer abundance

0 error: 32,33,34,34,35,36,40

1 error: 1,1,1,32,24,24,40

>1 error: 1,1,1,1,1,32,34

Digital Normalization

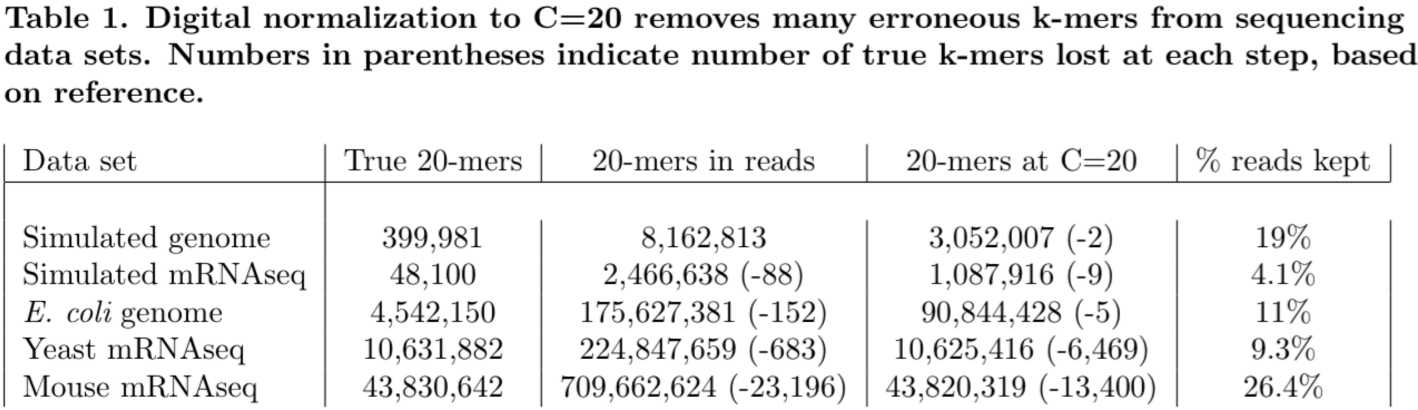


Table 1. Digital normalization to C=20 removes many erroneous k-mers from sequencing data sets. Numbers in parentheses indicate number of true k-mers lost at each step, based on reference.

Data set	True 20-mers	20-mers in reads	20-mers at C=20	% reads kept
Simulated genome	399,981	8,162,813	3,052,007 (-2)	19%
Simulated mRNAseq	48,100	2,466,638 (-88)	1,087,916 (-9)	4.1%
E. coli genome	4,542,150	175,627,381 (-152)	90,844,428 (-5)	11%
Yeast mRNAseq	10,631,882	224,847,659 (-683)	10,625,416 (-6,469)	9.3%
Mouse mRNAseq	43,830,642	709,662,624 (-23,196)	43,820,319 (-13,400)	26.4%

Digital Normalization

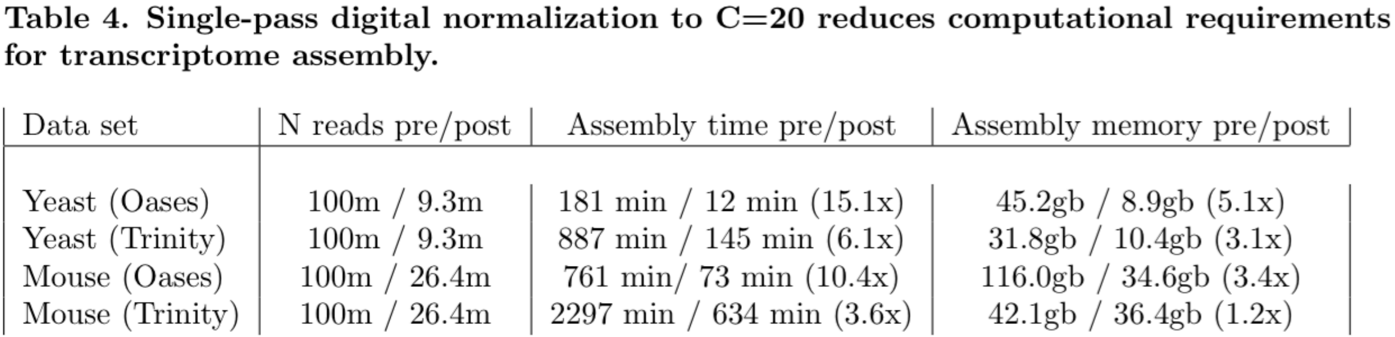


Table 4. Single-pass digital normalization to C=20 reduces computational requirements for transcriptome assembly.

