

# Genome assemblies

Isheng Jason Tsai

Introduction to NGS Data and Analysis  
Lecture 3



# References

Most of the slides are inspired from:

<http://www.slideshare.net/agbiotec/overview-of-genome-assembly-algorithms>

Michael Schatz

<http://schatzlab.cshl.edu/teaching/>

Ben Langmead

<http://www.langmead-lab.org/teaching-materials/>

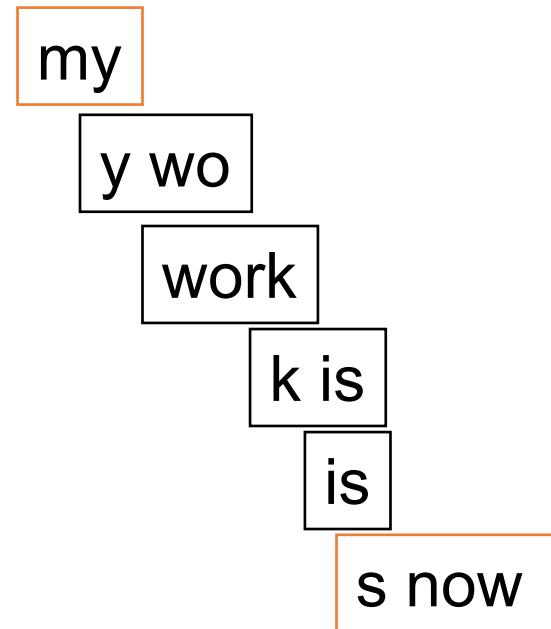
# Problem

**We accidentally printed my five copies of “Origin of Species”, and shredded into pieces**



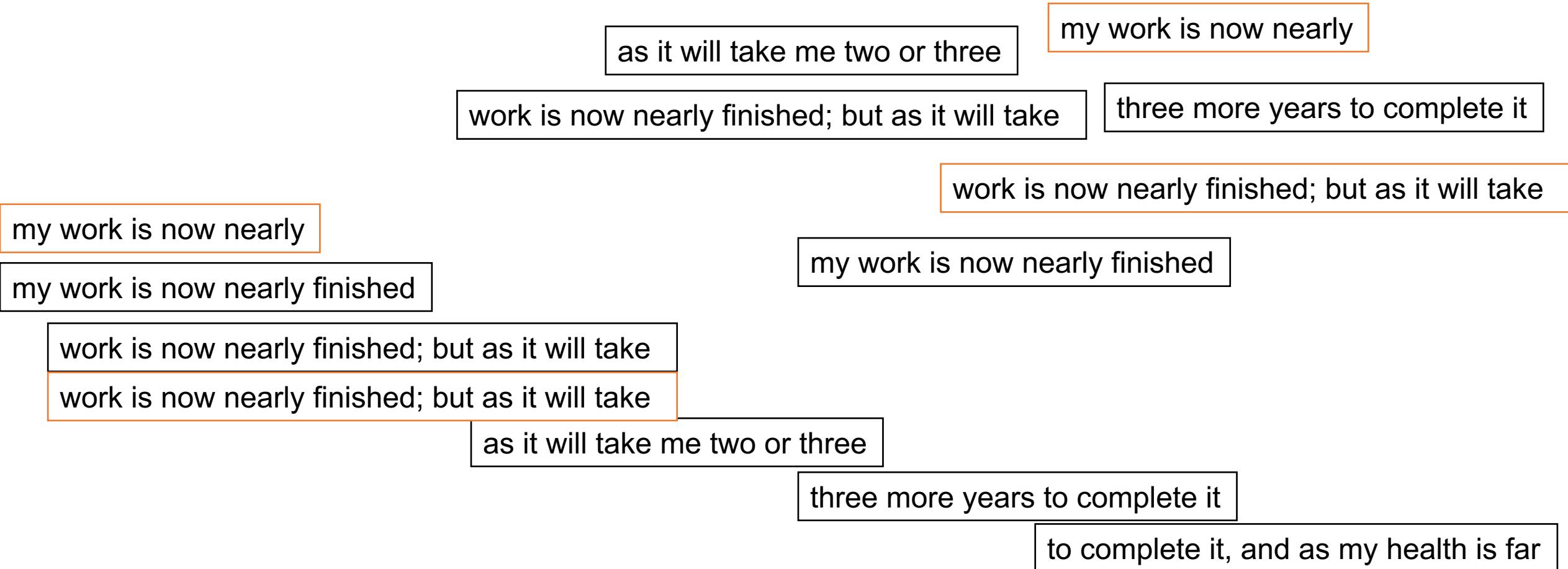
my work is now nearly finished; but as it will take me two or three more years to complete it, and as my health is far from strong, I have been urged to publish this Abstract. I have more especially been induced to do this, as Mr Wallace, who is now studying the natural history of the Malay archipelago, has arrived at almost exactly the same general conclusions that I have on the origin of species. Last year he sent to me a memoir on this subject, with a request that I would forward it to Sir Charles Lyell, who sent it to the Linnean Society, and it is published in the third volume of the journal of that Society. Sir C. Lyell and Dr Hooker, who both knew of my work -- the latter having read my sketch of 1844 -- honoured me by thinking it advisable to publish, with Mr Wallace's excellent memoir, some brief extracts from my manuscripts.

# Assemby = Piecing the pieces together



my work is now

# Long shredded pieces (read) = easier assembly



my work is now nearly finished; but as it will take me two or three more years to complete it, and as my health is far

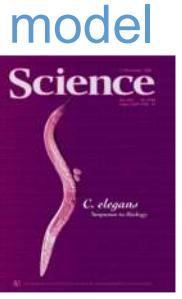
# Why sequence a genome?

# Genomics advance our understanding of organisms across tree of life

pathogens



TB



C. elegans



fruitfly



human

model  
NGS



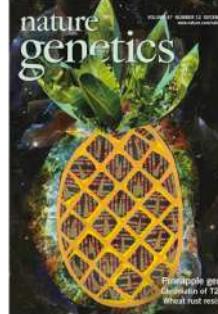
pathogens



Blood fluke



Tapeworms



Tapeworms



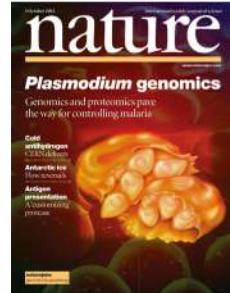
Avian genomics

1996



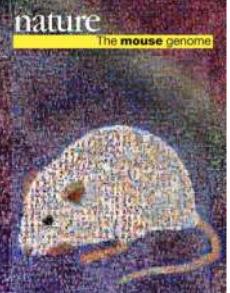
S. cerevisiae  
model

1997



Plasmodium  
pathogens

1998



model

1999



model

2000



Phytophthora infestans  
(potato blight)

2001



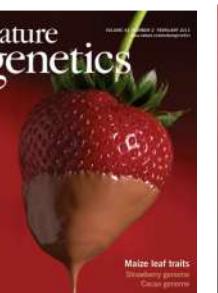
Black death  
(potato blight)

2002



model

2005



model

2006



pathogens

2007



pathogens

2008



2009



pathogens

2010



pathogens

2011



pathogens

2012



pathogens

2013



pathogens

2014

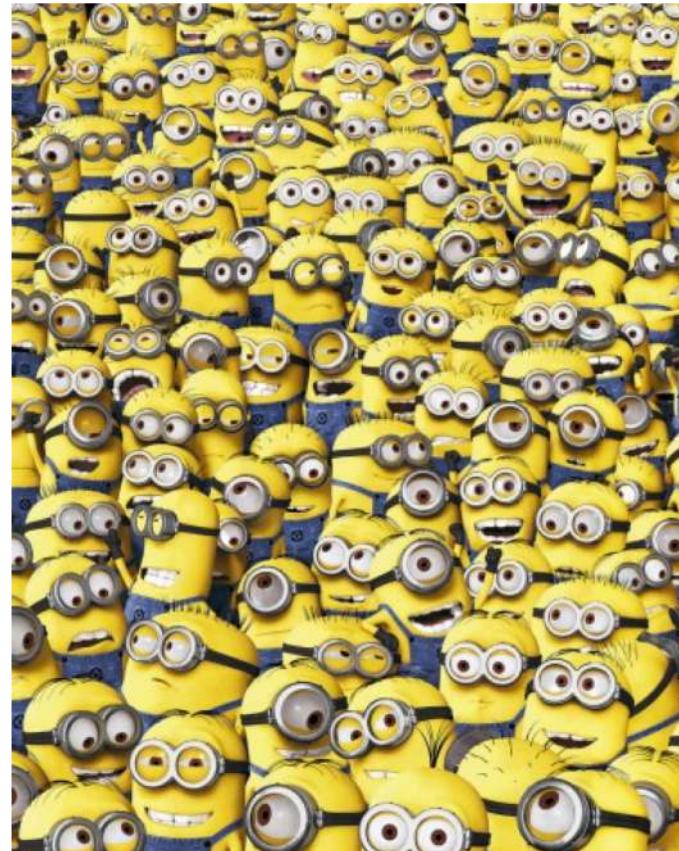


pathogens

# Assemble a genome is hard



Less complicated

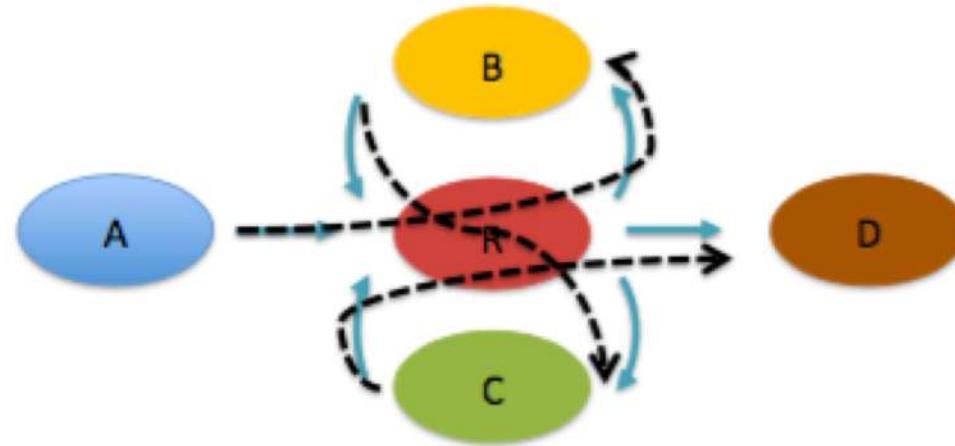
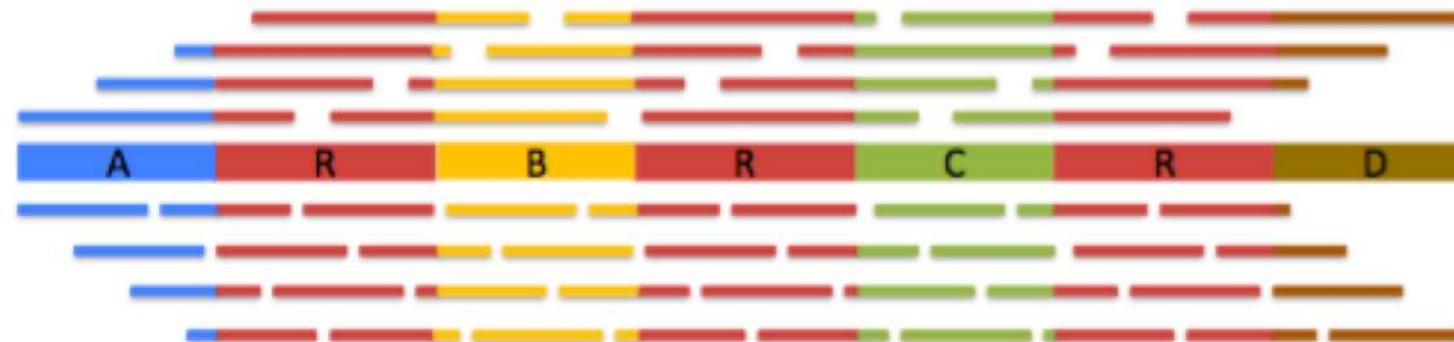


Complicated



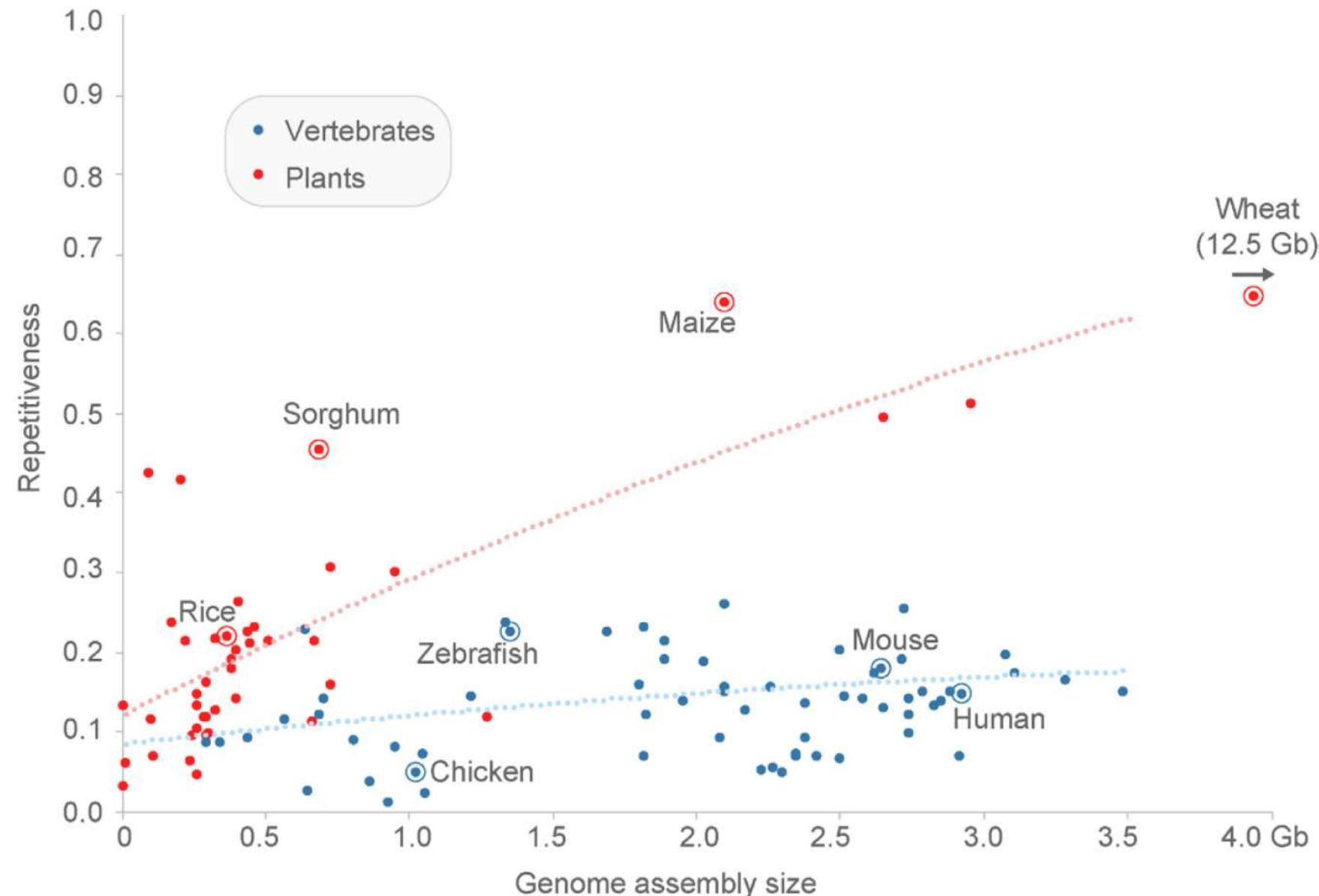
# Repeats

You can't 100% resolve repeats unless you have your sequence > repeat



Credit: Michael Schatz

# Repeat content of plant and vertebrate genomes



# QC and understand your data before assembly

Conversation:

Jason, we don't know why our assembly is not as good as yours. We have about 90X of Mate pair data..

**Email 2 weeks later:**

Dear Jason,

I run the Trimmomatic analysis for the raw data of mate-pair libraries (10kL2\_1, 10kL7\_1, 15kL3\_1 and 15kL8\_4 as examples) with a custom adapter file containing mate-pair adapter sequences (junction and external adaptors, you may find them in the attached technote pdf file) and found **that over 80% reads of the library were dropped out.**

# Always be careful of contamination

## Evidence for extensive horizontal gene transfer from the draft genome of a tardigrade

Thomas C. Boothby<sup>a,1</sup>, Jennifer R. Tenlen<sup>a,2</sup>, Frank W. Smith<sup>a</sup>, Jeremy R. Wang<sup>a,b</sup>, Kiera A. Patanella<sup>a</sup>, Erin Osborne Nishimura<sup>a</sup>, Sophia C. Tintori<sup>a</sup>, Qing Li<sup>c</sup>, Corbin D. Jones<sup>a</sup>, Mark Yandell<sup>c</sup>, David N. Messina<sup>d</sup>, Jarret Glasscock<sup>d</sup>, and Bob Goldstein<sup>a</sup>

17.5 % HGT

<sup>a</sup>Department of Biology, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599; <sup>b</sup>Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599; <sup>c</sup>Eccles Institute of Human Genetics, University of Utah, Salt Lake City, UT 84112; and <sup>d</sup>Cofactor Genomics, St. Louis, MO 63110

Edited by W. Ford Doolittle, Dalhousie University, Halifax, Canada, and approved September 28, 2015 (received for review May 28, 2015)

## No evidence for extensive horizontal gene transfer in the genome of the tardigrade *Hypsibius dujardini*

Georgios Koutsovoulos<sup>a</sup>, Sujai Kumar<sup>a</sup>, Dominik R. Laetsch<sup>a,b</sup>, Lewis Stevens<sup>a</sup>, Jennifer Daub<sup>a</sup>, Claire Conlon<sup>a</sup>, Habib Maroon<sup>a</sup>, Fran Thomas<sup>a</sup>, Aziz A. Aboobaker<sup>c</sup>, and Mark Blaxter<sup>a,1</sup>

0.4 % HGT

<sup>a</sup>Institute of Evolutionary Biology, University of Edinburgh, Edinburgh EH9 3FL, United Kingdom; <sup>b</sup>The James Hutton Institute, Dundee DD2 5DA, United Kingdom; and <sup>c</sup>Department of Zoology, University of Oxford, Oxford OX1 3PS, United Kingdom

Edited by W. Ford Doolittle, Dalhousie University, Halifax, Canada, and approved March 1, 2016 (received for review January 8, 2016)

# Always be careful of contamination

ARTICLE

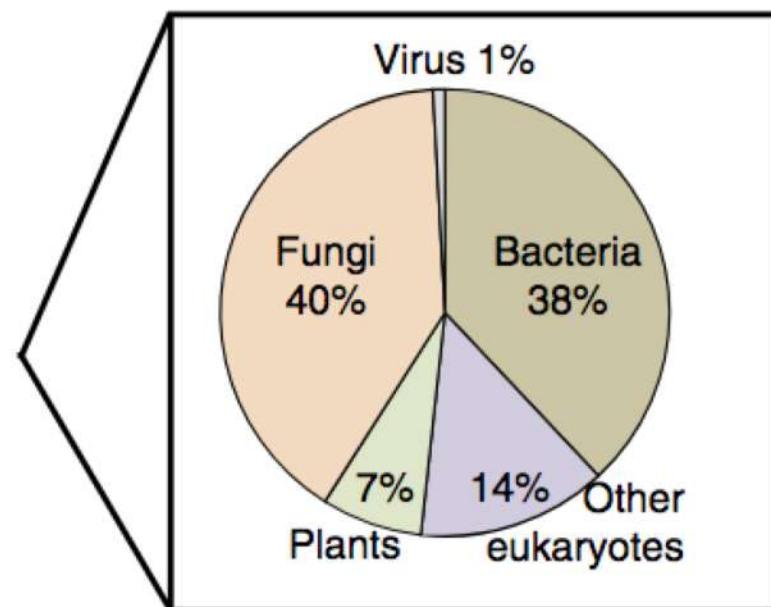
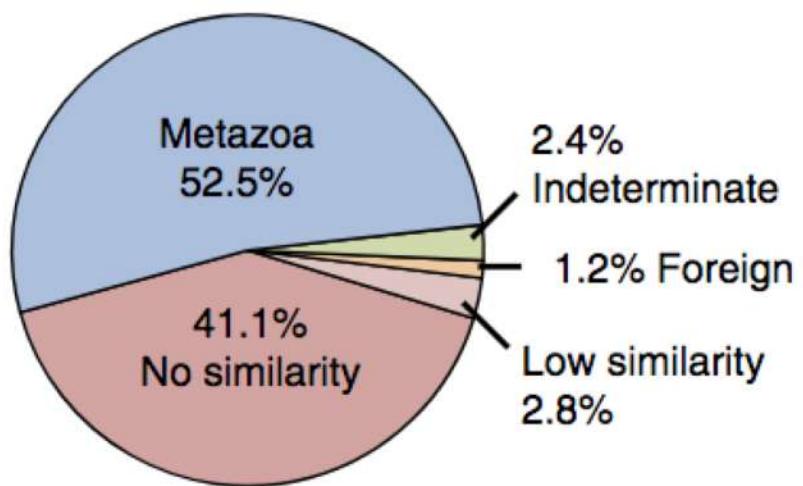
Received 21 Jun 2015 | Accepted 3 Aug 2016 | Published 20 Sep 2016

DOI: 10.1038/ncomms12808

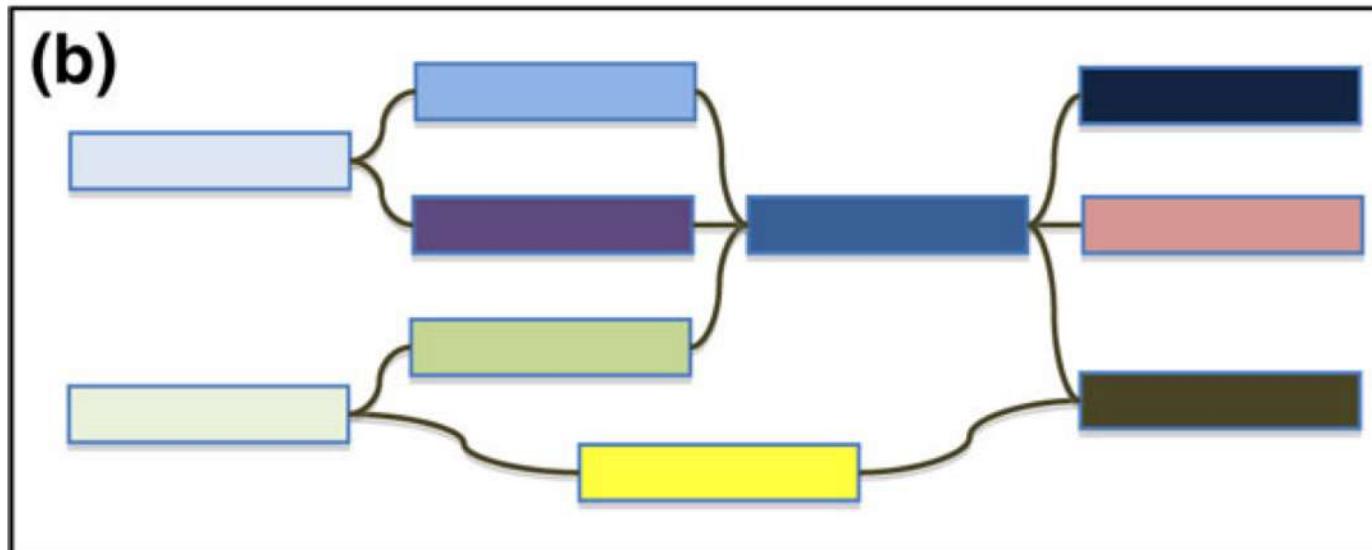
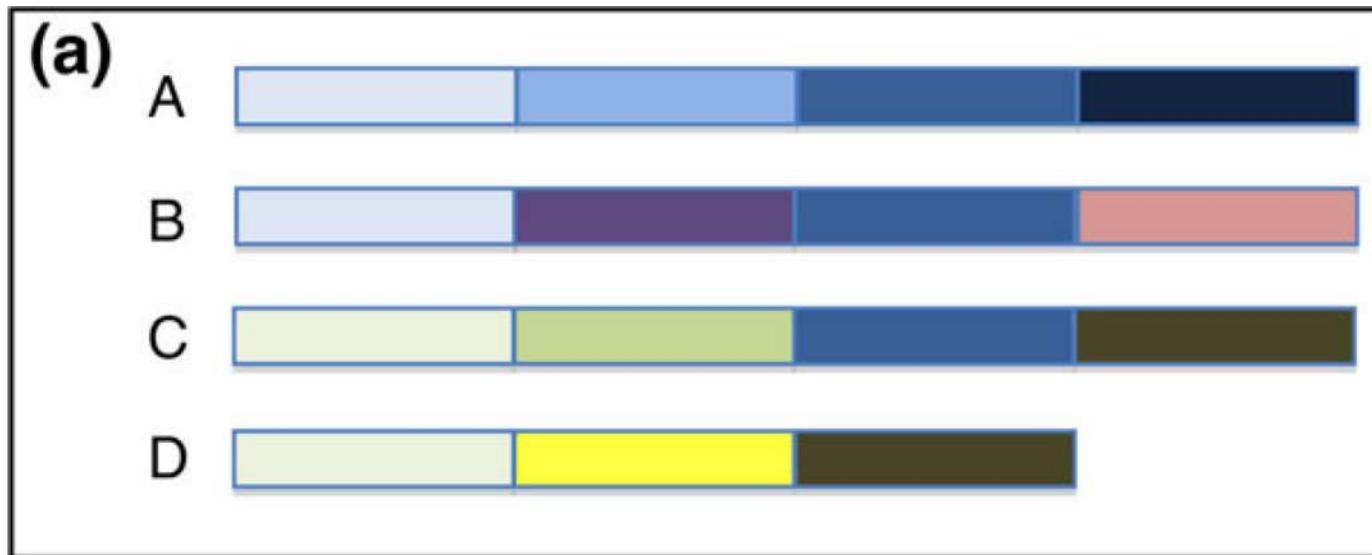
OPEN

## Extremotolerant tardigrade genome and improved radiotolerance of human cultured cells by tardigrade-unique protein

c



# Ploidy, heterozygosity and the assembly graph



# Assembly process

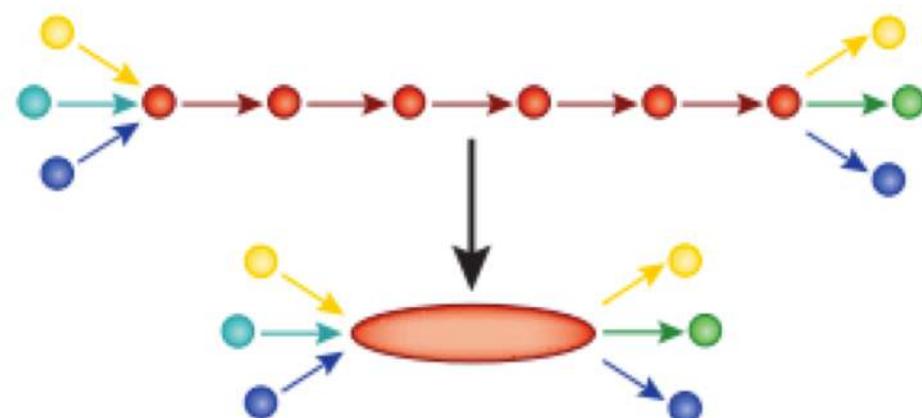
1. Fragment DNA and sequence



2. Find overlaps between reads

...AGCCTAGACCTACA**GGATGCGCGACACGT**  
**GGATGCGCGACACGT**CGCATATCCGGT...

3. Assemble overlaps into contigs



4. Assemble contigs into scaffolds



# Which program to choose?

EULER-SR	WhatsHap	GARM	SOAPdenovo		SSPACE	HaploMerger		mip
A5	Telescooper	Contrail	fermi	GABenchToB	QSRA	Opera	RAMPART	
SSAKE	SWAP-Assembler	Newbler	AutoAssemblyD	Dazzler	PCAP	Platanus	Arapan	
TIGR	Mapsembler 2	ALLPATHS-LG	HapCompass	VICUNA	Edena	CLC	Forge	SHEAR
gapfiller	CloudBrush	Cortex	REAPR	Ray	Tedna	TIGRA	PERGA	KmerGenie
Arachne	MIRA	dipSPAdes	MetAMOS	Geneious	SeqMan	Amos	Nesoni	ATAC
SCARPA	Celera	VCAKE	GAM	PASHA	NGen	bambus2	MetaVelvet-SL	Quast
GRIT	IDBA	Trinity	MHAP	Hapsembler	GAML	Sequencher	BESST	GGAKE
Phrap	MaSuRCA	HiPGA	PADENA	SeqPrep	Phusion	SGA	PE-Assembler	
CGAL	Curtain	SWiPS	KGBAssembler	Metassembler				
Pipeline	Pilot	SHRAP	Taipan	SILP3	IDBA-MTP	HGAP	PRICE	Pilon
OMACC	Anchor	Omega	SUTTA	ABySS	HyDA-Vista	SR-ASM	Velvet	MSR-CA
SOPRA	iMetAMOS	DNAexus		Ragout	SPAdes	Atlas		
DNA Dragon	CABOG	SAGE		Cerulean	Monument	SAT-Assembler	FRCBam	
FALCON	SuccinctAssembly	SHORTY		SHARCGS	GAGM	ngsShoRT	ABBA	
	GigAssembler			PBJelly	DecGPU	Khmer	GenoMiner	ELOPER

# Contiguity (good) is a genome? N50

Definition: 50% of genome in contigs/scaffolds of length **N50 bp** or greater

## Example: 1 Mbp genome

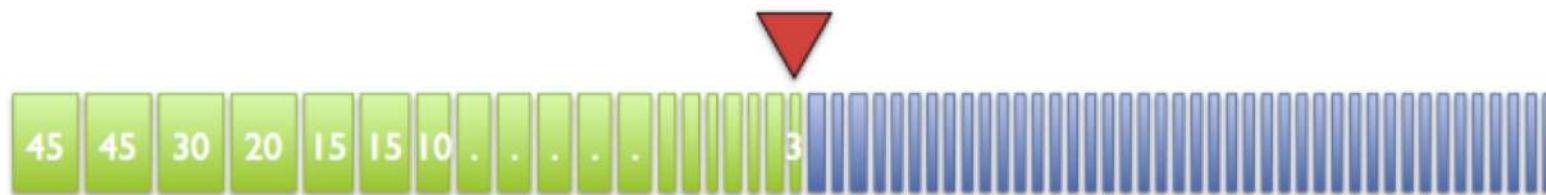


50%

1000

N50 size = 30 kbp

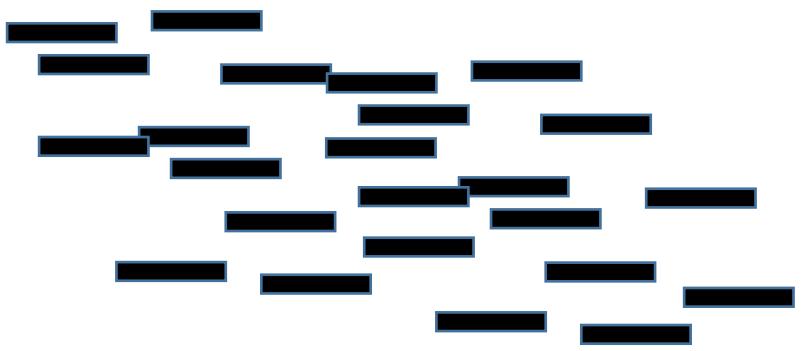
sorted



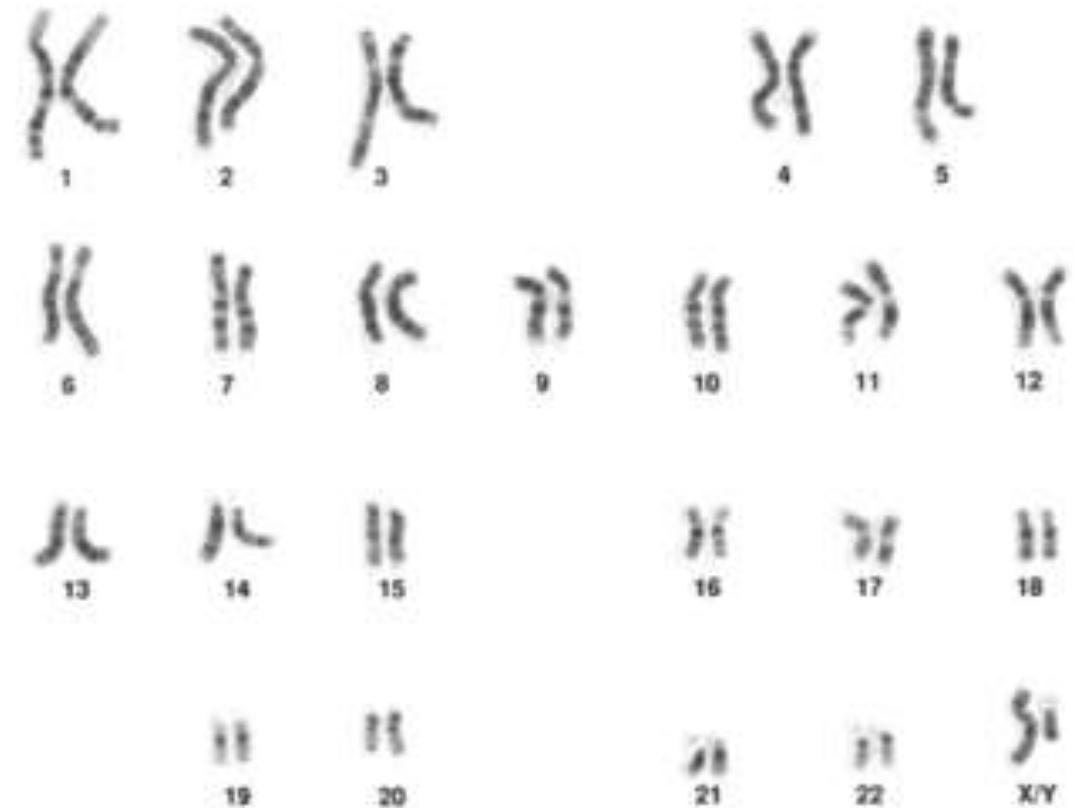
N50 size = 3 kbp

Credit: Michael Schatz

# Most assemblies are fragmented



!=

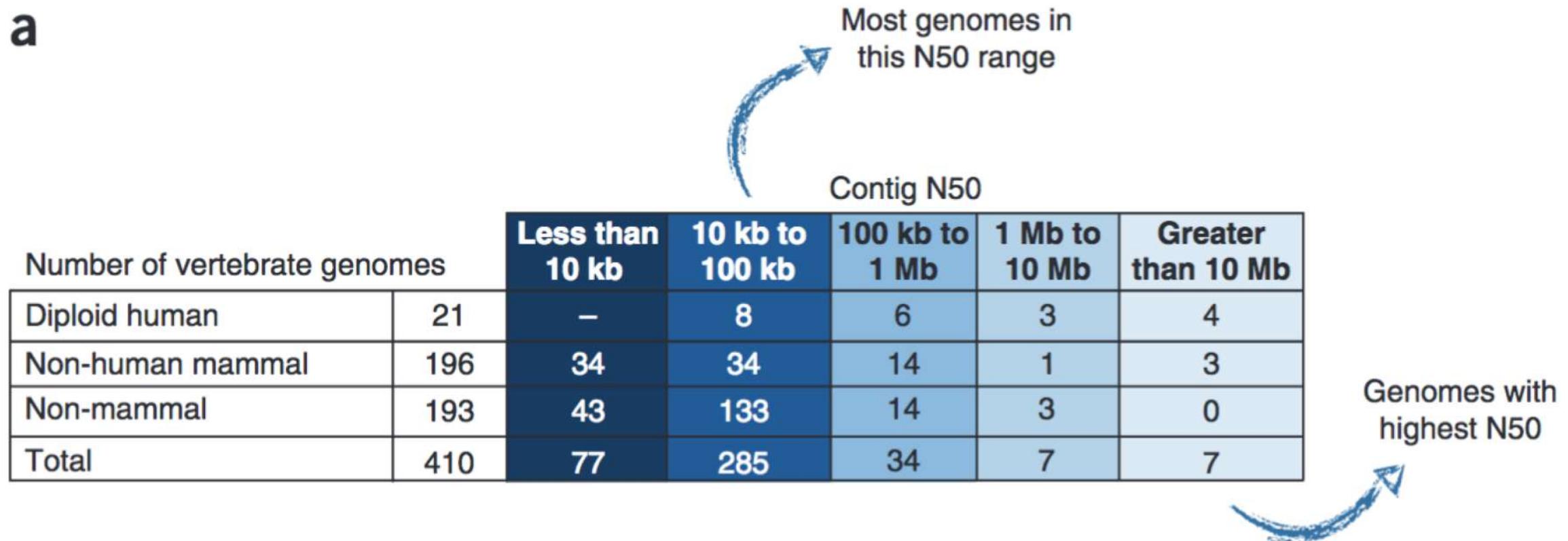


Contigs

chromosomes

# Statistics for current GenBank assemblies.

a



# Why do we need a good assembly?

It is easier to analyse

10000 pieces vs. 23 pieces (chromosomes)

Allows more accurate representation of genes locations on genome

Is gene A close to gene B? On the same chromosome?

Transposon dynamics

Missing in most assemblies (located in NNNNNNNNNN gaps)

Responsibility to contribute to your community

Do you want others to work on the same genome / species as well?

