Genome Assembly Background and Strategy

BIOL 7210: Computational Genomics - Spring 2018

Team-1 Members: Kunal Agarwal, Victoria Caban, Vasanta Chivukula, Seonggeon Cho, Siarhei Hladyshau, Hunter Seabolt, Nirav Shah, Tianze Song, Qinwei Zhuang

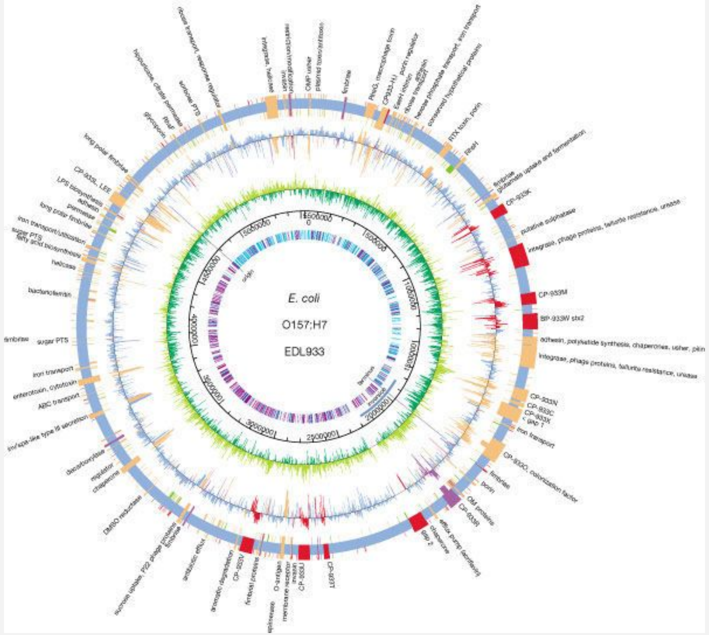
Bacterial Genomics

Bacterial genomics is the discipline concerning the genome of a bacteria and includes all hereditary information regarding that bacteria.

Bacterial genomics helps study bacterial evolution as well as determine the causative agent in disease outbreaks.

Helps identify bacterial pathogens (and antibiotic resistance) and how these pathogens interact with their host.

As Bioinformaticians, it is our job to decipher this information.



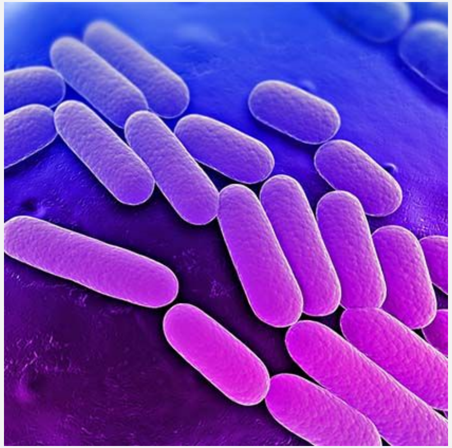
Picture source: Blattner, F. R. et al (2001) Genome sequence of enterohaemorrhagic Escherichia coli O157:H7. *Nature* **409**, 529

Klebsiella - General Characteristics

- Gram negative, non-motile, straight rods
- Singly, in pairs or short chains
- Capsule forming
- Both respiratory and fermentative metabolism (facultative)
- Oxidase negative
- Nosocomial and UTI

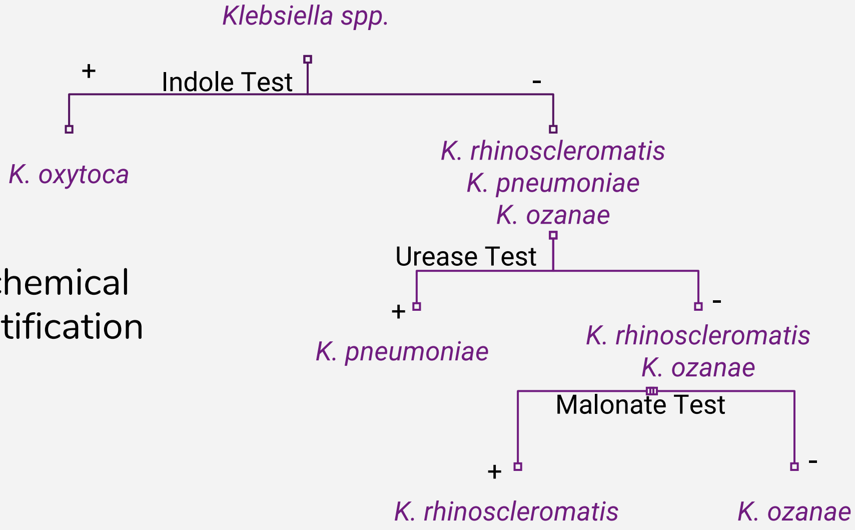
Classification

Proteobacteria
Gammaproteobacteria
Enterobacteriales
Enterobacteriaceae
Klebsiella



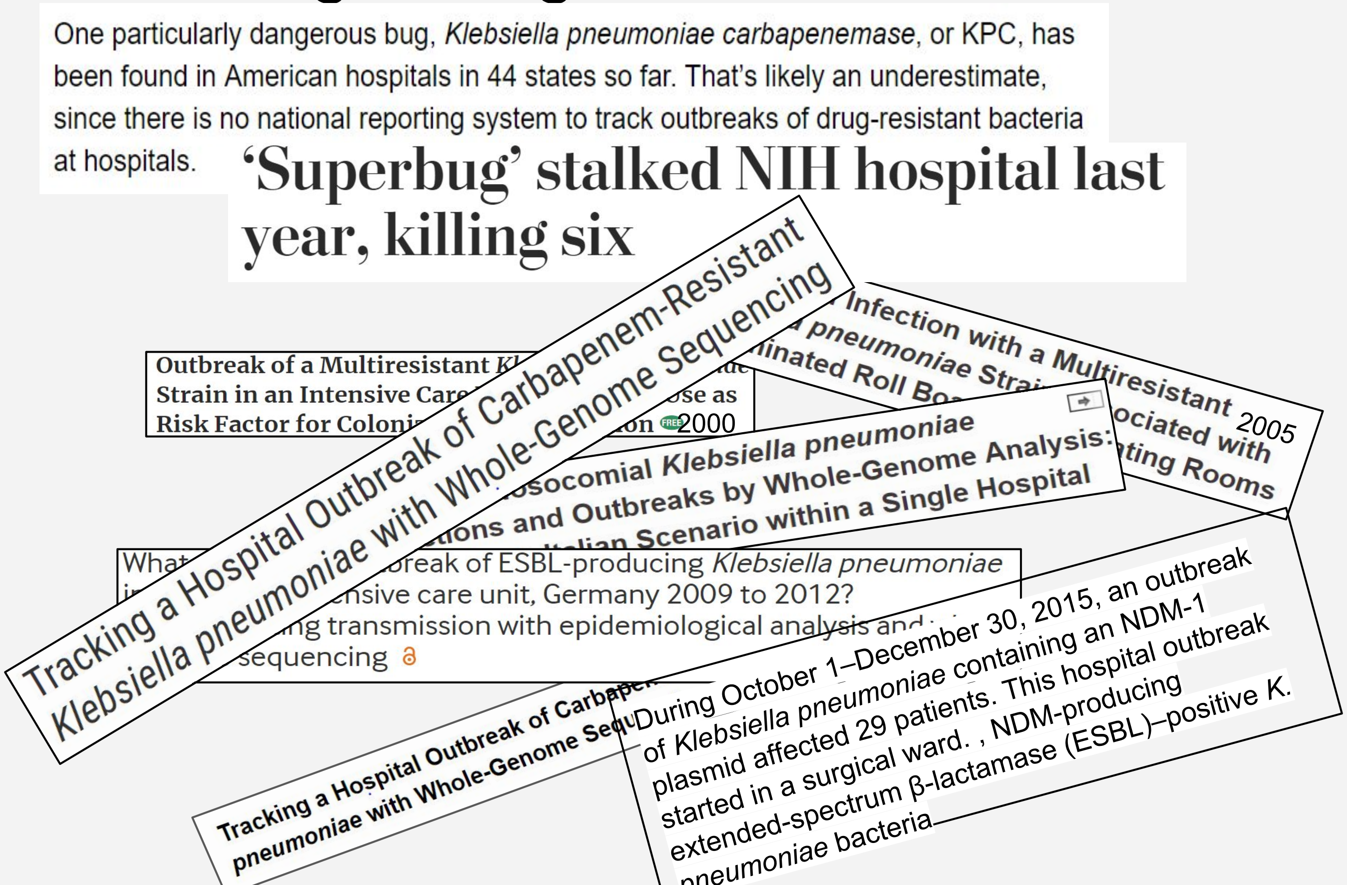
Picture source:
http://healthcare.bioquell.com

Biochemical identification



Source: Bergey’s Manual of Systematic Bacteriology

A Superbug Outbreak at NIH - 2013
Hunting the Nightmare Bacteria



One particularly dangerous bug, *Klebsiella pneumoniae carbapenemase*, or KPC, has been found in American hospitals in 44 states so far. That's likely an underestimate, since there is no national reporting system to track outbreaks of drug-resistant bacteria at hospitals.

'Superbug' stalked NIH hospital last year, killing six

Tracking a Hospital Outbreak of Carbapenem-Resistant *Klebsiella pneumoniae* with Whole-Genome Sequencing

Outbreak of a Multiresistant *K*
Strain in an Intensive Care
ose as
Risk Factor for Coloni
on FREE 2000

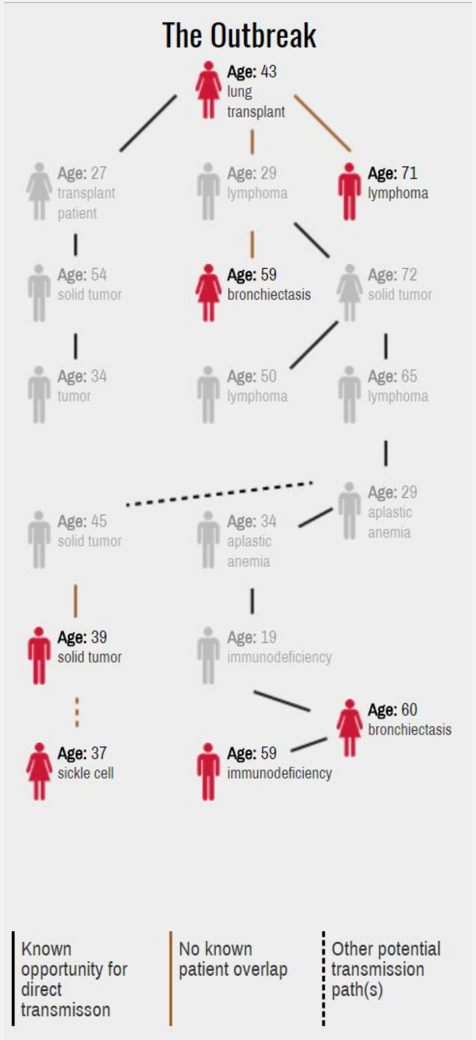
Infection with a Multiresistant
***pneumoniae* Strai**
ociated with
inated Roll Bo
ting Rooms
2005

osocomial *Klebsiella pneumoniae*
tions and Outbreaks by Whole-Genome Analysis:
Italian Scenario within a Single Hospital