Data set	N reads pre/post	Assembly time pre/post	Assembly memory pre/post
Yeast (Oases)	100m / 9.3m	181 min / 12 min (15.1x)	45.2gb / 8.9gb (5.1x)
Yeast (Trinity)	100m / 9.3m	887 min / 145 min (6.1x)	31.8gb / 10.4gb (3.1x)
Mouse (Oases)	100m / 26.4m	761 min/ 73 min (10.4x)	116.0gb / 34.6gb (3.4x)
Mouse (Trinity)	100m / 26.4m	2297 min / 634 min (3.6x)	42.1gb / 36.4gb (1.2x)

Genome Assembly

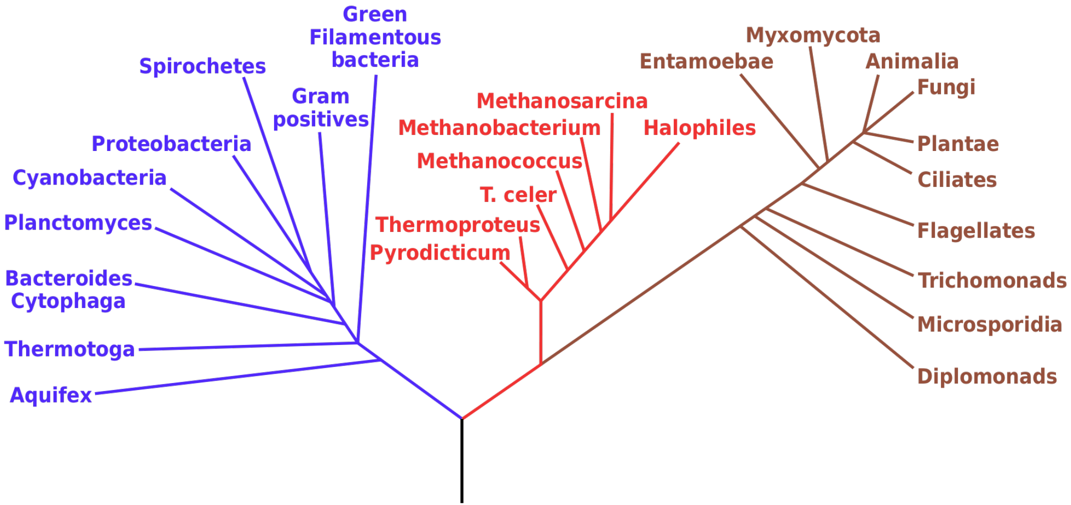
Why do you want to Assemble a genome?

What do you need to Assemble a genome?