**Bottom line:**

It's really no point to do one if you can't produce an accurate and useful one

# Assembly qualities

	<b>Whole Genome Representation</b>	<b>Sequence Status</b>	<b>Genes</b>	<b>Usability</b>
<b>1</b>	Incomplete for non-repetitive regions	Small scaffolds and contigs	Incomplete genes	Markers development
<b>2</b>	Complete for non-repetitive regions	Medium scaffolds and contigs	Complete but 1-2 genes/contig	Gene mining
<b>3</b>	Complete for non-repetitive regions	Large scaffolds and contigs	Several dozens of genes/contig	Microsynteny
<b>4</b>	Complete for almost the whole genome	Pseudomolecules	Hundreds of genes/contig	Any (Synteny, Candidate gene by QTLs)
<b>5</b>	Complete genome	Pseudomolecules	Thousands of genes/contig	

Credit: Aureliano Bombarely

# Approaches

# Approaches

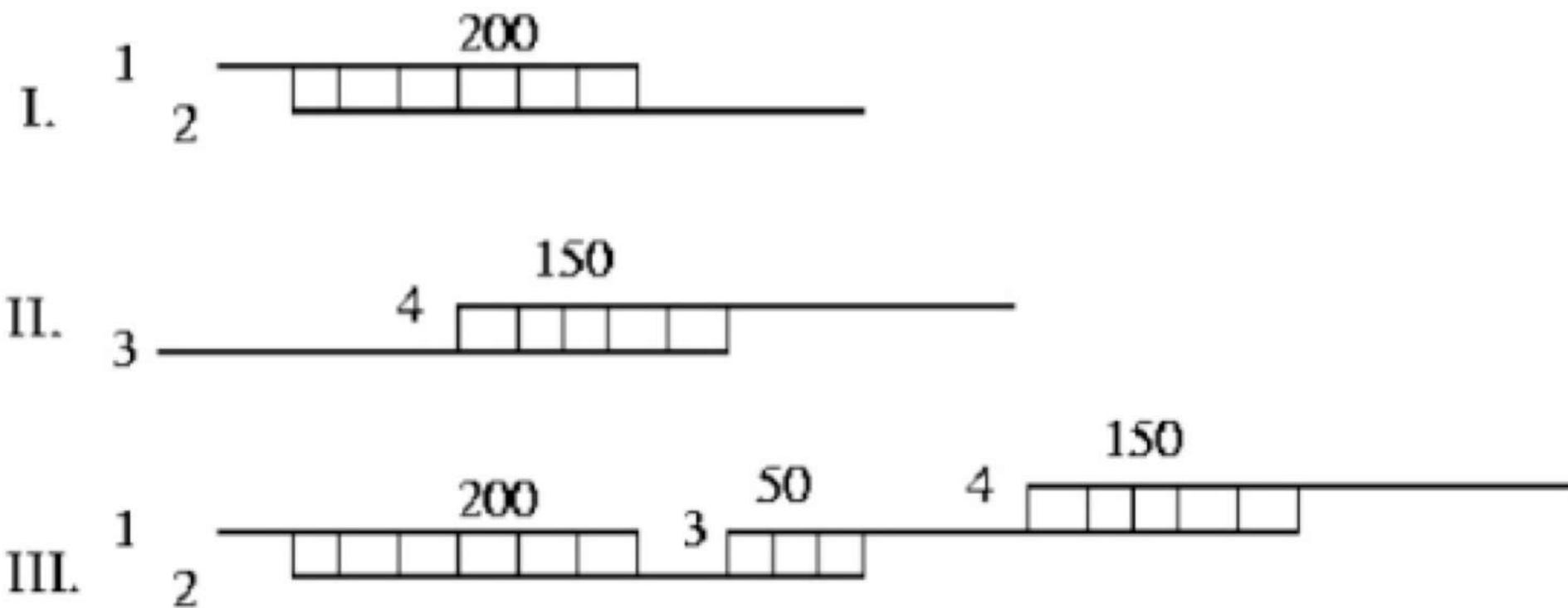
- Greedy extension
  - only mentioned for historical reasons
- Overlap – Layout – Consensus (**OLC**) assembly: 'traditional' and well established method, but challenging to implement at each stage
  - Most "old" and "newest" assemblies were produced using this approach
- de Bruijn graph (**DBG**) assembly

# Approaches

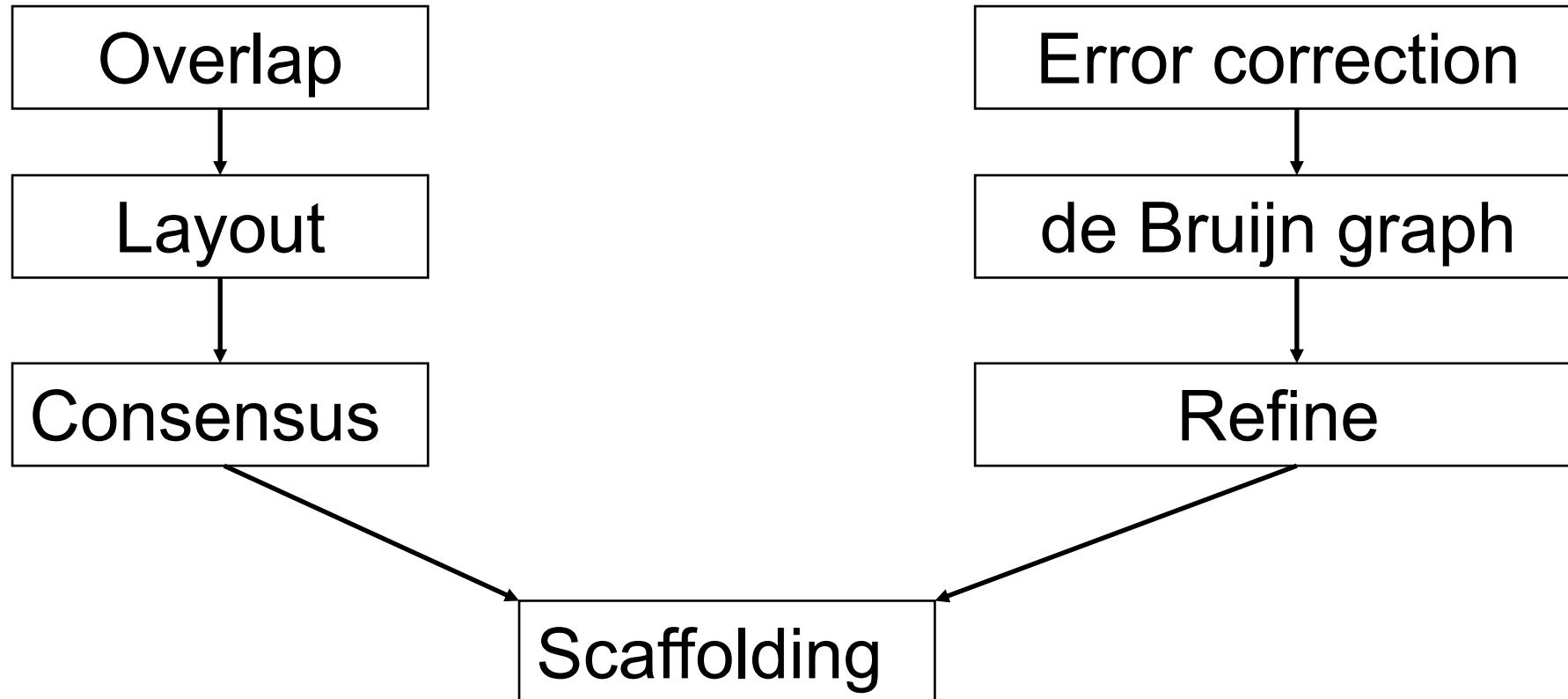
- Greedy extension
  - only mentioned for historical reasons
- Overlap – Layout – Consensus (**OLC**) assembly: 'traditional' and well established method, but challenging to implement at each stage
  - Most "old" and "newest" assemblies were produced using this approach
- de Bruijn graph (**DBG**) assembly

# Greedy extension

- Oldest and not really useful in most cases



# OLC and DBG assemblers



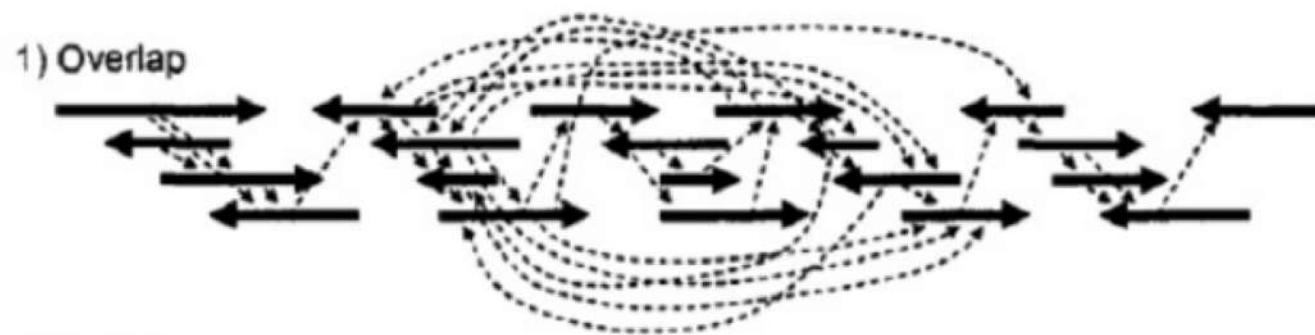
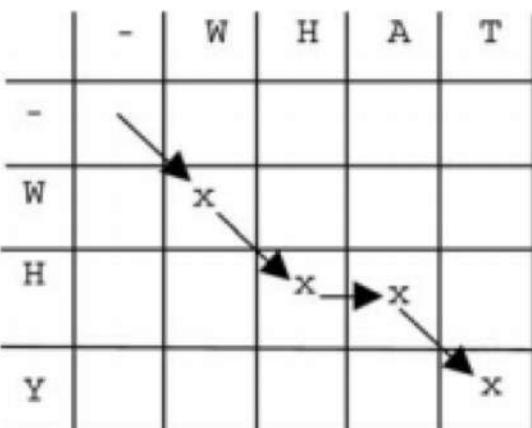
# OLC approach

- Pairwise alignment of all reads to find **overlaps**
- **Layout** the reads to decide which read align to which
- Get **consensus** by joining join the read sequences, merging overlaps
- All three are challenging



# Overlap

- All vs. all pairwise alignment
  - Smith-Waterman? Blast? Kmer based? Suffix tree? Dynamic programming?
- Computationally very intensive but can be parallelised
  - Need lots of CPUs!
  - Batch1 align Batch 1 in 1<sup>st</sup> CPU
  - Batch1 align Batch 2 in 2<sup>nd</sup> CPU



## Computational Example 8.1: Pseudocode for overlap alignment

Input sequences A, B

Set  $O_{i,0} = O_{0,j} = 0$  for all  $i, j$

for  $i = 1$  to  $n$

    for  $j = 1$  to  $m$

$O_{i,j} = \max\{O_{i-1,j} - \delta, O_{i-1,j-1} + s(a_i, b_j), O_{i,j-1} - \delta\}$

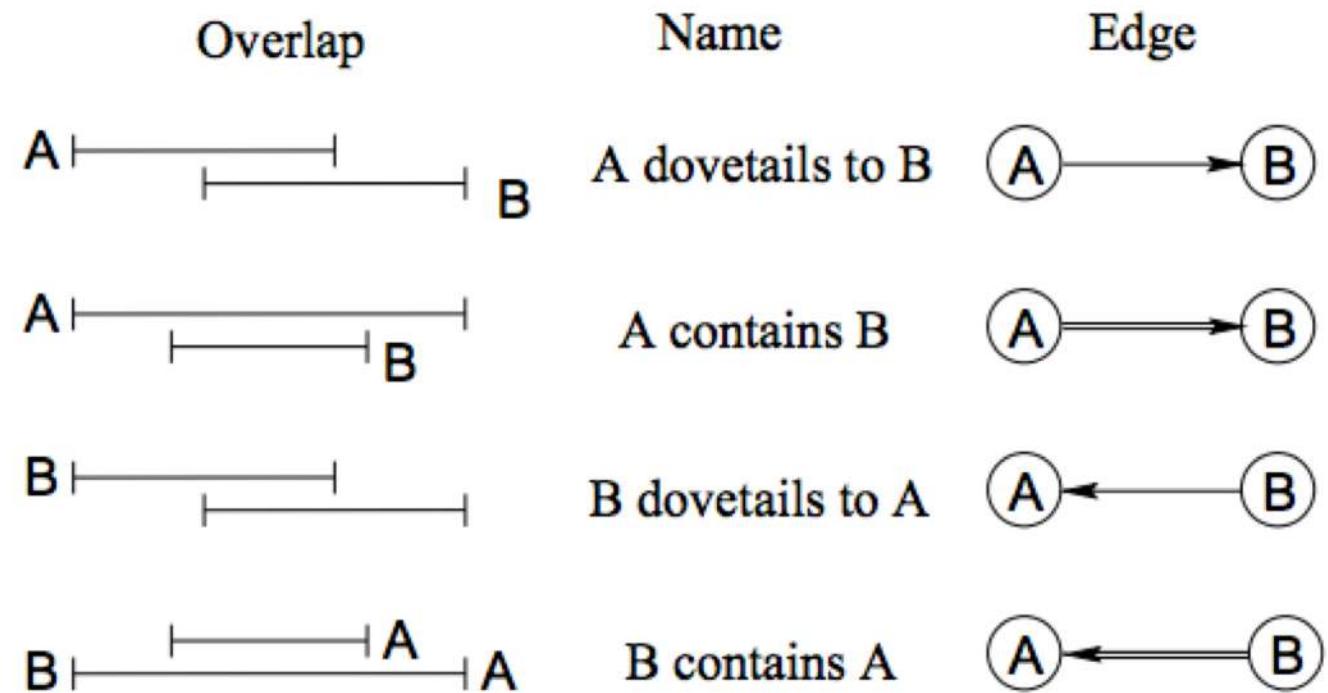
    end

end

Best overlap =  $\max\{O_{i,m}, O_{n,j}; 1 \leq i \leq n, 1 \leq j \leq m\}$

# Build overlap graph

- It's common practice to represent them in **graphs**
- The actual overlaps are the edges
- Now we create the genome assembly graph



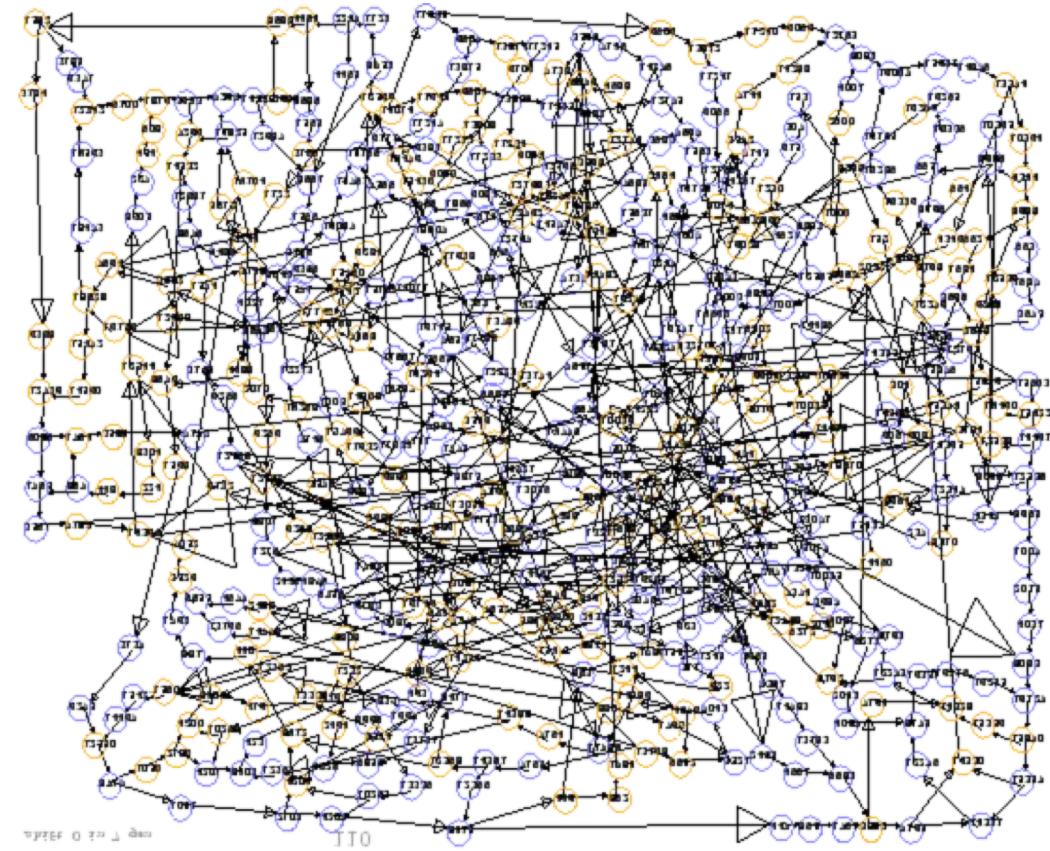
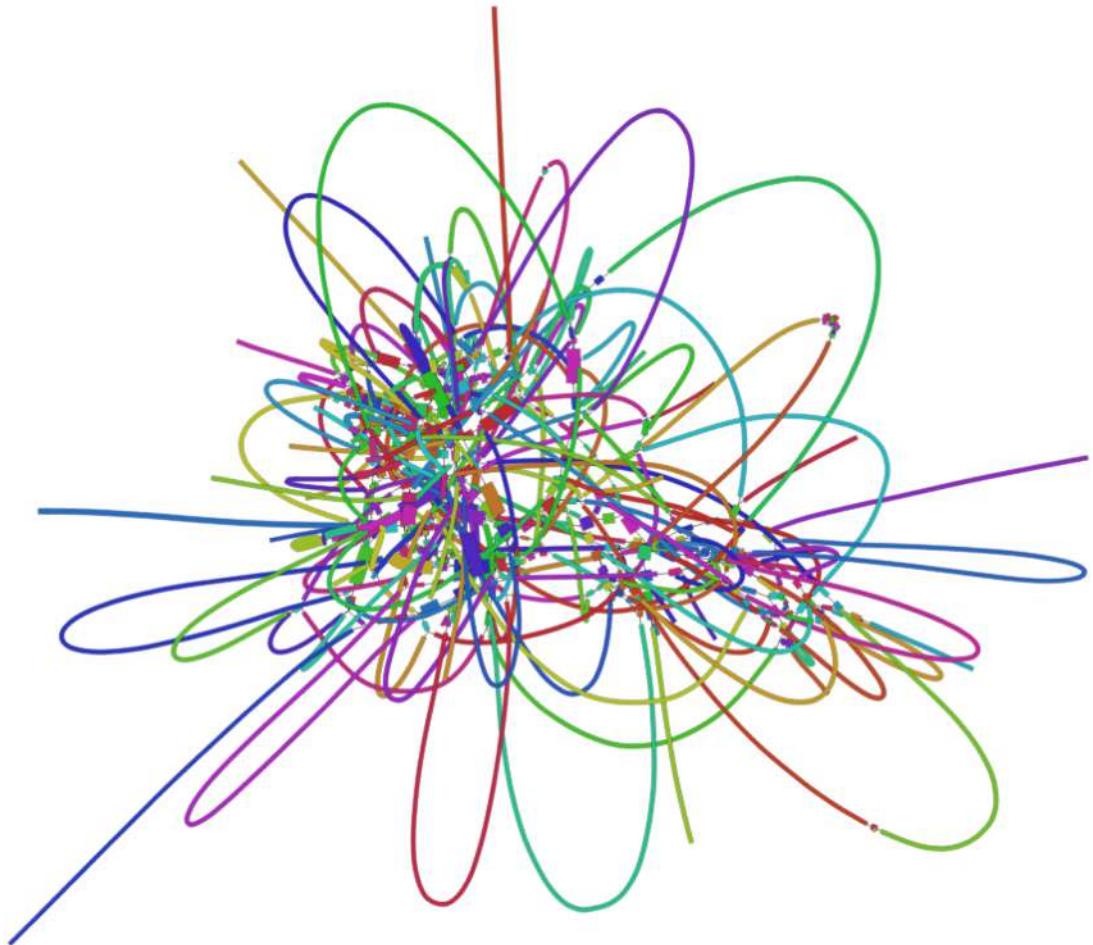
my work is now nearly finished

work is now nearly finished; but as it will take

my work is now nearly finished

work is now nearly finished; but as it will take

# Some assembly graph can be complicated...



<http://rrwick.github.io/Bandage/images/screenshots/screenshot02.png>

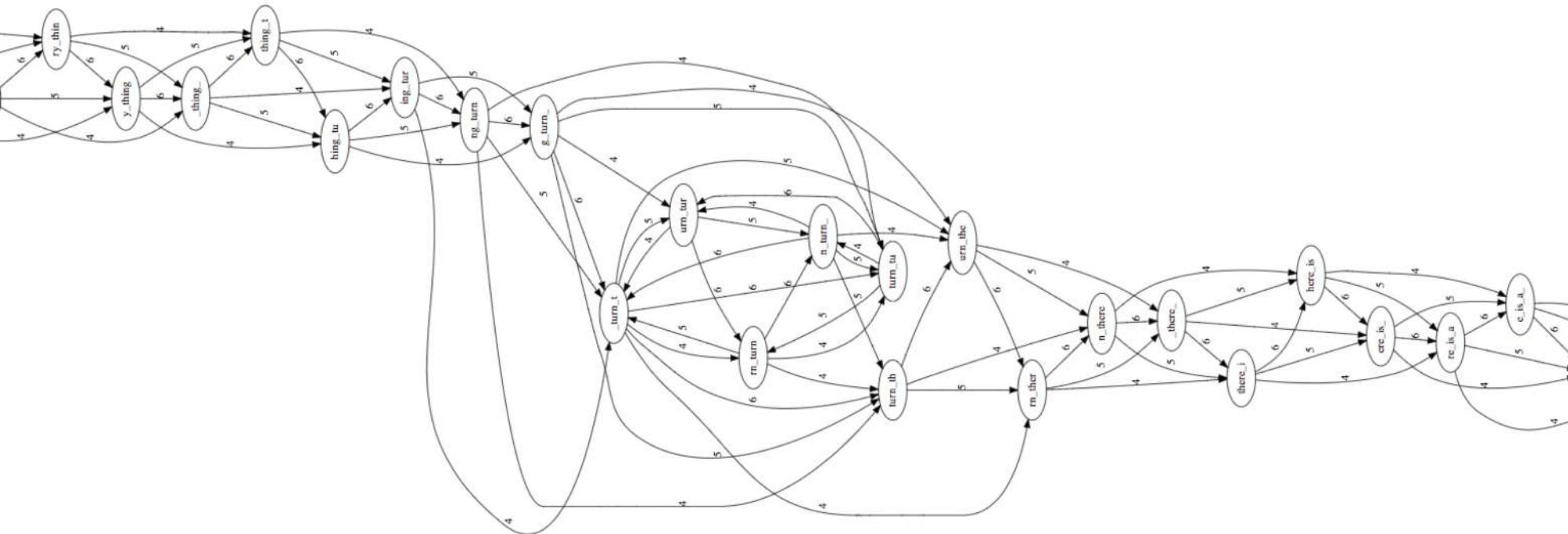
# Layout

Overlap graph is big and messy. Contigs don't "pop out" at us.

Below: part of the overlap graph for

[to\\_every\\_thing\\_turn\\_turn\\_turn\\_there\\_is\\_a\\_season](#)

$l = 4, k = 7$



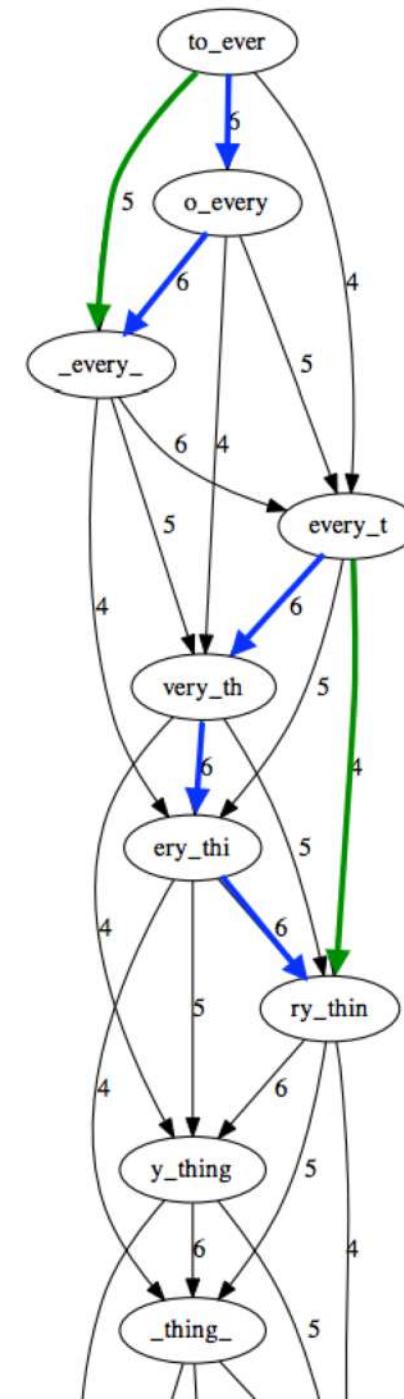
# Layout

Order the reads into a consistent manner

Anything redundant about this part of overlap graph?

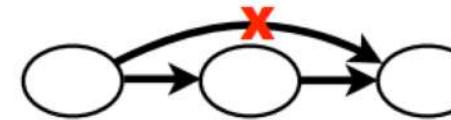
Some edges can be inferred from other edges

E.g., green edge can be inferred from blue

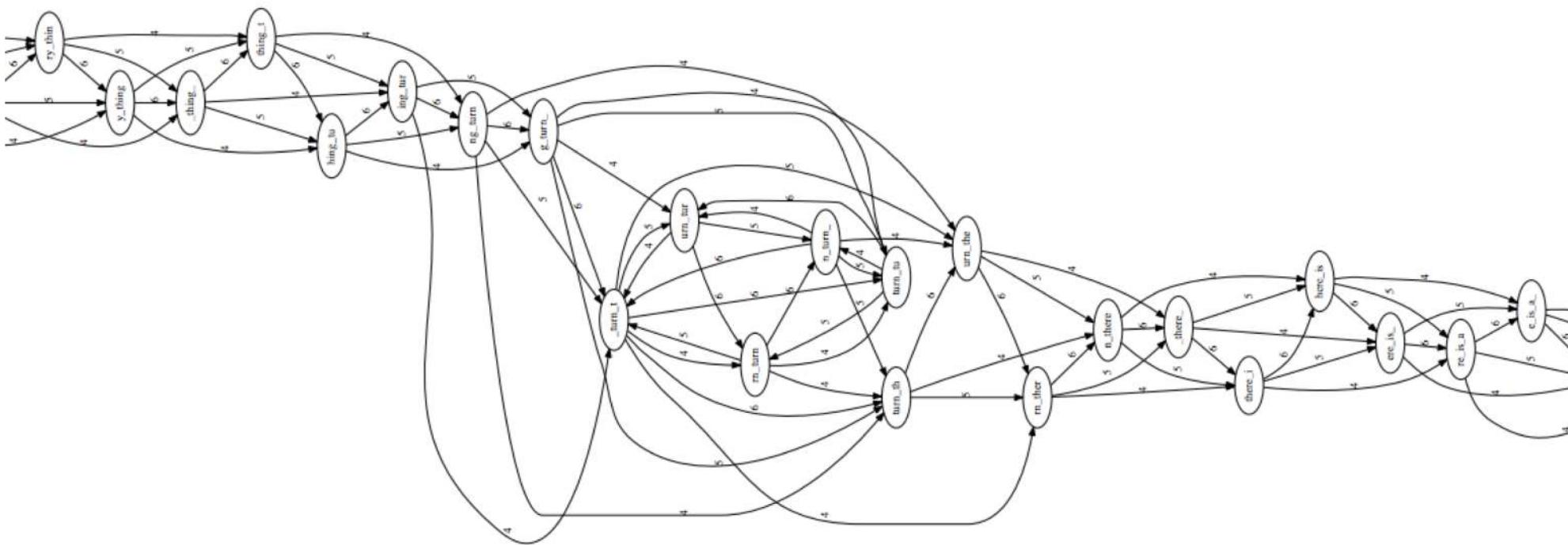


# Layout

Remove transitively-inferrible edges, starting with edges that skip one node:

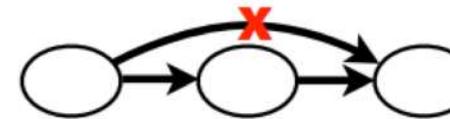


Before:

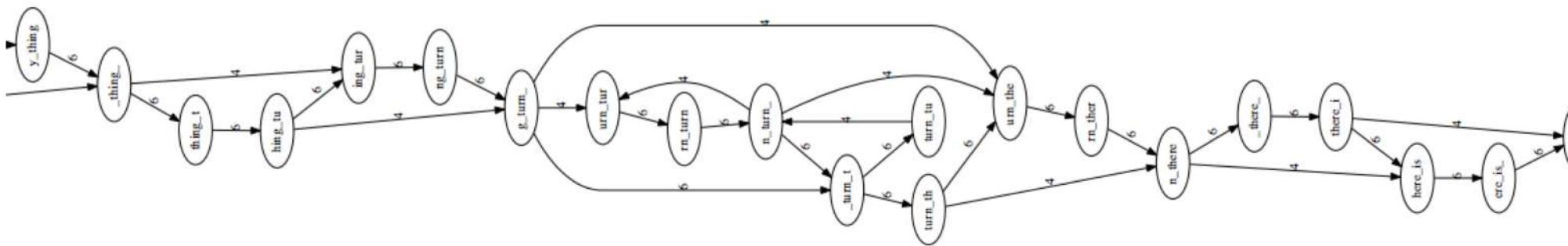


# Layout

Remove transitively-inferrible edges, starting with edges that skip one node:



After:

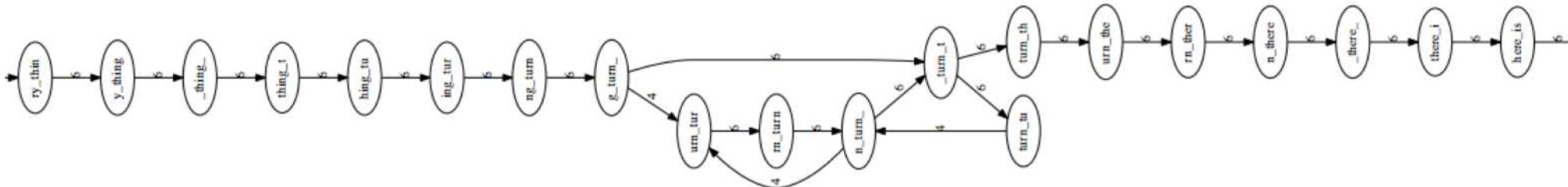


# Layout

Remove transitively-inferrible edges, starting with edges that skip one or two nodes:

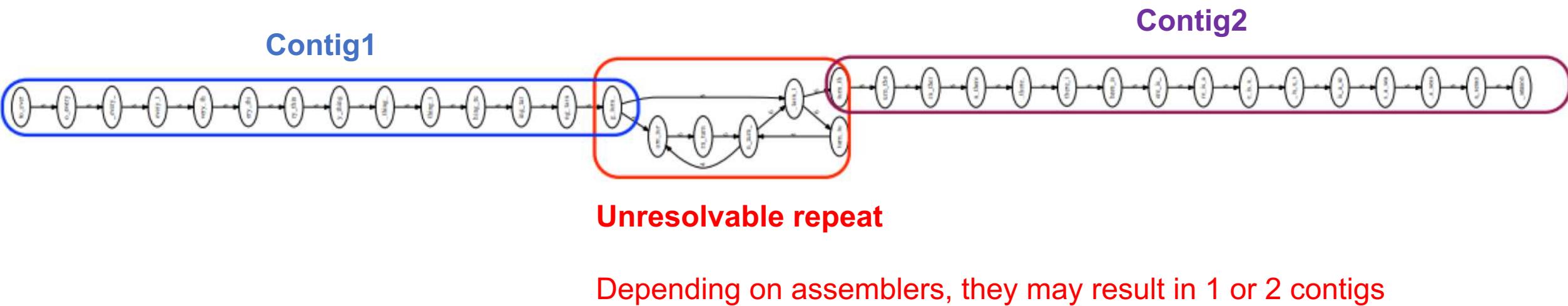


After:



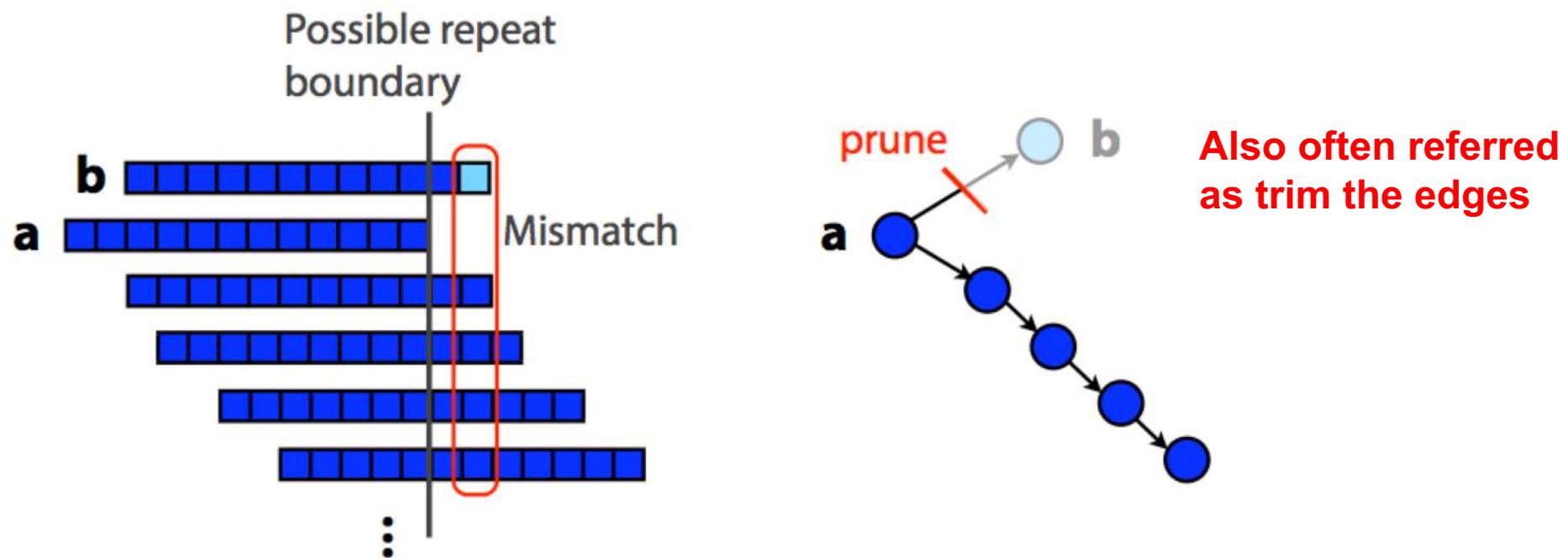
Even simpler

# Layout



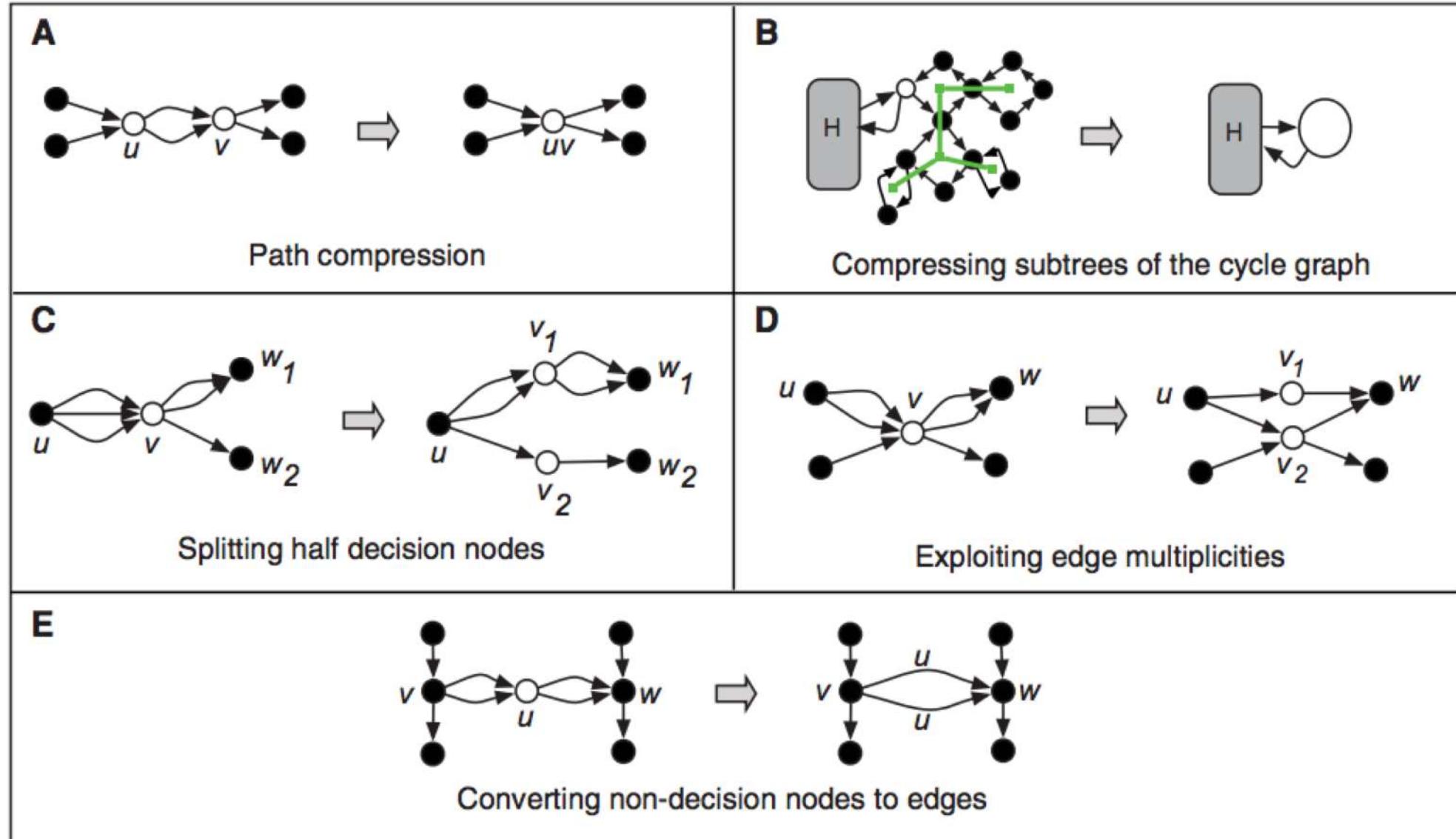
# Layout – can we do more?

In practice, layout step also has to deal with spurious subgraphs, e.g. because of sequencing error



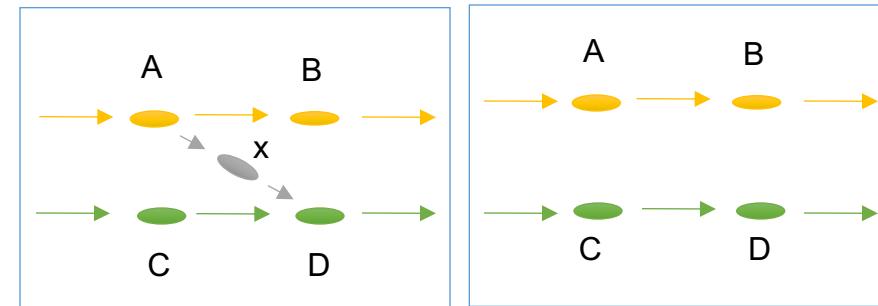
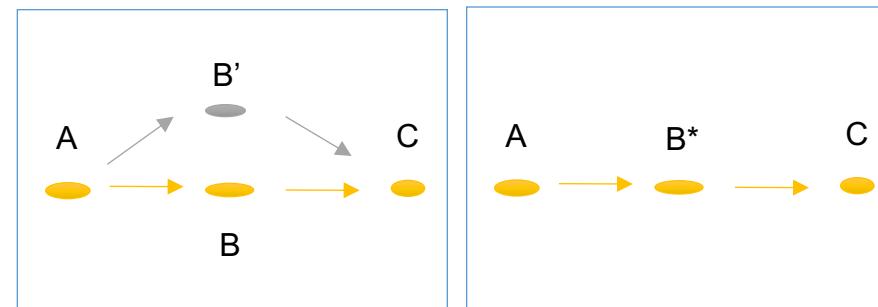
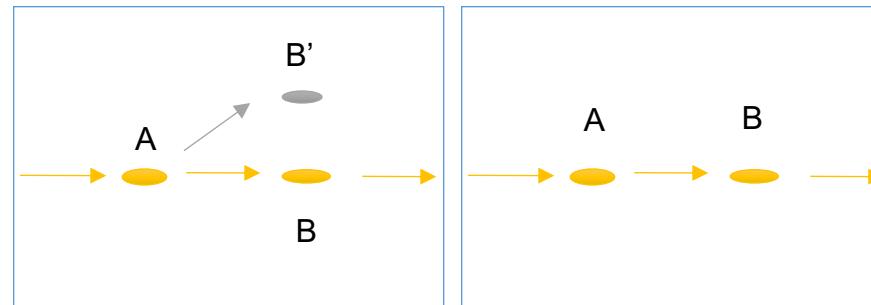
Mismatch could be due to sequencing error or repeat. Since the path through **b** ends abruptly we might conclude it's an error and prune **b**.