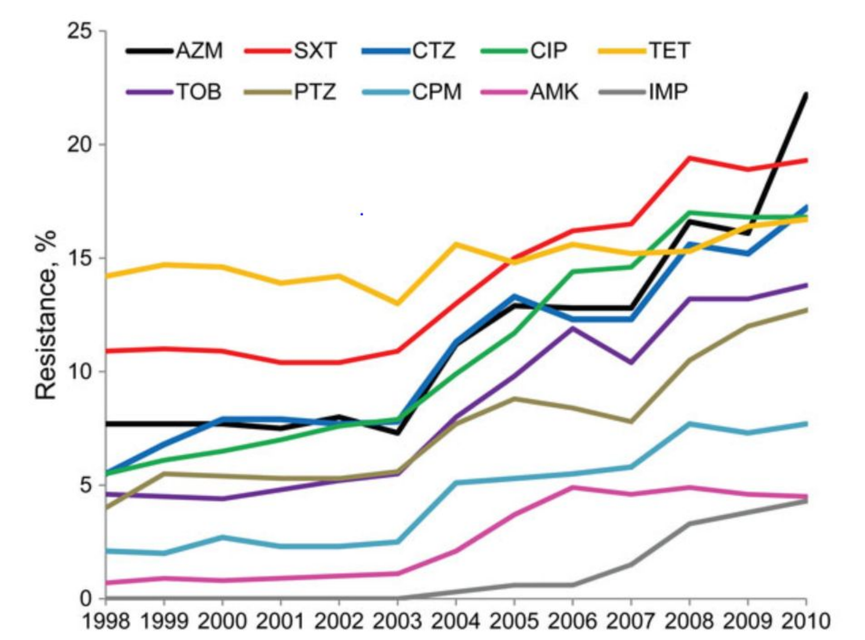
What
break of ESBL-producing *Klebsiella pneumoniae*
i
ensive care unit, Germany 2009 to 2012?
ng transmission with epidemiological analysis and
sequencing

Tracking a Hospital Outbreak of Carbape
***pneumoniae* with Whole-Genome Seq**

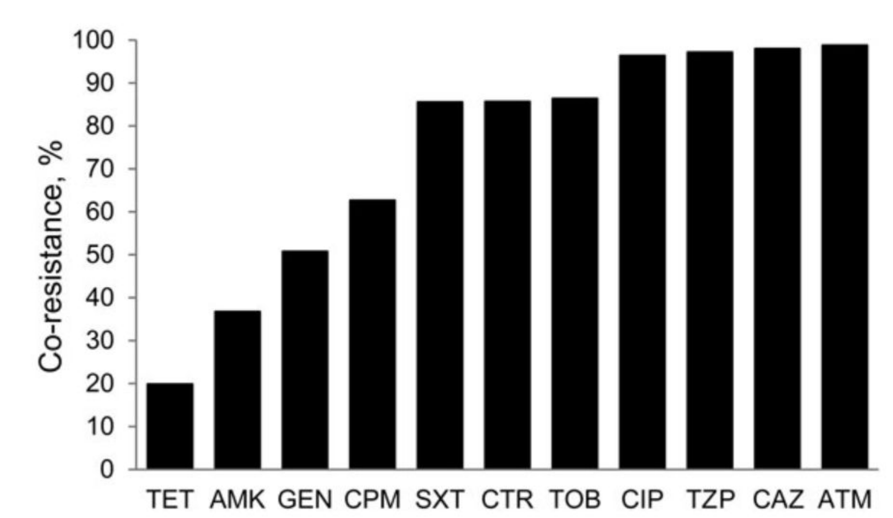
During October 1–December 30, 2015, an outbreak of *Klebsiella pneumoniae* containing an NDM-1 plasmid affected 29 patients. This hospital outbreak started in a surgical ward. , NDM-producing extended-spectrum β-lactamase (ESBL)–positive K. pneumoniae bacteria



Klebsiella pneumoniae Antimicrobial Drug Resistance, United States, 1998–2010



Klebsiella pneumoniae antimicrobial drug resistance, United States, 1998–2010

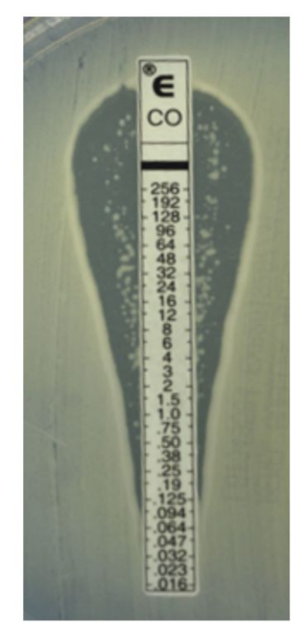


Prevalence of antimicrobial cross-resistance among imipenem-resistant *Klebsiella pneumoniae* isolates, United States, 2010

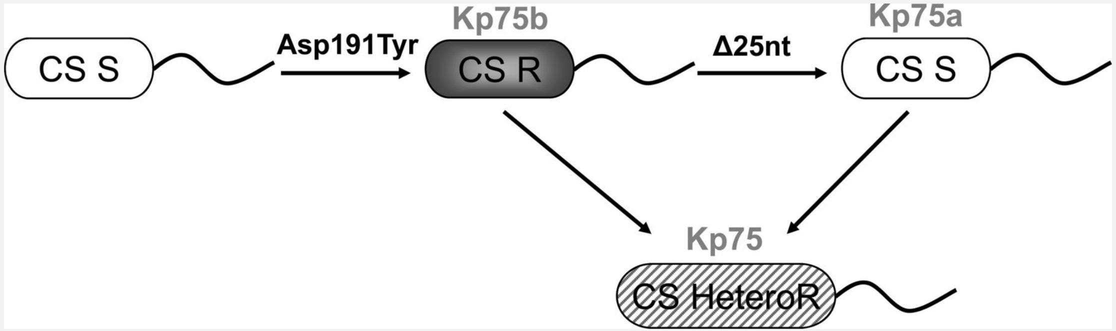
Source: Centers for Disease Control and Prevention

Colistin heteroresistance in *K. pneumoniae*

David Weiss - Genetically identical, but phenotypically distinct, subpopulation of colistin-resistant bacteria can mediate in vivo treatment failure



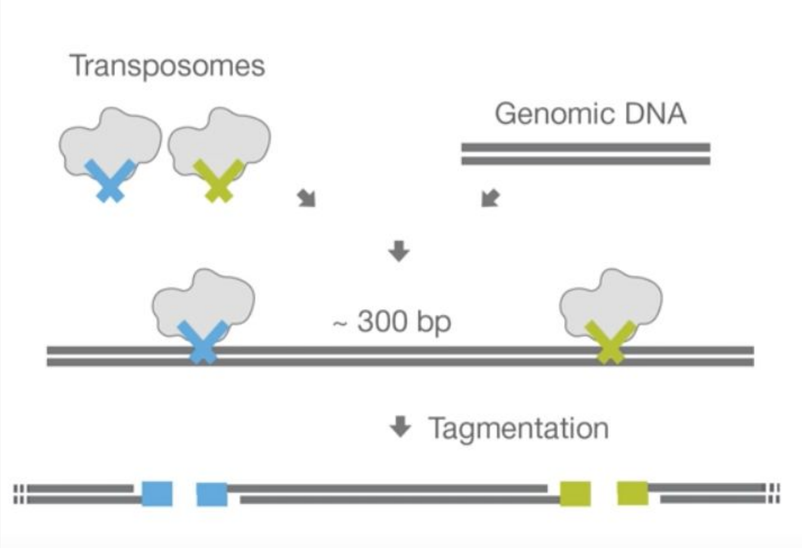
Heteroresistant subpopulation

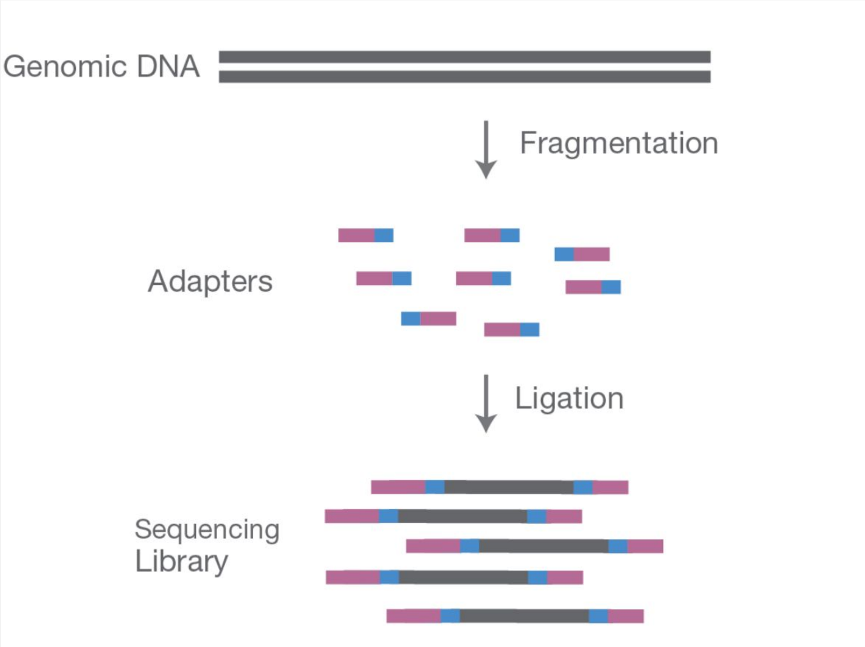


Schematic representation of mechanism leading to heteroresistance

Source: Poirel at al. 2015. Heteroresistance to colistin in *Klebsiella pneumoniae* associated with alteration sin the PHOPQ regulatory syste. Antimicrob Agents Chemother 59:2780-2784

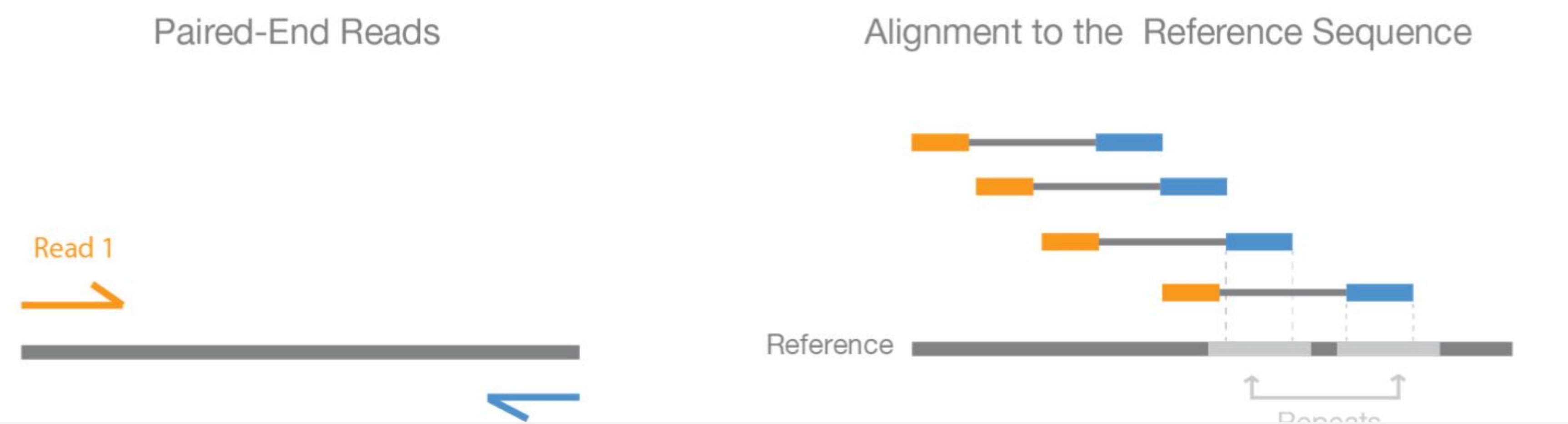
Library Preparation





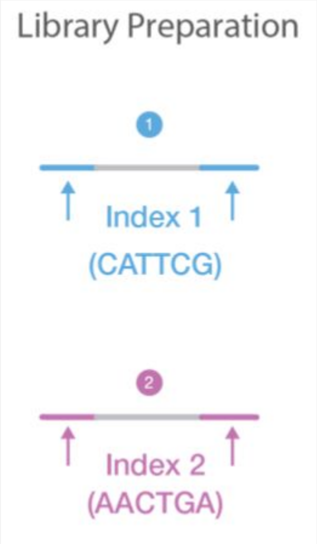
Source: Illumina, An Introduction to Next-Generation Sequencing Technology, 2017