Assemble a genome? General Strategies

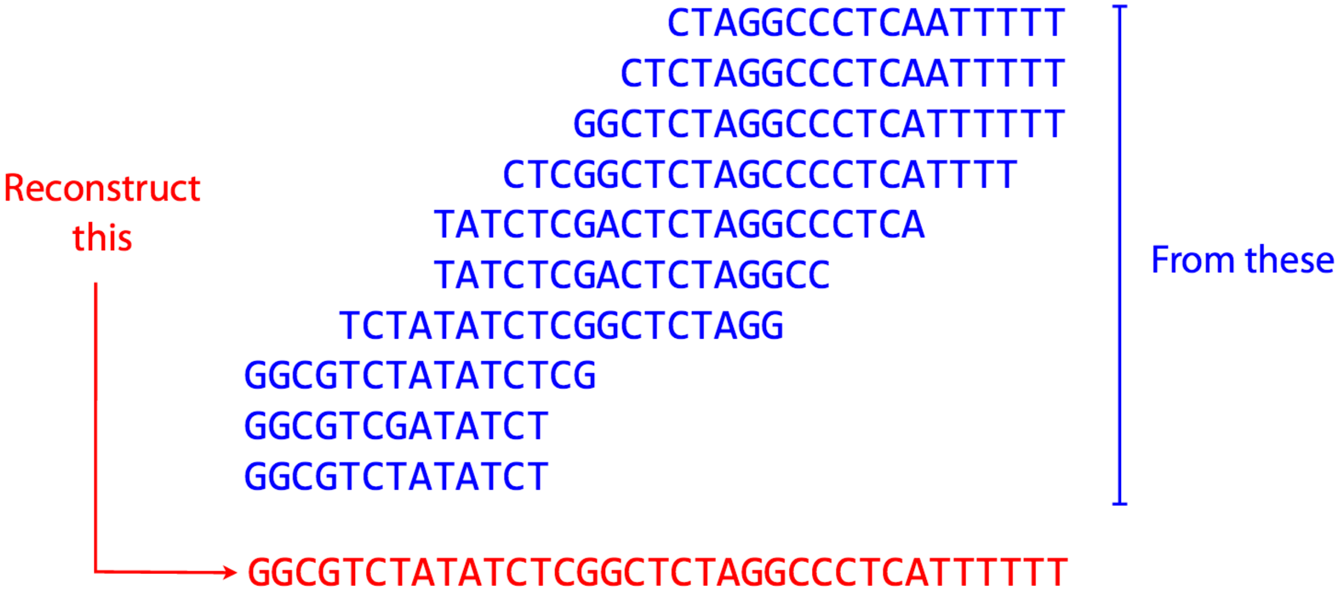
Genome size	Unlimited $$	Typical
>10Mb		
10Mb - 100Mb		
> 100 Mb		