# Usefulness of graph transformation



# Error correction in graph

- Errors at end of read
  - Trim off ‘dead-end’ tips
- Errors in middle of read
  - Pop Bubbles
- Chimeric Edges
  - Clip short, low coverage nodes



# Consensus

TAGATTACACAGATTACTGA TTGATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA  
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA

↓      ↓      ↓      ↓      ↓

TAGATTACACAGATTACTGACTTGATGGCGTAA CTA



Take reads that make up a contig and line them up

Take *consensus*, i.e. majority vote

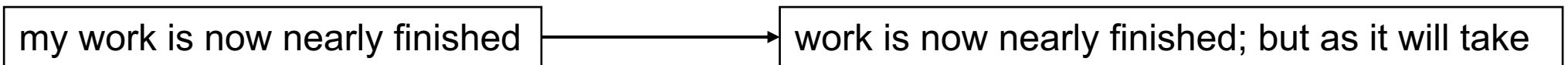
At each position, ask: what nucleotide (and/or gap) is here?

Complications: (a) sequencing error, (b) ploidy

Say the true genotype is AG, but we have a high sequencing error rate and only about 6 reads covering the position.

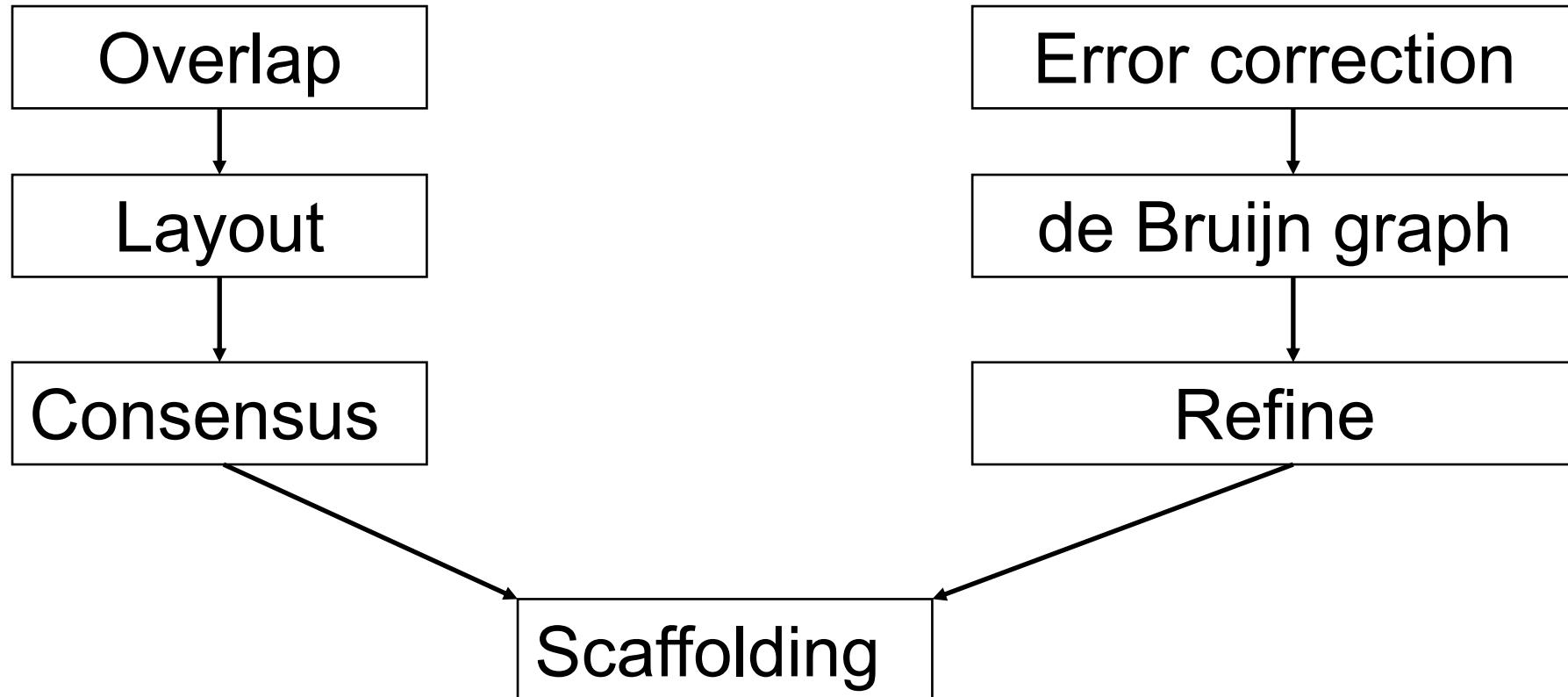
# OLC Assemblers

- Mostly used in the Sanger sequencing era
  - Celera, Phusion, PCAP, Arachne
- Disadvantages of OLC
  - Computing overlaps is slow
  - 5 billion reads -> takes **400 years** to compute overlaps if 1 million overlap per second
  - Overlap graph is big and complicated
    - One node per read
    - Number of edges grows superlinearly with number of reads



- When 2<sup>nd</sup> generation dataset first arrived
  - Millions and millions of reads
  - Short read length – difficult to build sufficient overlap

# OLC and DBG assemblers



# k-mer

“k-mer” is a substring of length k

S: **GGCGATTCA**T<sub>CG</sub>

A 4-mer of S: **ATTC**

All 3-mers of S:

**GGC**  
**GCG**  
**CGA**  
**GAT**  
**ATT**  
**TTC**  
**TCA**  
**CAT**  
**ATC**  
**TCG**

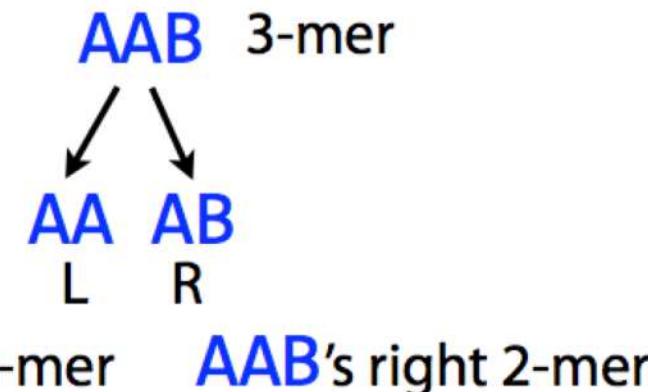
*mer*: from Greek meaning “part”

# De Bruijn graph

We start with a collection of reads of **3bp** from the reference genome **AAABBBA**

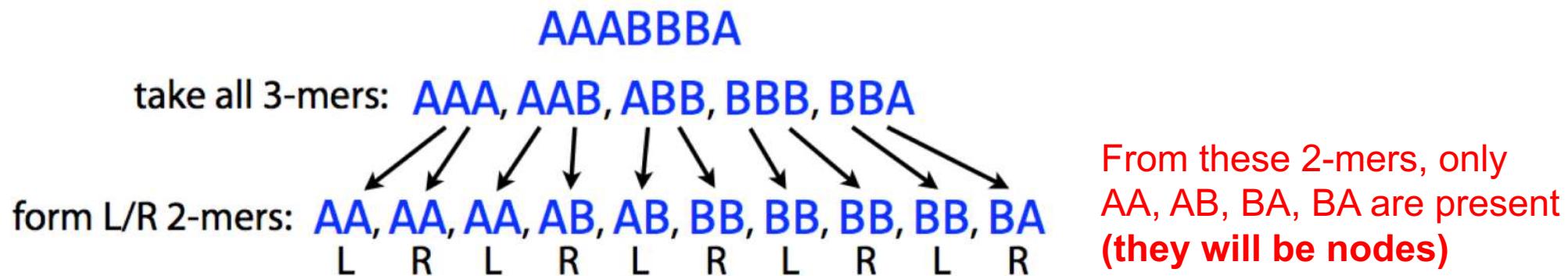
**AAA, AAB, ABB, BBB, BBA**

**AAB** is a  $k$ -mer ( $k = 3$ ). **AA** is its *left*  $k-1$ -mer, and **AB** is its right  $k-1$ -mer.



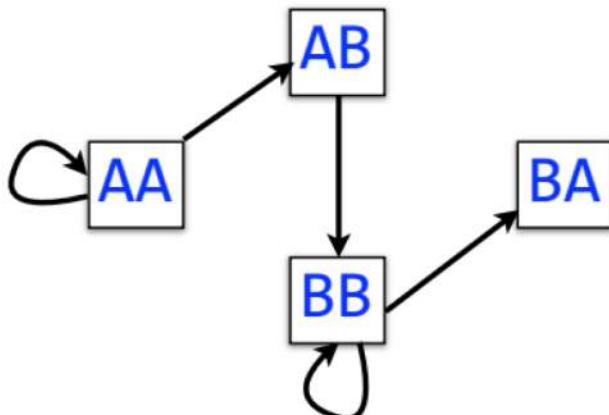
# De Bruijn graph

Take each length-3 input string and split it into two overlapping substrings of length 2. Call these the *left* and *right* 2-mers.



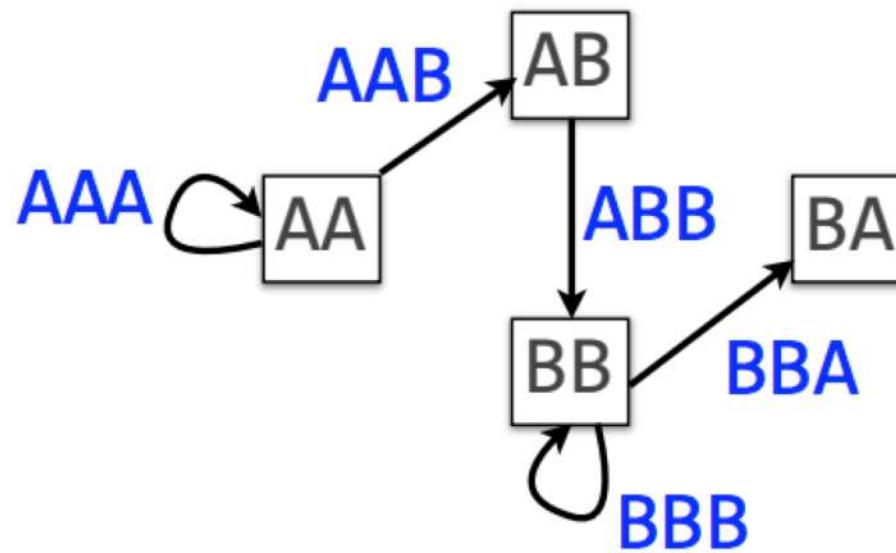
Let 2-mers be nodes in a new graph. Draw a directed edge from each left 2-mer to corresponding right 2-mer:

So AAB will be AA → AB



Each *edge* in this graph corresponds to a length-3 input string

# De Bruijn graph



How do we get contigs from the graph?

Intuitively we walk and visited all edges and node of the graph, but how?

An edge corresponds to an overlap (of length  $k-2$ ) between two  $k-1$  mers.  
More precisely, it corresponds to a  **$k$ -mer** from the input.

# De Bruijn graph is a directed multigraph

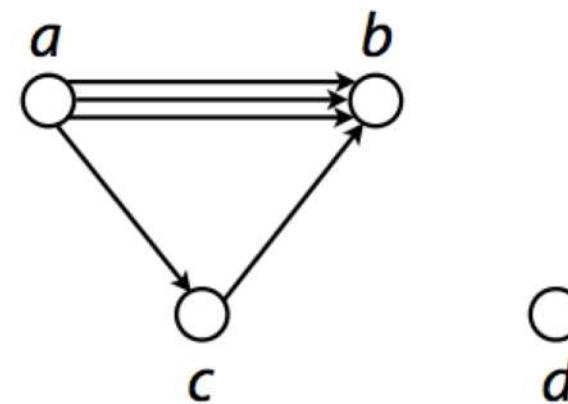
Directed **multigraph**  $G(V, E)$  consists of set of vertices,  $V$  and **multiset** of *directed edges*,  $E$

Otherwise, like a directed graph

Node's *indegree* = # incoming edges

Node's *outdegree* = # outgoing edges

De Bruijn graph is a directed multigraph



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

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

Repeated

# Eulerian walk definitions and statement

1.A **directed graph** is a *graph* in which each edge has a direction, usually represented as an arrow from a node  $v$  to a node  $w$ .

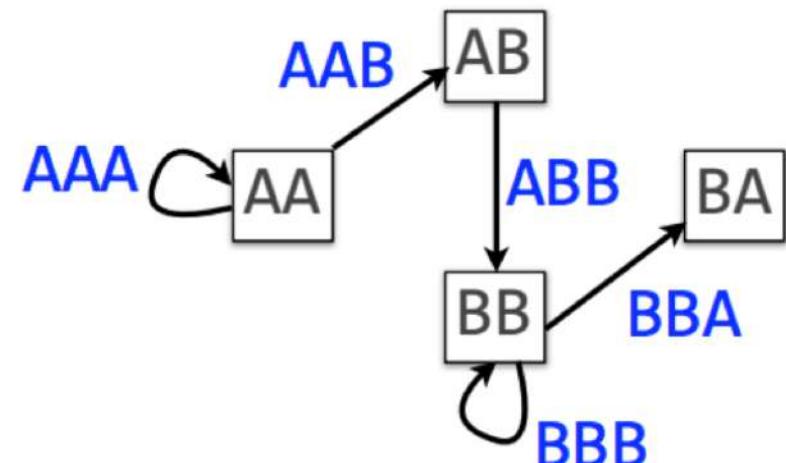
Yes

2.Graph is connected if each node can be reached by some other node.

Yes

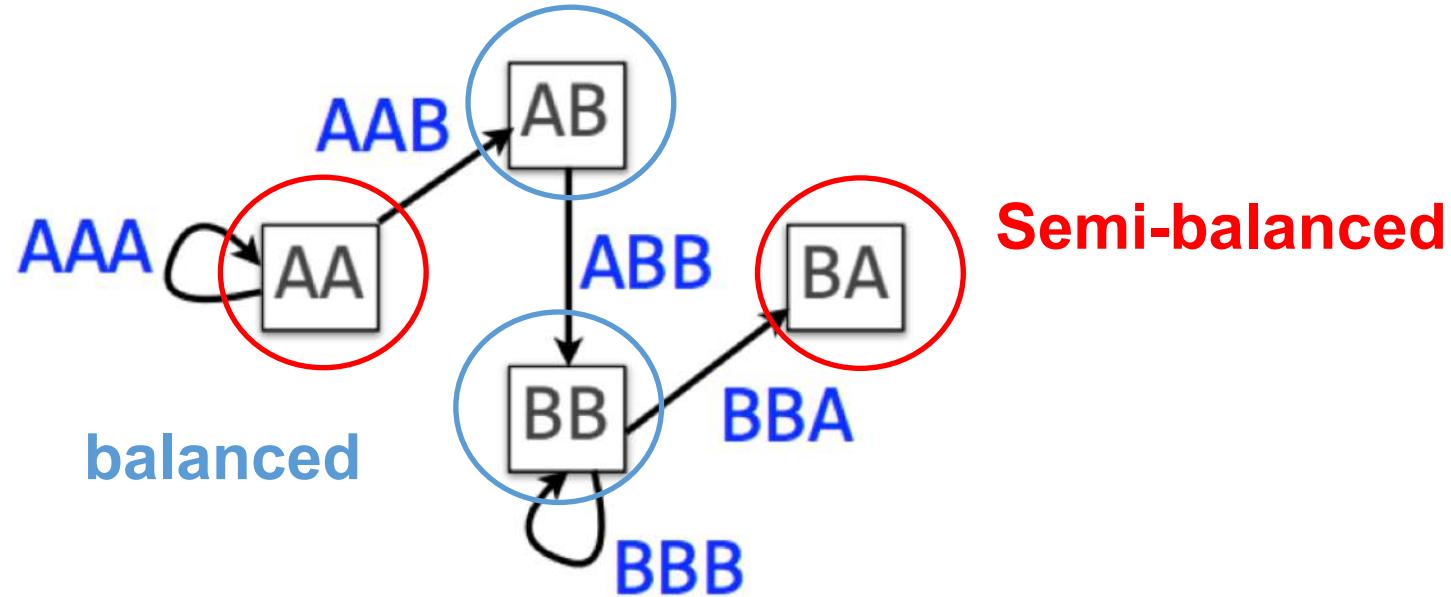
3.Node is **balanced** if indegree equals outdegree Node is **semi-balanced** if indegree differs from outdegree by 1

4.A directed, connected graph is Eulerian if and only if it has **at most 2 semi-balanced nodes** and all other nodes are balanced



# Eulerian walk definitions and statement

A **directed, connected graph** is Eulerian if and only if **it has at most 2 semi-balanced nodes and all other nodes are balanced**



Is it Eulerian? Yes

Argument 1:  $\text{AA} \rightarrow \text{AA} \rightarrow \text{AB} \rightarrow \text{BB} \rightarrow \text{BB} \rightarrow \text{BA}$

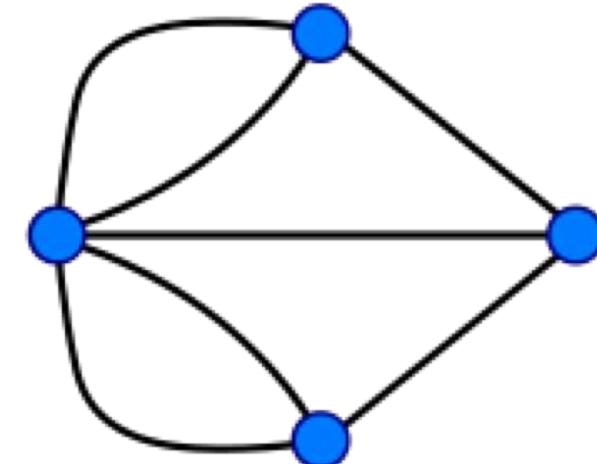
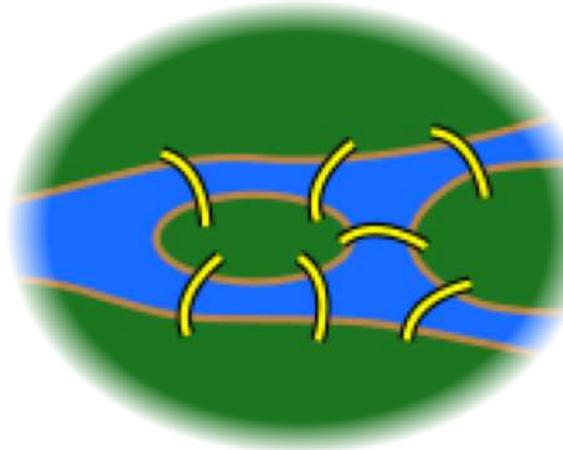
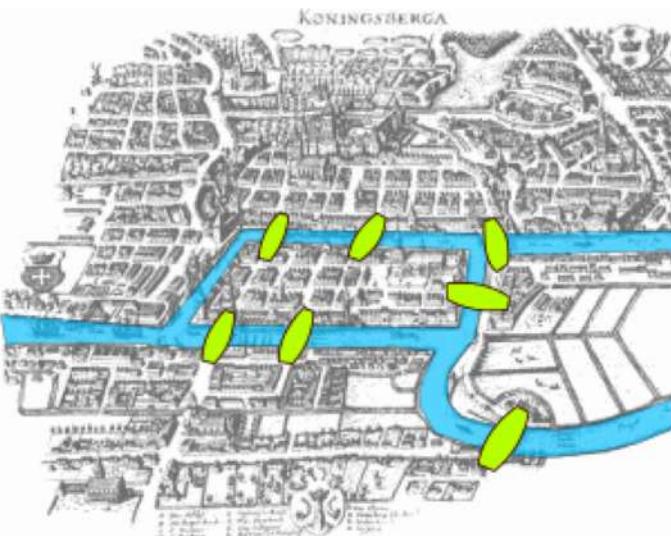
Argument 2: **AA** and **BA** are semi-balanced, **AB** and **BB** are balanced

Then we can search for Eulerian path from the graph:  
Eulerian walk visits each edge exactly once \*\*\*\*

# Eulerian walk

In [graph theory](#), an **Eulerian trail** (or **Eulerian path**) is a [trail](#) in a graph which visits every [edge](#) exactly once. Similarly, an **Eulerian circuit** or **Eulerian cycle** is an Eulerian trail which starts and ends on the same [vertex](#). They were first discussed by [Leonhard Euler](#) while solving the famous [Seven Bridges of Königsberg](#) problem in 1736. Mathematically the problem can be stated like this:

Given the graph in the image, is it possible to construct a path (or a [cycle](#), i.e. a path starting and ending on the same vertex) which visits each edge exactly once?



PS. This graph is  
Not Eulerian

# De Bruijn graph

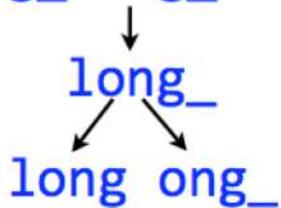
A procedure for making a De Bruijn graph  
for a genome

Assume *perfect sequencing* where each length- $k$   
substring is sequenced exactly once with no errors

Pick a substring length  $k$ : 5

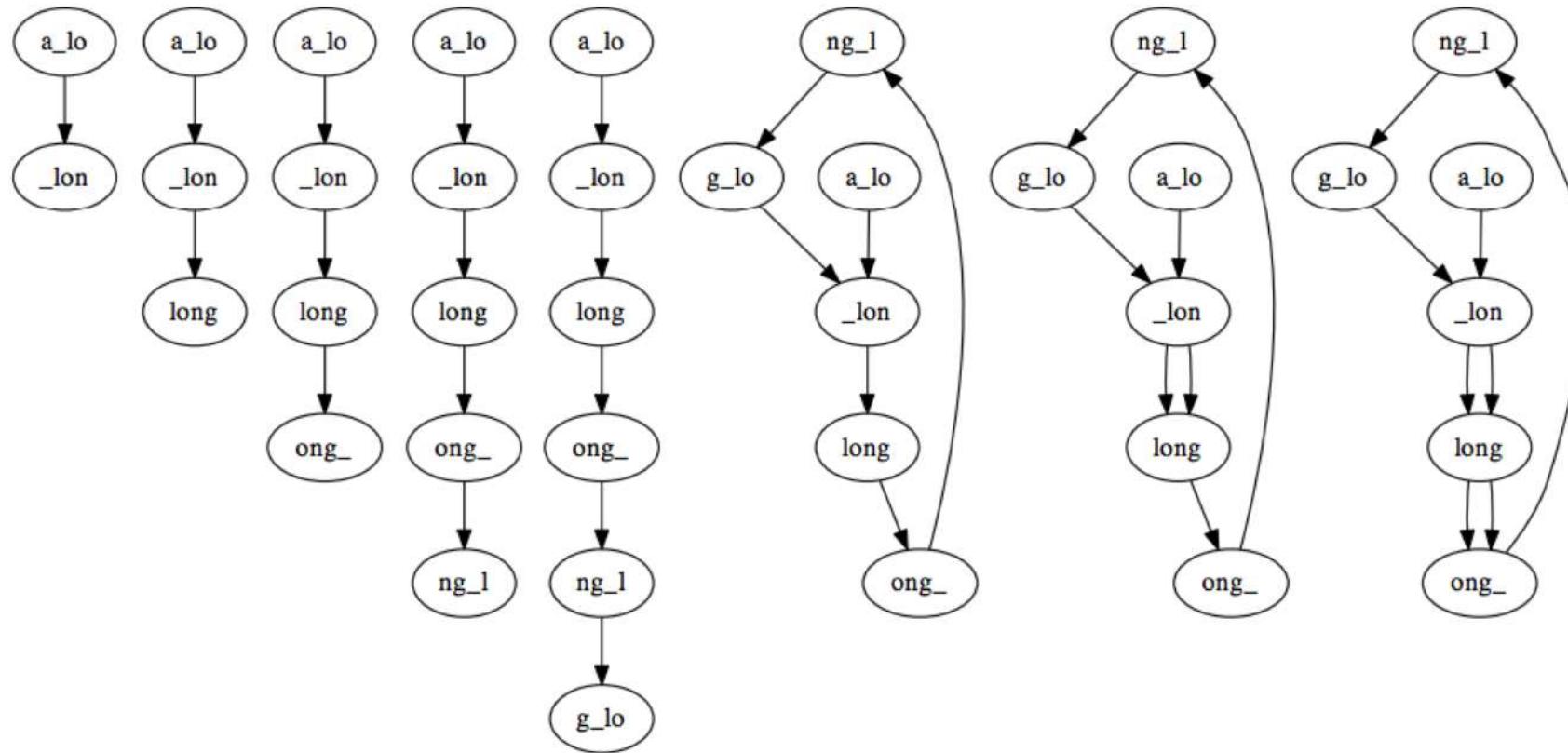
Start with an input string: a\_long\_long\_long\_time

Take each  $k$  mer and split  
into left and right  $k-1$  mers



Add  $k-1$  mers as nodes to De Bruijn graph  
(if not already there), add edge from left  $k-1$   
mer to right  $k-1$  mer

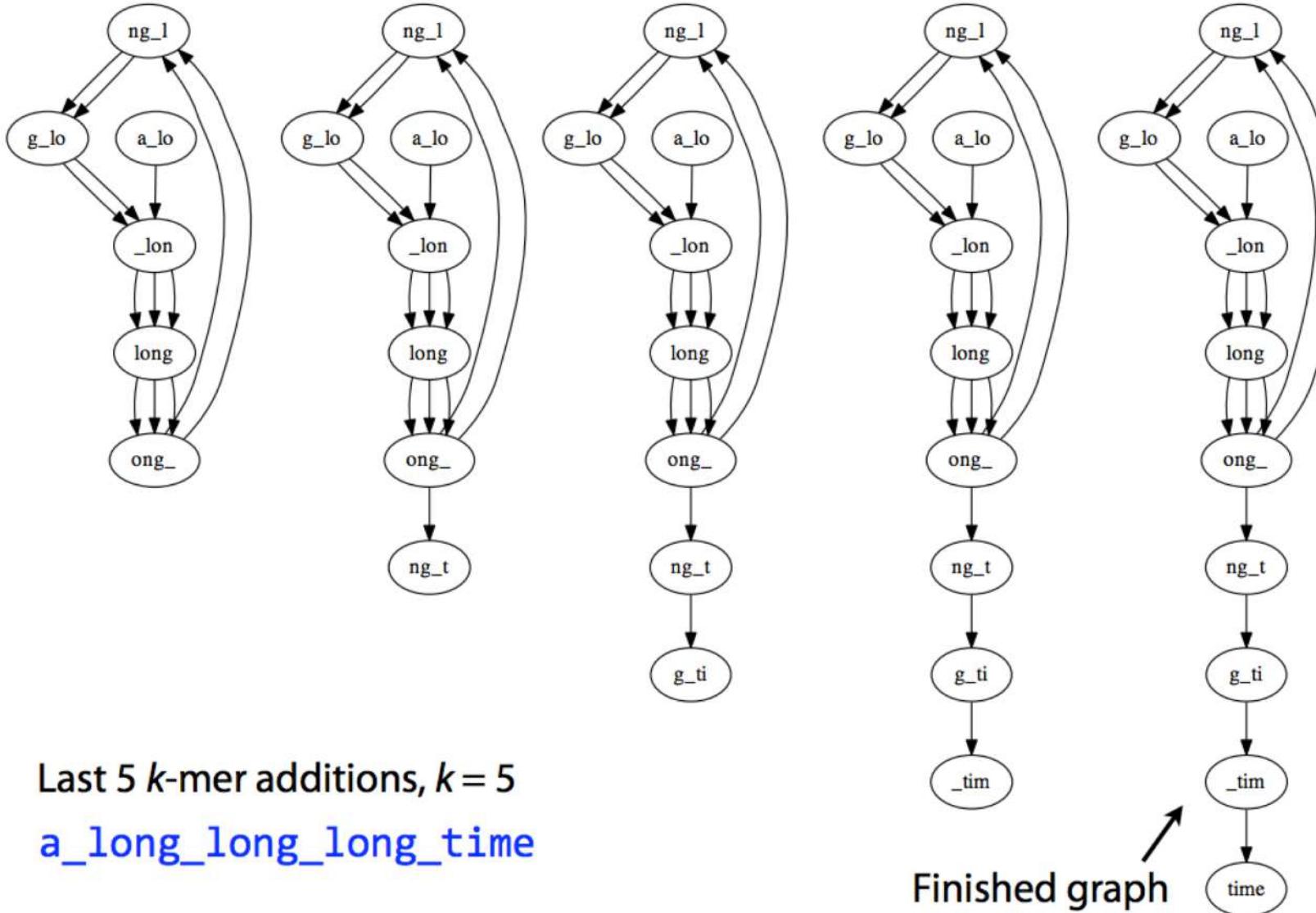
# De Bruijn graph



First 8  $k$ -mer additions,  $k = 5$

a\_long\_long\_long\_time

# De Bruijn graph



# De Bruijn graph

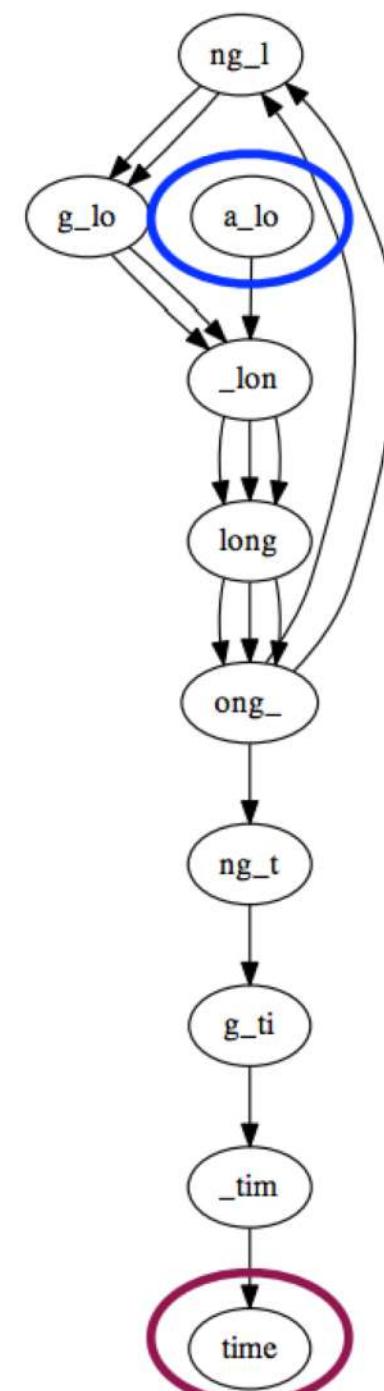
With perfect sequencing, this procedure always yields an Eulerian graph. Why?

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

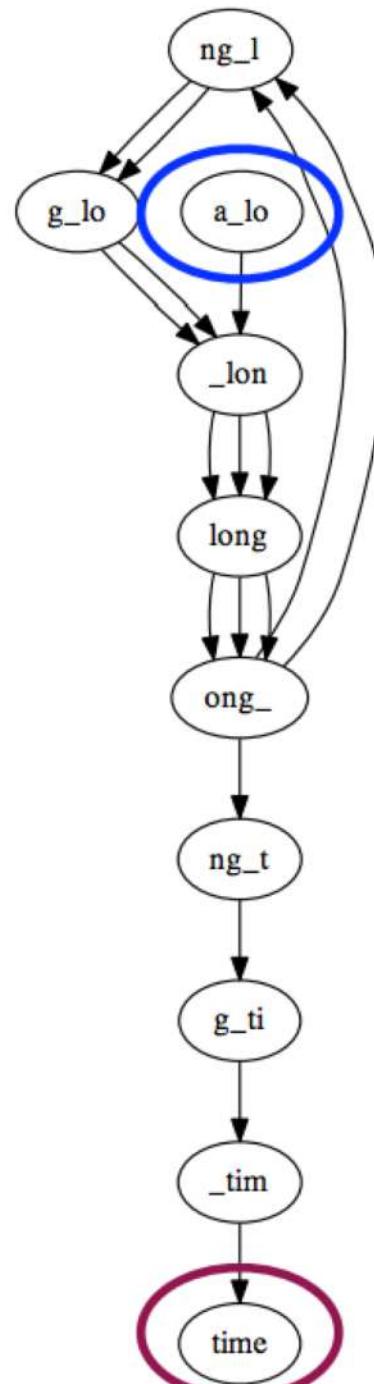
\* Unless genome is circular



# De Bruijn graph with actual data

Assuming perfect sequencing, procedure yields graph with Eulerian walk that can be found efficiently.

We saw cases where Eulerian walk corresponds to the original superstring. Is this always the case?



# When k-mer is repeat

**No:** graph can have multiple Eulerian walks, only one of which corresponds to original superstring

Right: graph for **ZABCDABEFABY**,  $k=3$

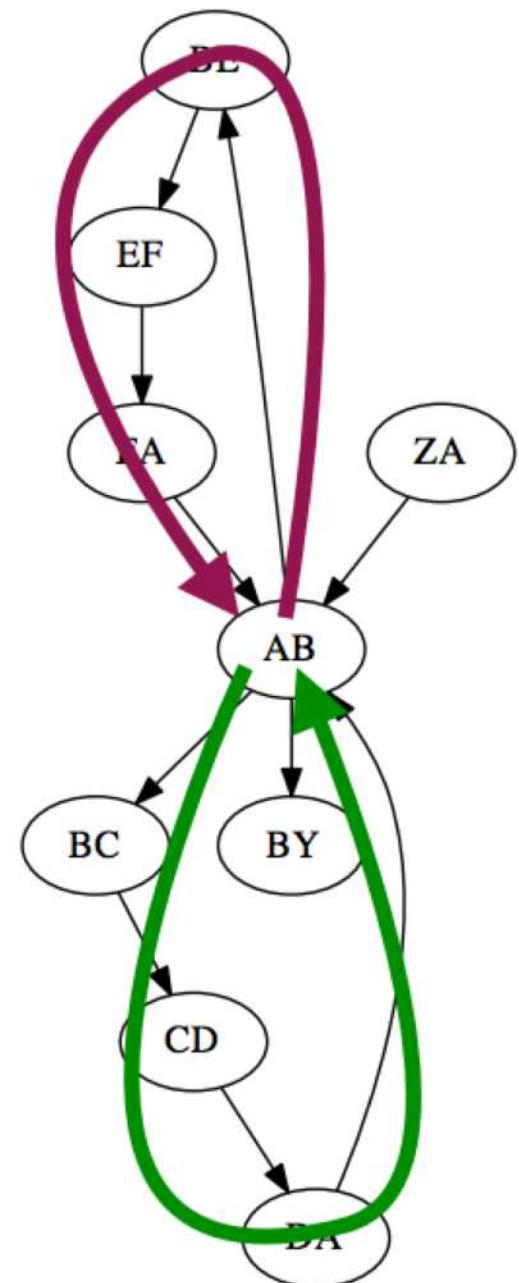
Alternative Eulerian walks:

**ZA** → **AB** → **BE** → **EF** → **FA** → **AB** → **BC** → **CD** → **DA** → **AB** → **BY**

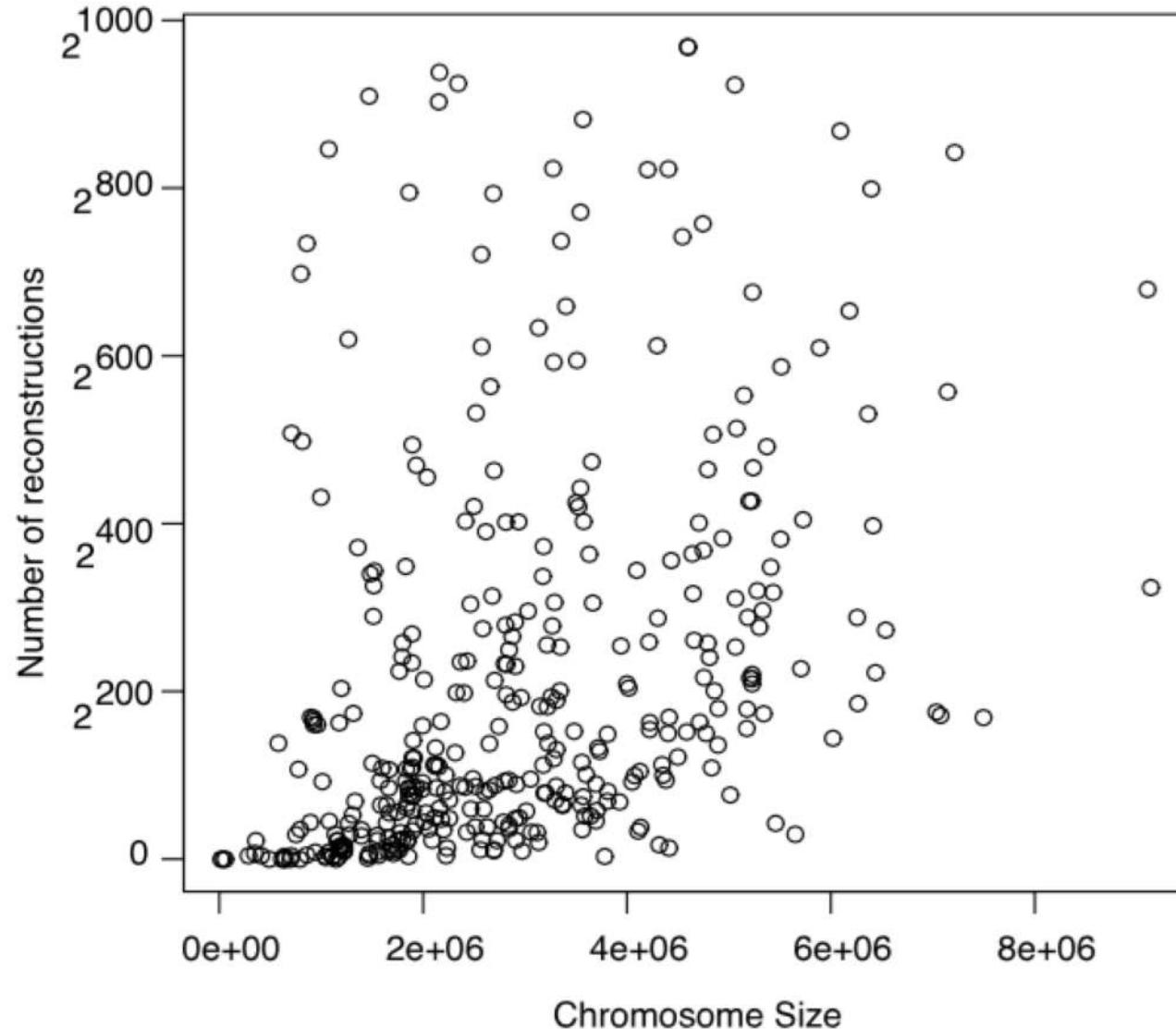
**ZA** → **AB** → **BC** → **CD** → **DA** → **AB** → **BE** → **EF** → **FA** → **AB** → **BY**

These correspond to two edge-disjoint directed cycles joined by node **AB**

**AB** is a repeat: **ZABCDABEFABY**



# When k-mer is repeat (in practice)



**Figure 2** Number of words consistent with genome graphs. The

# Impact of changing kmer size

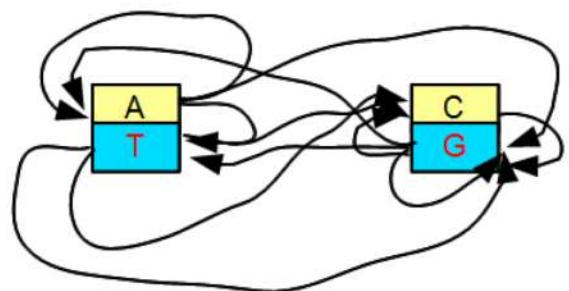
sequence

ATGGAAAGTCGCGGAATC

1mers

A  
T  
G  
G  
A  
A  
G  
T  
C  
G  
C  
G  
G  
A  
A  
T  
C

de Bruijn graph



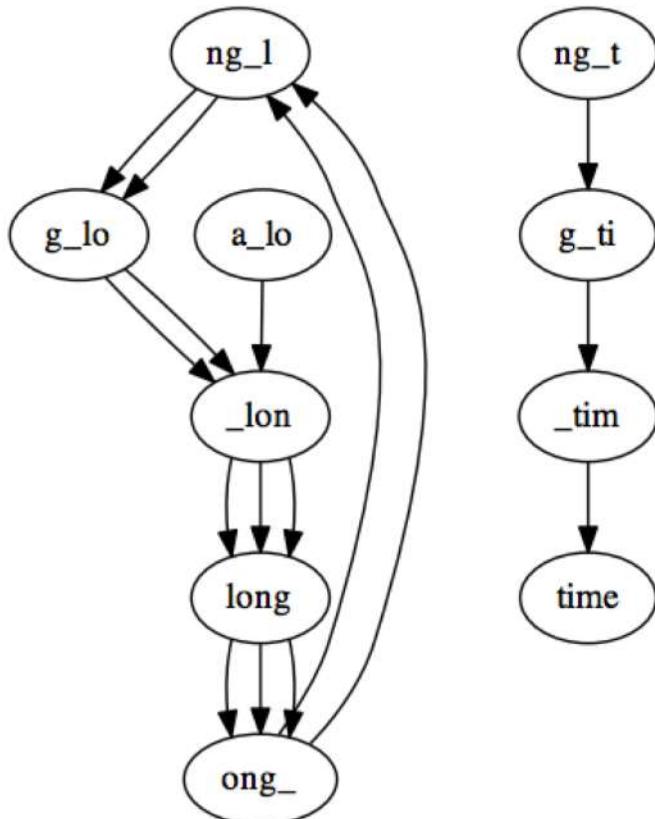
**Example** of a kmer of 1 basically means A,C,T,G are all repeats..