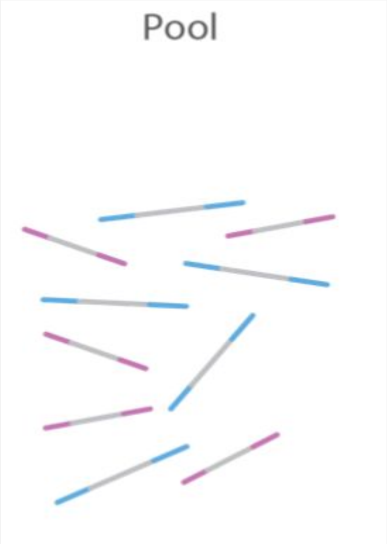
Sequencing: Paired-end

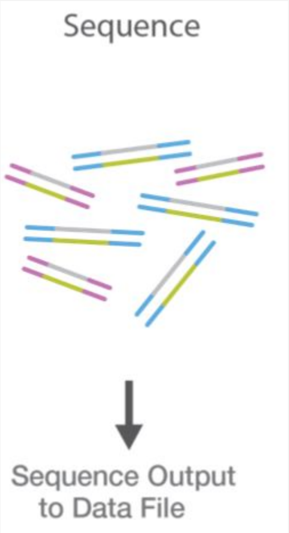


Source: Illumina, An Introduction to Next-Generation Sequencing Technology, 2017

Sequencing: Output File





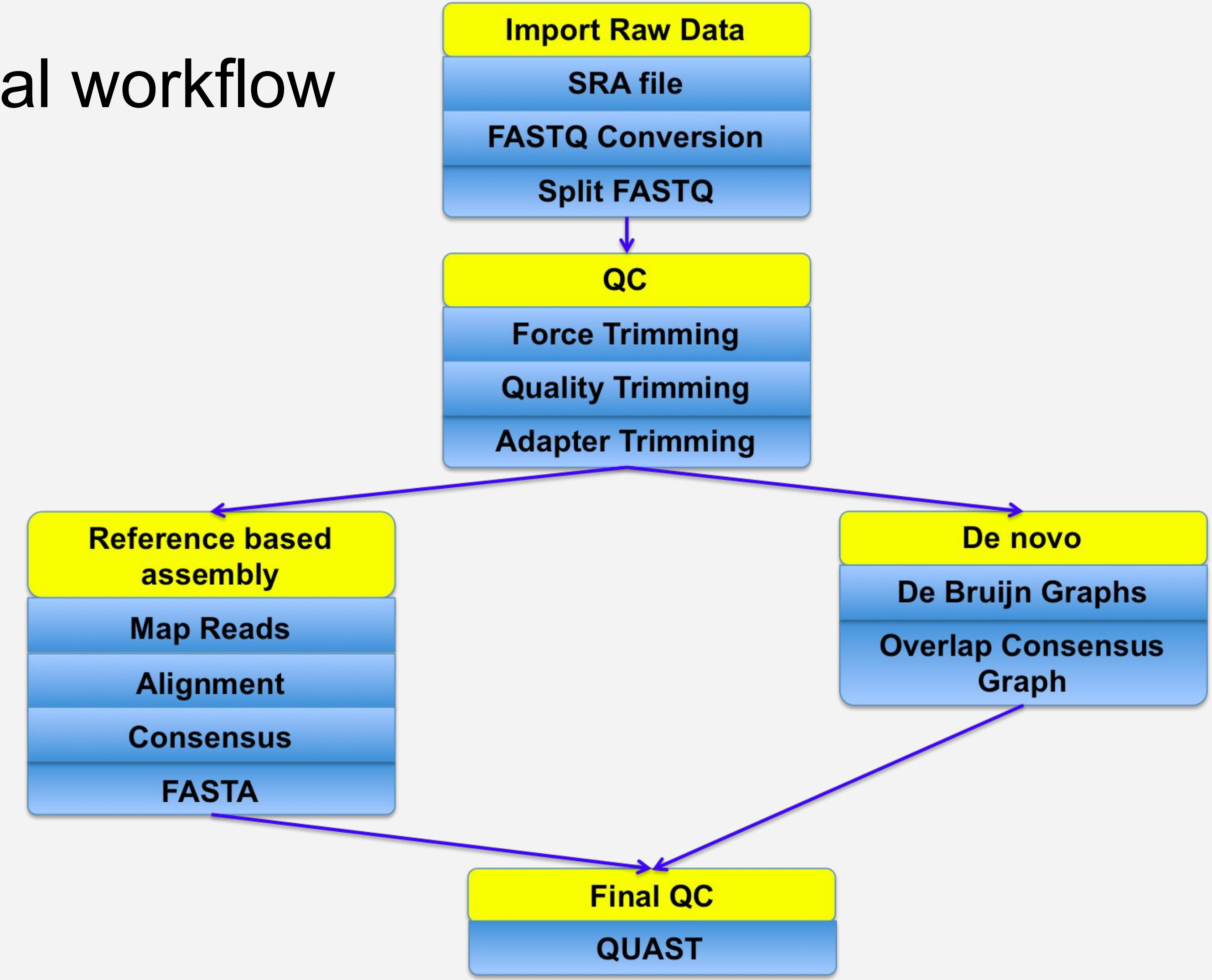


```
CATTCGACGGATCG
AACTGAGTCCGATA
AACTGATCGGATCC
CATTCGTGGCAGTC
AACTGAACCTGATG
AACTGAGATTACAA
CATTCGCAGTTCATT
CATTCGAACTTCGA
```