Genome Sizes



Assembly

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



Reconstruct this

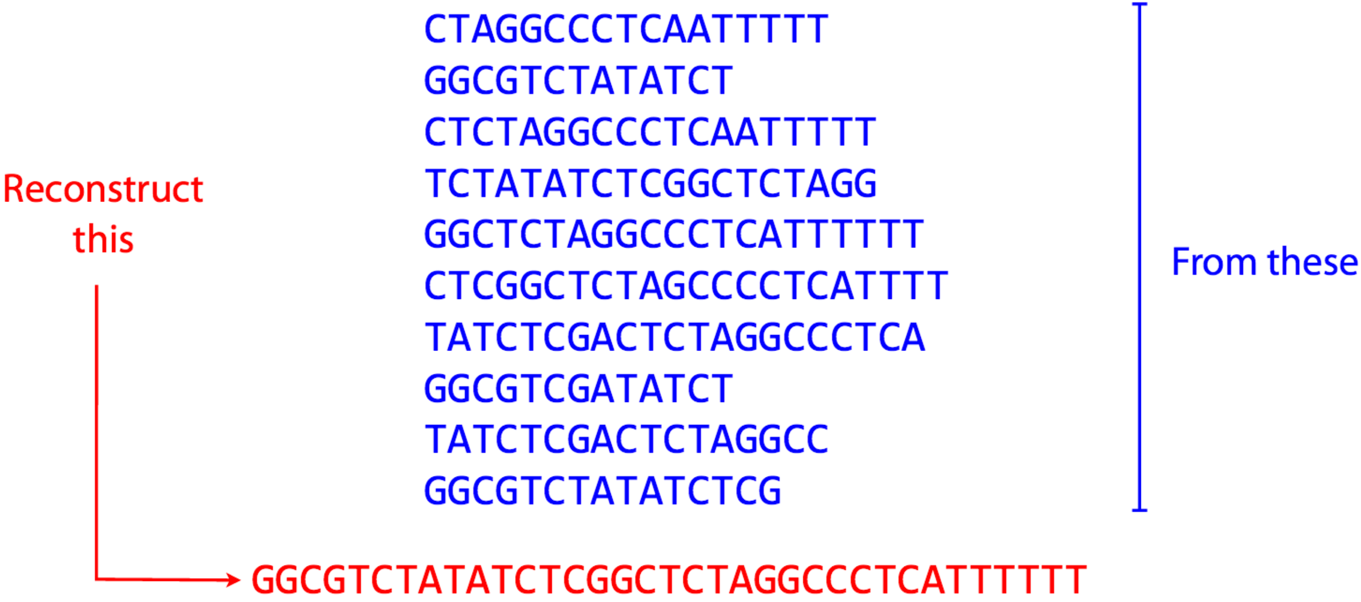
From these

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

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
           CTCGGCTCTAGCCCTCATTTT
        TATCTCGACTCTAGGCCCTCA             177 nucleotides
        TATCTCGACTCTAGGCC
    TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT       35 nucleotides
```

Average coverage = 177 / 35 ≈ 7x

Other Assembly Terms

Unitig

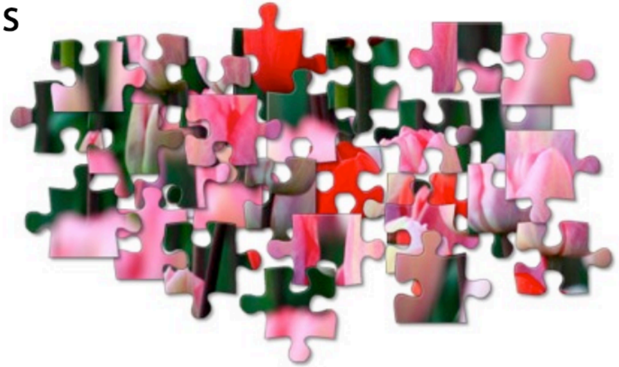
Contig

scaffold

Assembly

- Complicated by:

Reads



+

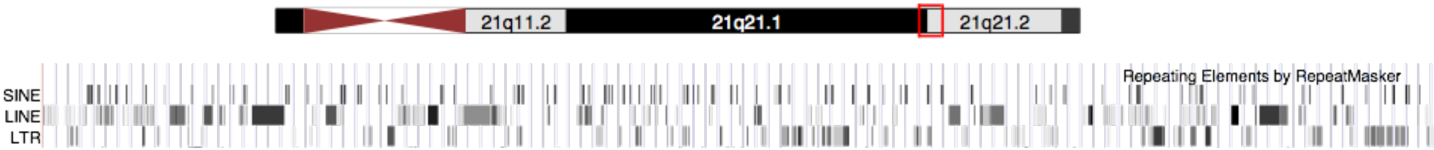
Reference genome

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

Input DNA

Assembly

- Complicated by:



ASSEMBLY

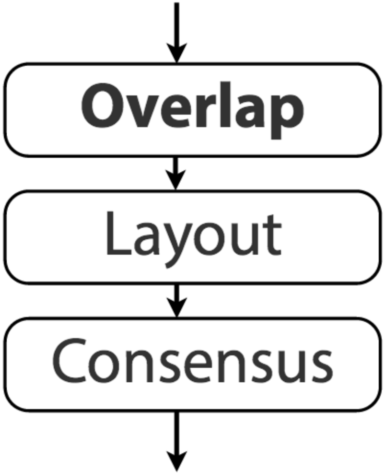
- Work flow:

Assembly

- 3 assembly strategies:

ASSEMBLY

- OLC Assembly



Overlap

Build overlap graph

Layout

Bundle stretches of the overlap graph into *contigs*

Consensus

Pick most likely nucleotide sequence for each contig

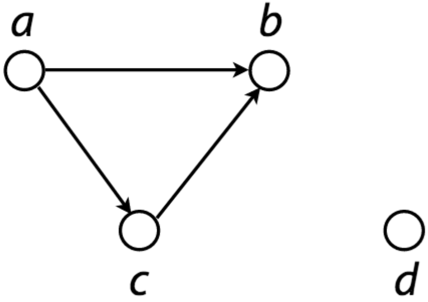
Assembly

- 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

Vertex also called *node* or *point*

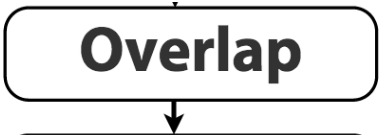
$$V = \{ a, b, c, d \}$$

$$E = \{ (\underset{\uparrow \atop \text{Source}}{a}, \underset{\uparrow \atop \text{Sink}}{b}), (a, c), (c, b) \}$$

Edge also called *arc* or *line*

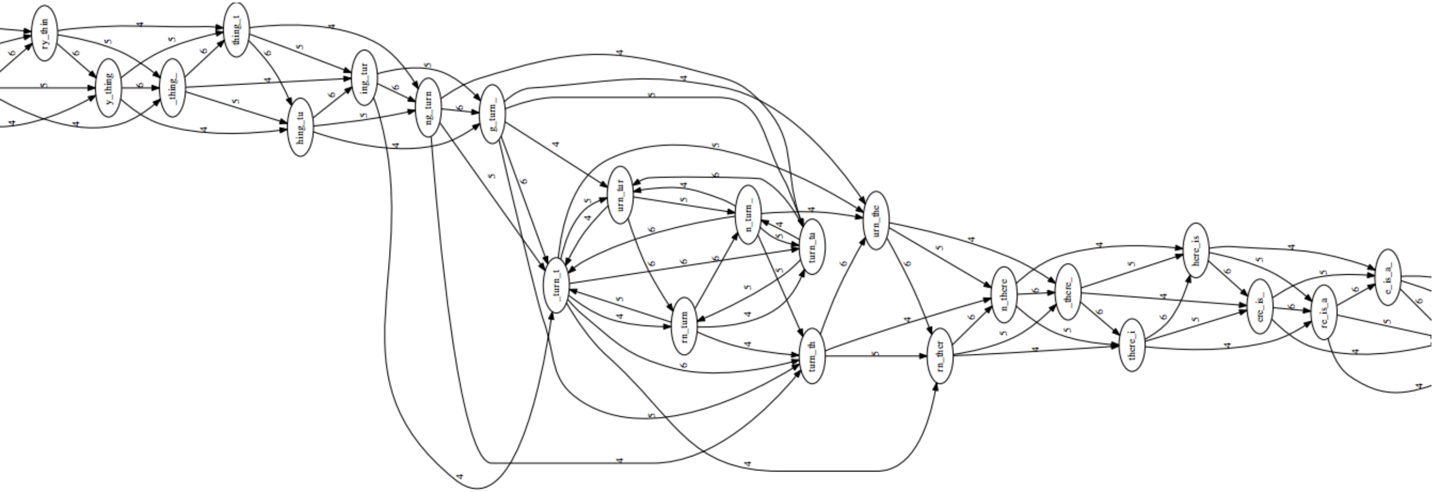
Directed graph also called *digraph*

ASSEMBLY

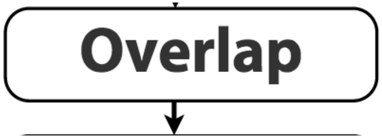

Overlap

Build overlap graph

to_every_thing_turn_turn_turn_there_is_a_season
L=4, k=7



ASSEMBLY

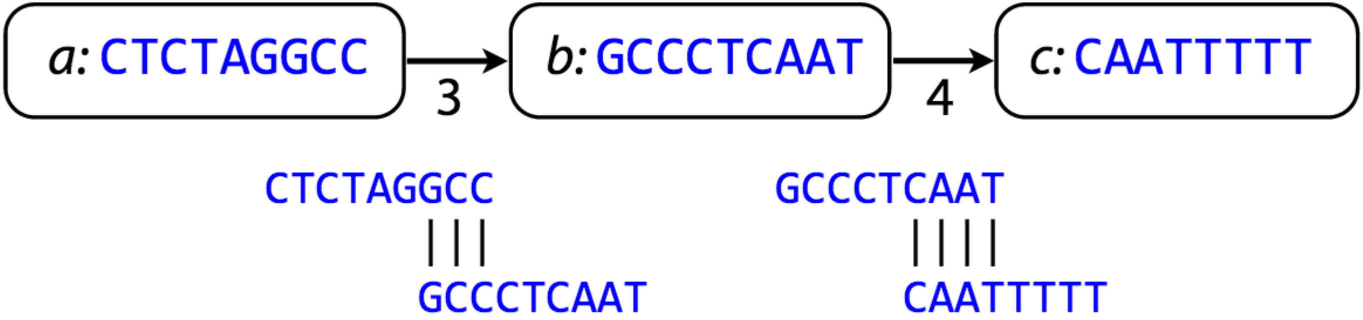


Overlap

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