Larger kmer will span more small repeat less than kmer size, but likely to have less overlap

# Low coverage = disconnected graph

Gaps in coverage can lead to *disconnected* graph

Graph for `a_long_long_long_time`,  $k = 5$  but *omitting* `ong_t`:



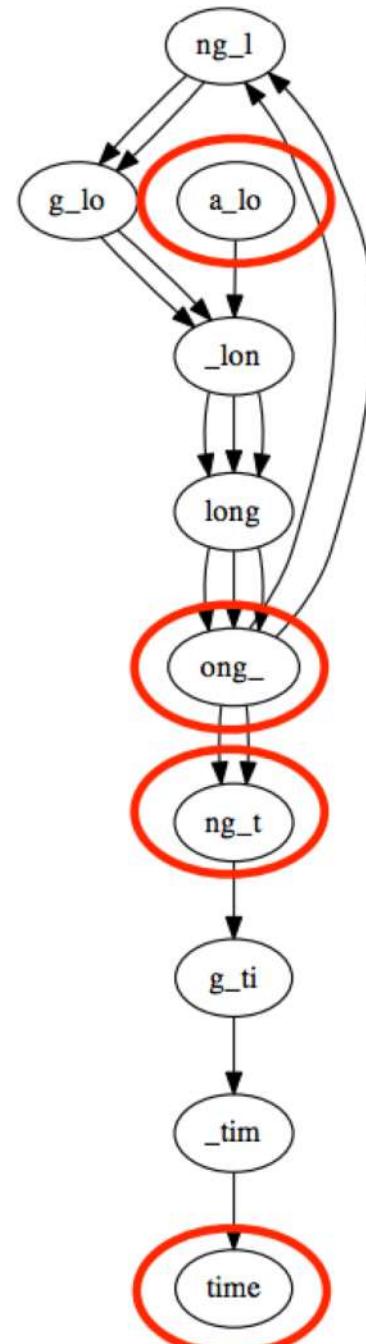
Connected components are individually Eulerian, overall graph is not

# Coverage difference = not Eulerian

Differences in coverage also lead to non-Eulerian graph

Graph for `a_long_long_long_time`,  
 $k = 5$  but with extra copy of `ong_t`:

Graph has 4 semi-balanced nodes,  
isn't Eulerian



# De Bruijn graph

Gaining assembly as Eulerian walk is appealing, not many practical cases impede this:

- Uneven coverage, sequencing errors, make graph non-Eulerian
- Repeats produces many possible walks

But there is one major advantage of De Bruijn graph over OLC

- Computationally efficient

# Efficiency

Assume you have 5 billion reads to assemble

- Not uncommon nowadays in some plant species

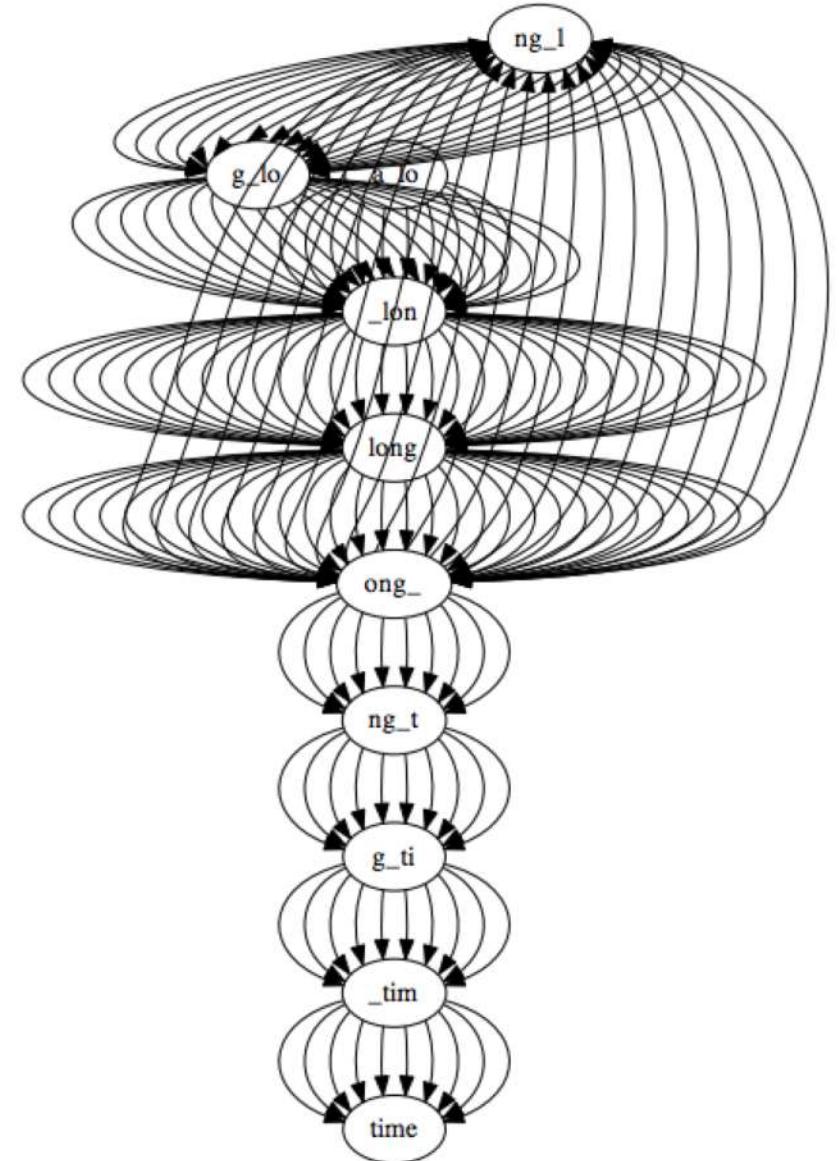
OLC: 12.5 quadrillion overlaps to compute first

Even 1 million overlap per sec will equate to **400 years**

DBG: depends on genome size

# Size of De Bruijn graph depends on genome size

Usually ~50X paired end reads in coverage



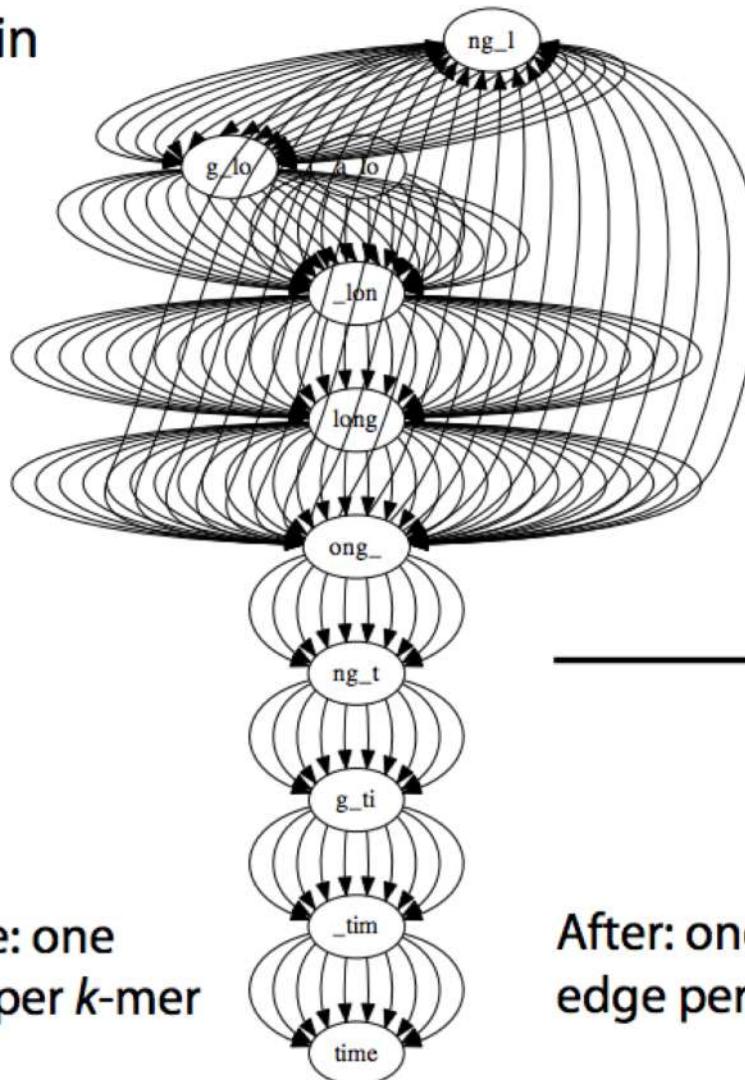
# Size of De Bruijn graph depends on genome size

Same edge might appear in dozens of copies; let's use edge *weights* instead

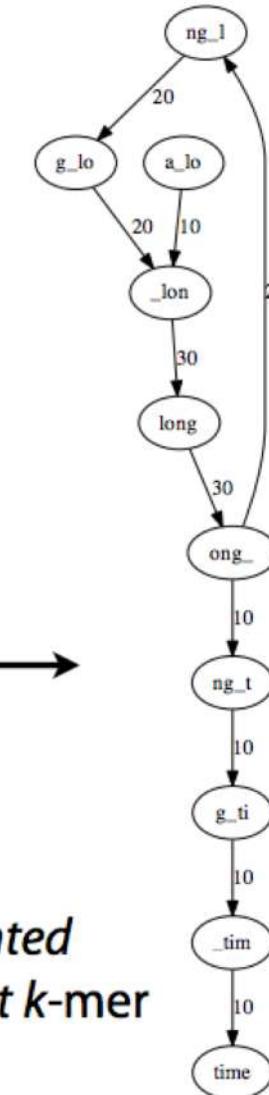
Weight = # times k-mer occurs

Using weights, there's one *weighted* edge for each *distinct* k-mer

Before: one edge per k-mer



After: one *weighted* edge per *distinct* k-mer

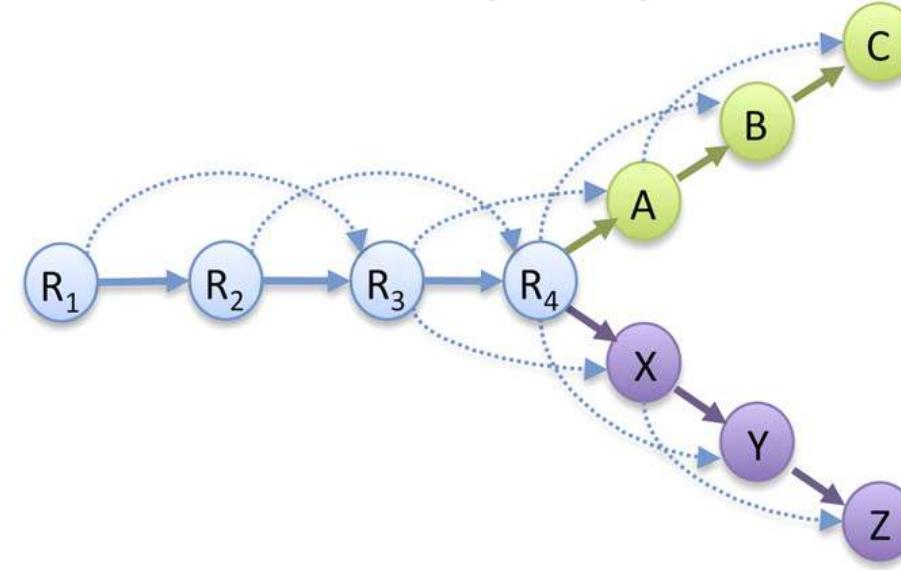


# Summary

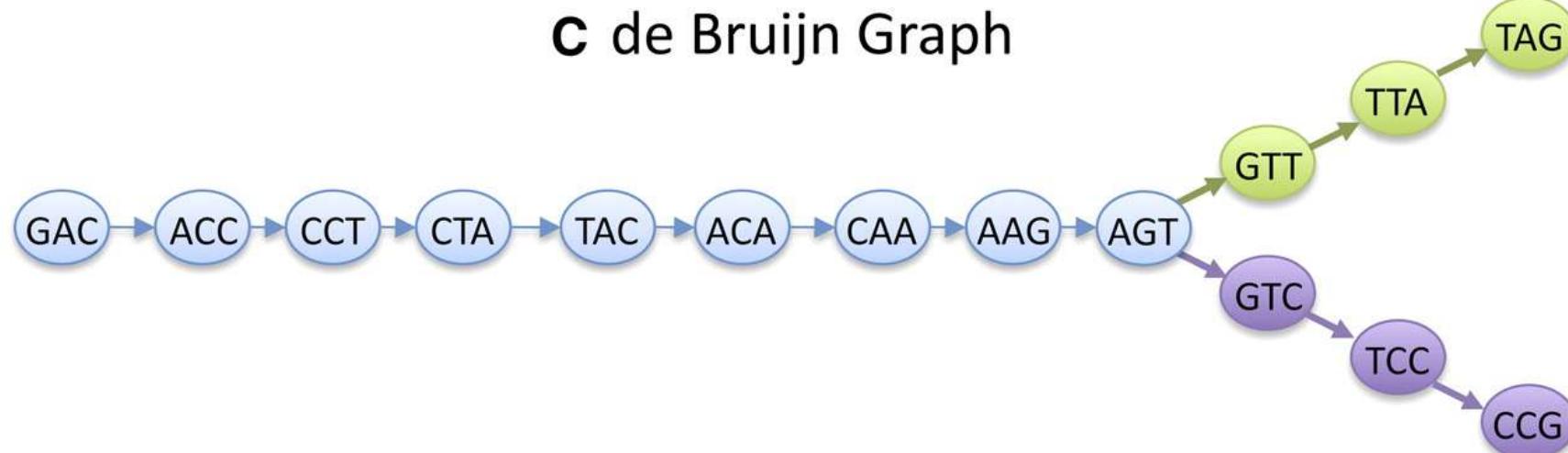
## A Read Layout

R <sub>1</sub> :	GACCTACA
R <sub>2</sub> :	ACCTACAA
R <sub>3</sub> :	CCTACAAAG
R <sub>4</sub> :	CTACAAGT
A:	TACAAGTT
B:	ACAAGTTA
C:	CAAGTTAG
X:	TACAAGTC
Y:	ACAAGTCC
Z:	CAAGTCCG

## B Overlap Graph



## C de Bruijn Graph



# Summary

Advantage of DBG:

Time to build based on Genome size (G) or total length of reads (N)

For OLC: time to build overlap graph is based on number of reads

But:

DBG not flexible: only overlap of **fixed length** (=kmer)  
can't solve repeat with repeat > kmer

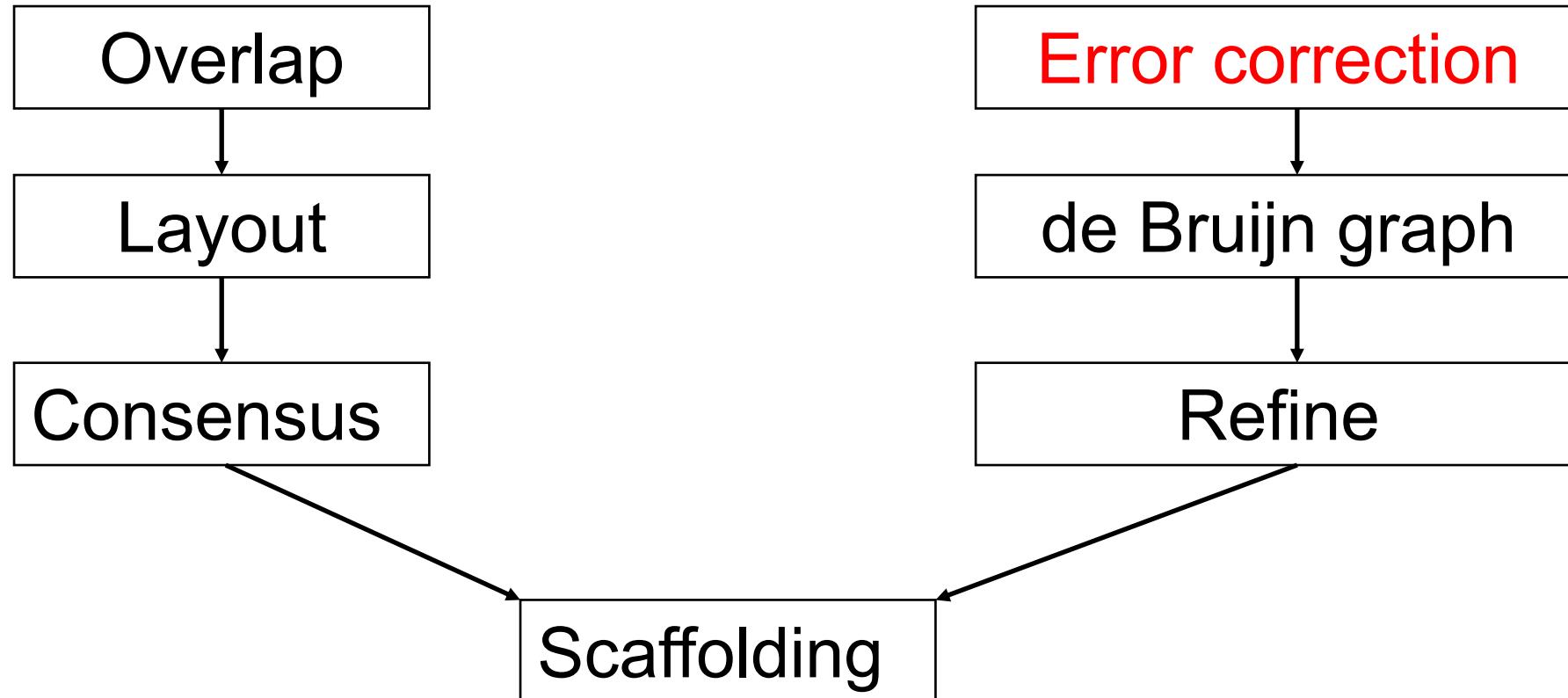
Read information is lost: All reads are split into kmers.  
(A lot of work on later DBG assemblers are put in this)

Tradeoff between DBG and OLC needed

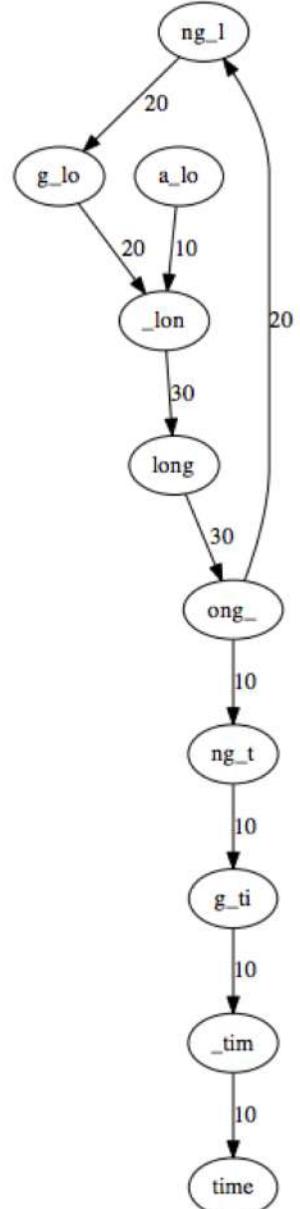
Some improve over existing approach: Spades

Some combine both: Maserca

# OLC and DBG assemblers



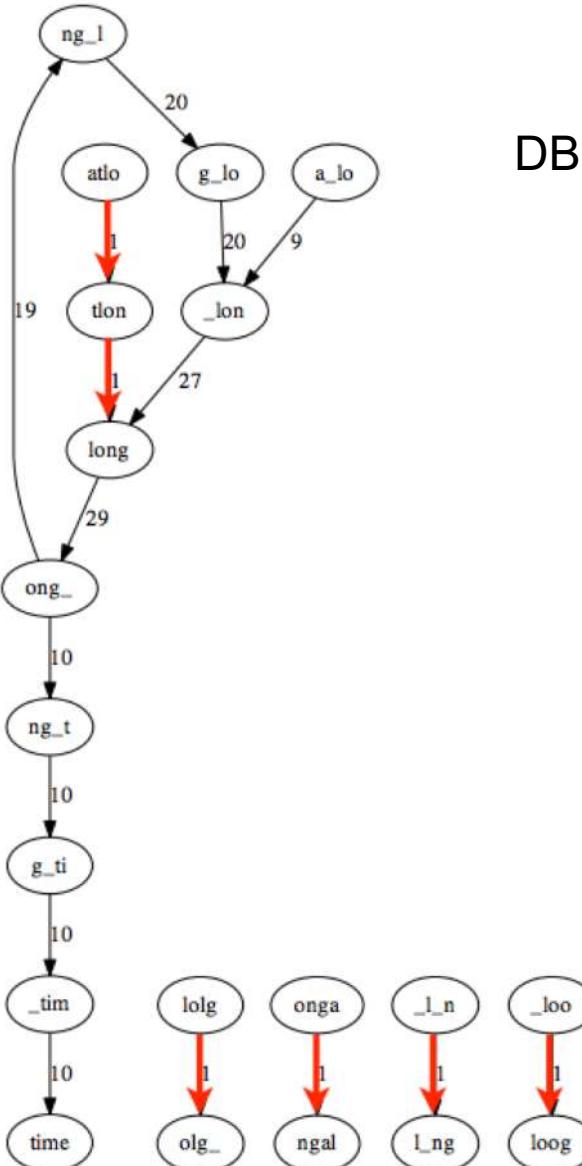
# Error in graphs



DBG from perfect reads (10X)

As you can see all weighted edges have high coverage

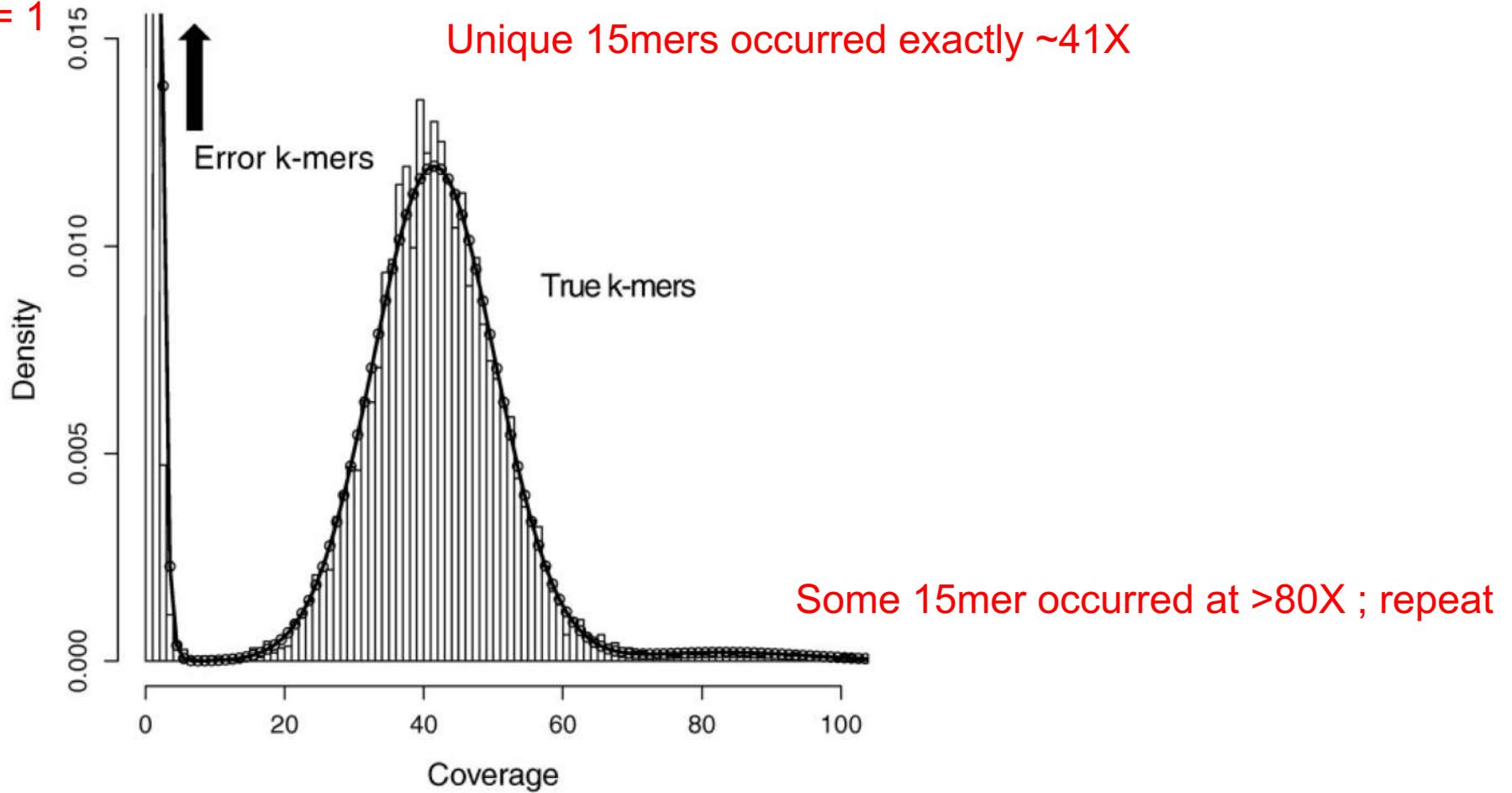
Some higher than the other; this is obviously repeat



DBG from reads with some errors

# Kmer coverage

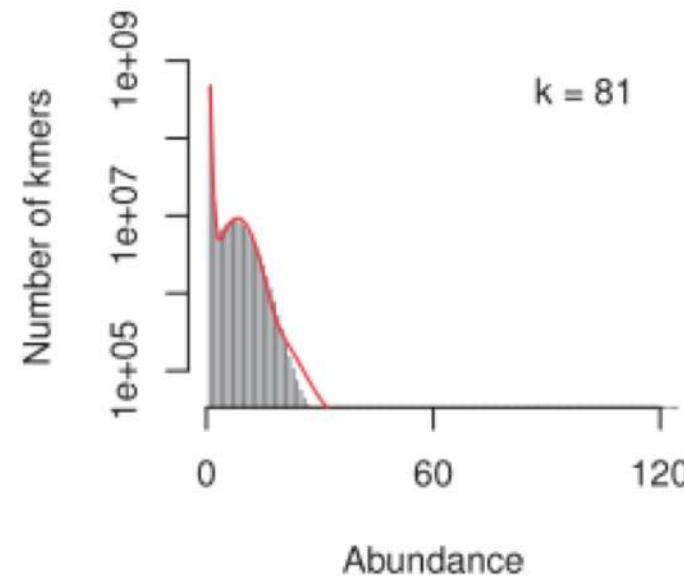
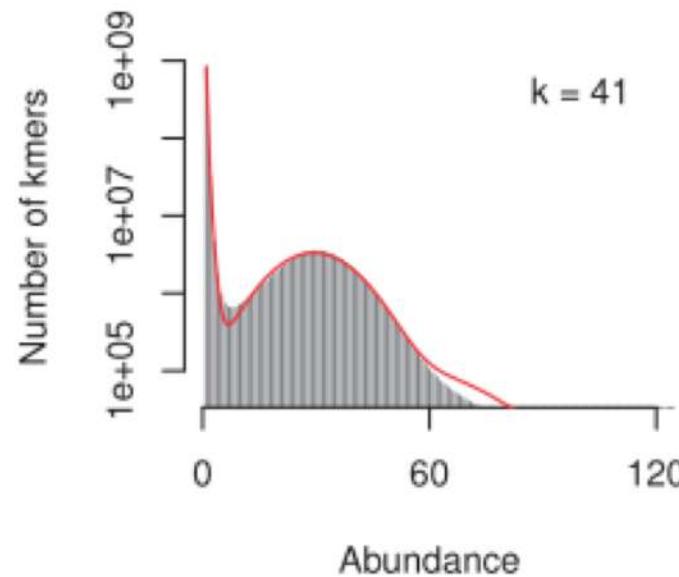
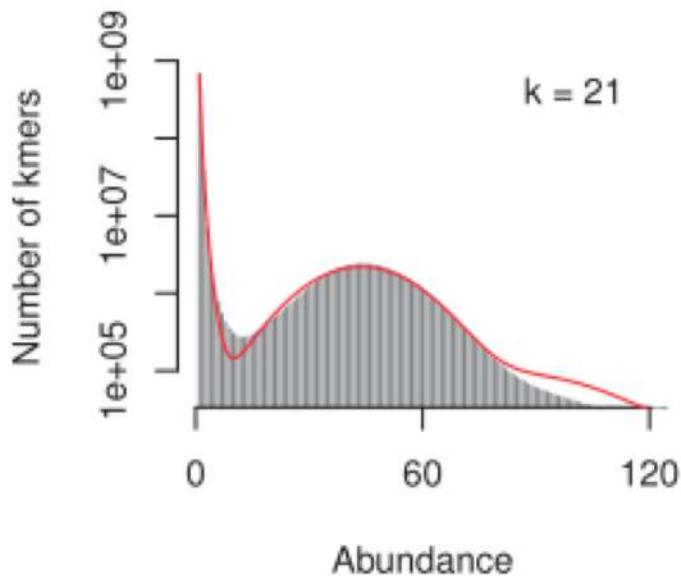
Indeed most erroneous kmer  
has kmer = 1



Variance = 77X

Almost twice of mean suggest it's Poisson distribution

# Choosing the right kmer



**Fig. 2.** The abundance histograms for *chr14* with  $k$  values of 21, 41 and 81 (on a y log scale). Each plot also shows a curve corresponding to the optimized statistical model (haploid)

# Error correction: rationale

Idea: errors tend to turn frequent  $k$ -mers to infrequent  $k$ -mers, so corrections should do the reverse

Say we have a collection of reads where each distinct 8-mer occurs an average of ~10 times, and we have the following read:

Read: GCGTATTACGCGTCTGGCCT (20 nt)

8-mers:	
GCGTATTAA	8
CGTATTAC	8
GTATTACG	9
TATTACGC	9
ATTACGCG	9
TTACGCGT	12
TACGCGTC	9
ACGCGTCT	8
CGCGTCTG	10
GCGTCTGG	10
CGTCTGGC	11
GTCTGGCC	9
TCTGGCCT	8

# times each 8-mer occurs in the dataset.  
"k-mer count profile"

All 8-mer counts are around the average, suggesting read is error-free

# Error correction: rationale

Suppose there's an **error**

Read: GCGTACTACGCGTCTGGCCT

GCGTACTA:	1
CGTACTAC:	3
GTACTACG:	1
TACTACGC:	1
ACTACGCG:	2
CTACGCGT:	1
TACGCGTC:	9
ACGCGTCT:	8
CGCGTCTG:	10
GCGTCTGG:	10
CGTCTGGC:	11
GTCTGGCC:	9
TCTGGCCT:	8

Below average

Around average

k-mer count profile has corresponding stretch of below-average counts

# Error correction: rationale

k-mer count profiles when errors are in different parts of the read:

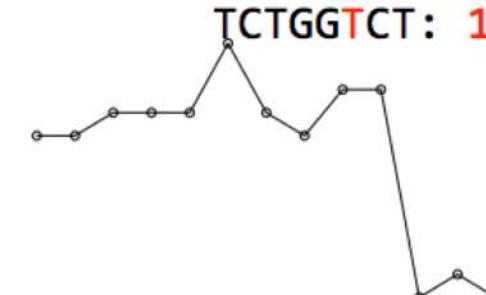
GCGTACTACGCGTCTGGCCT  
GCGTACTA: 1  
CGTACTAC: 3  
GTACTACG: 1  
TACTACGC: 1  
ACTACGCG: 2  
CTACGCGT: 1  
TACGCGTC: 9  
ACGCGTCT: 8  
CGCGTCTG: 10  
GCGTCTGG: 10  
CGTCTGGC: 11  
GTCTGGCC: 9  
TCTGGCCT: 8



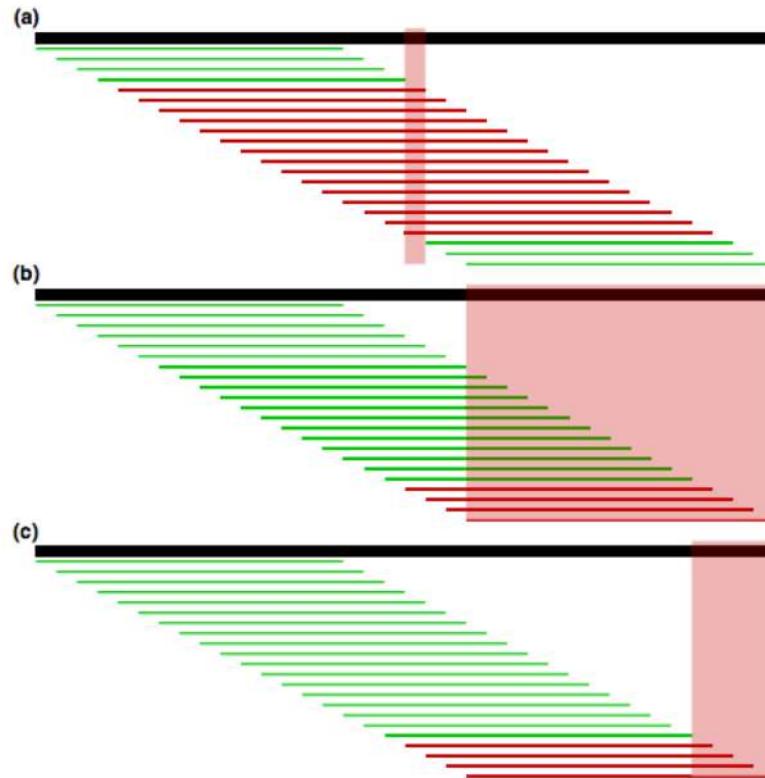
GCGTATTACACGTCTGGCCT  
GCGTATTA: 8  
CGTATTAC: 8  
GTATTACA: 1  
TATTACAC: 1  
ATTACACG: 1  
TTACACGT: 1  
TACACGTC: 1  
ACACGTCT: 2  
CACGTCTG: 1  
GCGTCTGG: 10  
CGTCTGGC: 11  
GTCTGGCC: 9  
TCTGGCCT: 8



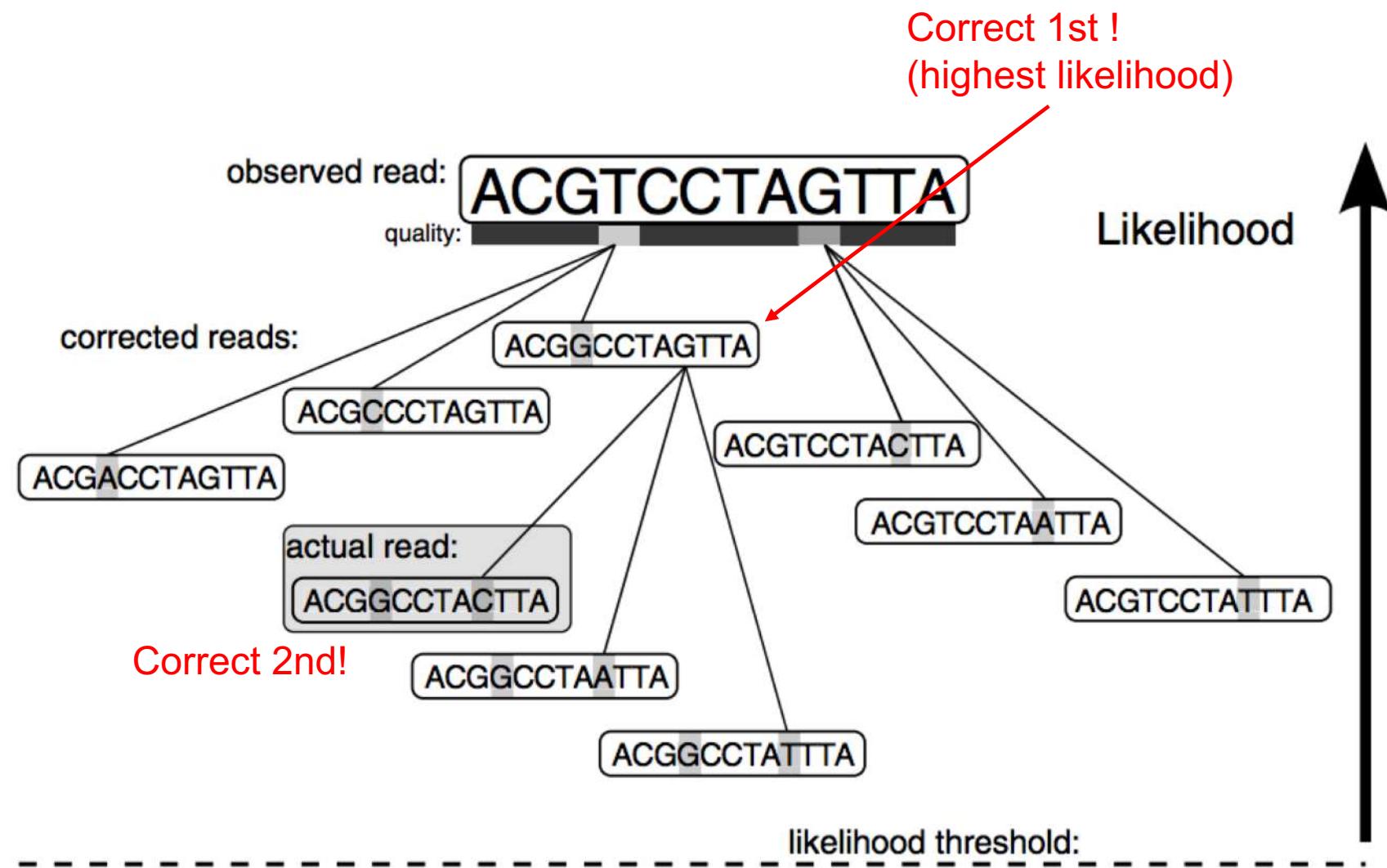
GCGTATTACGCGTCTGGTCT  
GCGTATTA: 8  
CGTATTAC: 8  
GTATTACG: 9  
TATTACGC: 9  
ATTACGCG: 9  
TTACGCGT: 12  
TACGCGTC: 9  
ACGCGTCT: 8  
CGCGTCTG: 10  
GCGTCTGG: 10  
CGTCTGGT: 1  
GTCTGGTC: 2  
TCTGGTCT: 1



# Localize error and correct



**Figure 4 Localize errors.** Trusted (green) and untrusted (red) 15-mers are drawn against a 36 bp read. In (a), the intersection of the untrusted  $k$ -mers localizes the sequencing error to the highlighted column. In (b), the untrusted  $k$ -mers reach the edge of the read, so we must consider the bases at the edge in addition to the intersection of the untrusted  $k$ -mers. However, in most cases, we can further localize the error by considering all bases covered by the right-most trusted  $k$ -mer to be correct and removing them from the error region as shown in (c).