- 🟦 Library 1 Barcode
- 🟪 Library 2 Barcode
- 🟩 Sequencing Reads

Source: Illumina, An Introduction to Next-Generation Sequencing Technology, 2017

General workflow



Import Raw Data

SRA file

FASTQ Conversion

Split FASTQ

QC

Force Trimming

Quality Trimming

Adapter Trimming

Reference based assembly

Map Reads

Alignment

Consensus

FASTA

De novo

De Bruijn Graphs

Overlap Consensus Graph

Final QC

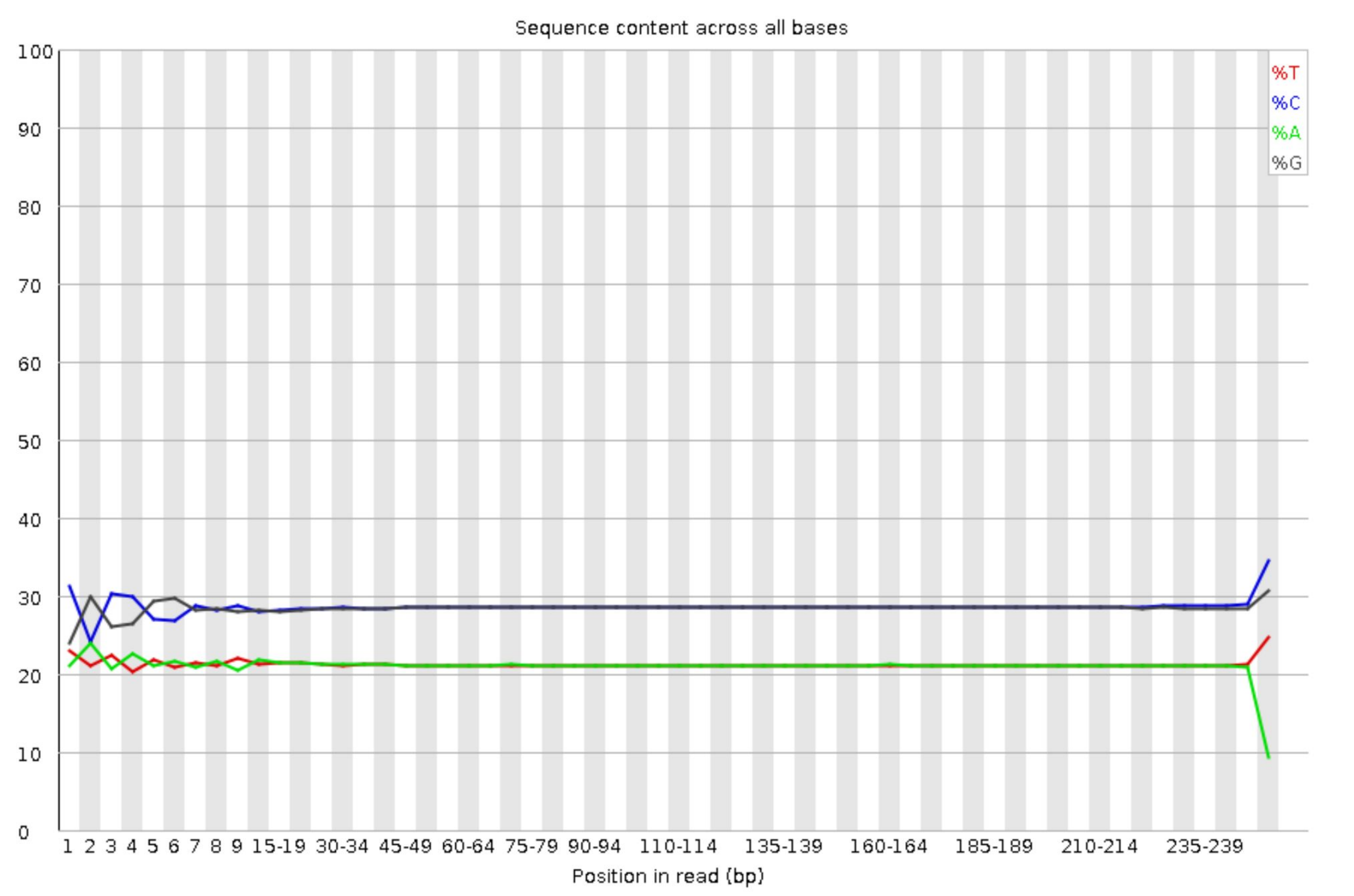
QUAST

QC——FastQC

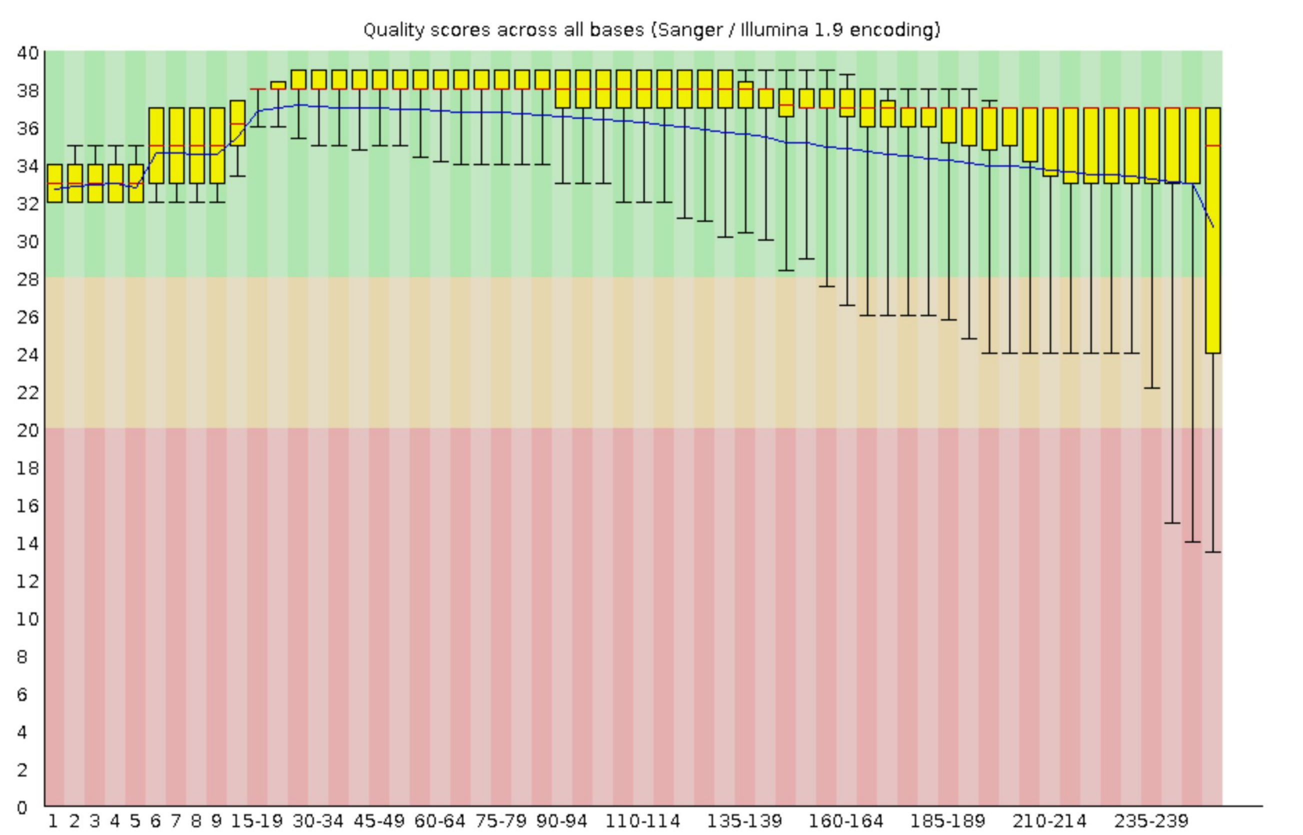
Forced Trimming

Beginning: 15-20bp

End: 5bp

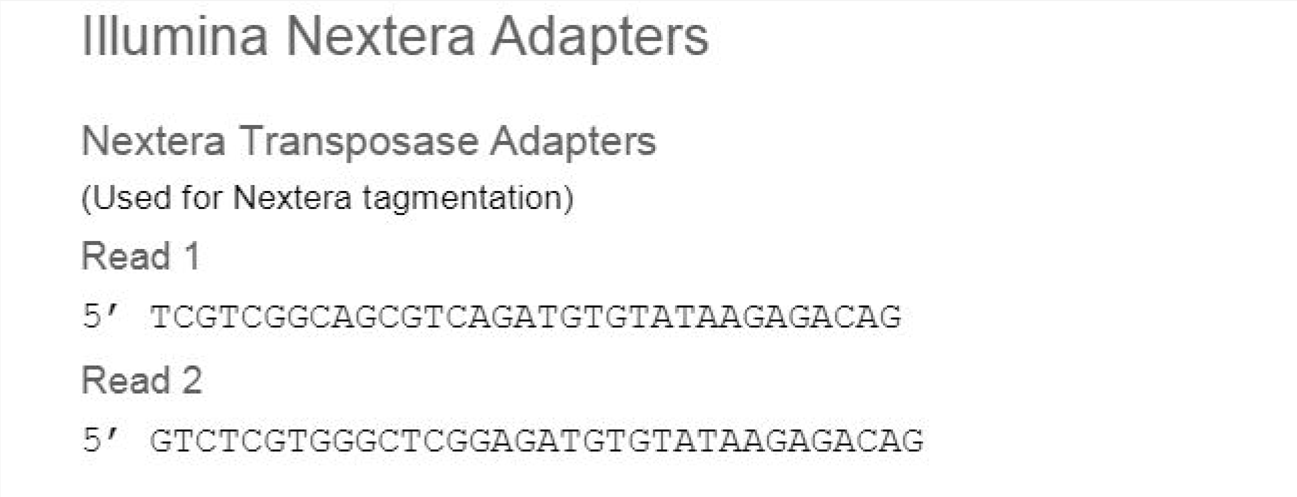


Quality Trimming



Quality Trimming
Quality score < 20 trimmed

Adapter Trimming



Illumina Nextera Adapters

Nextera Transposase Adapters

(Used for Nextera tagmentation)

Read 1

```
5′ TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG
```

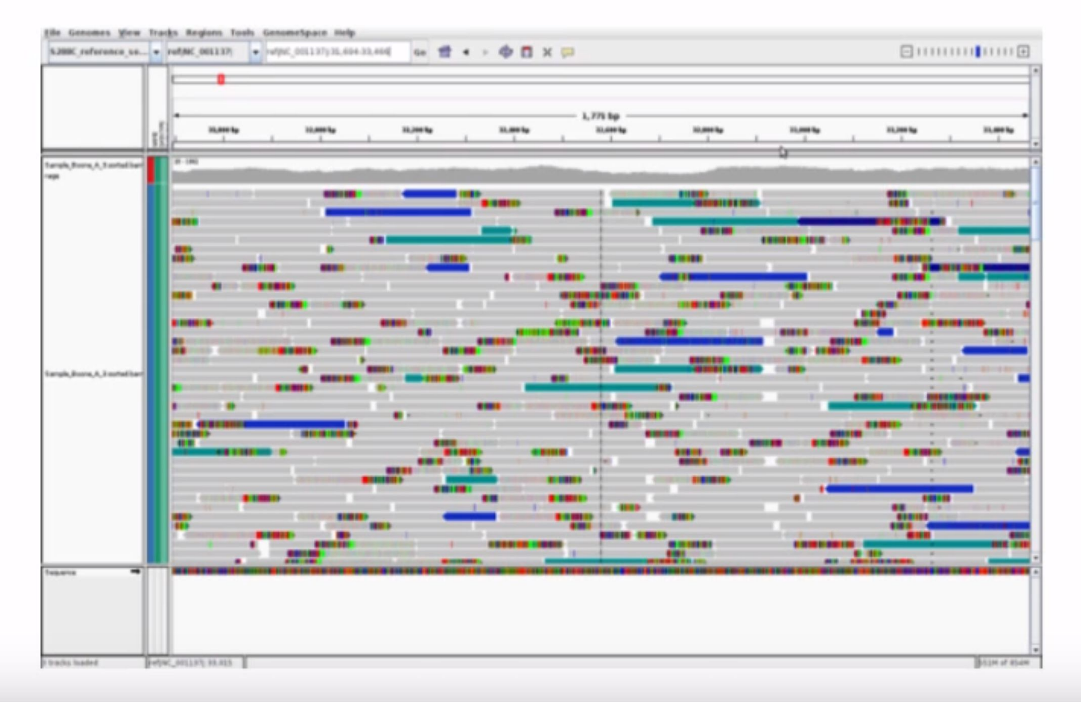
Read 2

```
5′ GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG
```

Source: Protocol, Illumina Adapter Sequences

Adapters can ruin the Assembly

They look like very high copy repeats



Source: De novo Assembly, Bioinformatics DotCa

Burrows–Wheeler transformation (BWT)

- BWT is used in mapping short reads to a reference.
- Intuition of how BWT reduces running(mapping) time.
- Tools implementing BWT: BWA, Bowtie.
- Topics we are going to talk about today:
  - How does it work? (A step-wise tutorial)
  - Brief introduction of annotation for matched position on the reference of patterns (suffix array) and inexact matching (error counting array).

How does BWT work?

Steps:

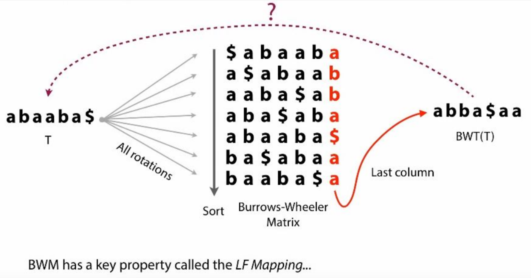
(a) Sort all rotations of the text into lexicographic order ($ always as the first row). Only keep the first and last column.

(b) Invert the BWT matrix (BWM).

(c) Map patterns to the data structure

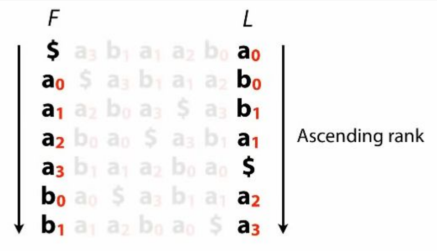
Intuitions:

The first and last column include order information while “$” marks the end of the original sequence.

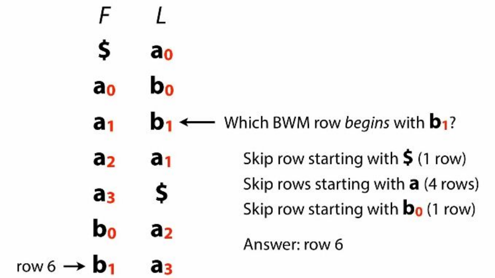


$ a b a a b a
a $ a b a a b
a a b a $ a b
a b a $ a b a
a b a a b a $
b a $ a b a a
b a a b a $ a

BWM has a key property called the *LF Mapping*...