# Velvet: first de Bruijn graph assembler

- Cited 6811 times
- Still being used in some metagenomics dataset

## Resource

---

### Velvet: Algorithms for de novo short read assembly using de Bruijn graphs

Daniel R. Zerbino and Ewan Birney<sup>1</sup>

*EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom*

We have developed a new set of algorithms, collectively called “Velvet,” to manipulate de Bruijn graphs for genomic sequence assembly. A de Bruijn graph is a compact representation based on short words ( $k$ -mers) that is ideal for high coverage, very short read (25–50 bp) data sets. Applying Velvet to very short reads and paired-ends information only, one can produce contigs of significant length, up to 50-kb N50 length in simulations of prokaryotic data and 3-kb N50 on simulated mammalian BACs. When applied to real Solexa data sets without read pairs, Velvet generated contigs of ~8 kb in a prokaryote and 2 kb in a mammalian BAC, in close agreement with our simulated results without read-pair information. Velvet represents a new approach to assembly that can leverage very short reads in combination with read pairs to produce useful assemblies.

[Supplemental material is available online at [www.genome.org](http://www.genome.org). The code for Velvet is freely available, under the GNU Public License, at <http://www.ebi.ac.uk/~zerbino/velvet>.]

# Summary

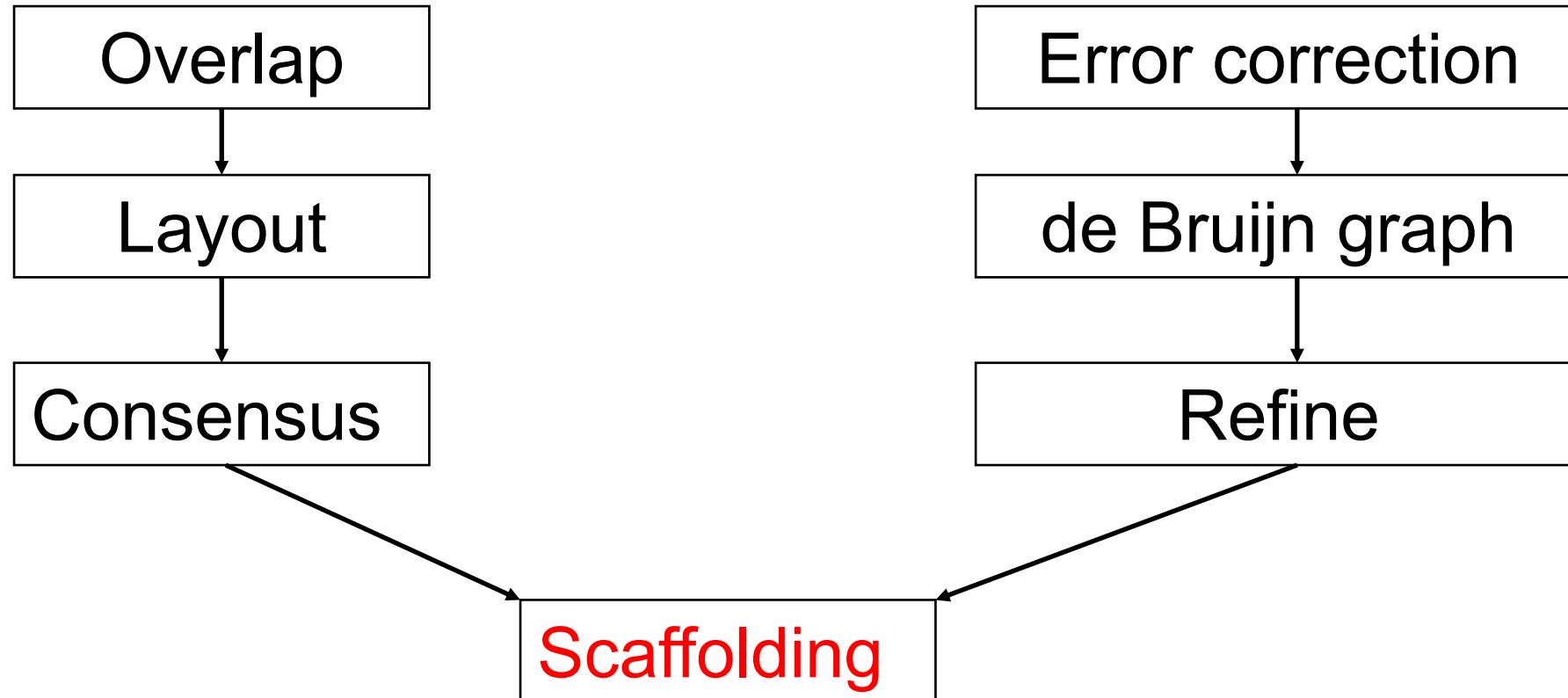
Error correction will **definitely** improve assembly

But, for it to work well:

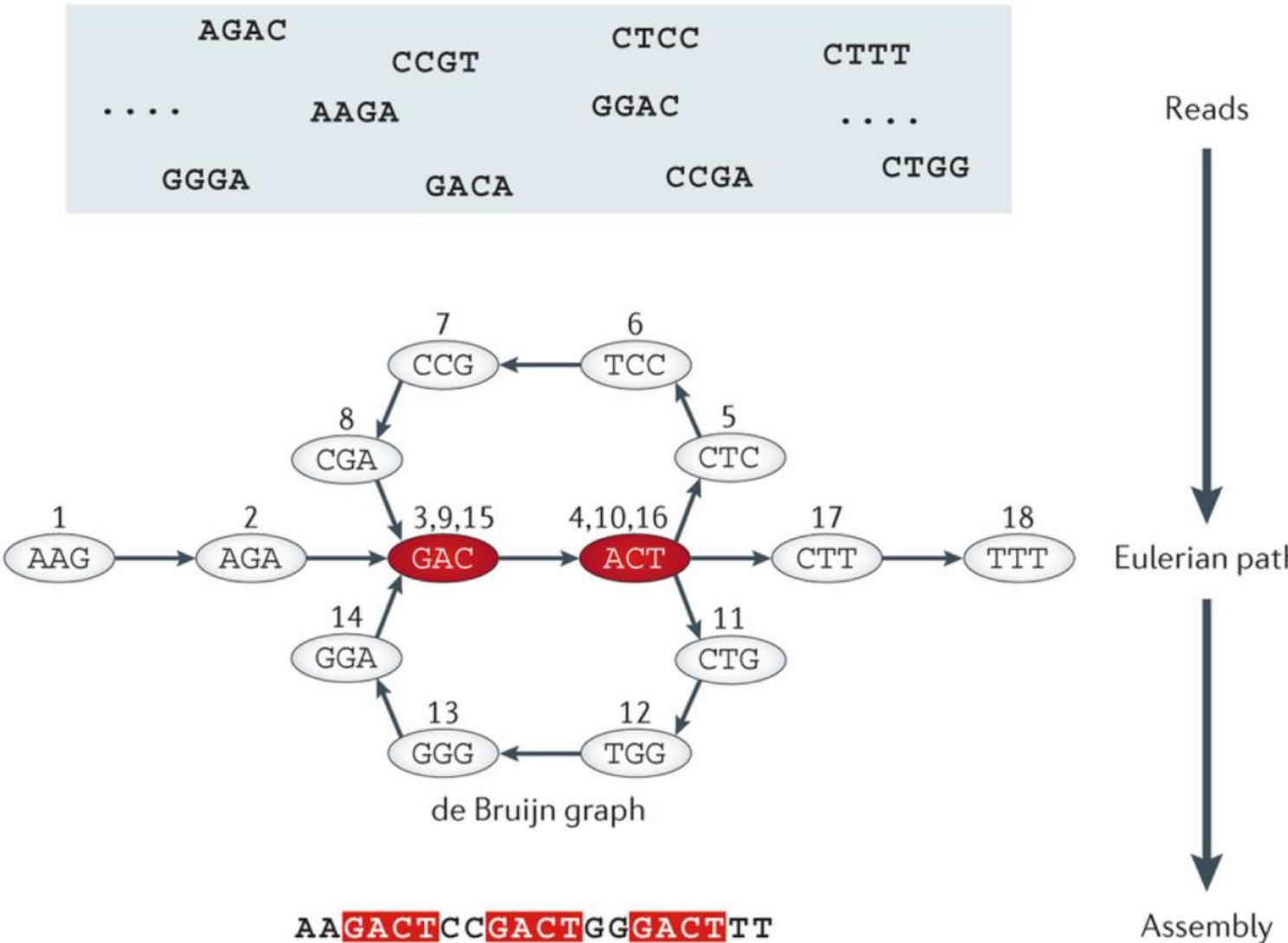
- Sequenced coverage should be high enough

- Choose kmer wisely otherwise we can't distinguish erroneous kmer from frequent kmers

# OLC and DBG assemblers



# Summary I



# Summary II

a

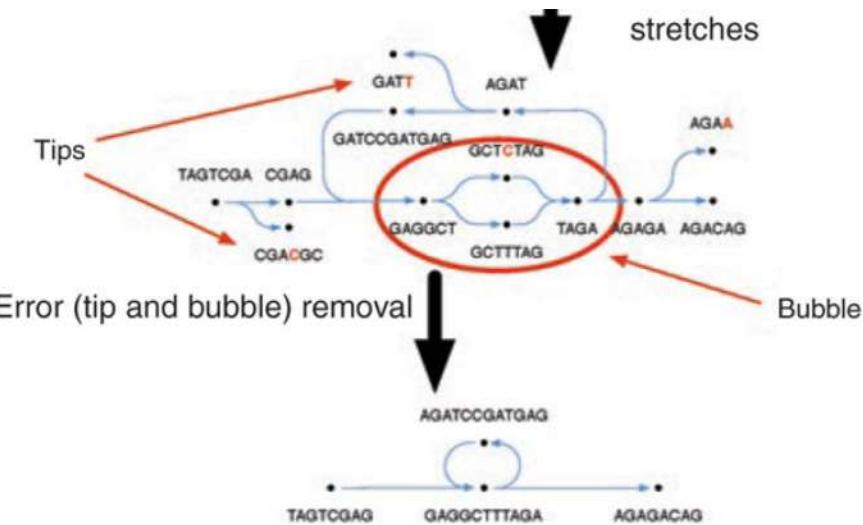
TAGTCGAGGCTTAGATCCGATGAGGCTTAGAGACAG  
AGTCGAG CTTAGA CGATGAG CTTAGA  
GTCGGG TTAGATC ATGAGGC GAGACAG  
GAGGCTC ATCCGAT AGGCTT GAGACAG  
AGTCGAG TAGATCC ATGAGGC TAGAGAA  
TAGTCGA CTTAGA CCGATGA TTAGAGA  
CGAGGCT AGATCCG TGAGGCT AGAGACA  
TAGTCGA GCTTTAG TCCGATG GCTCTAG  
TCGACGC GATCCGA GAGGCTT AGAGACA  
TAGTCGA TTAGATC GATGAGG TTTAGAG  
GTCGAGG TCTAGAT ATGAGGC TAGAGAC  
AGGCTT ATCCGAT AGGCTT GAGACAG  
AGTCGAG TTAGATT ATGAGGC AGAGACA  
GGCTTA TCCGATG TTTAGAG  
CGAGGCT TAGATCC TGAGGCT GAGACAG  
AGTCGAG TTAGATC ATGAGGC TTAGAGA  
GAGGCTT GATCCGA GAGGCTT GAGACAG

1. Sequencing  
(for example, Solexa or 454)

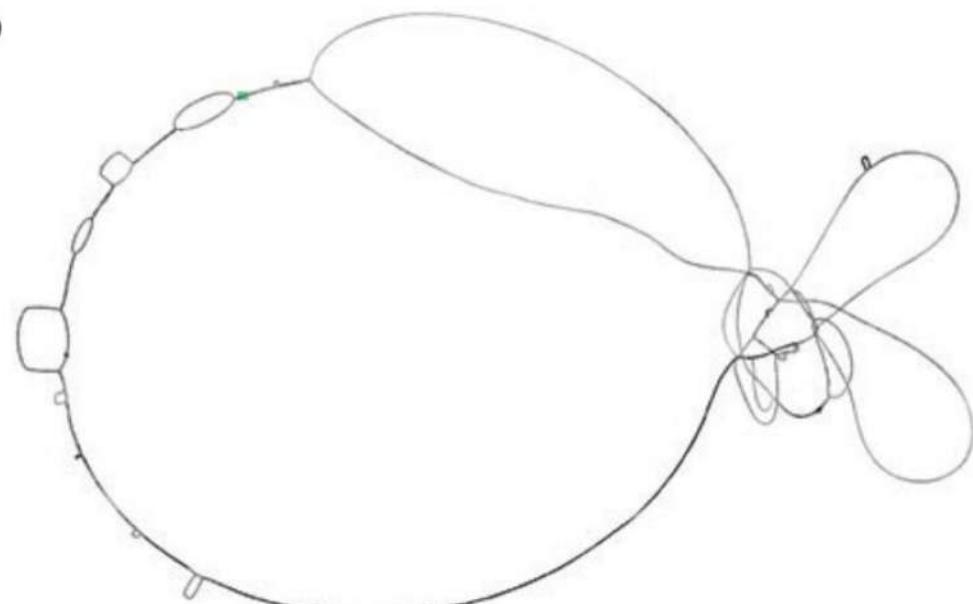
2. Hashing

TAGT AGTC GTCG TCGA (3x) (7x) (9x) (10x)  
CGAG GAGG AGGC GGCT GGCT (8x) (16x) (16x) (16x) (11x)  
CGAC GACG ACAC (1x) (1x) (1x)  
Linear stretches  
GATT (1x) (9x) (8x) (5x) (6x) (7x) (7x) (7x) (8x) (8x)  
ATGA (8x) CGAT (6x) CGA (7x) TCGG (7x) ATCC (7x) GATC (8x) AGAT (8x)  
GCTC (2x) CTCT (1x) TCTA (2x) CTAG (2x)  
GCTT (8x) CTTT (8x) TTTA (8x) TTAG (12x)  
TAGA (16x) AGAG (9x) GAGA (12x) AGAC (9x) GACA (8x) ACAG (5x)

3. Simplification of linear stretches



b



# Scaffolding

OLC and DBG attempt to construct longest and most accurate **contigs** (contiguous stretch of assembled bases)

Scaffolding is to order and orient contigs with respect to each other

Various data types:

Paired ends / Mate pairs

Genetic map

Additional long range information

# Scaffolding: Paired end sequencing



Because of technology limitation (usually ~150bp at each end), whole fragment is not sequenced. But the distance between two mates equals to length of fragment (**insert size**)

# Scaffolding: Paired end sequencing

- DNA fragment (200-800 bp)



- Single end



- Paired end (up to 800 bp span)



- Mate pair (up to 40 kbp span)



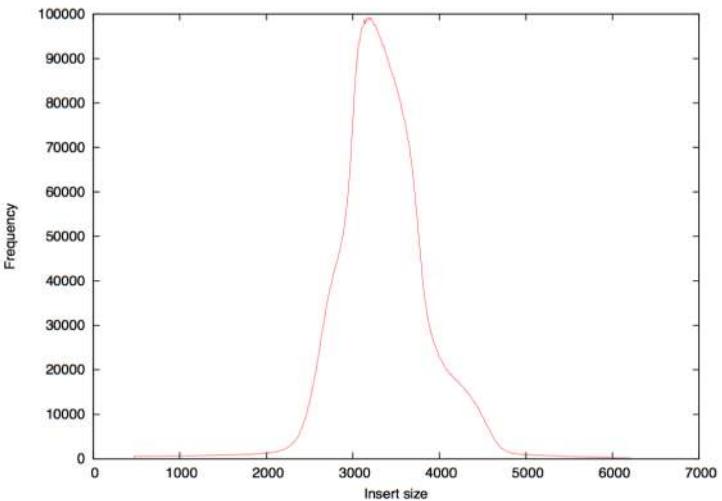
# Examples

How to check insert size?

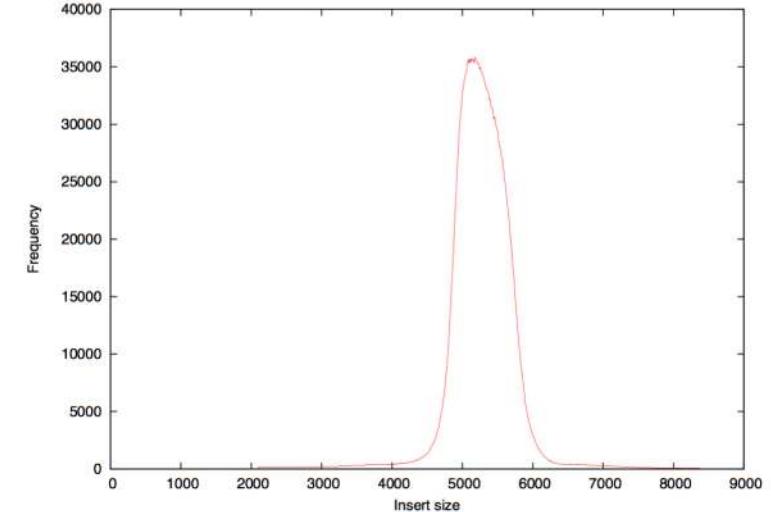
Remap the data back to the assembly

Problem can arise in larger insert sizes

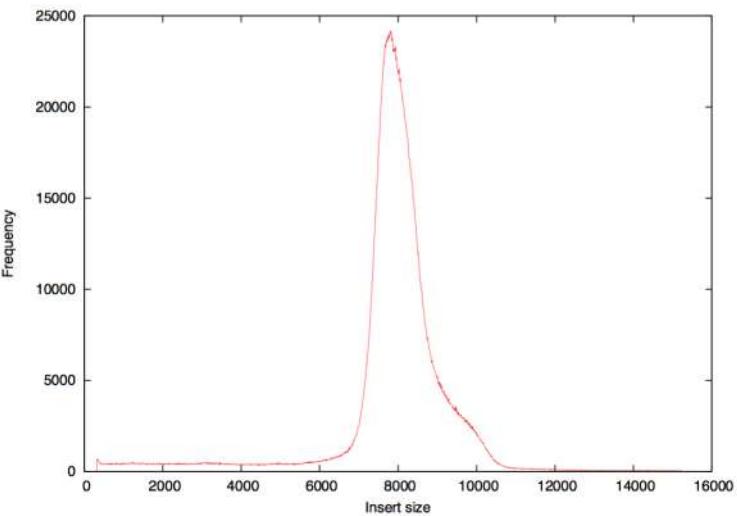
3kb



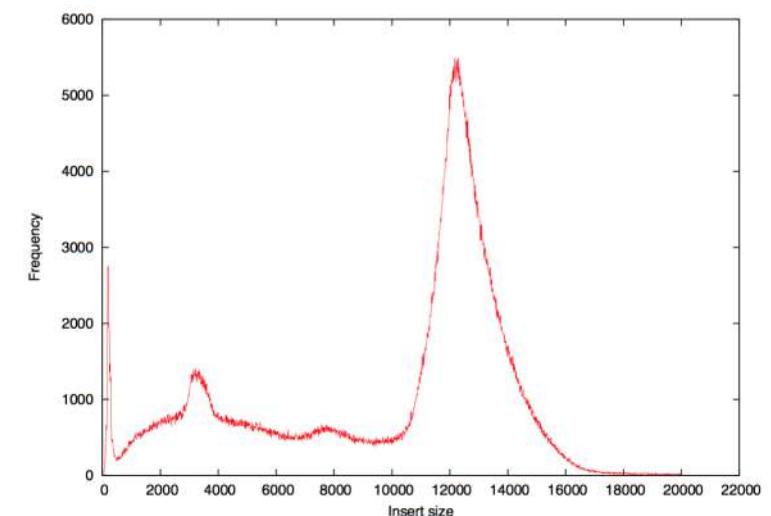
5kb



8kb



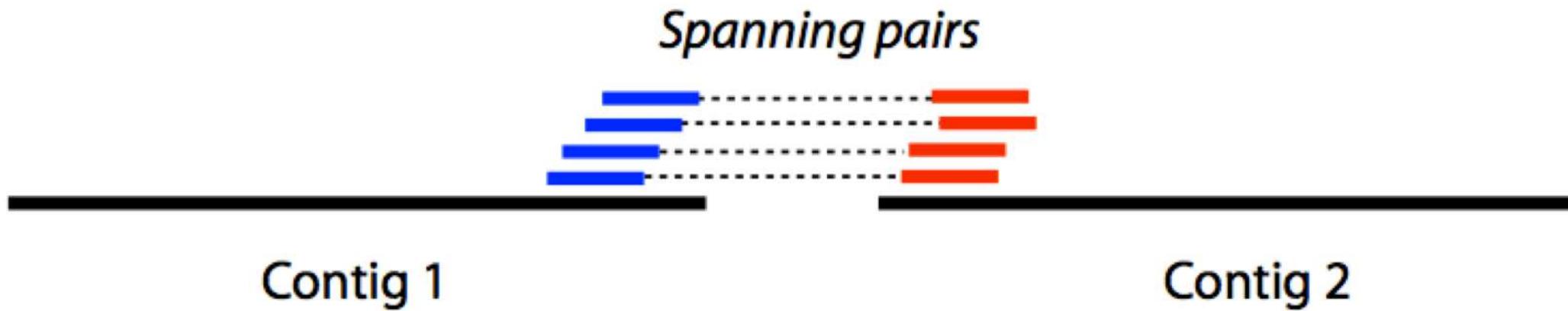
12kb



# Scaffolding

Say we have a collection of pairs and we assemble them as usual

Assembly yields two contigs:

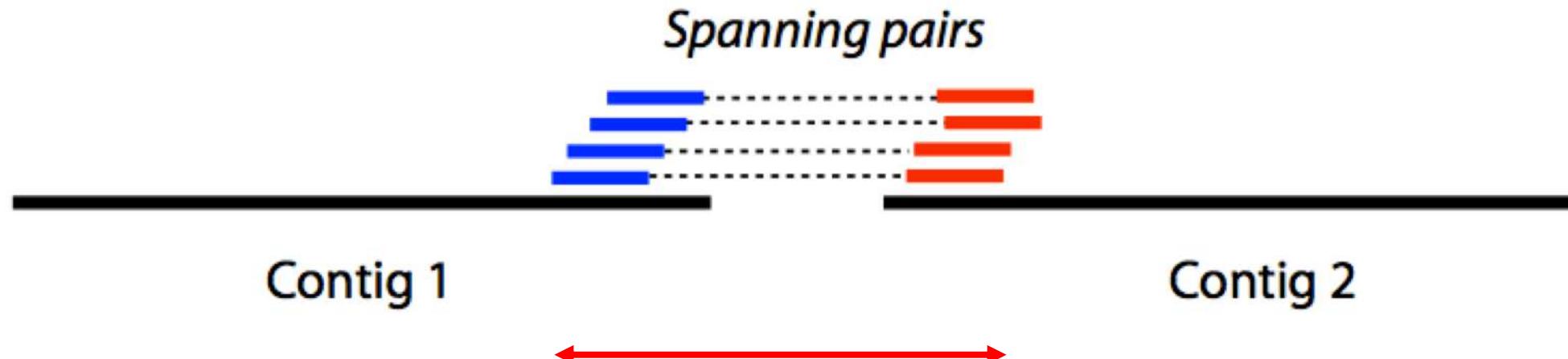


...and we discover that some of the mates at one edge of contig 1 are paired with mates in contig 2

Call these *spanning pairs*

# Scaffolding

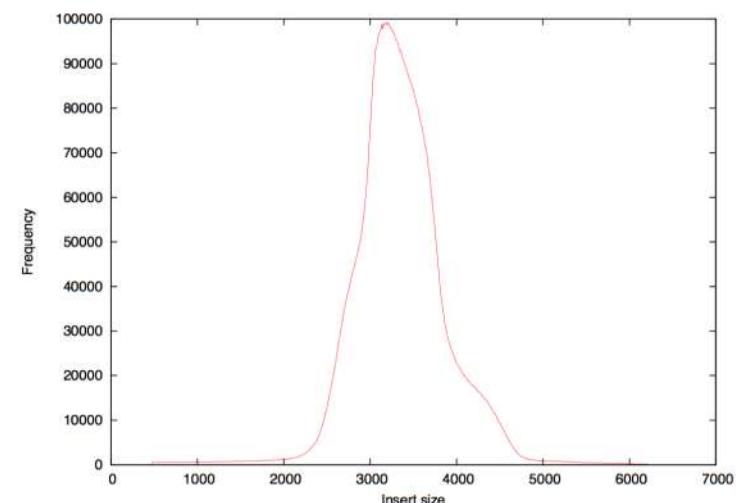
---



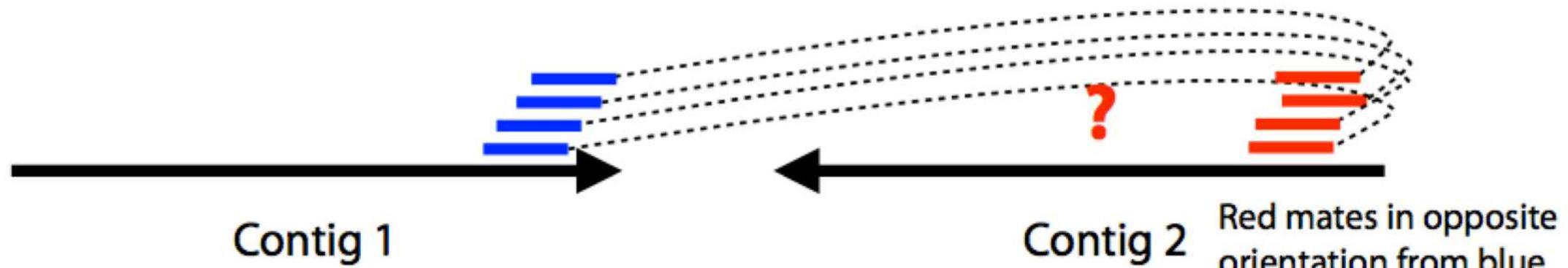
Does the distance equal to the insert size distribution mapped to the rest the genome?

Yes = Accept and Contig1NNNNContig2

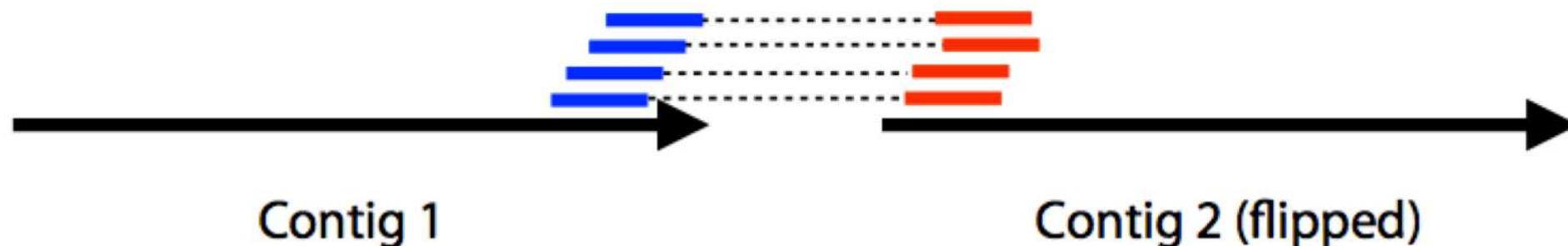
No = Reject



# Scaffolding



What does the picture look like if contigs 1 and 2 are close, but we assembled contig 2 “backwards” (i.e. reverse complemented)

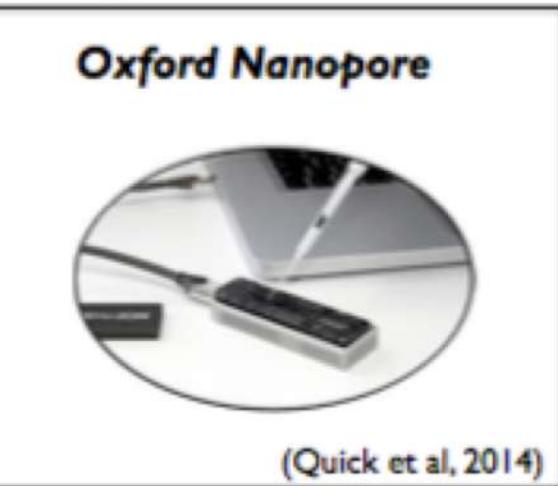
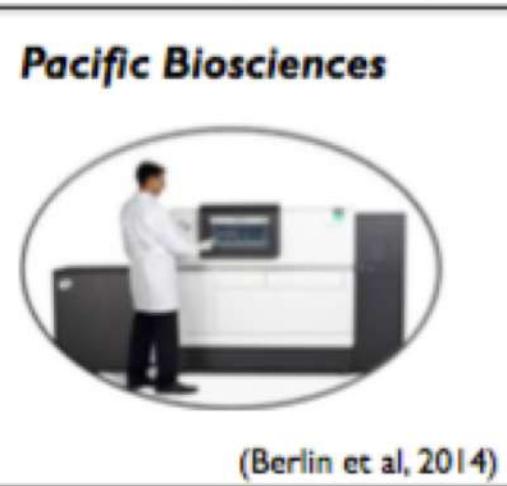
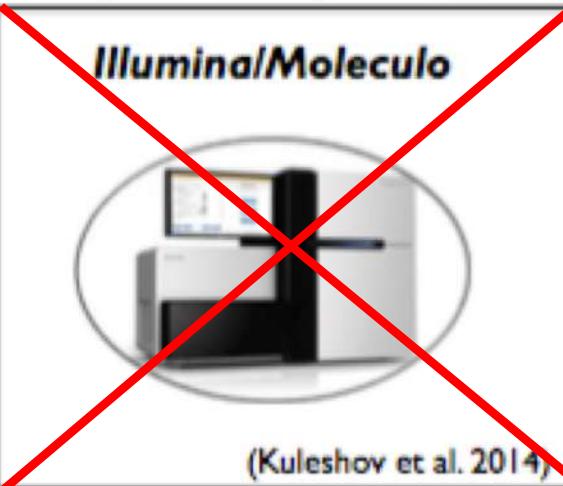


Pairs also tell us about contigs' relative *orientation*

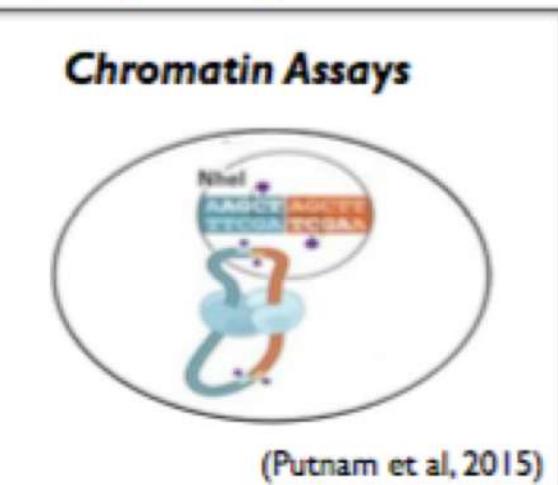
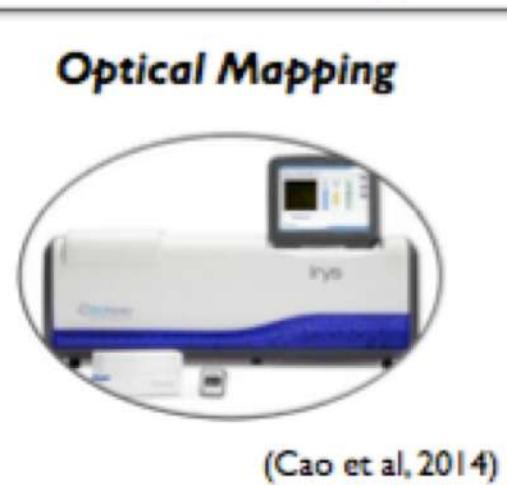
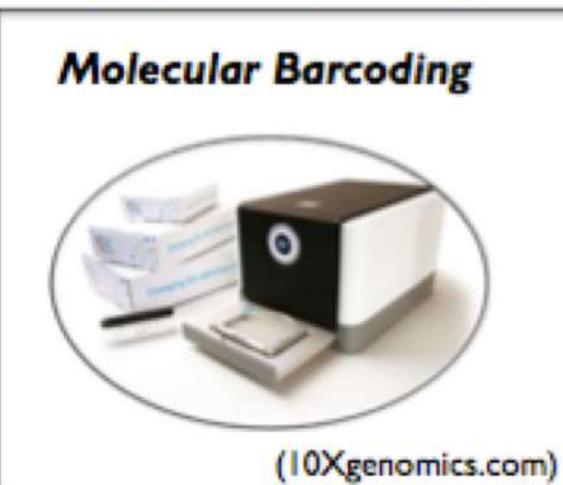
# Long read technologies

# New technology to the rescue!

## Long Read Sequencing: De novo assembly, SV analysis, phasing

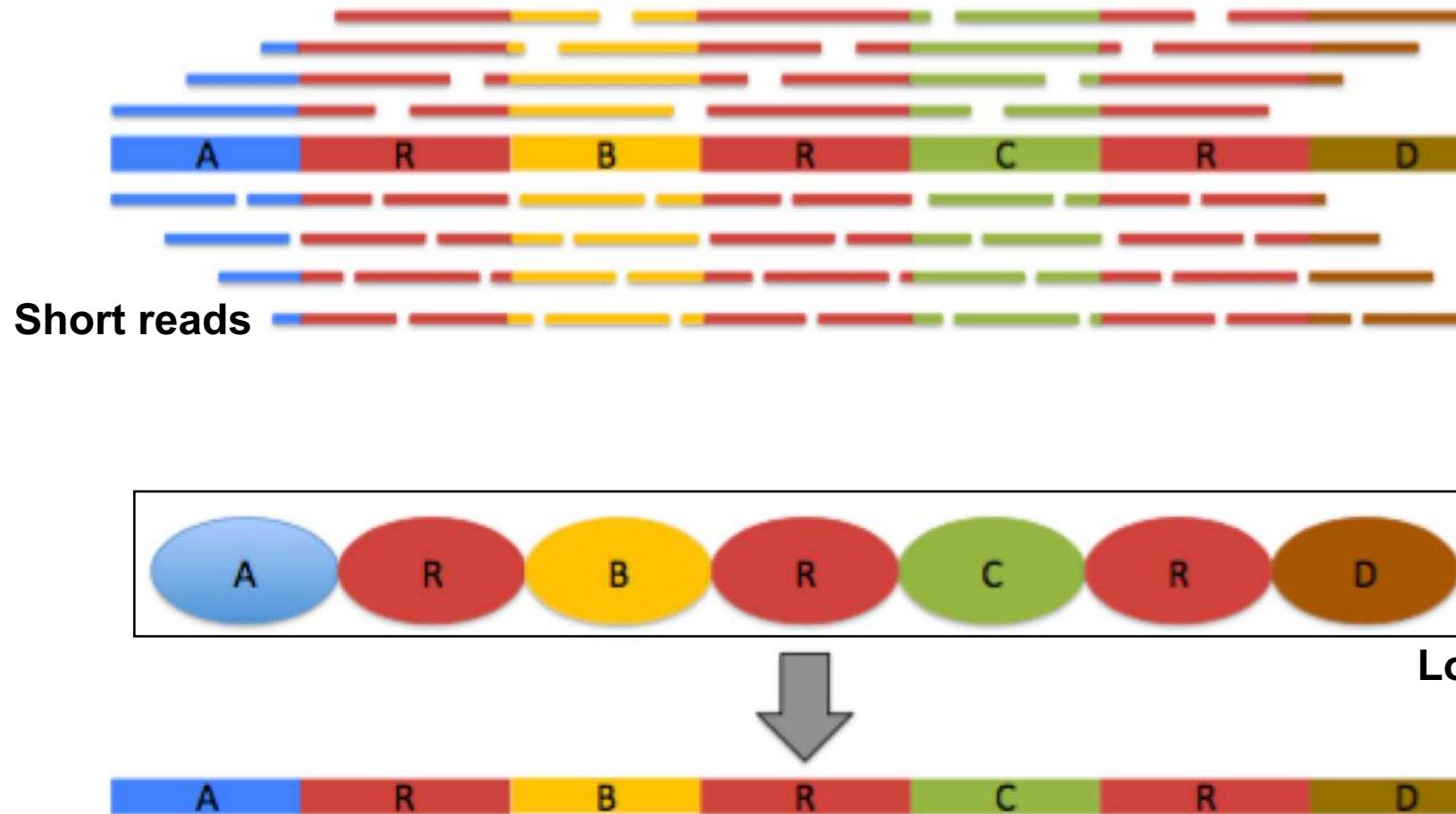


## Long Span Sequencing: Chromosome Scaffolding, SV analysis, phasing



Credit: Michael Schatz

# Now you can with long reads



Credit: Michael Schatz

# Simpler graphs

(a)



K=100  
Contigs=98

(b)



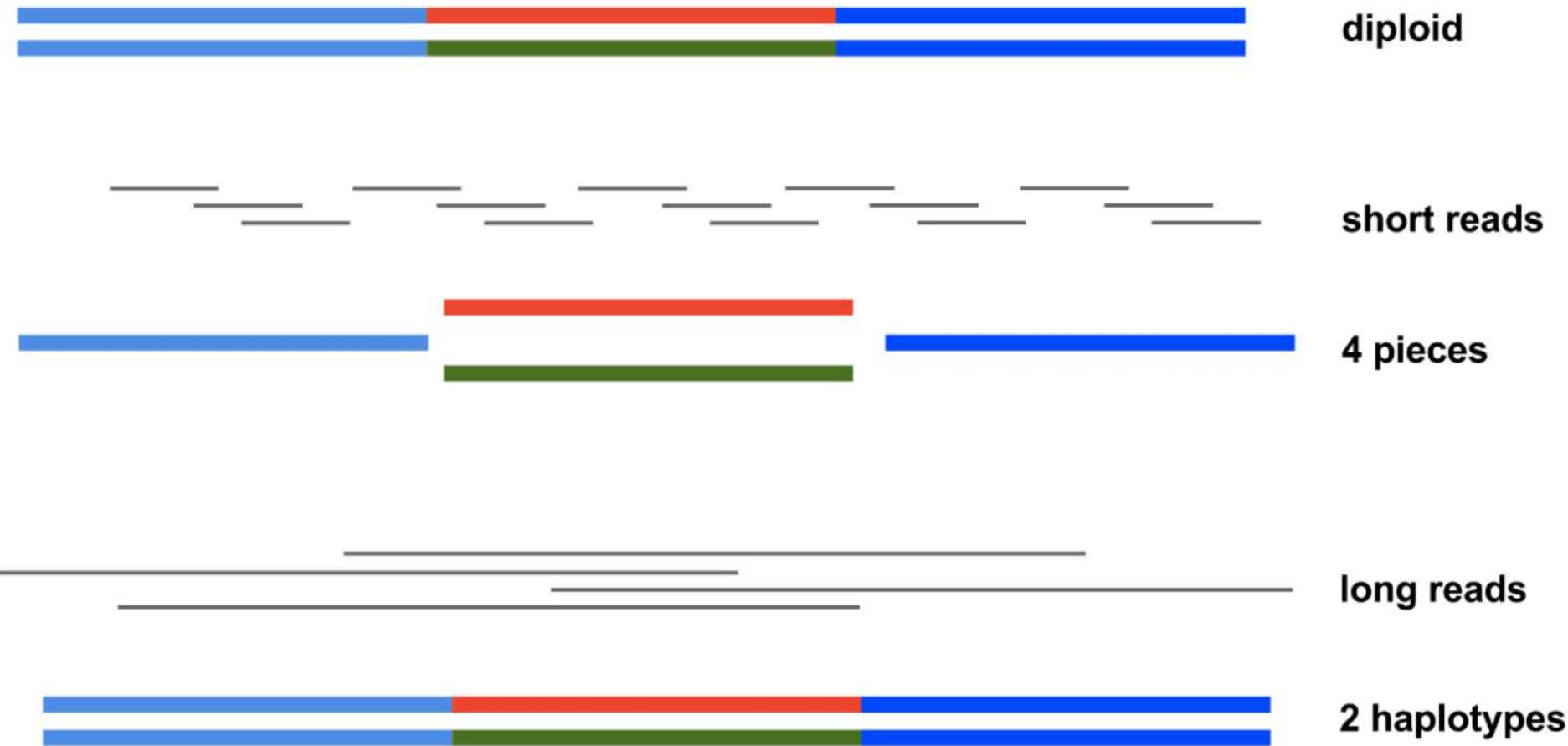
K=1,000  
Contigs=31

(c)



K=5,000  
Contigs=1

# Long reads can also resolve haplotypes (with sufficient coverage)



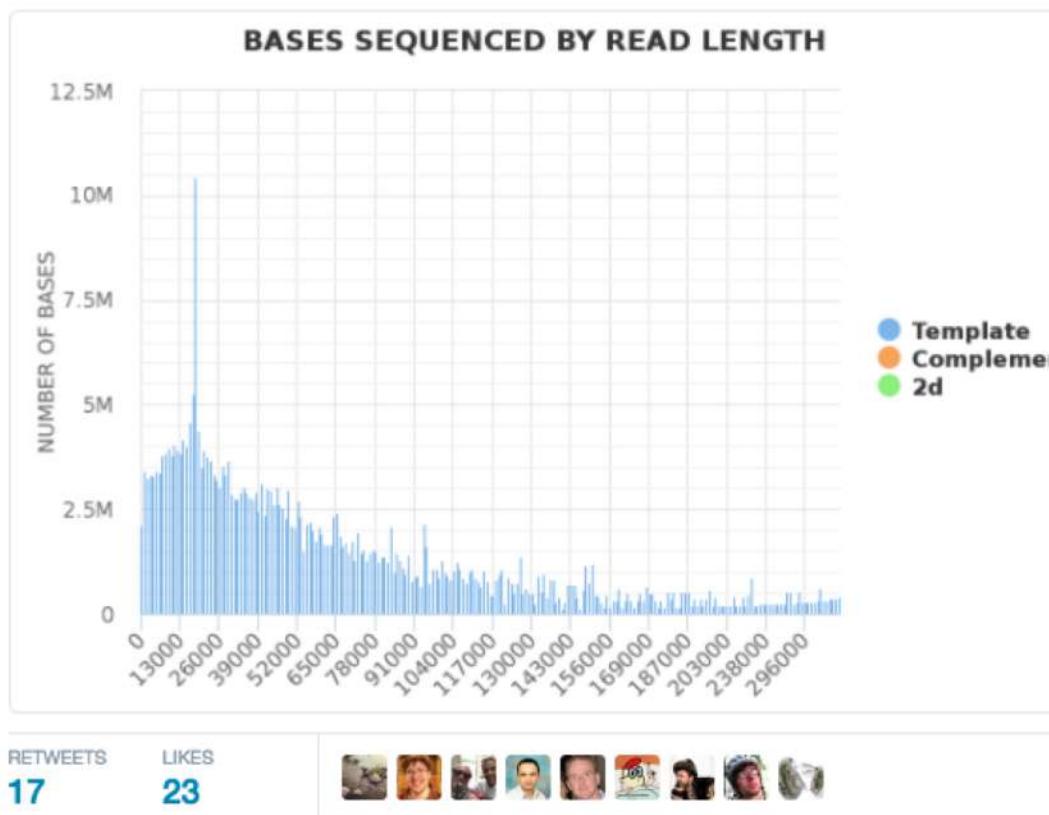


Dominik Handler  
@86Dominik



Follow

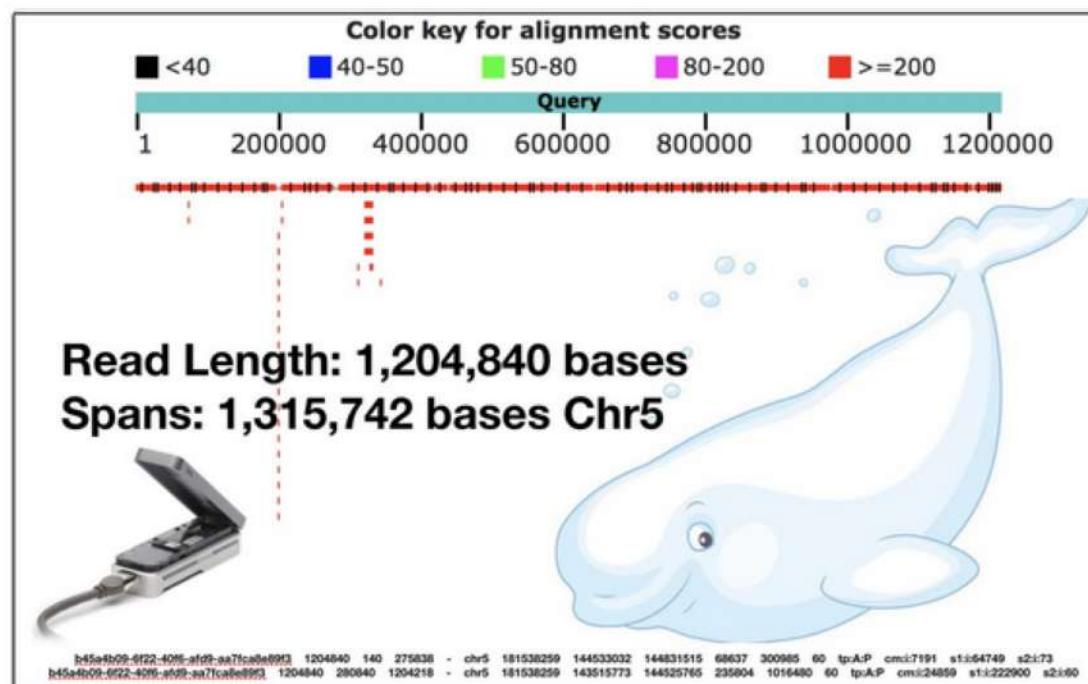
new awesome @nanopore R9-rapid-1D run:  
max=380kb; 33reads>200kb; N50=45kb  
(considering only genome matching)  
graph by MinoTour @mattloose



Matt Loose  
@mattloose

Following

Amazing what you can see on an evenings whale watching with @nanopore - I saw a big one this tonight! Thanks to Nadine, @alexomics and of course @scalene . Protocol tweaks will be available shortly on lab.loman.net/protocols

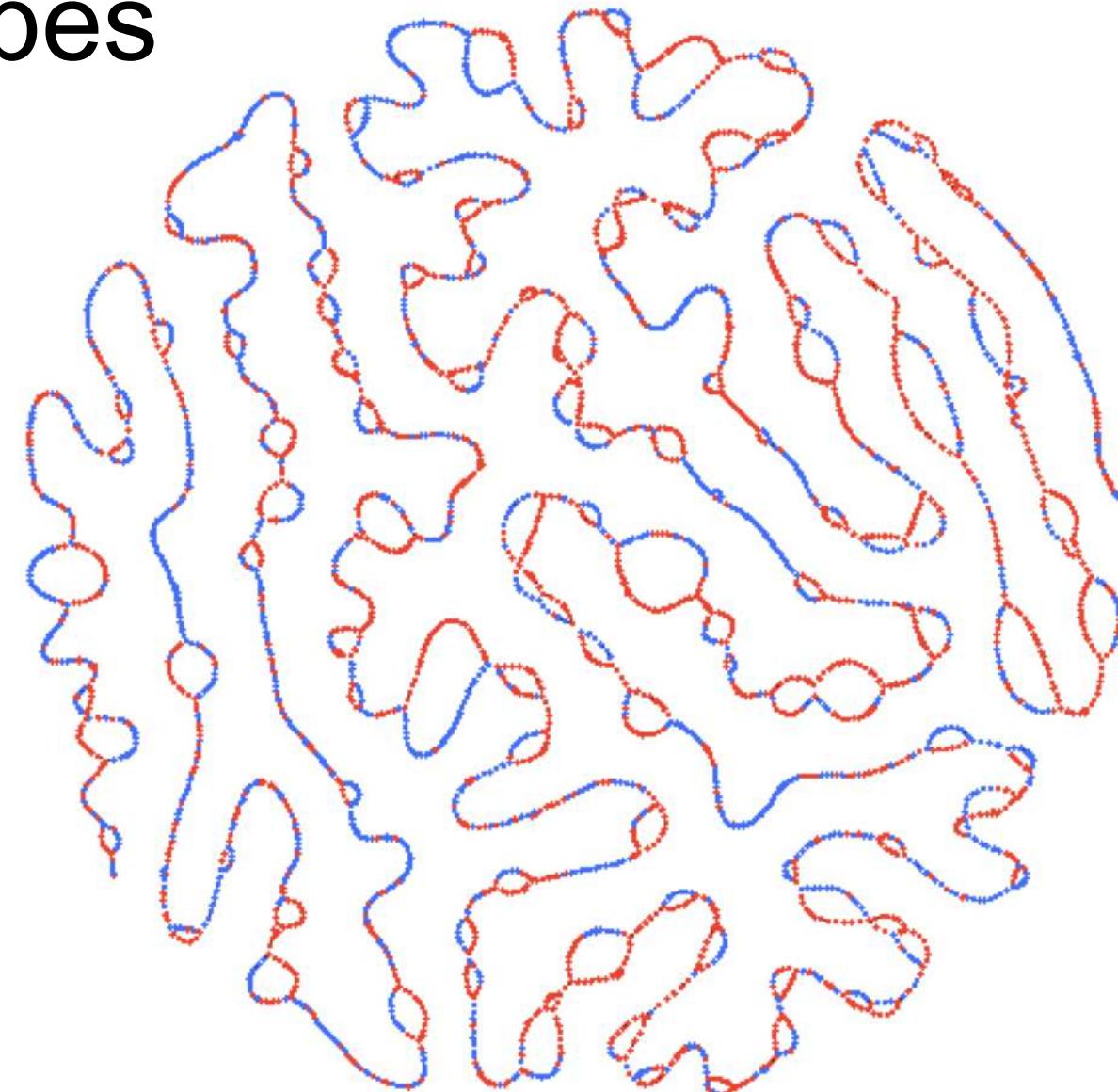


# What you can do with long reads: Resolving haplotypes

Credit: Jason Chin

Two genomes. One  
assembly. Two colors.  
Many bubbles. Game  
against entropy.

<http://t.co/uCPmxCRiZ6>



# Phasing / resolving haplotypes

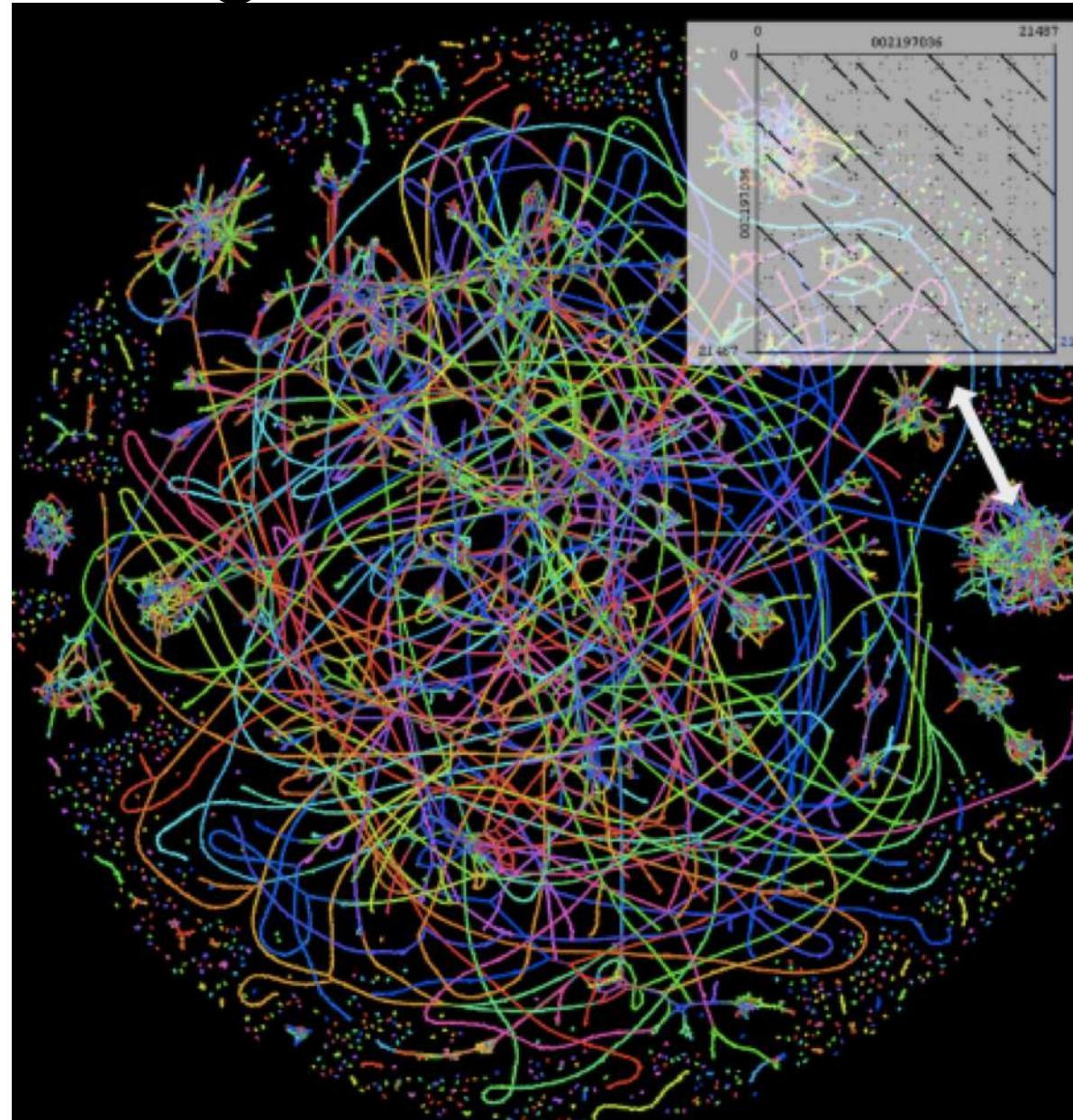
A paper on resolving SV  
And a paper on wheat genomics / ploidy?

# Current limitation of long reads..

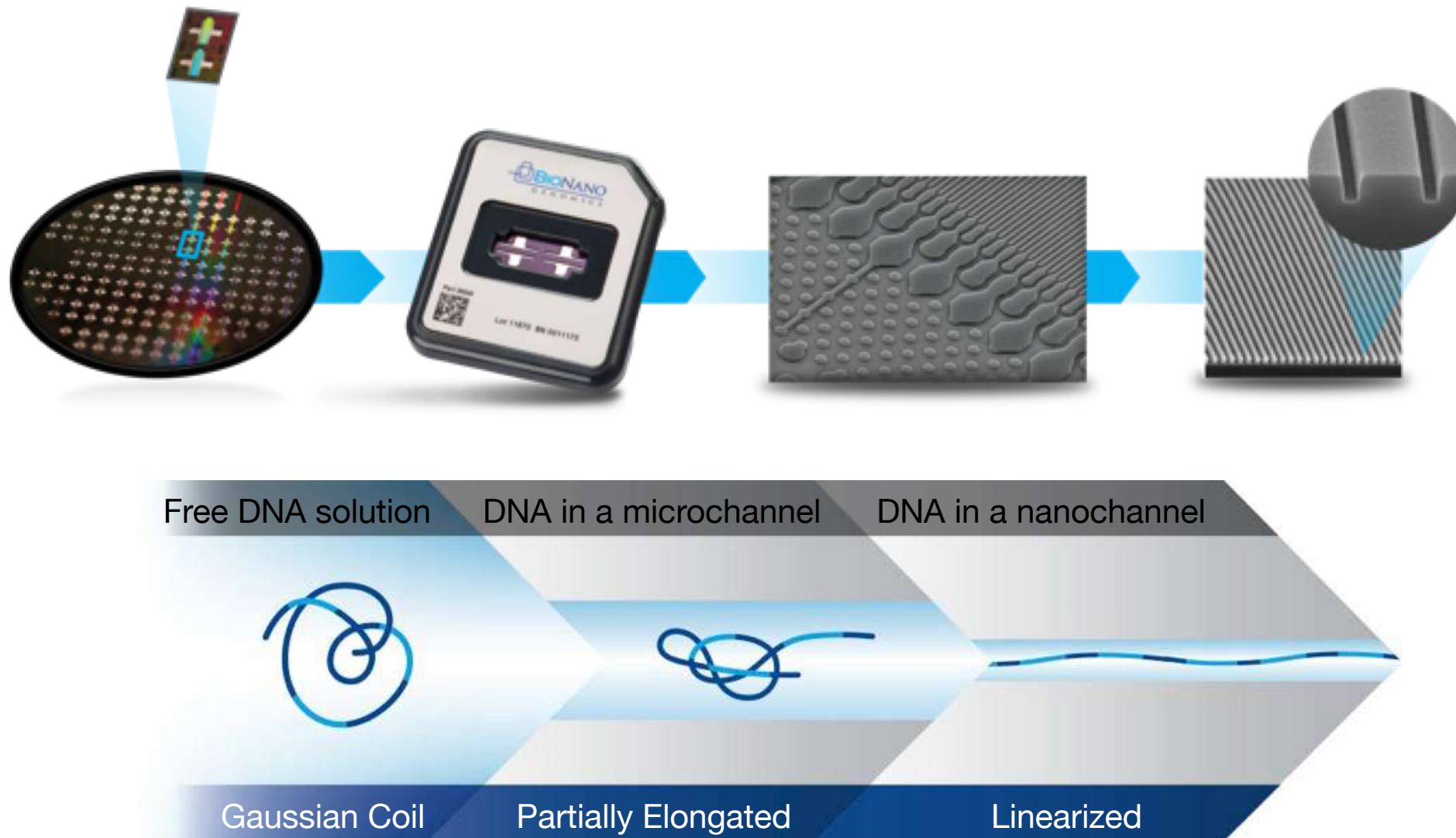
Credit: Jason Chin

What are those blobs in the genome assembly graph?  
Intriguing repeats that have no NCBI blast hit.

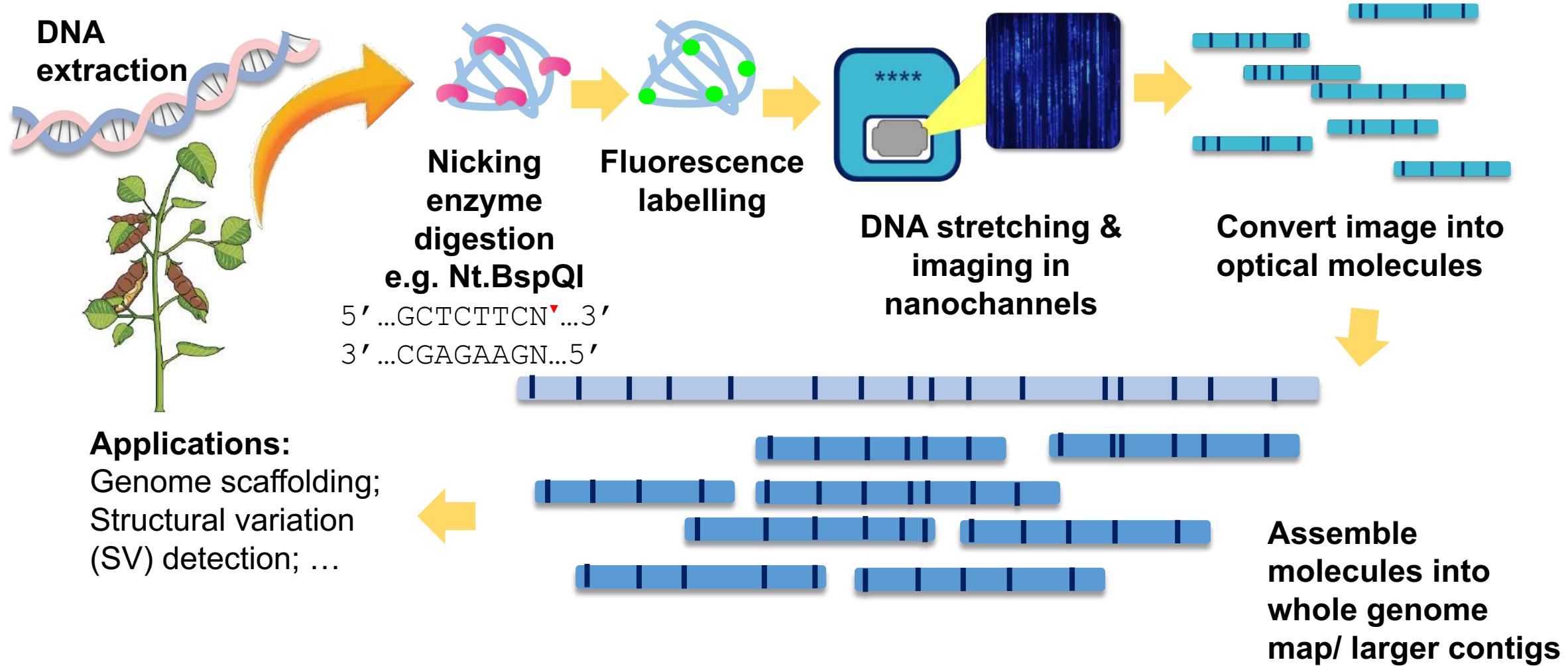
<http://t.co/2y7stBGs4W>



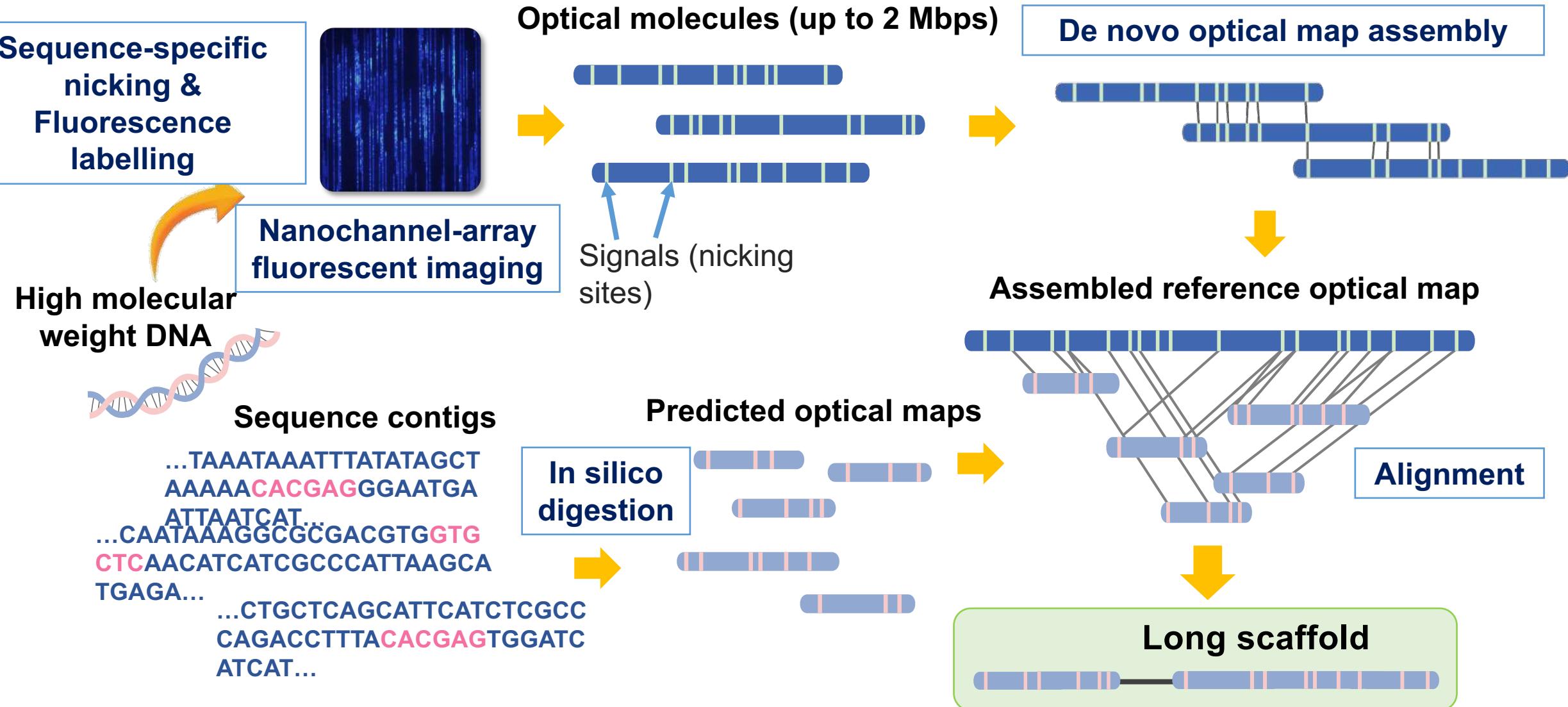
# Optical map: DNA linearized in nanochannel array



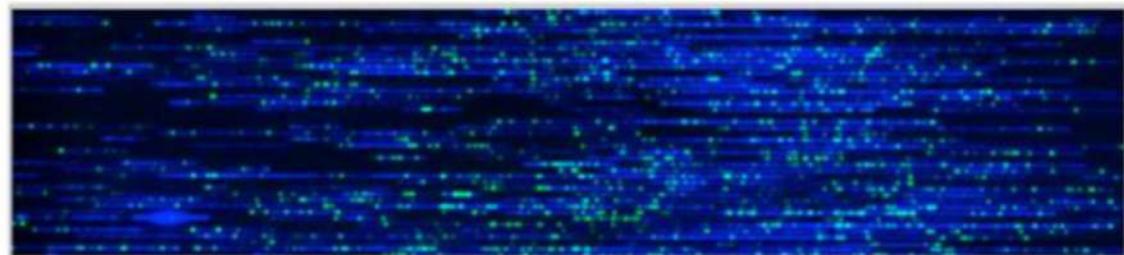
# Workflow of an optical mapping procedure



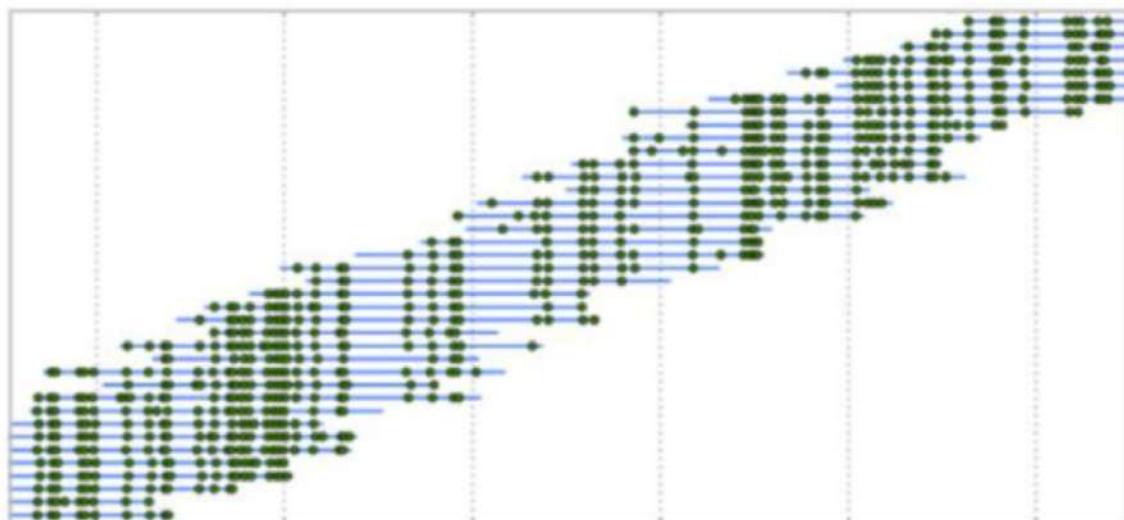
# Optical mapping-assisted scaffolding principles



Raw Image Data



Conversion to  
Molecules

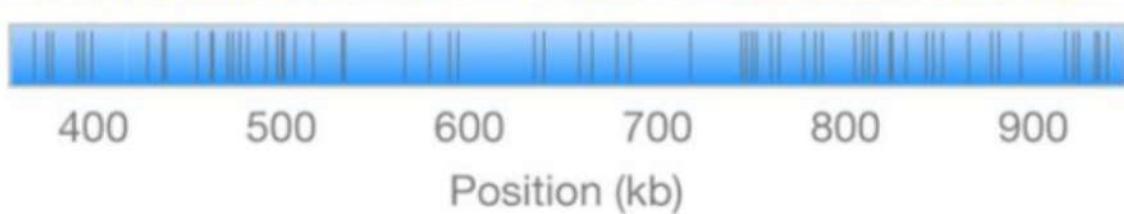


Population of Molecules

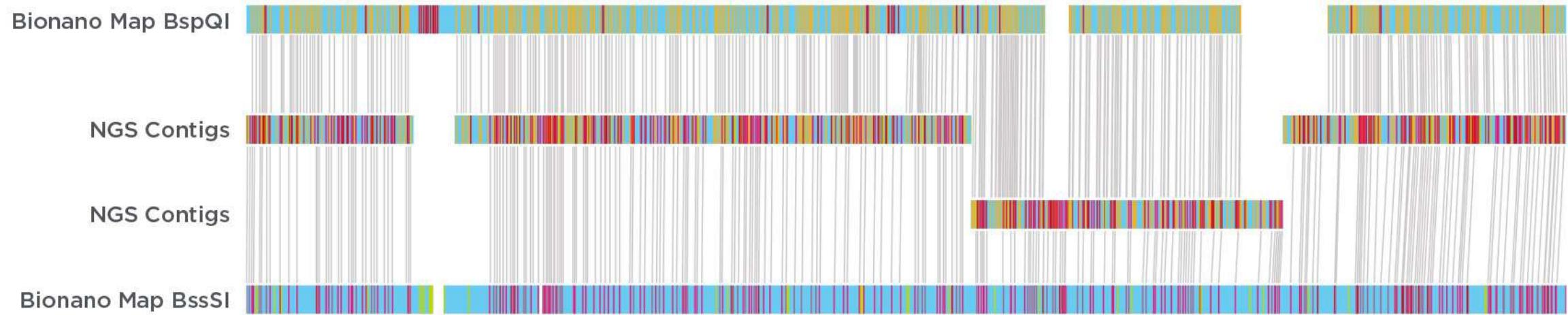
Aggregation &  
Assembly



Consensus Genome Map

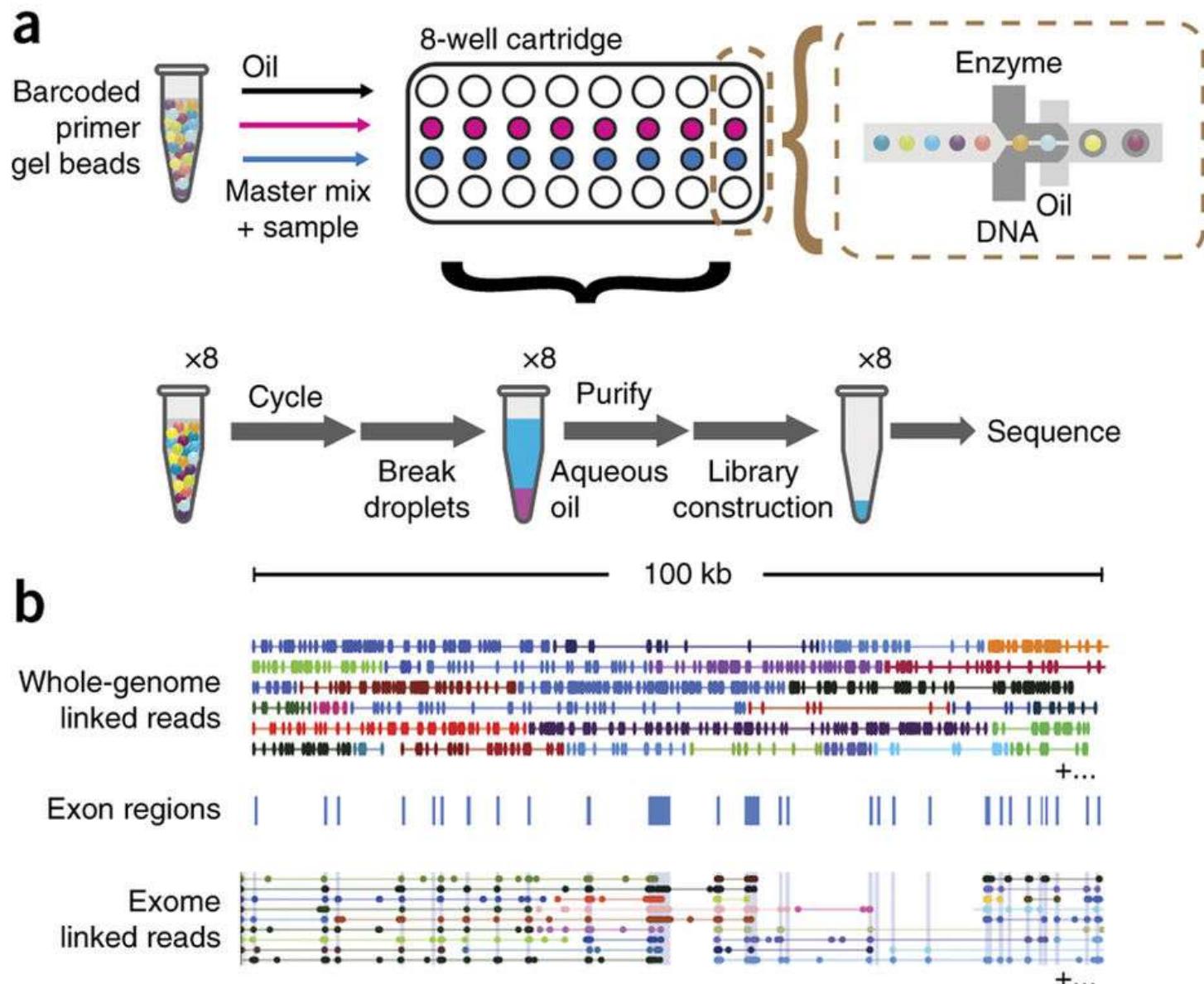


## TWO ENZYME HYBRID SCAFFOLDING



# Haplotyping germline and cancer genomes with high-throughput linked-read sequencing

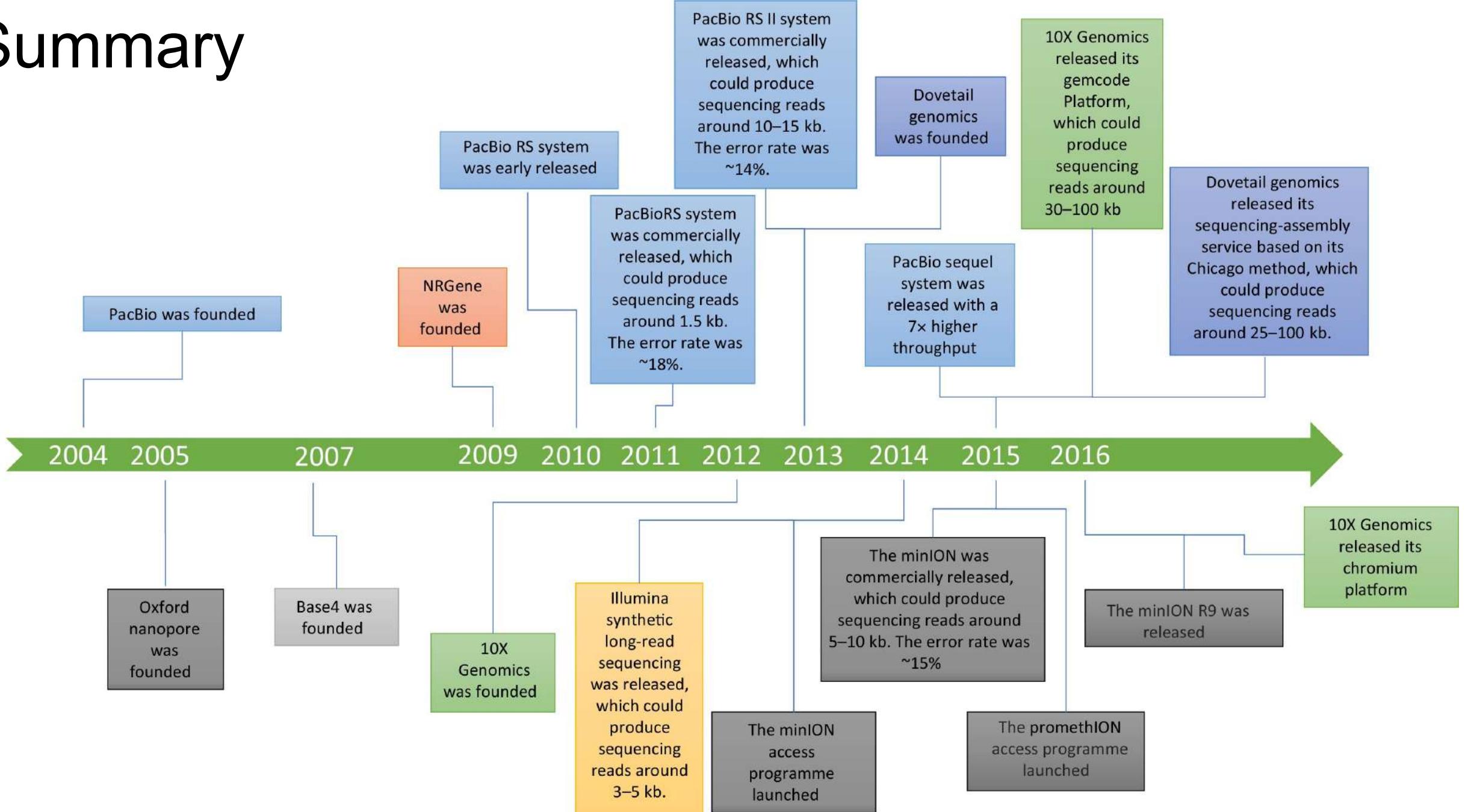
- Long range information from short reads using 14bp barcodes
- Very low input DNA (ng) and 20 mins preparation time
- 1ng of DNA is split across 100,000 Gel coated beads
- Single-cell available