$\$\ a_3\ b_1\ a_1\ a_2\ b_0\ a_0$
$a_0\ \$\ a_3\ b_1\ a_1\ a_2\ b_0$
$a_1\ a_2\ b_0\ a_3\ \$\ a_3\ b_1$
$a_2\ b_0\ a_0\ \$\ a_3\ b_1\ a_1$
$a_3\ b_1\ a_1\ a_2\ b_0\ a_0\ \$$
$b_0\ a_0\ \$\ a_3\ b_1\ a_1\ a_2$
$b_1\ a_1\ a_2\ b_0\ a_0\ \$\ a_3$



F	L
$	a_0
a_0	b_0
a_1	b_1
a_2	a_1
a_3	$
b_0	a_2
b_1	a_3

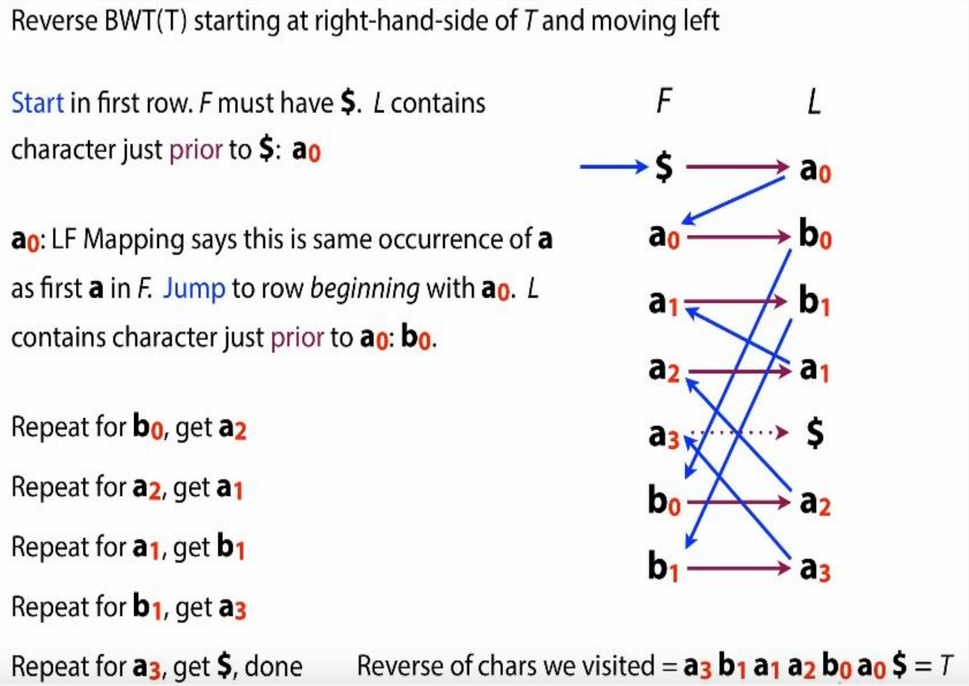
Which BWM row *begins* with **b_1**?

Skip row starting with **$** (1 row)
Skip rows starting with **a** (4 rows)
Skip row starting with **b_0** (1 row)

Answer: row 6

https://www.youtube.com/watch?time_continue=733&v=4n7NPk5lwbI

How does BWT work?



Reverse BWT(T) starting at right-hand-side of *T* and moving left

Start in first row. *F* must have $\mathbf{\$}$. *L* contains character just prior to $\mathbf{\$}$: $\mathbf{a}_0$

$\mathbf{a}_0$: LF Mapping says this is same occurrence of **a** as first **a** in *F*. Jump to row *beginning* with $\mathbf{a}_0$. *L* contains character just prior to $\mathbf{a}_0$: $\mathbf{b}_0$.

Repeat for $\mathbf{b}_0$, get $\mathbf{a}_2$

Repeat for $\mathbf{a}_2$, get $\mathbf{a}_1$

Repeat for $\mathbf{a}_1$, get $\mathbf{b}_1$

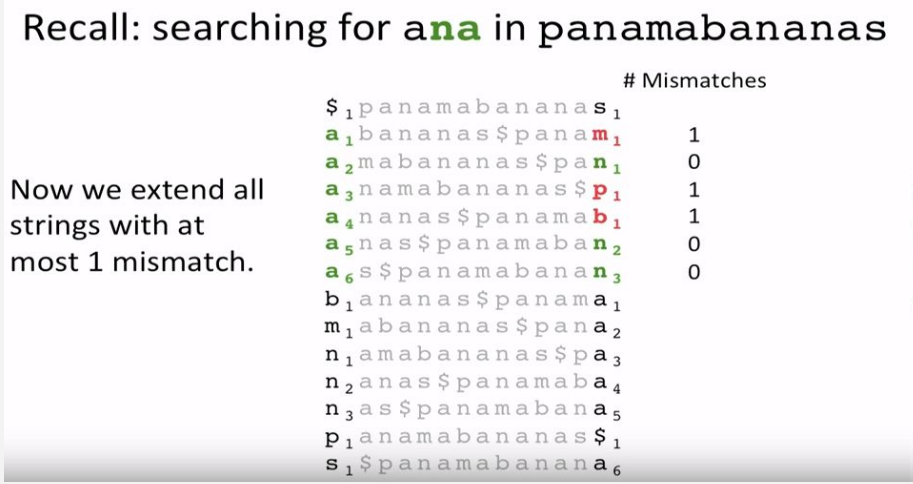
Repeat for $\mathbf{b}_1$, get $\mathbf{a}_3$

Repeat for $\mathbf{a}_3$, get $\mathbf{\$}$, done

Reverse of chars we visited = $\mathbf{a}_3\ \mathbf{b}_1\ \mathbf{a}_1\ \mathbf{a}_2\ \mathbf{b}_0\ \mathbf{a}_0\ \mathbf{\$} = T$

www.langmead-lab.org/teaching-materials

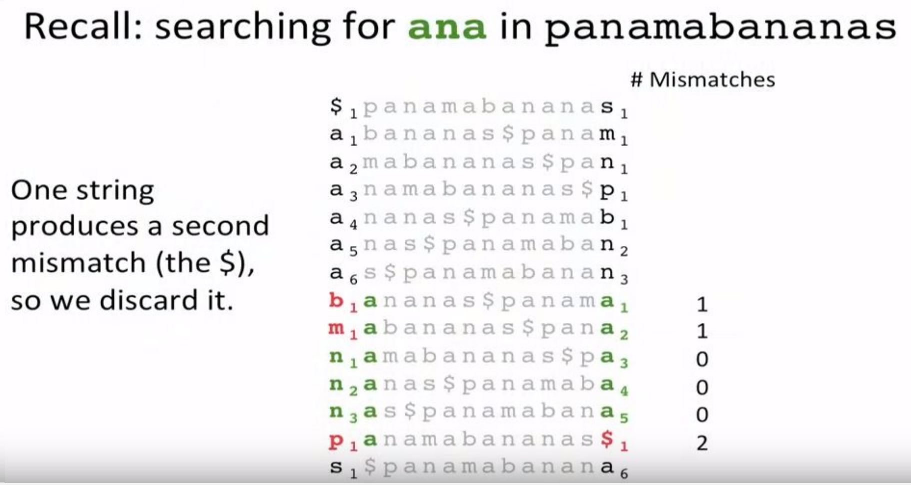
How does BWT work?



Recall: searching for ana in panamabananas

Now we extend all strings with at most 1 mismatch.

	# Mismatches
$\$_1$ panamabanana s_1	
a_1 bananas\$pana m_1	1
a_2 mabananas\$pa n_1	0
a_3 namabananas\$ p_1	1
a_4 nanas\$panama b_1	1
a_5 nas\$panamaba n_2	0
a_6 s\$panamabana n_3	0
b_1 ananas\$panam a_1	
m_1 abananas\$pan a_2	
n_1 amabananas\$p a_3	
n_2 anas\$panamab a_4	
n_3 as\$panamaban a_5	
p_1 anamabananas $\$_1$	
s_1 \$panamabanan a_6	



Recall: searching for ana in panamabananas

One string produces a second mismatch (the $), so we discard it.

	# Mismatches
$\$_1$ panamabanana s_1	
a_1 bananas\$pana m_1	
a_2 mabananas\$pa n_1	
a_3 namabananas\$ p_1	
a_4 nanas\$panama b_1	
a_5 nas\$panamaba n_2	
a_6 s\$panamabana n_3	
b_1 ananas\$panam a_1	1
m_1 abananas\$pan a_2	1
n_1 amabananas\$p a_3	0
n_2 anas\$panamab a_4	0
n_3 as\$panamaban a_5	0
p_1 anamabananas $\$_1$	2
s_1 \$panamabanan a_6	

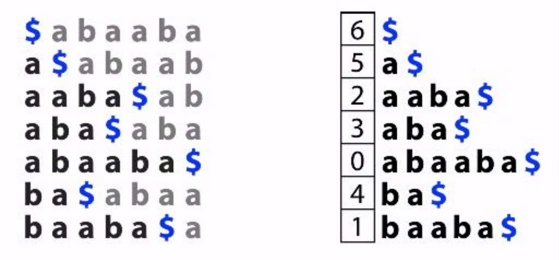
https://www.youtube.com/watch?v=Vjnm-jF1PBQ

Brief introduction of annotation for matched position matched patterns (suffix array) and inexact matching (error counting array)

Suffix array (SA) can be precalculated and is used to annotate the matched position found on the reference.

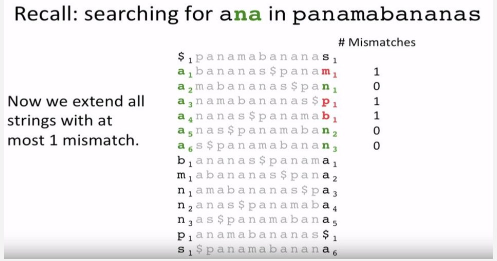
So far all we talked about was exacting matching. However, BWT can be modified to work for inexact matching. The basic idea is to carry an array for counting the number of unmatched bp. (example of “panamabanana”)

If interested, see the video: https://www.youtube.com/watch?v=Vjnm-jF1PBQ



$ a b a a b a	6	$
a $ a b a a b	5	a $
a a b a $ a b	2	a a b a $
a b a $ a b a	3	a b a $
a b a a b a $	0	a b a a b a $
b a $ a b a a	4	b a $
b a a b a $ a	1	b a a b a $

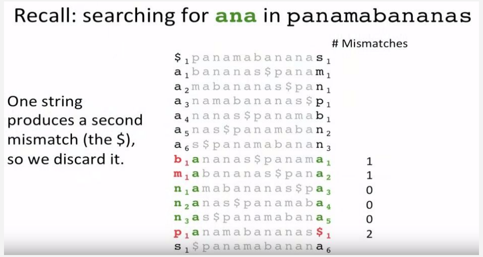
https://www.youtube.com/watch?v=kvVGj5V65io



Recall: searching for ana in panamabananas

Now we extend all strings with at most 1 mismatch.

			# Mismatches
$\$_1$	panamabanana	s_1	
a_1	bananas$panam	m_1	1
a_2	mabananas$pan	n_1	0
a_3	namabananas$	p_1	1
a_4	nanas$panama	b_1	1
a_5	nas$panamaba	n_2	0
a_6	s$panamabana	n_3	0
b_1	ananas$panam	a_1	
m_1	abananas$pan	a_2	
n_1	amabananas$p	a_3	
n_2	anas$panamab	a_4	
n_3	as$panamaban	a_5	
p_1	anamabananas	$\$_1$	
s_1	$panamabanan	a_6	



Recall: searching for ana in panamabananas

One string produces a second mismatch (the $), so we discard it.