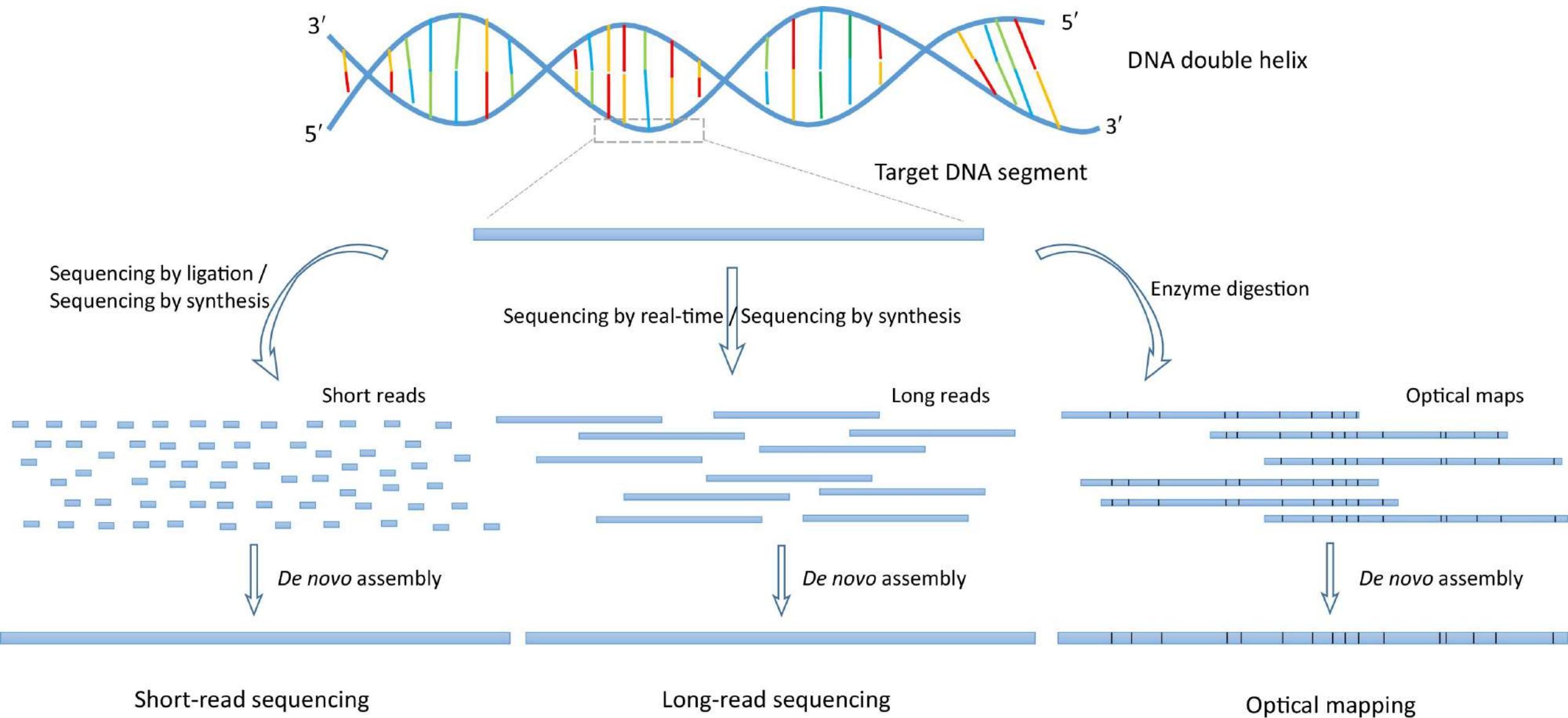
# 10X Genomics

<https://www.youtube.com/watch?v=aUyFzwRFWJQ>

# Summary

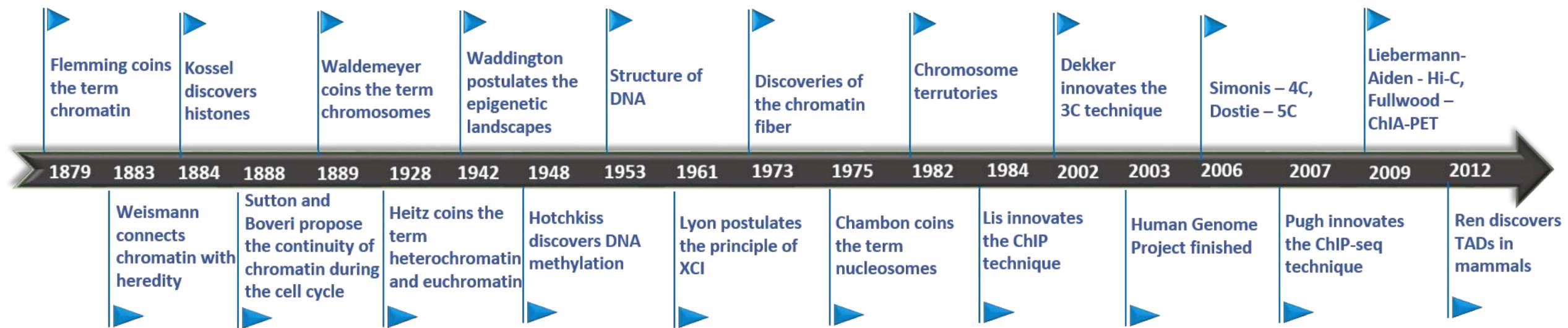


# Summary

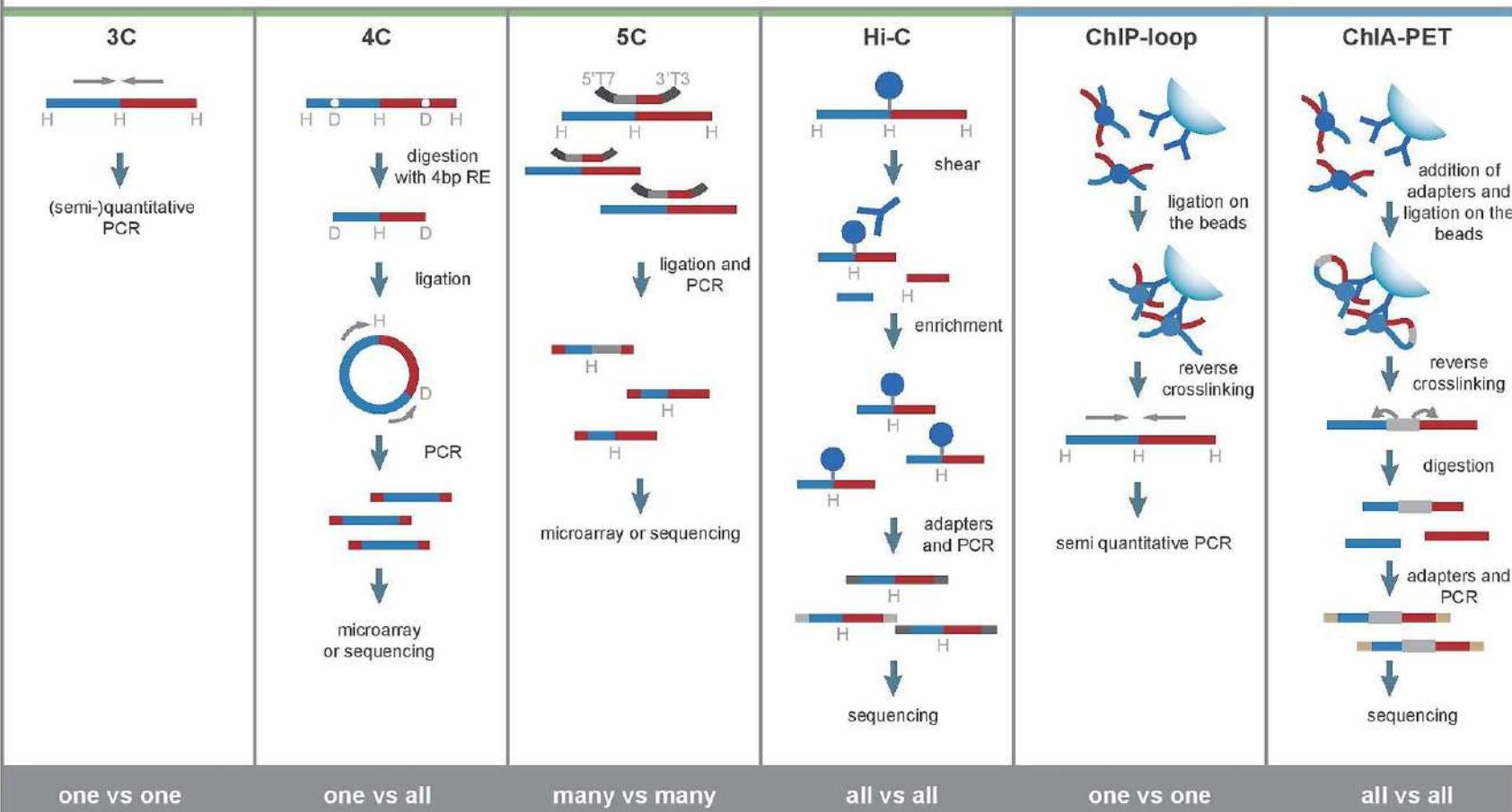
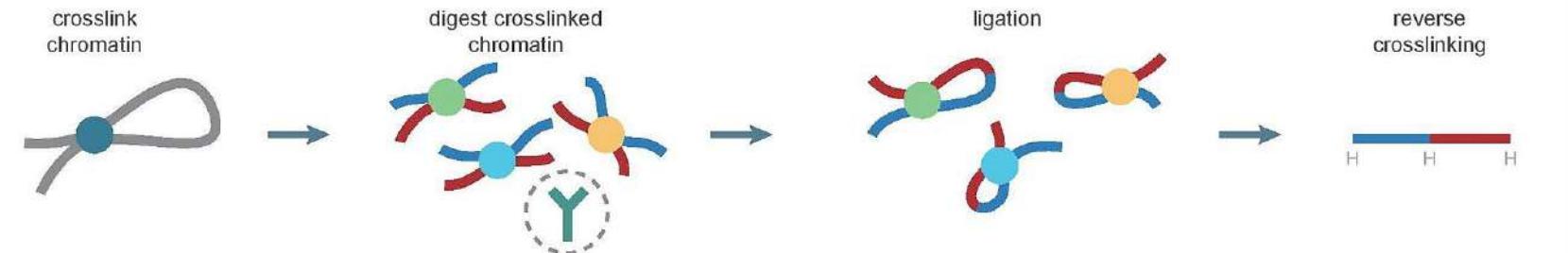


# Chromosome conformation capture

Chromosome conformation capture techniques (often abbreviated to 3C technologies or 3C-based methods) are a set of molecular biology methods used to analyze the spatial organization of chromatin in a cell. These methods quantify the **number of interactions between genomic loci that are nearby in 3-D space**, but may be **separated by many nucleotides** in the linear genome



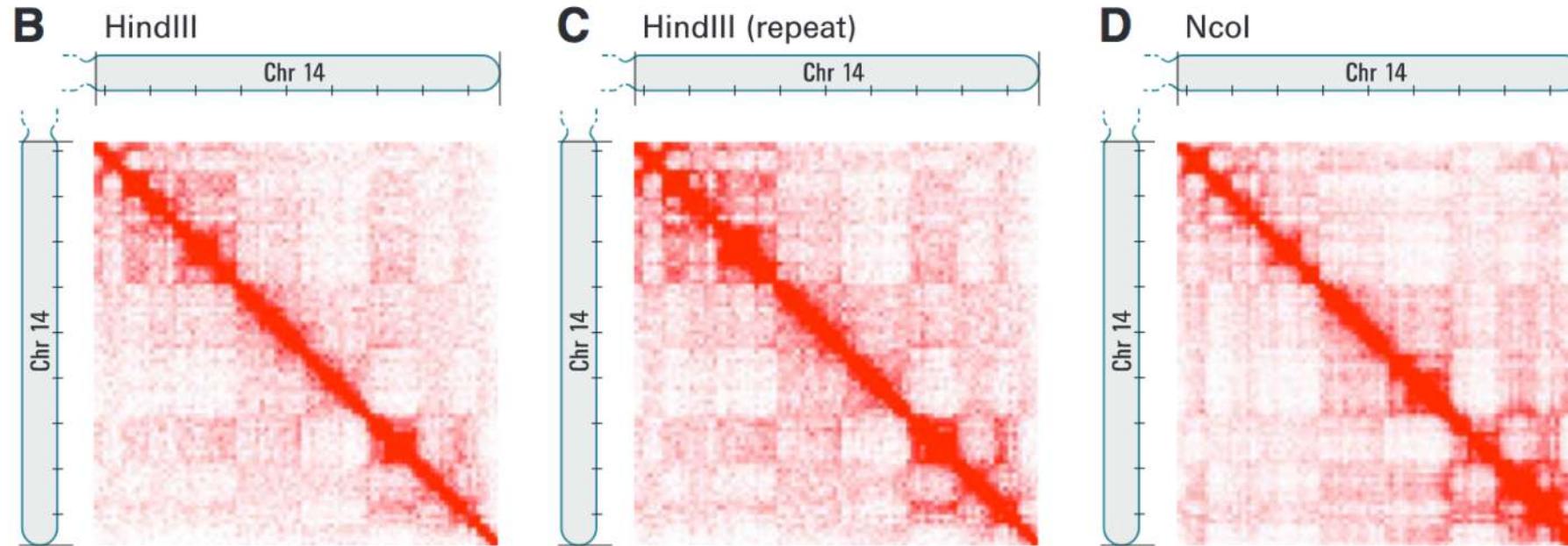
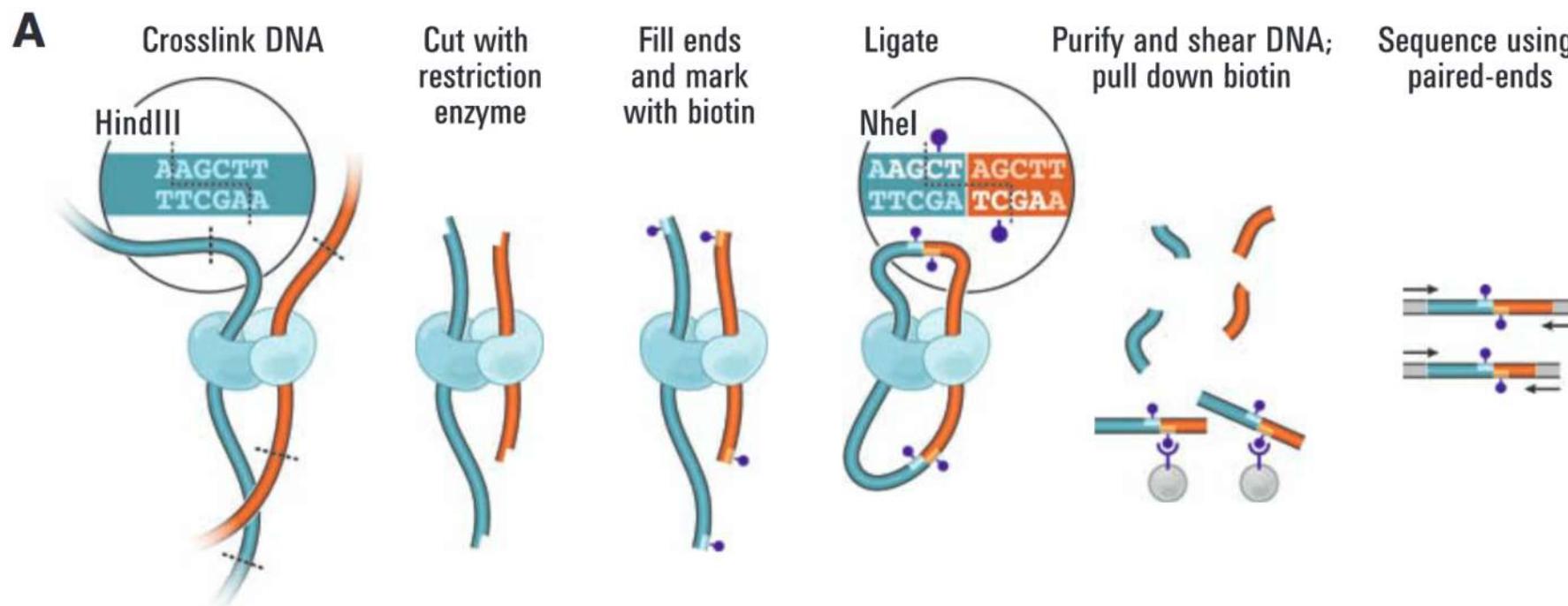
# Chromosome Conformation Technologies

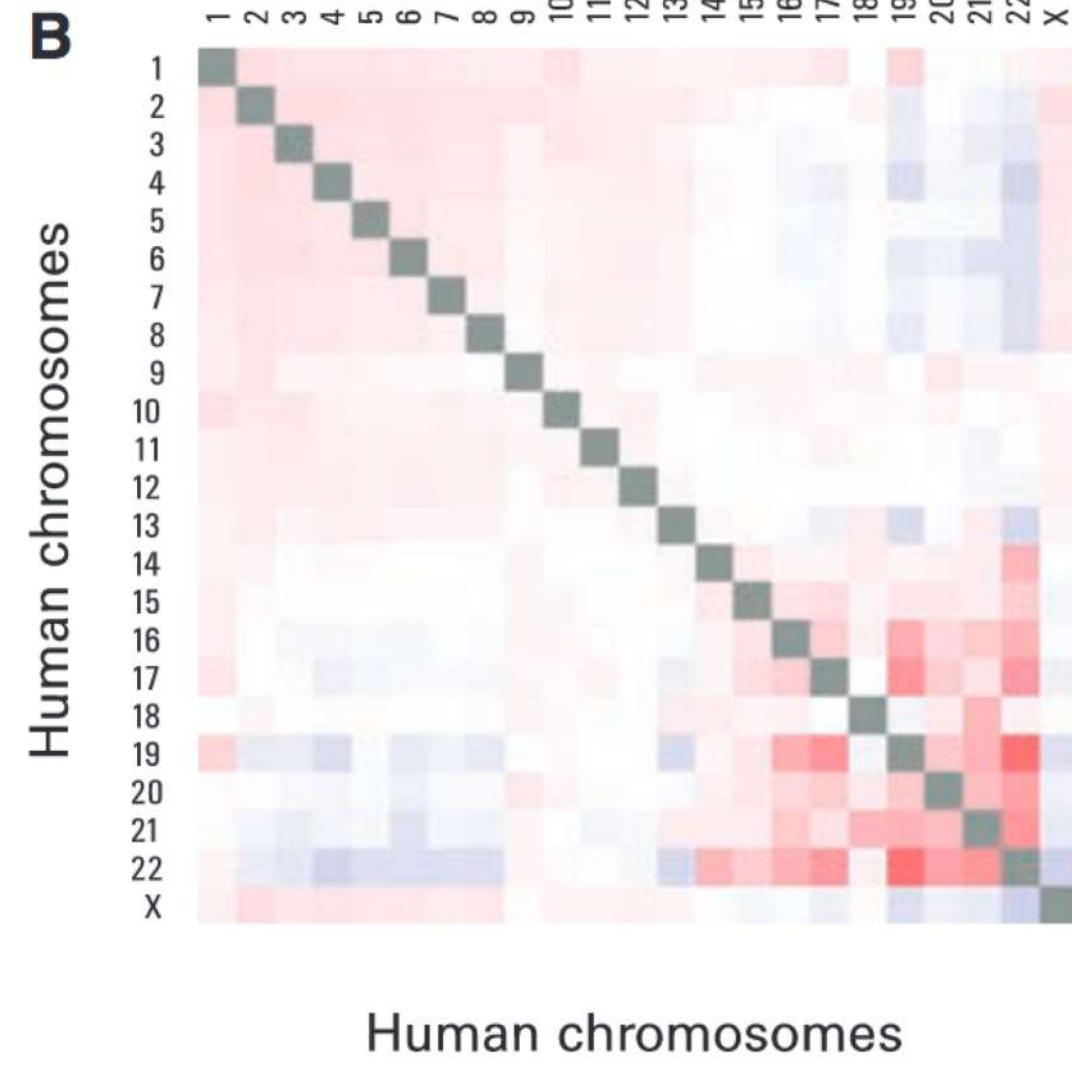
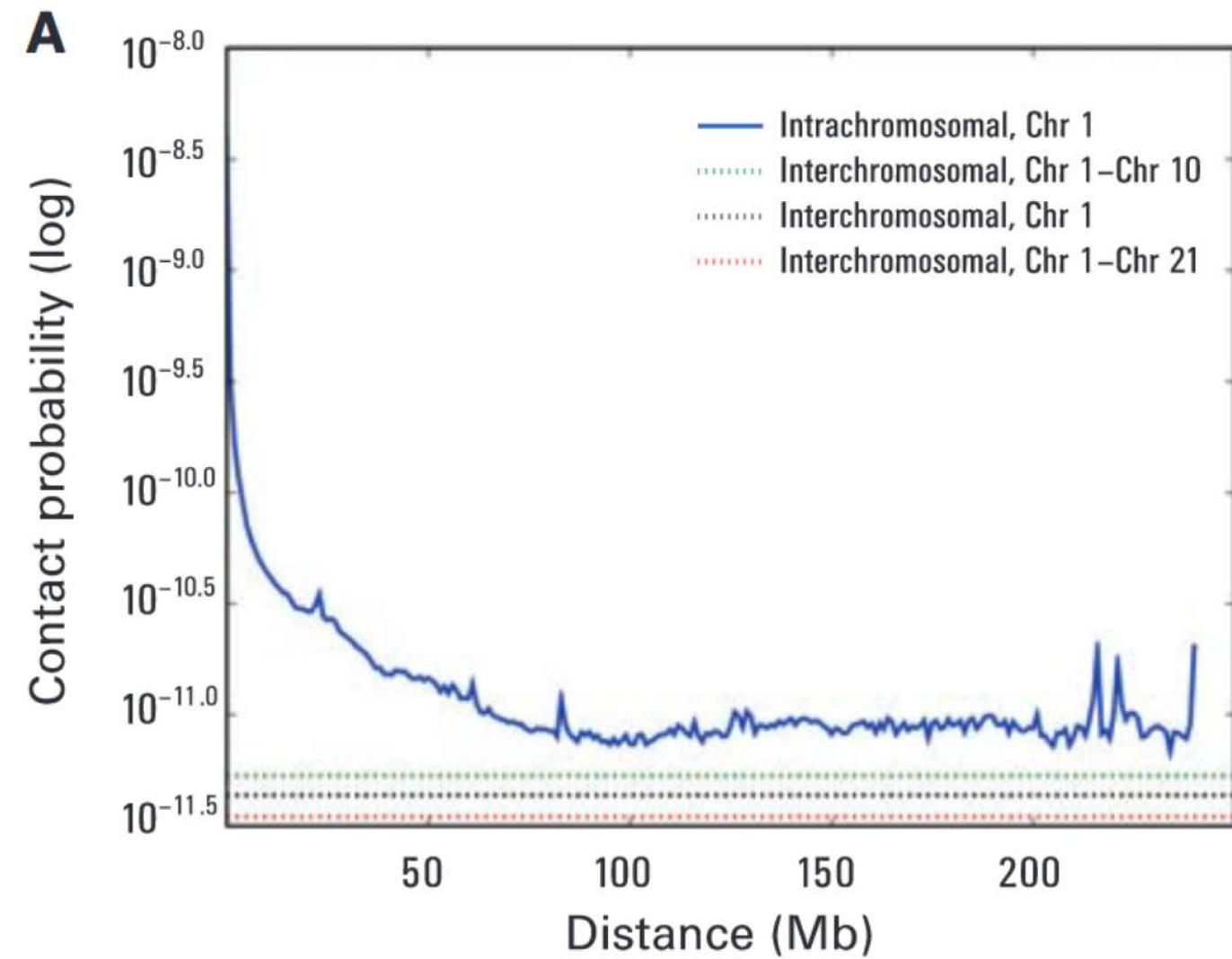


# Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome

Erez Lieberman-Aiden,<sup>1,2,3,4\*</sup> Nynke L. van Berkum,<sup>5\*</sup> Louise Williams,<sup>1</sup> Maxim Imakaev,<sup>2</sup> Tobias Ragoczy,<sup>6,7</sup> Agnes Telling,<sup>6,7</sup> Ido Amit,<sup>1</sup> Bryan R. Lajoie,<sup>5</sup> Peter J. Sabo,<sup>8</sup> Michael O. Dorschner,<sup>8</sup> Richard Sandstrom,<sup>8</sup> Bradley Bernstein,<sup>1,9</sup> M. A. Bender,<sup>10</sup> Mark Groudine,<sup>6,7</sup> Andreas Gnirke,<sup>1</sup> John Stamatoyannopoulos,<sup>8</sup> Leonid A. Mirny,<sup>2,11</sup> Eric S. Lander,<sup>1,12,13†</sup> Job Dekker<sup>5†</sup>

We describe Hi-C, a method that probes the three-dimensional architecture of whole genomes by coupling proximity-based ligation with massively parallel sequencing. We constructed spatial proximity maps of the human genome with Hi-C at a resolution of 1 megabase. These maps confirm the presence of chromosome territories and the spatial proximity of small, gene-rich chromosomes. We identified an additional level of genome organization that is characterized by the spatial segregation of open and closed chromatin to form two genome-wide compartments.





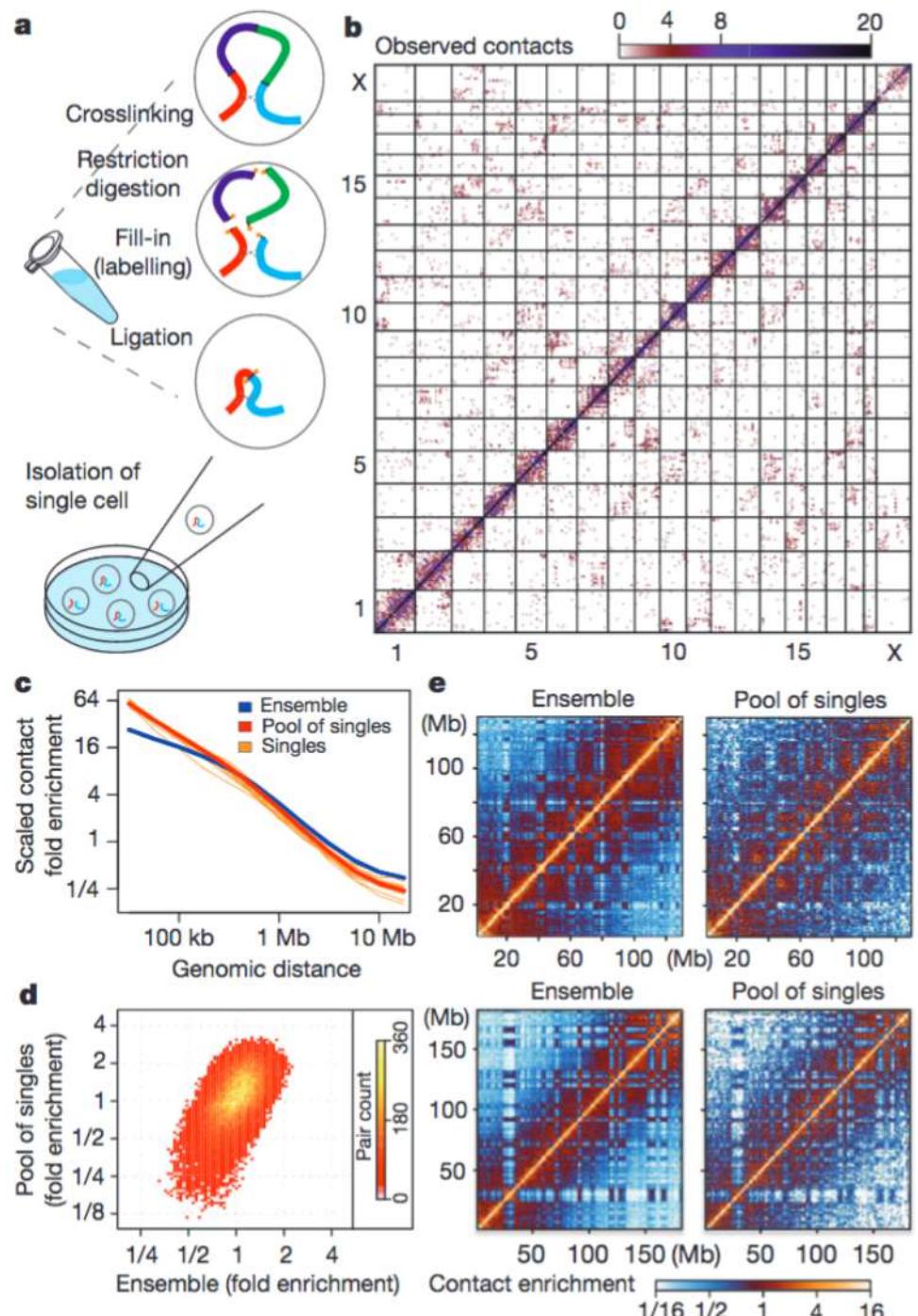
# Hi-C at single cell level

## ARTICLE

doi:10.1038/nature12593

### Single-cell Hi-C reveals cell-to-cell variability in chromosome structure

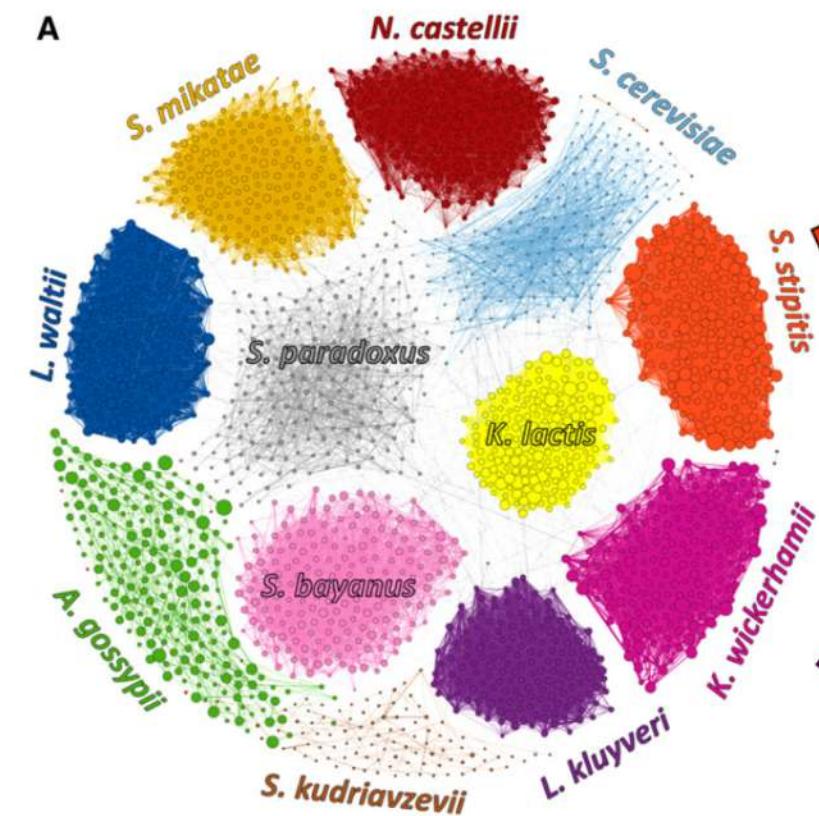
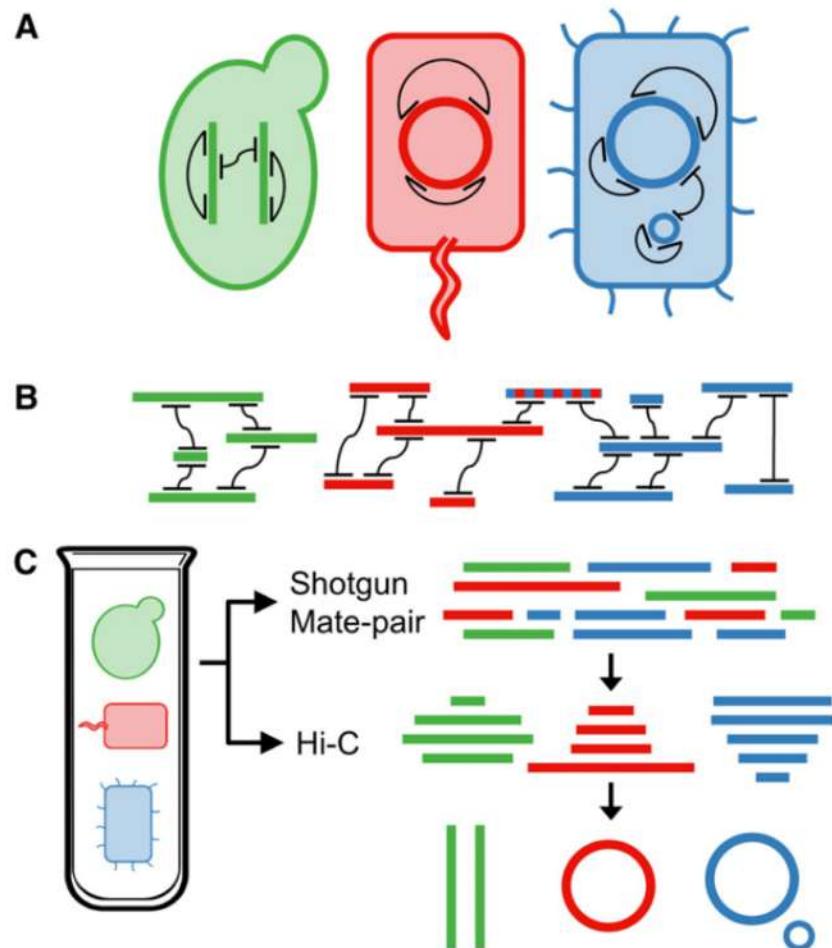
Takashi Nagano<sup>1\*</sup>, Yaniv Lubling<sup>2\*</sup>, Tim J. Stevens<sup>3\*</sup>, Stefan Schoenfelder<sup>1</sup>, Eitan Yaffe<sup>2</sup>, Wendy Dean<sup>4</sup>, Ernest D. Laue<sup>3</sup>, Amos Tanay<sup>2</sup> & Peter Fraser<sup>1</sup>



# Species-Level Deconvolution of Metagenome Assemblies with Hi-C-Based Contact Probability Maps

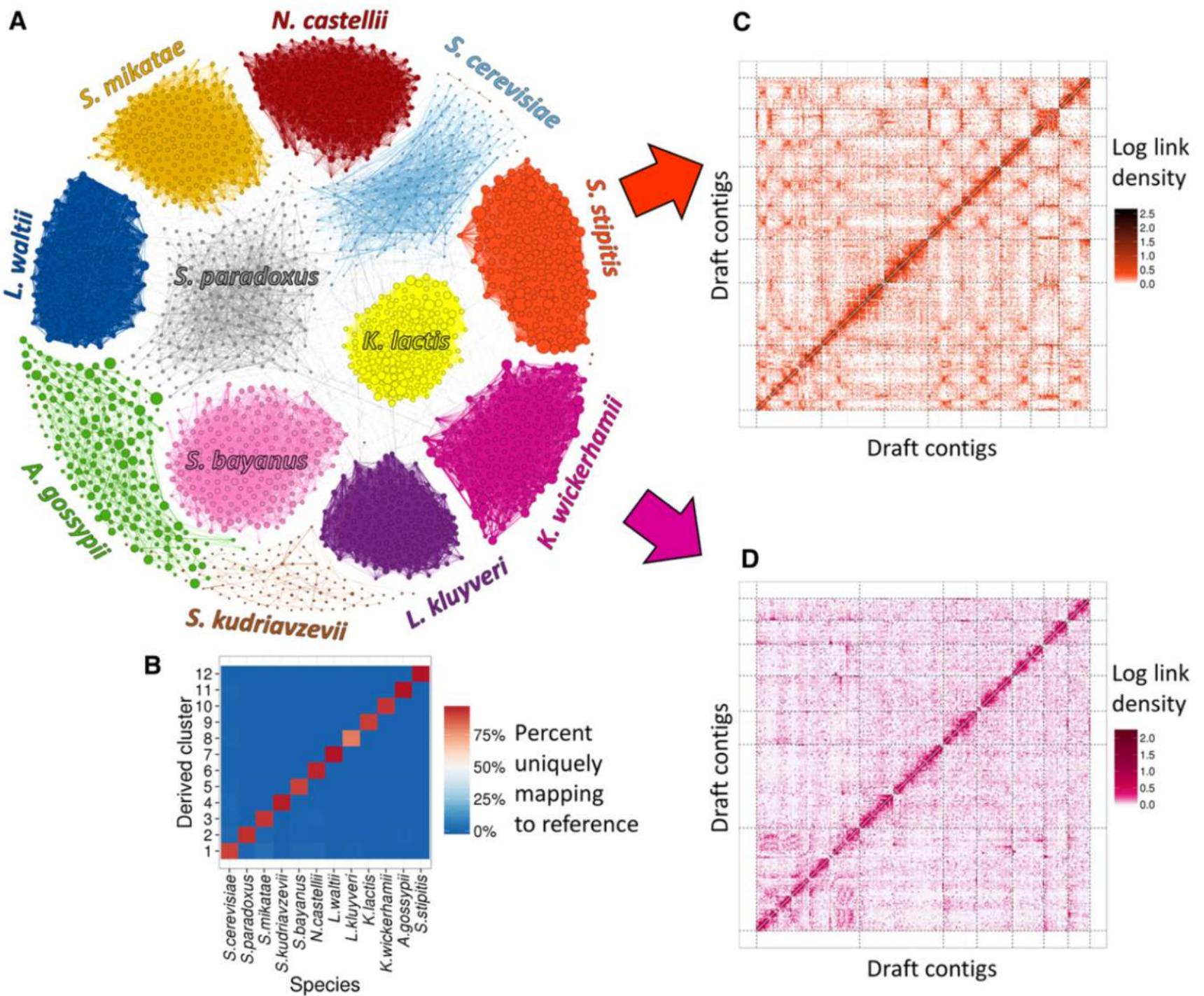
Joshua N. Burton,<sup>1</sup> Ivan Liachko,<sup>1</sup> Maitreya J. Dunham,<sup>2</sup> and Jay Shendure<sup>2</sup>