			# Mismatches
$\$_1$	panamabanana	s_1	
a_1	bananas$panam	m_1	
a_2	mabananas$pan	n_1	
a_3	namabananas$	p_1	
a_4	nanas$panama	b_1	
a_5	nas$panamaba	n_2	
a_6	s$panamabana	n_3	
b_1	ananas$panam	a_1	1
m_1	abananas$pan	a_2	1
n_1	amabananas$p	a_3	0
n_2	anas$panamab	a_4	0
n_3	as$panamaban	a_5	0
p_1	anamabananas	$\$_1$	2
s_1	$panamabanan	a_6	

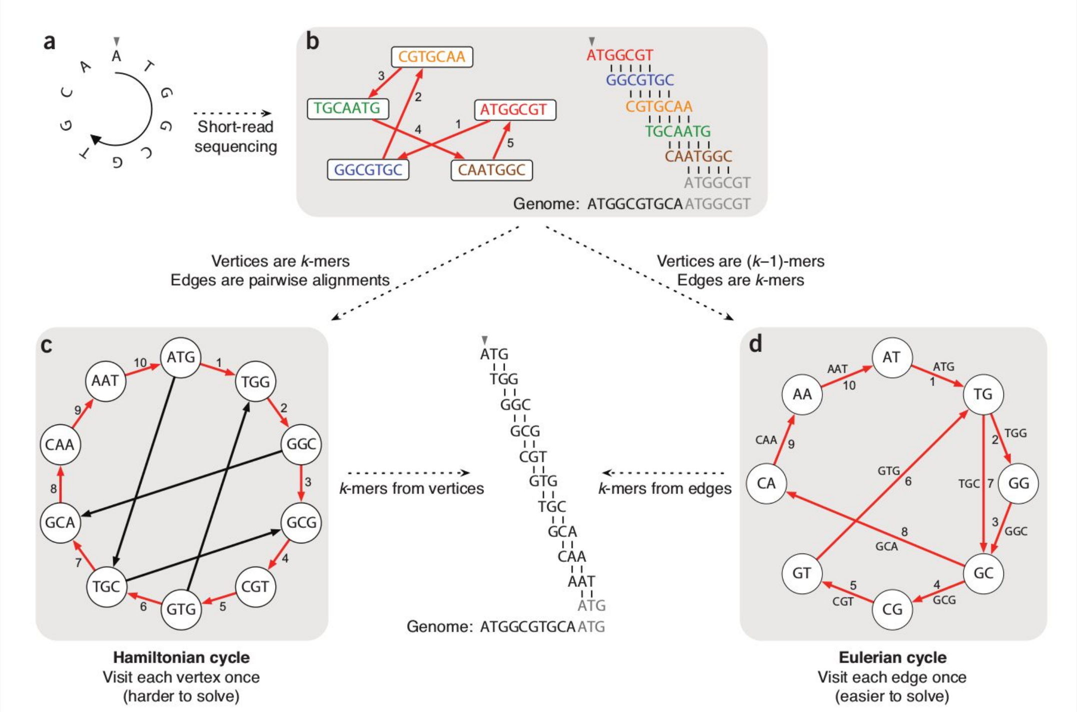
https://www.youtube.com/watch?v=Vjnm-jF1PBQ

Genome assembly alternatives

b. Traditional Sanger sequencing algorithms (reads represent as nodes, edges represent alignments between reads)

c. Overlapping k-mers

d. Building de Bruijn graph (k-mer prefixes and suffixes are nodes, edges represent k-mers having a particular prefix and suffix)

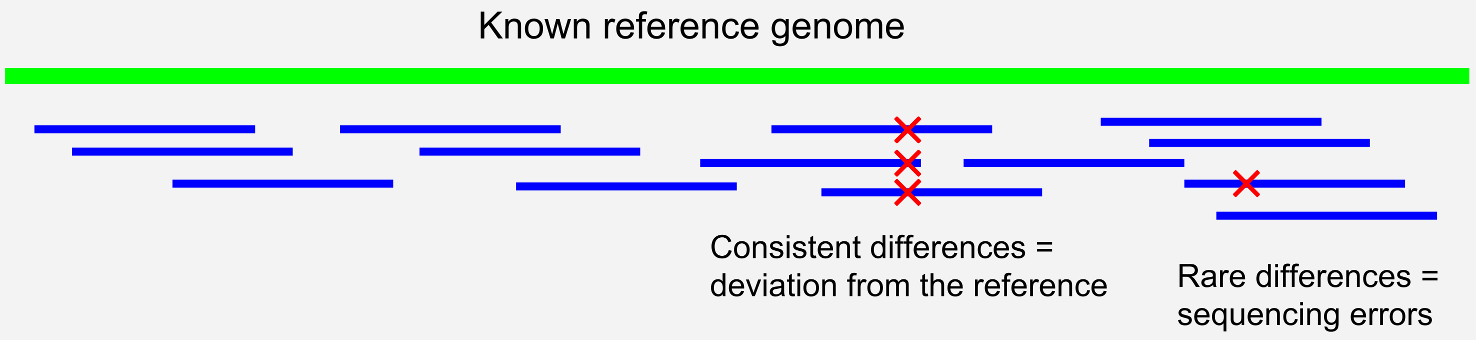


Hamiltonian cycle
Visit each vertex once
(harder to solve)

Eulerian cycle
Visit each edge once
(easier to solve)

Nat Biotechnol. 2011 Nov 8, Compeau PE, Pevzner PA, Tesler G.

Genome assembly, reference-based approach

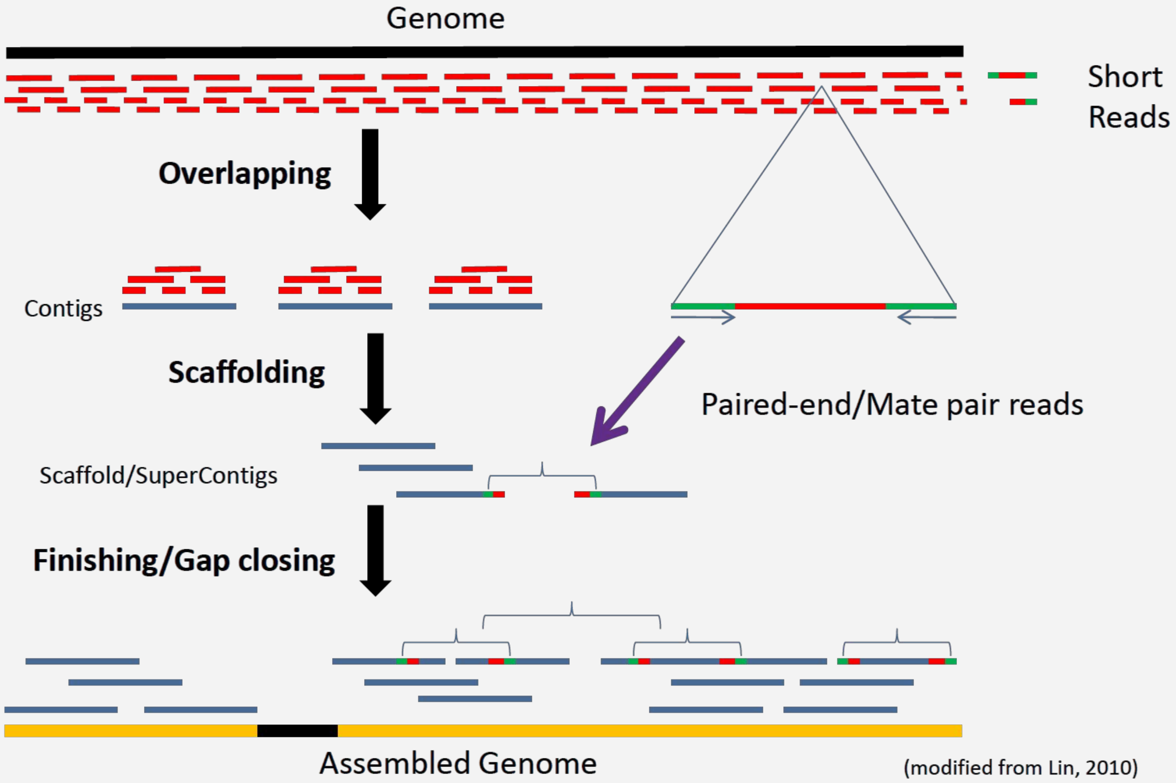

Known reference genome
Consistent differences = deviation from the reference
Rare differences = sequencing errors

Problems:

- Large scale differences:
  - Insertions,
  - Deletions
  - Rearrangements
- Repeats
- Reference bias

Possible solution: combination of mapping and de novo assembly

General workflow of genome assembly



(modified from Lin, 2010)

Measures of assembly quality

- Number of contigs/scaffolds
  - Fewer is better, one is ideal
- Contig sizes
  - Maximum
  - Average
  - Median
  - N50
- Total size
  - Should be close to expected genome size
  - Repeats may only be counted once
- Number of “N”s
  - N is the ambiguous base, fewer is better
- Genes, that must present in this genome (BUSCO)

- The N50 of a set of contigs is the size of the largest contig for which half the total size is contained in that contigs and those larger
  - The weighted median contig size
- Example:
  - 7 contigs totalling 20 units: 7, 4, 3, 2, 2, 1, 1
  - N50 is 4, as 7+4=11, which is > 50% of 20

Dr Torsten Seemann, IMB – Winter School 2011

Bioinformatics. 2015 Oct 1, Simao FA et. al.

Citation

Perna, Nicole T., Guy Plunkett III, Valerie Burland, Bob Mau, Jeremy D. Glasner, Debra J. Rose, George F. Mayhew et al. "Genome sequence of enterohaemorrhagic Escherichia coli O157: H7." Nature 409, no. 6819 (2001): 529.

Bergey, David Hendricks, Robert Stanley Breed, Everitt George Dunne Murray, and A. Parker Hitchens. Bergey's manual of determinative bacteriology. Baltimore: Williams & Wilkins, 1934. Harvard

Sanchez, Guillermo V., Ronald N. Master, Richard B. Clark, Madiha Fyyaz, Padmaraj Duvvuri, Gupta Ekta, and Jose Bordon. "Klebsiella pneumoniae antimicrobial drug resistance, United States, 1998–2010." Emerging infectious diseases 19, no. 1 (2013): 133.

Sanchez, Guillermo V., Ronald N. Master, Richard B. Clark, Madiha Fyyaz, Padmaraj Duvvuri, Gupta Ekta, and Jose Bordon. "Klebsiella pneumoniae antimicrobial drug resistance, United States, 1998–2010." Emerging infectious diseases 19, no. 1 (2013): 133.

Jayol, Aurélie, Patrice Nordmann, Adrian Brink, and Laurent Poirel. "Heteroresistance to colistin in Klebsiella pneumoniae associated with alterations in the PhoPQ regulatory system." Antimicrobial agents and chemotherapy 59, no. 5 (2015): 2780-2784.

Compeau, Phillip EC, Pavel A. Pevzner, and Glenn Tesler. "How to apply de Bruijn graphs to genome assembly." Nature biotechnology 29, no. 11 (2011): 987.

Dr Torsten Seemann, IMB – Winter School 2011

Simão, Felipe A., Robert M. Waterhouse, Panagiotis Ioannidis, Evgenia V. Kriventseva, and Evgeny M. Zdobnov. "BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs." Bioinformatics 31, no. 19 (2015): 3210-3212.

https://www.illumina.com/science/technology/next-generation-sequencing.html

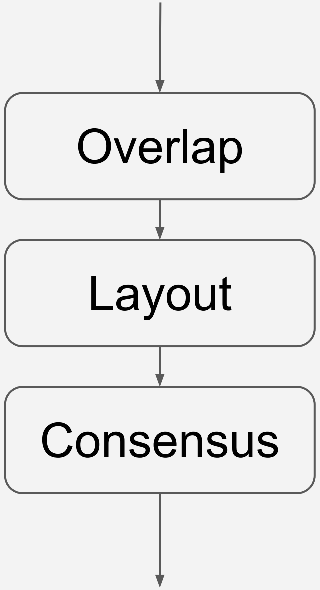
https://www.youtube.com/watch?time_continue=733&v=4n7NPk5lwbI

www.langmead-lab.org/teaching-materials

https://www.youtube.com/watch?v=Vjnm-jF1PBQ

Thank you for your attention!

Overlap–layout–consensus approach



Overlap

Build overlap graph

Layout

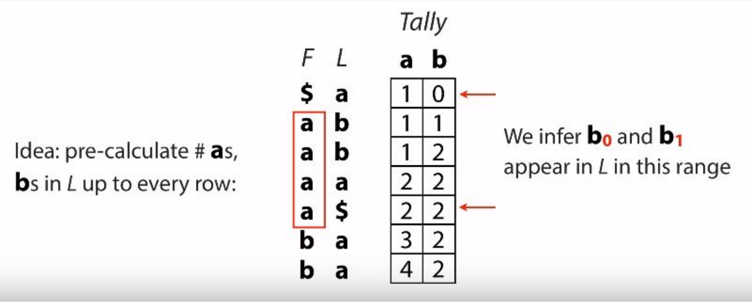
Bundle stretches of the overlap graph into contigs

Consensus

Pick most likely nucleotide sequence for each contig

How does BWT work?

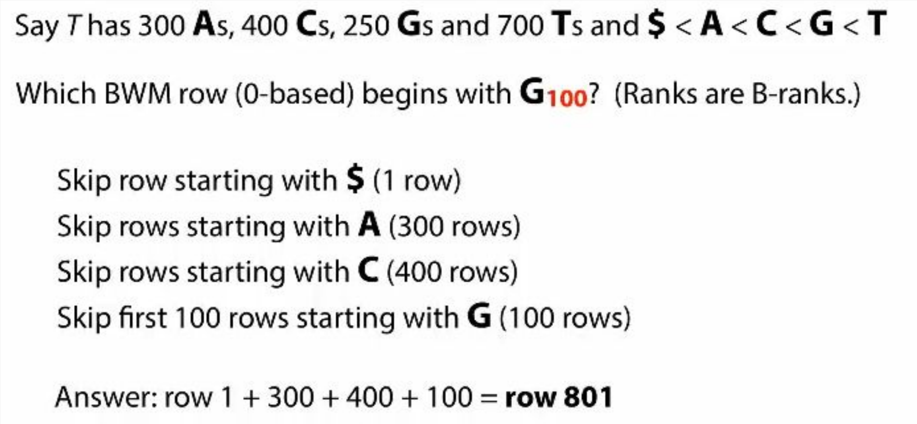
The key idea is to avoid linear searching.



Idea: pre-calculate # **a**s, **b**s in *L* up to every row:

		Tally	
F	*L*	**a**	**b**
$	**a**	1	0
a	**b**	1	1
a	**b**	1	2
a	**a**	2	2
a	**$**	2	2
b	**a**	3	2
b	**a**	4	2

We infer $\mathbf{b}_0$ and $\mathbf{b}_1$ appear in *L* in this range



Say *T* has 300 **A**s, 400 **C**s, 250 **G**s and 700 **T**s and **$** < **A** < **C** < **G** < **T**

Which BWM row (0-based) begins with $\mathbf{G}_{100}$? (Ranks are B-ranks.)

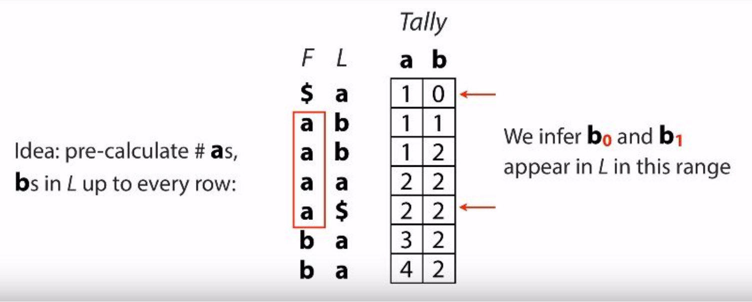
Skip row starting with **$** (1 row)
Skip rows starting with **A** (300 rows)
Skip rows starting with **C** (400 rows)
Skip first 100 rows starting with **G** (100 rows)

Answer: row 1 + 300 + 400 + 100 = **row 801**

www.langmead-lab.org/teaching-materials

How does it reduce running (mapping) time?

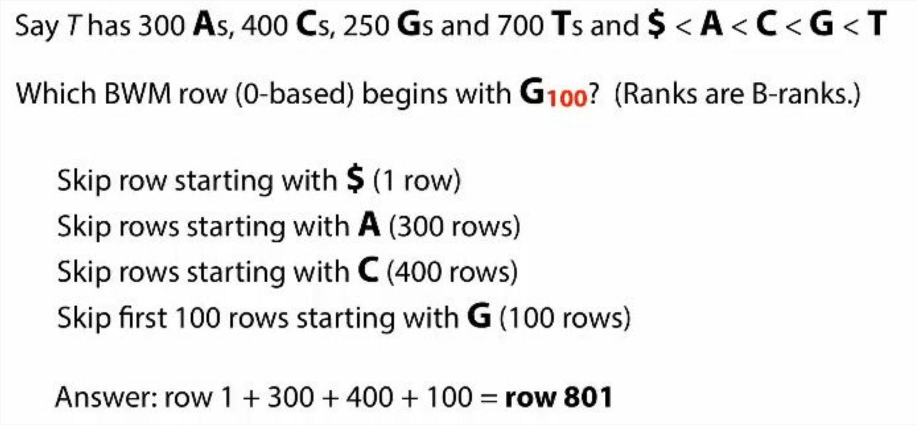
The key idea is to avoid linear searching.



Idea: pre-calculate # **a**s, **b**s in *L* up to every row:

		Tally	
F	*L*	**a**	**b**
$	**a**	1	0
a	**b**	1	1
a	**b**	1	2
a	**a**	2	2
a	**$**	2	2
b	**a**	3	2
b	**a**	4	2

We infer $\mathbf{b}_0$ and $\mathbf{b}_1$ appear in *L* in this range



Say *T* has 300 **A**s, 400 **C**s, 250 **G**s and 700 **T**s and **$** < **A** < **C** < **G** < **T**

Which BWM row (0-based) begins with $\mathbf{G}_{100}$? (Ranks are B-ranks.)

Skip row starting with **$** (1 row)

Skip rows starting with **A** (300 rows)

Skip rows starting with **C** (400 rows)

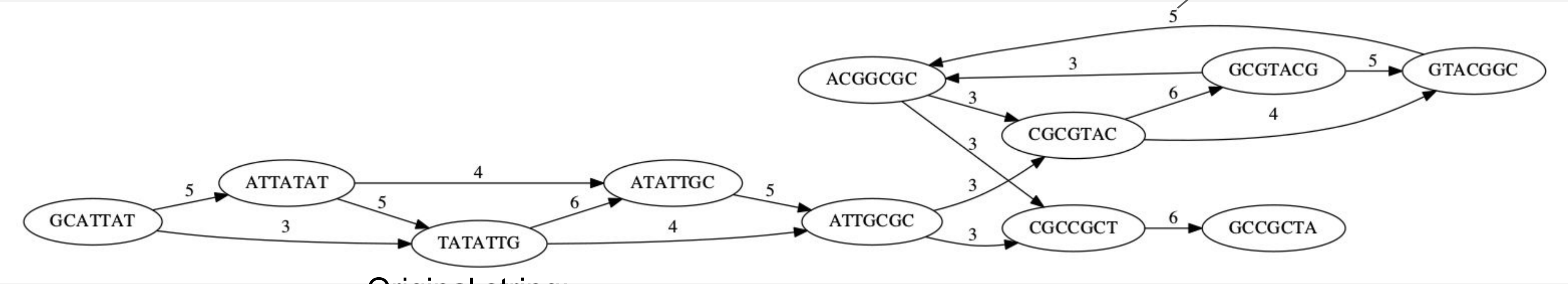
Skip first 100 rows starting with **G** (100 rows)

Answer: row 1 + 300 + 400 + 100 = **row 801**

www.langmead-lab.org/teaching-materials

Overlap graph

Edge label is overlap length



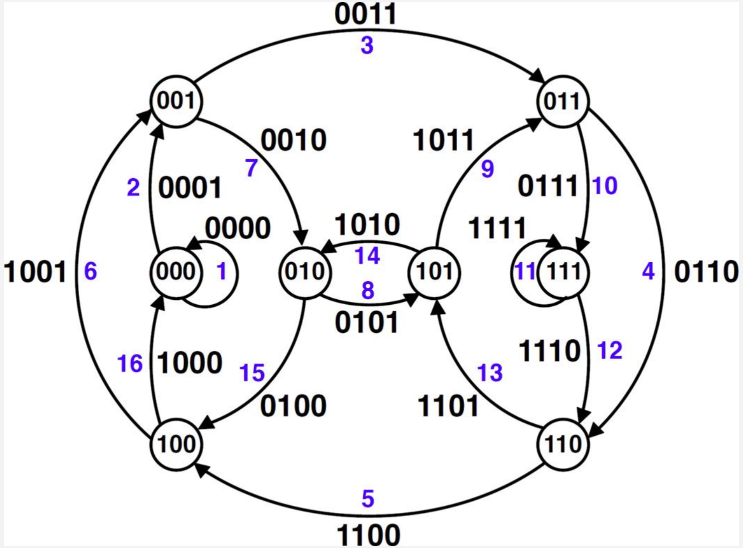
Original string:
GCATTATATATTGCGCGTACGGCGCCGCTACA

Shortest common superstring (visit every node once, minimize cost) = Traveling Salesman Problem - NP-hard

Hamiltonian Cycle (visit every node once) - NP-compete

Greedy algorithms can help (but no guarantee of optimal solution)

Genome assembly, De Bruijn graphs



Superstring: **0000110010111101**

Hierholzer's algorithm:

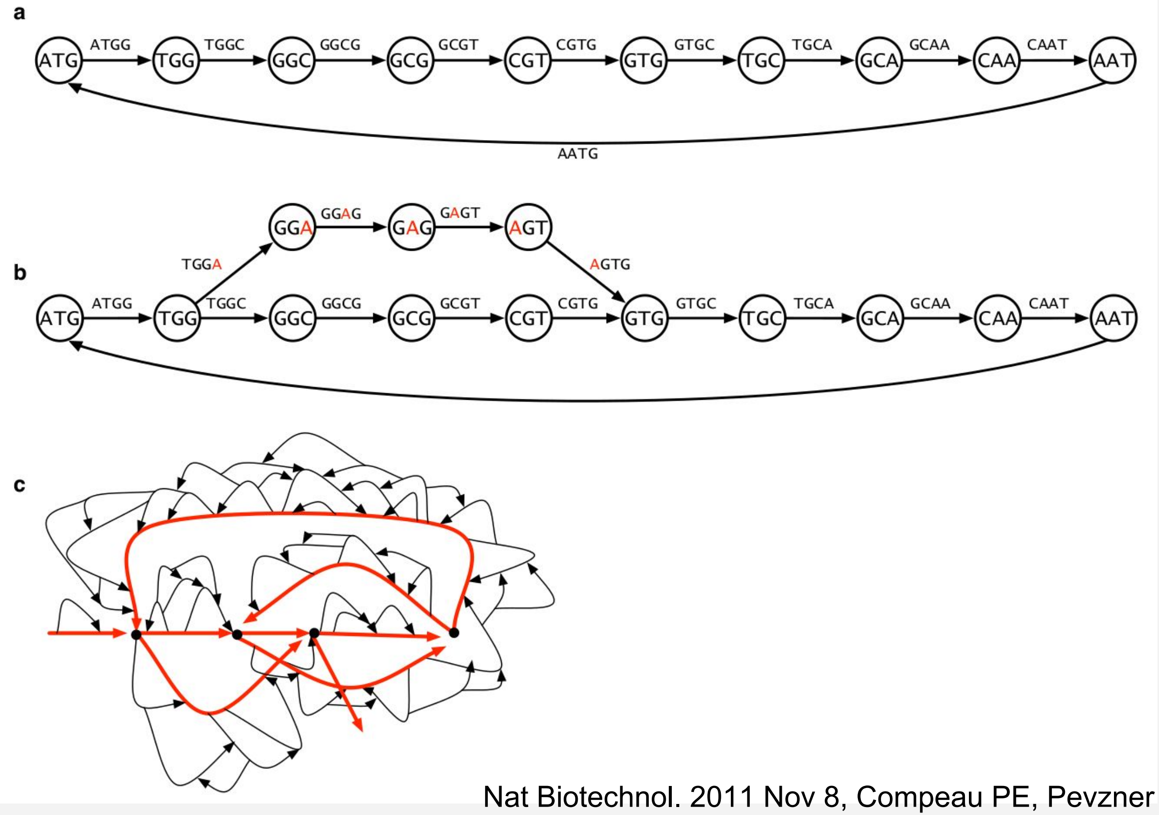
- Choose any starting vertex *v*, follow a trail of edges from that vertex until returning to *v* (the tour may not cover all the vertices and edges of the initial graph).
- As long as there exists a vertex *u* that belongs to the current tour but that has adjacent edges not part of the tour, start another trail from *u*, following unused edges until returning to *u*, and join the tour formed in this way to the previous tour.

Complexity of the algorithm: O(|E|)

Nat Biotechnol. 2011 Nov 8, Compeau PE, Pevzner PA, Tesler G.

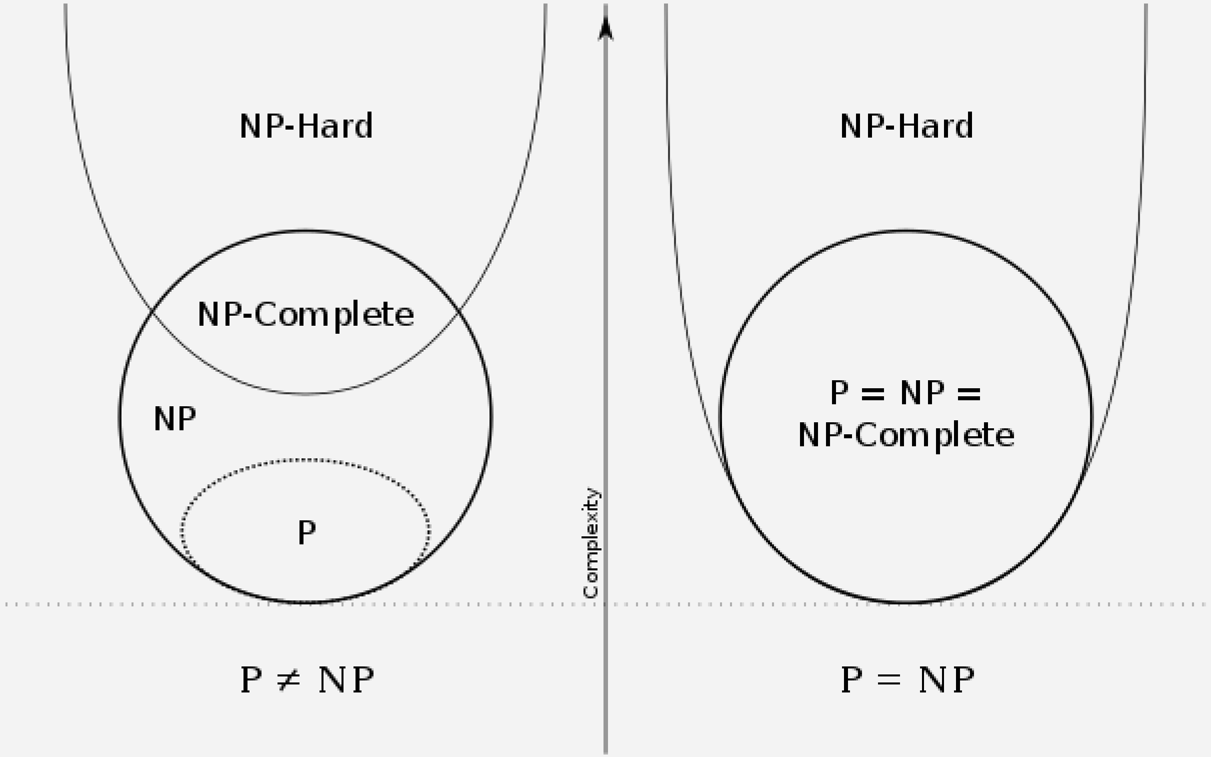
https://en.wikipedia.org/wiki/Eulerian_path

De Bruijn graph from reads with sequencing errors



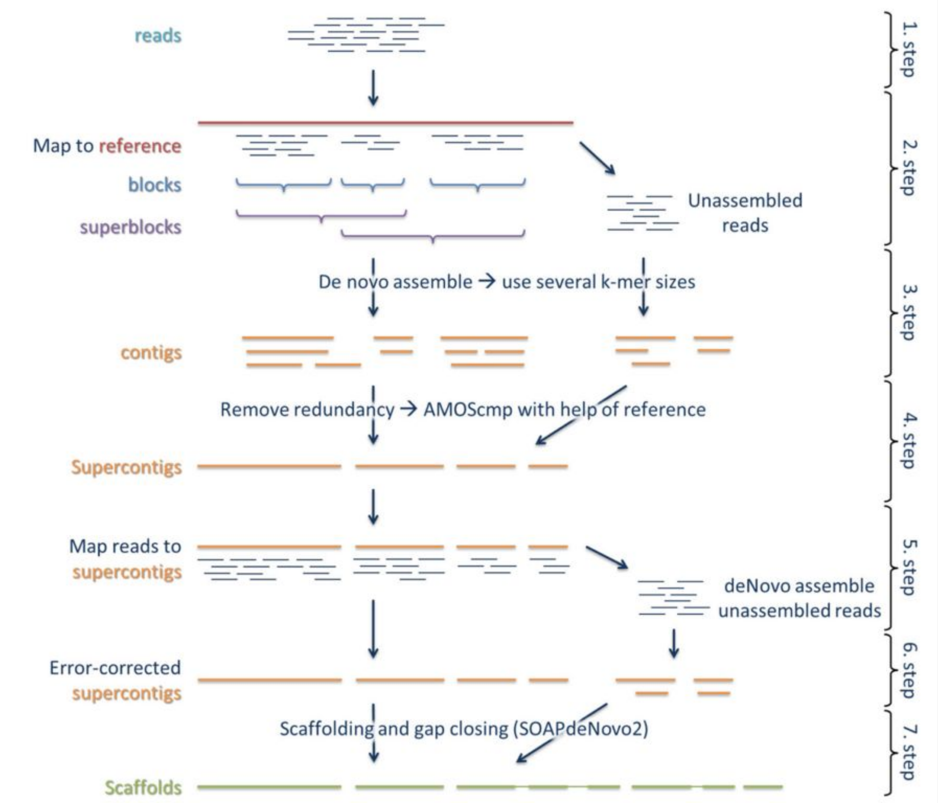
Nat Biotechnol. 2011 Nov 8, Compeau PE, Pevzner PA, Tesler G.

NP-hardness



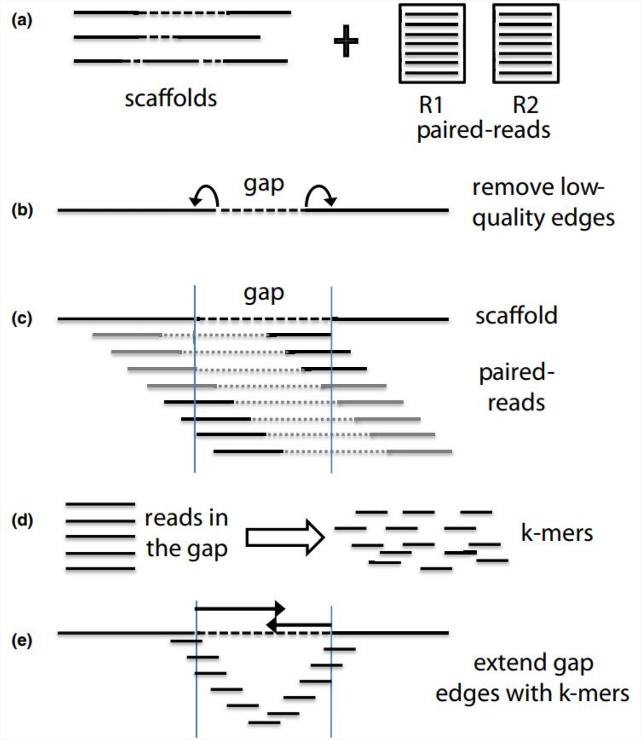
https://en.wikipedia.org/wiki/NP_(complexity)

Reference-guided de novo assembly approach


reads
1. step
Map to reference
blocks
superblocks
Unassembled reads
2. step
De novo assemble → use several k-mer sizes
contigs
3. step
Remove redundancy → AMOScmp with help of reference
Supercontigs
4. step
Map reads to supercontigs
deNovo assemble unassembled reads
5. step
Error-corrected supercontigs
6. step
Scaffolding and gap closing (SOAPdeNovo2)
Scaffolds
7. step

BMC Bioinformatic 2017, Heidi EL Lischer, Kentaro K Shimizu

Gap filling



Genome Biology 2012, Marten Boetzer, Walter Pirovano

General workflow

- Get data
- QC FastQC
    - Trimming with Trimmomatic (Read to the manual)
        - Library bias (forced trimming) choose 15-20bp
        - Quality trimming get rid of the data with Quality score lower than 20
        - Adapter trimming
- Assembly
    - De novo
        - Overlap consensus Graph
        - De Bruijn graphs
            - SPades
            - Skesa
    - Reference
        - Map Reads BWA The output is an alignment file
        - Alignment File SamTools Process the file and output is a consensus file
        - Consensus File SamTools Process to get a FASTA file using Sam tools and SeqTk
        - FASTA File SeqTK
- QC
    - QUAST
    - BUSCO