Department of Genome Sciences, University of Washington, Seattle, Washington 98195-5065



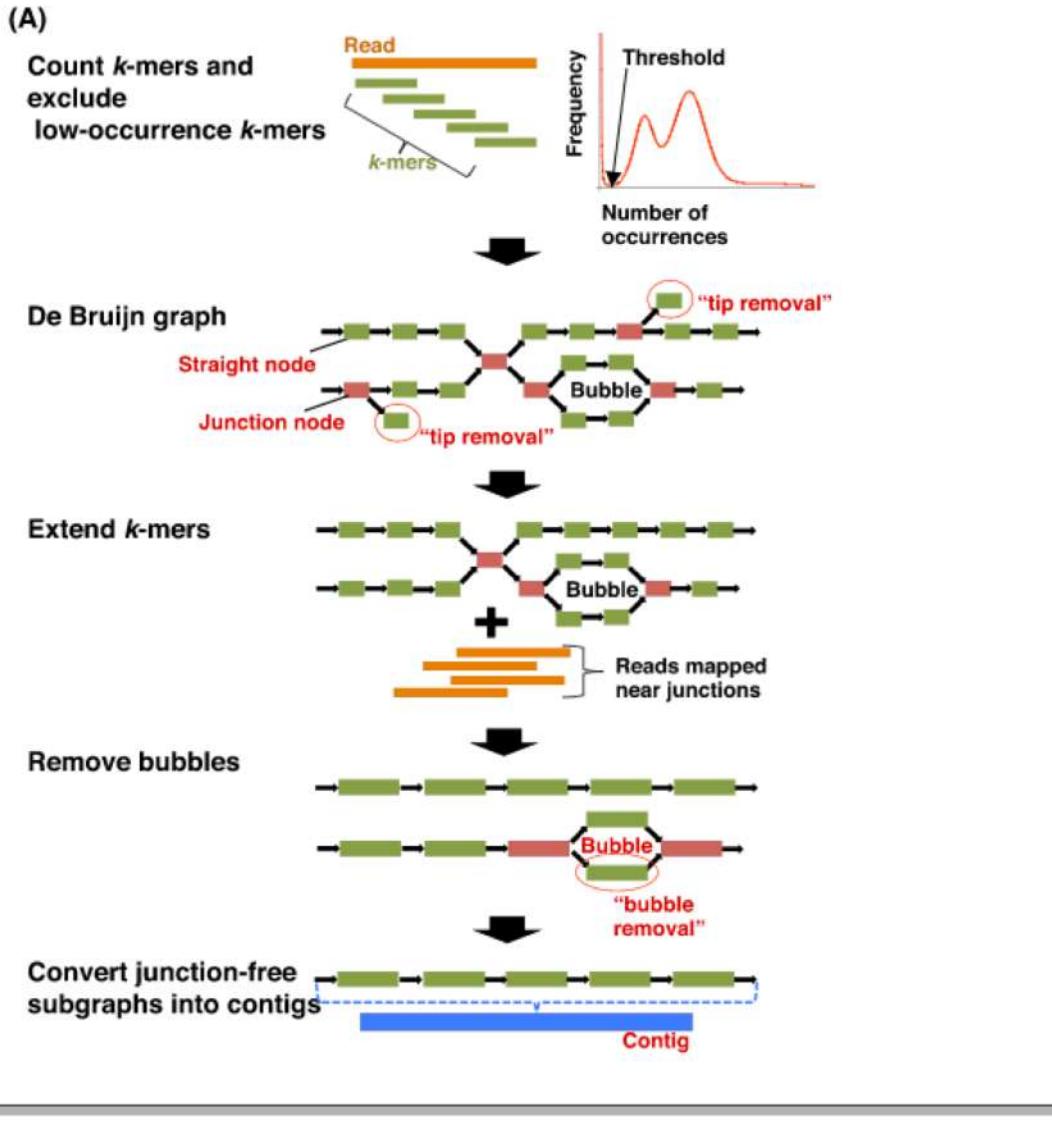
■ Table 2 Sequencing libraries used in MetaPhase analyses

Sample	Library Type	Read Length, bp	Read Pairs, millions
M-Y	Shotgun	101	85.7
	Mate-pair	100	9.2
	Hi-C	100	81.0

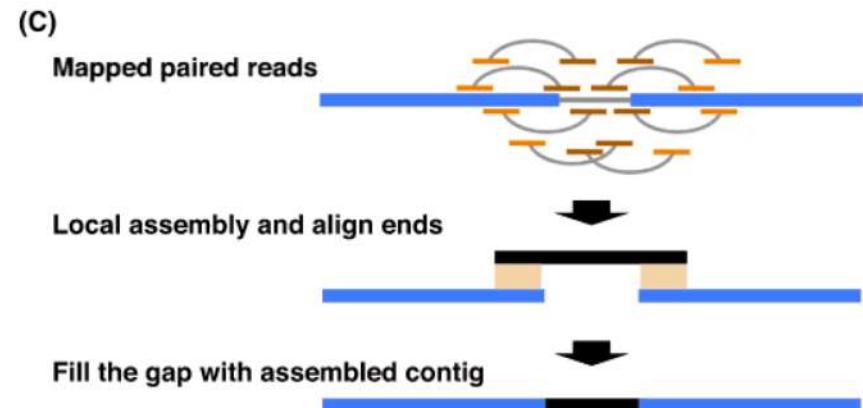
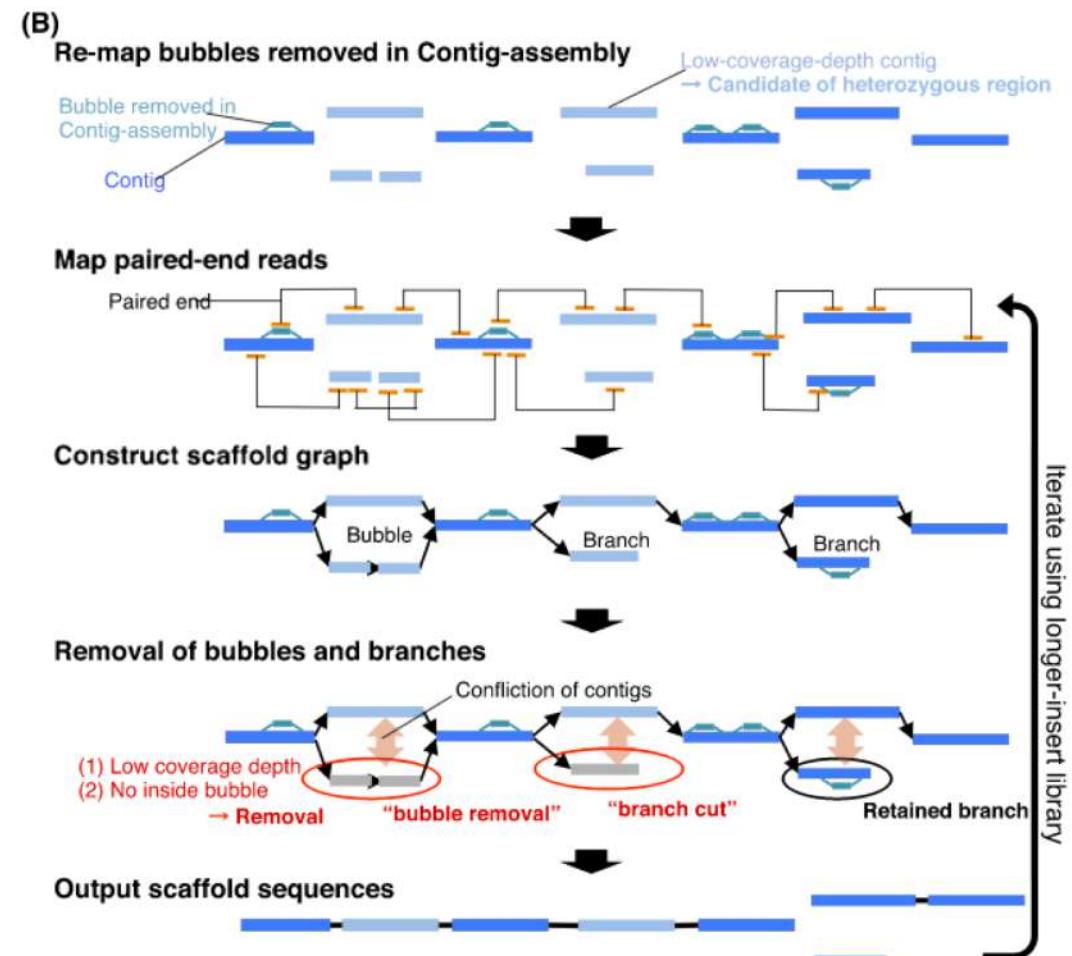


# Case studies

# Would you understand everything in this paper?



Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads



# Why sequence a genome? (2018 version)

- **Genomics advance our understanding of organisms across tree of life**
  - All previous published genomes that were fragmented are being redone again
    - To reveal greater insights and ease of use for community
  - New genomes are expected to be of good quality
- **Reveal more variations within species**
  - Population genomics is not just **remapping** anymore
  - More accurate inference of structure variation
    - Gene level
- Better analysis power in a genomics world

## Improved maize reference genome with single-molecule technologies

Yinping Jiao<sup>1</sup>, Paul Peluso<sup>2</sup>, Jinghua Shi<sup>3</sup>, Tiffany Liang<sup>3</sup>, Michelle C. Stitzer<sup>4</sup>, Bo Wang<sup>1</sup>, Michael S. Campbell<sup>1</sup>, Joshua C. Stein<sup>1</sup>, Xuehong Wei<sup>1</sup>, Chen-Shan Chin<sup>2</sup>, Katherine Guill<sup>5</sup>, Michael Regulski<sup>1</sup>, Sunita Kumari<sup>1</sup>, Andrew Olson<sup>1</sup>, Jonathan Gent<sup>6</sup>, Kevin L. Schneider<sup>7</sup>, Thomas K. Wolfgruber<sup>7</sup>, Michael R. May<sup>8</sup>, Nathan M. Springer<sup>9</sup>, Eric Antoniou<sup>1</sup>, W. Richard McCombie<sup>1</sup>, Gernot G. Presting<sup>7</sup>, Michael McMullen<sup>5</sup>, Jeffrey Ross-Ibarra<sup>10</sup>, R. Kelly Dawe<sup>6</sup>, Alex Hastie<sup>3</sup>, David R. Rank<sup>2</sup> & Doreen Ware<sup>1,11</sup>

## RESEARCH ARTICLE

## Open Access



CrossMark

## An improved genome assembly uncovers prolific tandem repeats in Atlantic cod

Ole K. Tørresen<sup>1\*</sup>, Bastiaan Star<sup>1</sup>, Sissel Jentoft<sup>1,2</sup>, William B. Reinar<sup>1</sup>, Harald Grove<sup>3</sup>, Jason R. Miller<sup>4</sup>, Brian P. Walenz<sup>5</sup>, James Knight<sup>6</sup>, Jenny M. Ekholm<sup>7</sup>, Paul Peluso<sup>7</sup>, Rolf B. Edvardsen<sup>8</sup>, Ave Tooming-Klunderud<sup>1</sup>, Morten Skage<sup>1</sup>, Sigbjørn Lien<sup>3</sup>, Kjetill S. Jakobsen<sup>1</sup> and Alexander J. Nederbragt<sup>1,9\*</sup>

## An improved assembly of the loblolly pine mega-genome using long-read single-molecule sequencing

Aleksey V. Zimin<sup>1,2</sup>, Kristian A. Stevens<sup>3</sup>, Marc W. Crepeau<sup>3</sup>, Daniela Puiu<sup>2</sup>, Jill L. Wegrzyn<sup>4</sup>, James A. Yorke<sup>1</sup>, Charles H. Langley<sup>3</sup>, David B. Neale<sup>5</sup> and Steven L. Salzberg<sup>2,6,\*</sup>

## Research

## Improved genome assembly of American alligator genome reveals conserved architecture of estrogen signaling

Edward S. Rice<sup>1</sup>, Satomi Kohno<sup>2</sup>, John St. John<sup>3</sup>, Son Pham<sup>4</sup>, Jonathan Howard<sup>5</sup>, Liana F. Lareau<sup>6</sup>, Brendan L. O'Connell<sup>1,7</sup>, Glenn Hickey<sup>1</sup>, Joel Armstrong<sup>1</sup>, Alden Deran<sup>1</sup>, Ian Fiddes<sup>1</sup>, Roy N. Platt II<sup>8</sup>, Cathy Gresham<sup>9</sup>, Fiona McCarthy<sup>10</sup>, Colin Kern<sup>11</sup>, David Haan<sup>1</sup>, Tan Phan<sup>12</sup>, Carl Schmidt<sup>13</sup>, Jeremy R. Sanford<sup>14</sup>, David A. Ray<sup>8</sup>, Benedict Paten<sup>15</sup>, Louis J. Guillette Jr.,<sup>16,†</sup> and Richard E. Green<sup>1,6,7</sup>

## Single-molecule sequencing and optical mapping yields an improved genome of woodland strawberry (*Fragaria vesca*) with chromosome-scale contiguity

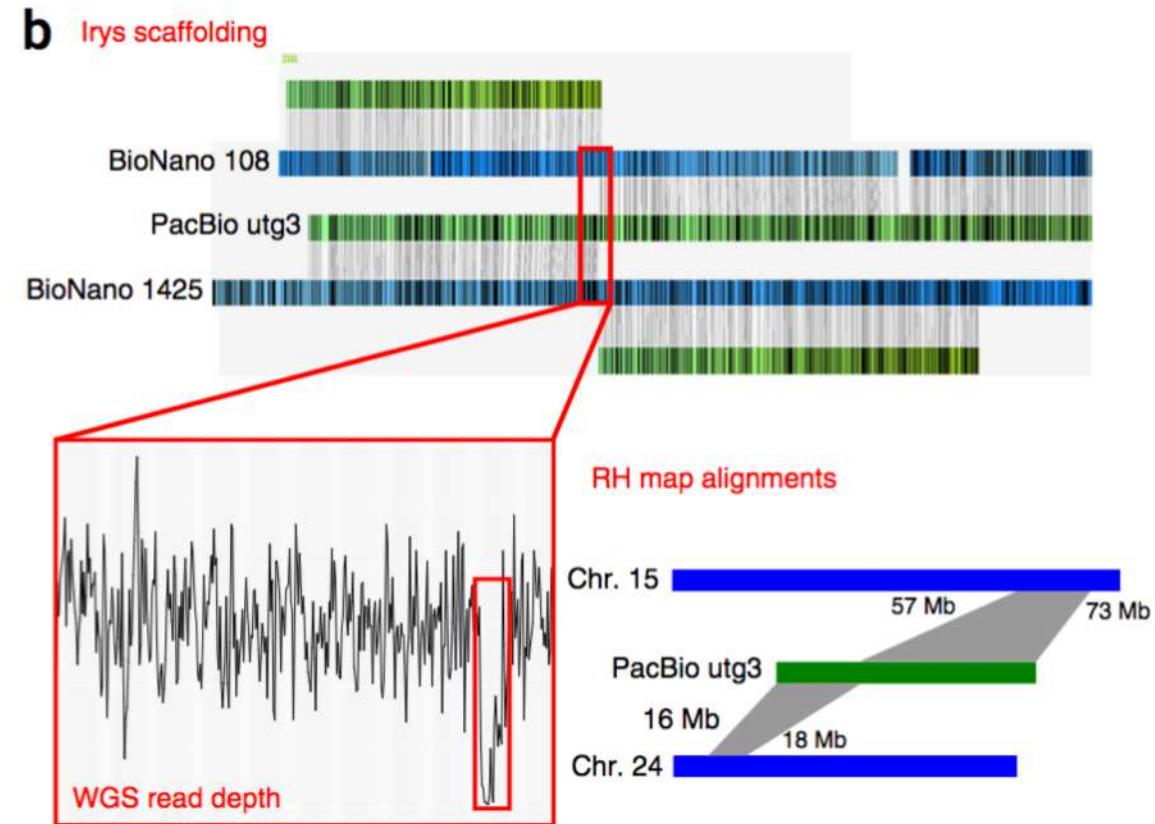
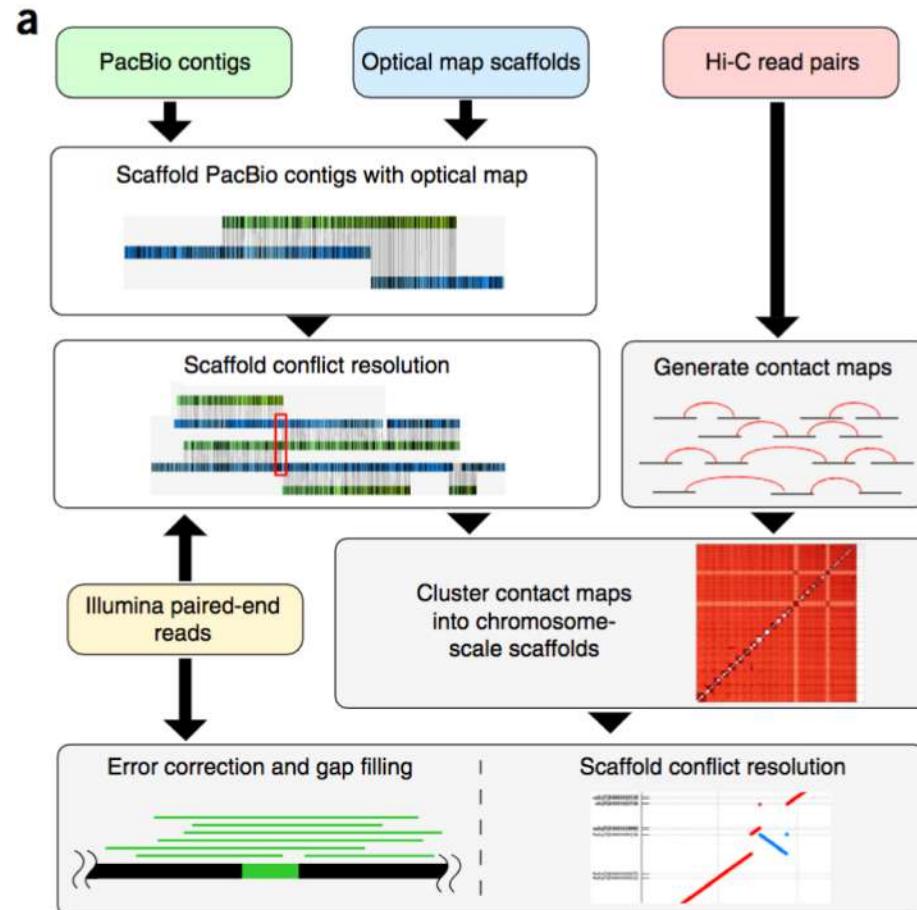
Patrick P. Edger<sup>1,2,\*†</sup>, Robert VanBuren<sup>1,†</sup>, Marivi Colle<sup>1</sup>, Thomas J. Poorten<sup>3</sup>, Ching Man Wai<sup>1</sup>, Chad E. Niederhuth<sup>4</sup>, Elizabeth I. Alger<sup>1</sup>, Shujun Ou<sup>1,2</sup>, Charlotte B. Acharya<sup>3</sup>, Jie Wang<sup>5</sup>, Pete Callow<sup>1</sup>, Michael R. McKain<sup>6</sup>, Jinghua Shi<sup>7</sup>, Chad Collier<sup>7</sup>, Zhiyong Xiong<sup>8</sup>, Jeffrey P. Mower<sup>9</sup>, Janet P. Slovin<sup>10</sup>, Timo Hytönen<sup>11</sup>, Ning Jiang<sup>1,2</sup>, Kevin L. Childs<sup>5,12</sup> and Steven J. Knapp<sup>3,\*</sup>

## An improved assembly and annotation of the allohexaploid wheat genome identifies complete families of agronomic genes and provides genomic evidence for chromosomal translocations

Bernardo J. Clavijo<sup>1,9</sup>, Luca Venturini<sup>1,9</sup>, Christian Schudoma<sup>1</sup>, Gonzalo Garcia Accinelli<sup>1</sup>, Gemy Kaithakottil<sup>1</sup>, Jonathan Wright<sup>1</sup>, Philippa Borrill<sup>2</sup>, George Kettleborough<sup>1</sup>, Darren Heavens<sup>1</sup>, Helen Chapman<sup>1</sup>, James Lipscombe<sup>1</sup>, Tom Barker<sup>1</sup>, Fu-Hao Lu<sup>2</sup>, Neil McKenzie<sup>2</sup>, Dina Raats<sup>1</sup>, Ricardo H. Ramirez-Gonzalez<sup>1,2</sup>, Aurore Coince<sup>1</sup>, Ned Peel<sup>1</sup>, Lawrence Percival-Alwyn<sup>1</sup>, Owen Duncan<sup>3</sup>, Josua Trösch<sup>3</sup>, Guotai Yu<sup>2</sup>, Dan M. Bolser<sup>4</sup>, Guy Namaati<sup>4</sup>, Arnaud Kerhennou<sup>4</sup>, Manuel Spannagl<sup>5</sup>, Heidrun Gundlach<sup>5</sup>, Georg Haberer<sup>5</sup>, Robert P. Davey<sup>1,6</sup>, Christine Fosker<sup>1</sup>, Federica Di Palma<sup>1,6</sup>, Andrew L. Phillips<sup>7</sup>, A. Harvey Millar<sup>3</sup>, Paul J. Kersey<sup>4</sup>, Cristobal Uauy<sup>2</sup>, Ksenia V. Krasileva<sup>1,6,8</sup>, David Swarbreck<sup>1,6</sup>, Michael W. Bevan<sup>2</sup> and Matthew D. Clark<sup>1,6</sup>

# Single-molecule sequencing and chromatin conformation capture enable *de novo* reference assembly of the domestic goat genome

Derek M Bickhart<sup>1,18</sup>, Benjamin D Rosen<sup>2,18</sup>, Sergey Koren<sup>3,18</sup>, Brian L Sayre<sup>4</sup>, Alex R Hastie<sup>5</sup>, Saki Chan<sup>5</sup>, Joyce Lee<sup>5</sup>, Ernest T Lam<sup>5</sup>, Ivan Liachko<sup>6</sup>, Shawn T Sullivan<sup>7</sup>, Joshua N Burton<sup>6</sup>, Heather J Huson<sup>8</sup>, John C Nystrom<sup>8</sup>, Christy M Kelley<sup>9</sup>, Jana L Hutchison<sup>2</sup>, Yang Zhou<sup>2,10</sup>, Jiajie Sun<sup>11</sup>, Alessandra Crisà<sup>12</sup>, F Abel Ponce de León<sup>13</sup>, John C Schwartz<sup>14</sup>, John A Hammond<sup>14</sup>, Geoffrey C Waldbieser<sup>15</sup>, Steven G Schroeder<sup>2</sup>, George E Liu<sup>2</sup>, Maitreya J Dunham<sup>6</sup>, Jay Shendure<sup>6,16</sup>, Tad S Sonstegard<sup>17</sup>, Adam M Phillippy<sup>3</sup>, Curtis P Van Tassell<sup>2</sup> & Timothy P L Smith<sup>9</sup>



Correcting misassemblies

... ARS1 comprises just 31 scaffolds and 649 gaps covering 30 of the 31 haploid, acrocentric goat chromosomes (excluding only the Y chromosome),

**Table 1 Assembly statistics**

Assembly <sup>a</sup>	Contigs <sup>b</sup>	Scaffolds	Unplaced contigs <sup>c</sup>	Degenerate contigs <sup>d</sup>	Contig NG50 (Mb) <sup>e</sup>	Scaffold NG50 (Mb) <sup>e,f</sup>	Assembly size (Gb)	Assembly in scaffolds (%)
PacBio	3,074	–	–	30,693	3.795	–	2.914	N/A
Optical Map	–	2,944	–	–	–	1.487	2.748	N/A
PacBio + Optical Map	1,109	333	1,242	30,693	10.197	20.623	2.910	90.89
PacBio + Hi-C	2,115	31	959	30,693	3.795	88.799	2.910	87.97
PacBio + Optical Map + Hi-C	1,780	31	571	30,693	10.197	87.347	2.910	89.05
ARS1	680	31	654	29,315	18.702	87.277	2.924	88.32

<sup>a</sup>Assemblies are listed in order of inclusion of scaffolding technologies toward the final assembly (ARS1), with the original contigs (PacBio) scaffolded using different technologies (Optical Map and Hi-C). Because the optical map program (Irys Scaffold) generates an assembly from the consensus of labeled DNA molecules, we have included scaffold statistics from these data (optical map) for comparison. <sup>b</sup>The number of continuous stretches of sequence within the scaffold without gaps >3 bases in length of at least 10 bases. <sup>c</sup>Unplaced contigs are defined as input contigs or scaffolds that were not placed by the optical map or Hi-C in a scaffold were excluded from the scaffold counts. <sup>d</sup>Degenerate contigs were assembled unitigs that had less than 50 PacBio reads supporting their assembly (**Supplementary Note**). Differences in degenerate contig counts in the final ARS1 assembly are due to PBJelly merging of degenerate contigs (538 contigs) or removal due to no supporting PacBio read alignments (840). <sup>e</sup>All NG50 values are based on the ARS1 assembly size (2.924 Gb). <sup>f</sup>No scaffolds were generated for the PacBio entry.

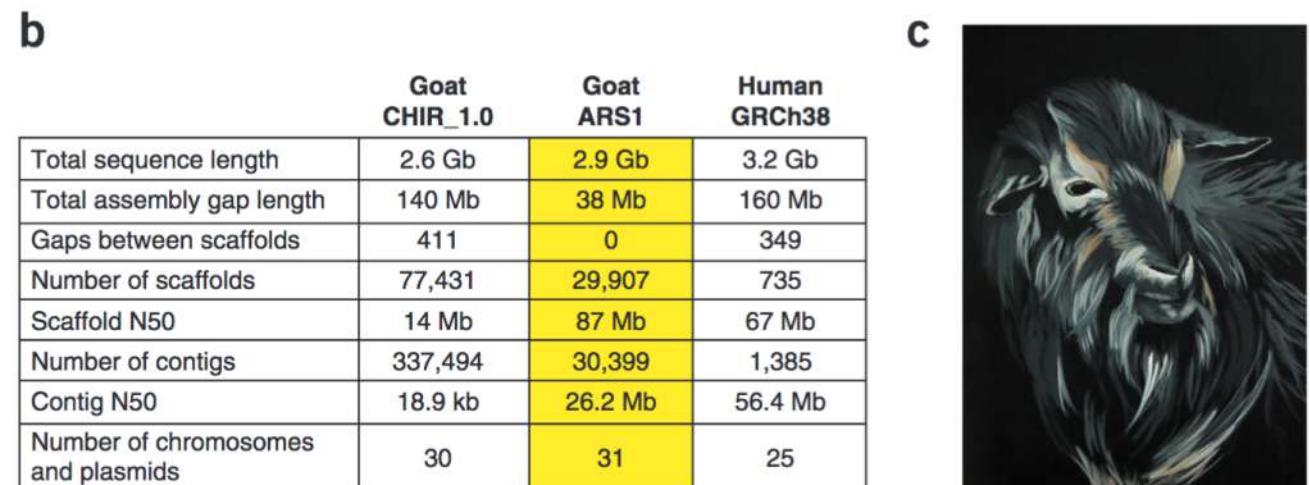
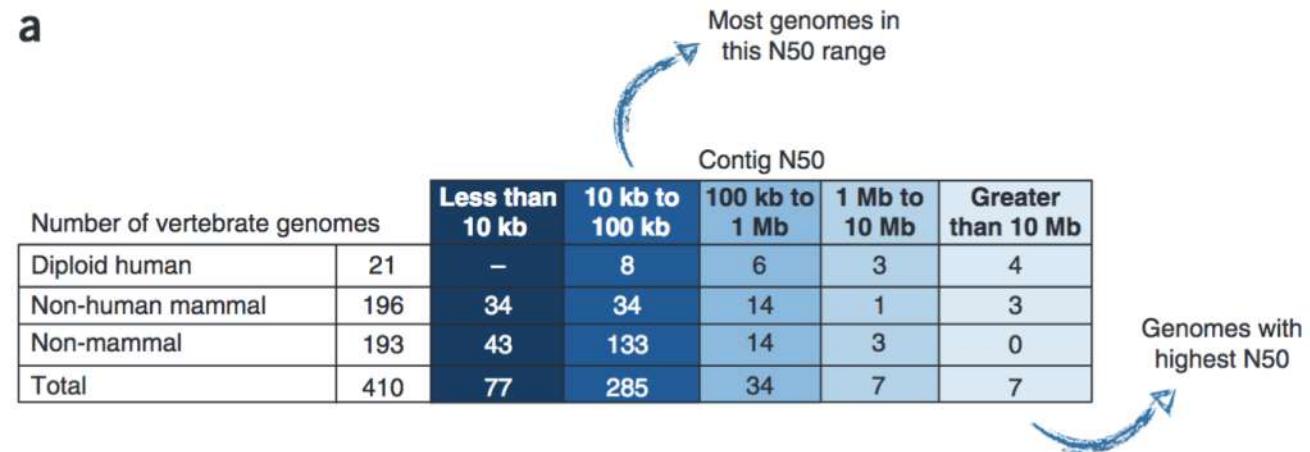
# A golden goat genome

Kim C Worley

The newly described *de novo* goat genome sequence is the most contiguous diploid vertebrate assembly generated thus far using whole-genome assembly and scaffolding methods. The contiguity of this assembly is approaching that of the finished human and mouse genomes and suggests an affordable roadmap to high-quality references for thousands of species.

.... This report generates sighs of relief from researchers frustrated with the highly fragmented genome sequences available for most species....

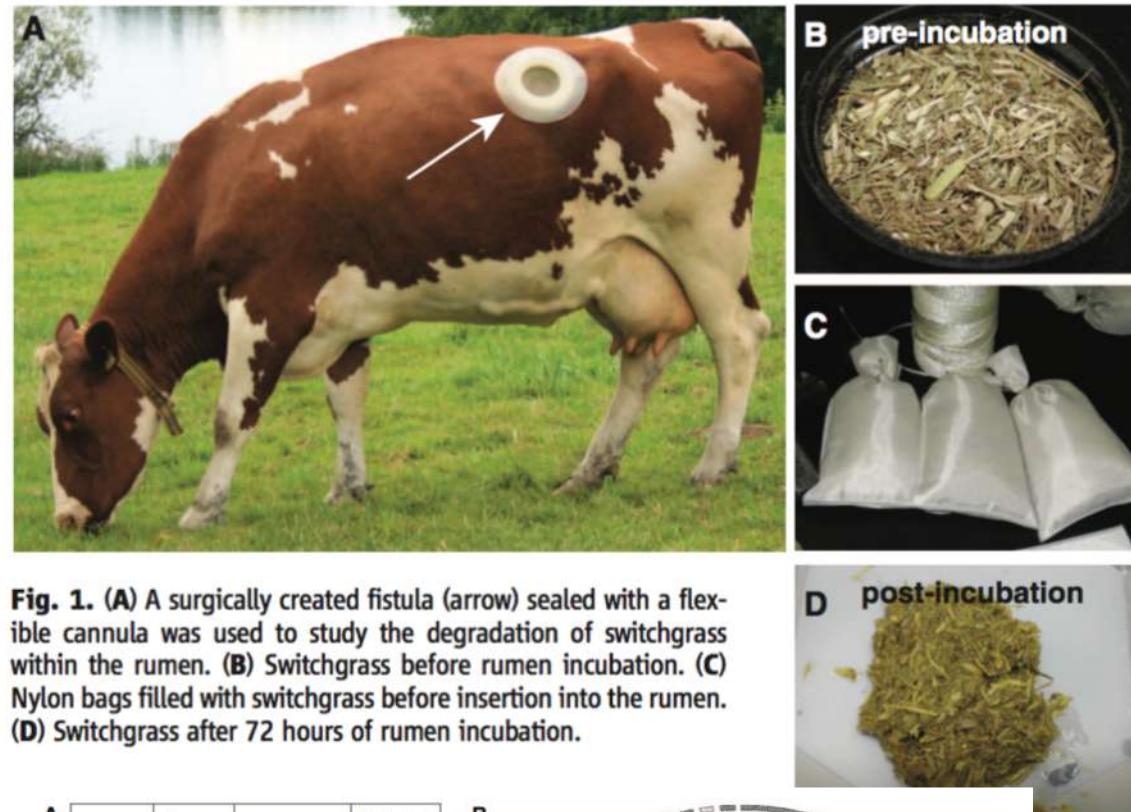
... The lower costs and greater accessibility of these methods bring potential for wider impact....



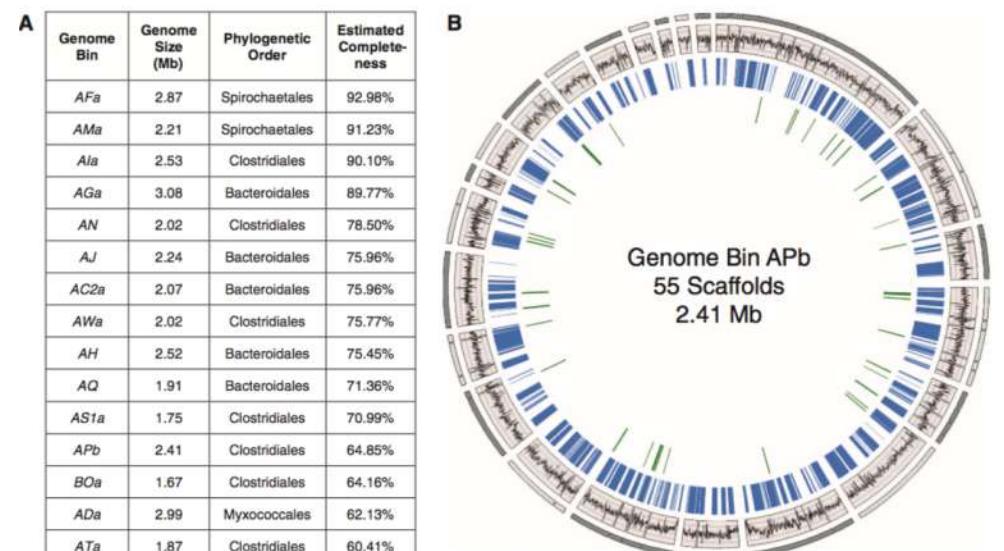
# Metagenomic Discovery of Biomass-Degrading Genes and Genomes from Cow Rumen

Matthias Hess,<sup>1,2\*</sup> Alexander Sczyrba,<sup>1,2\*</sup> Rob Egan,<sup>1,2</sup> Tae-Wan Kim,<sup>3</sup> Harshal Chokhawala,<sup>3</sup> Gary Schroth,<sup>4</sup> Shujun Luo,<sup>4</sup> Douglas S. Clark,<sup>3,5</sup> Feng Chen,<sup>1,2</sup> Tao Zhang,<sup>1,2</sup> Roderick I. Mackie,<sup>6</sup> Len A. Pennacchio,<sup>1,2</sup> Susannah G. Tringe,<sup>1,2</sup> Axel Visel,<sup>1,2</sup> Tanja Woyke,<sup>1,2</sup> Zhong Wang,<sup>1,2</sup> Edward M. Rubin<sup>1,2†</sup>

- 268Gb of metagenomics data
- Identified 27,755 putative carbohydrate-active genes from a cow rumen metagenome
- Expressed 90 candidates of which 57% had enzymatic activity against cellulosic substrates
- Assembled 15 uncultured microbial genomes



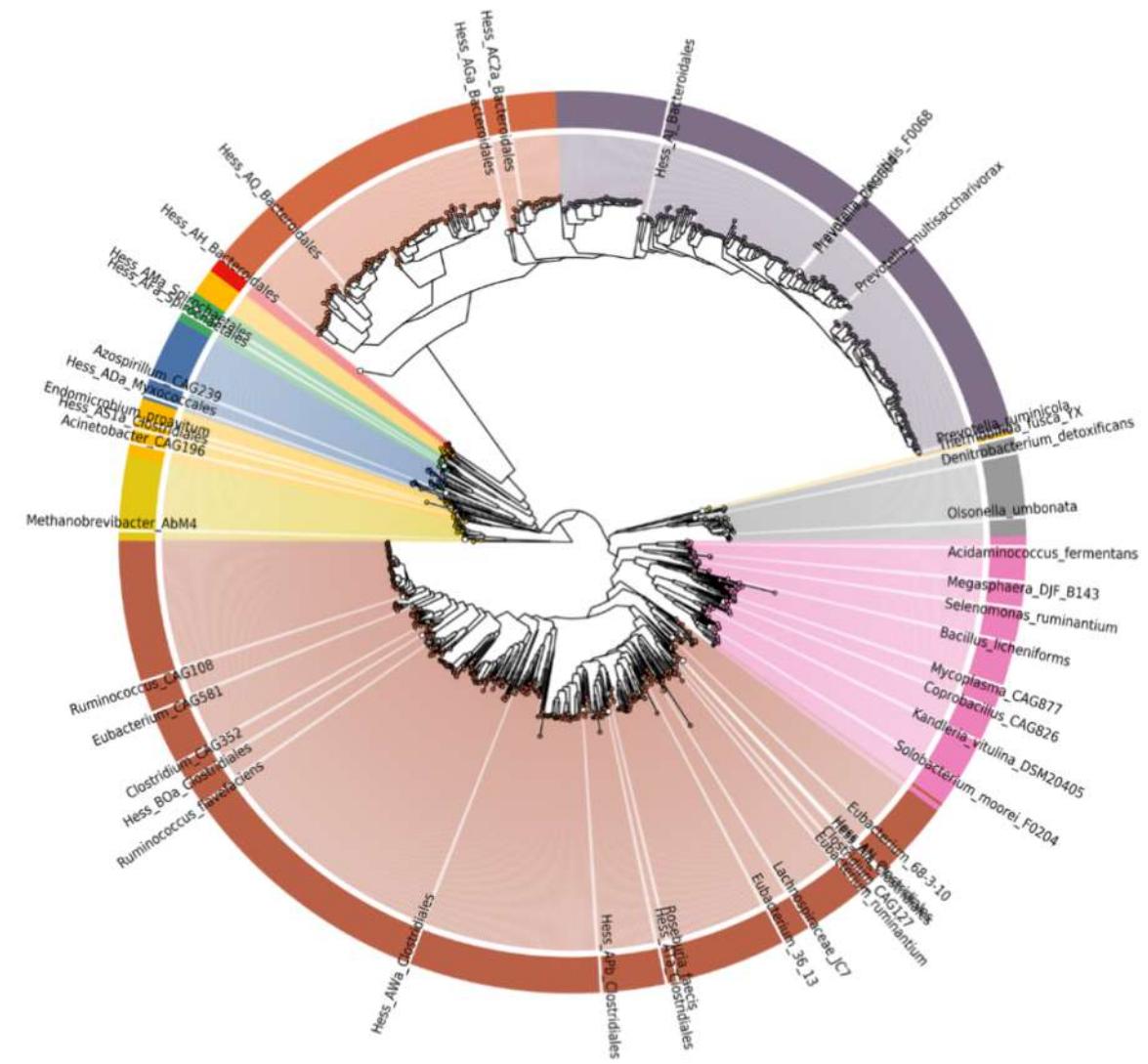
**Fig. 1.** (A) A surgically created fistula (arrow) sealed with a flexible cannula was used to study the degradation of switchgrass within the rumen. (B) Switchgrass before rumen incubation. (C) Nylon bags filled with switchgrass before insertion into the rumen. (D) Switchgrass after 72 hours of rumen incubation.



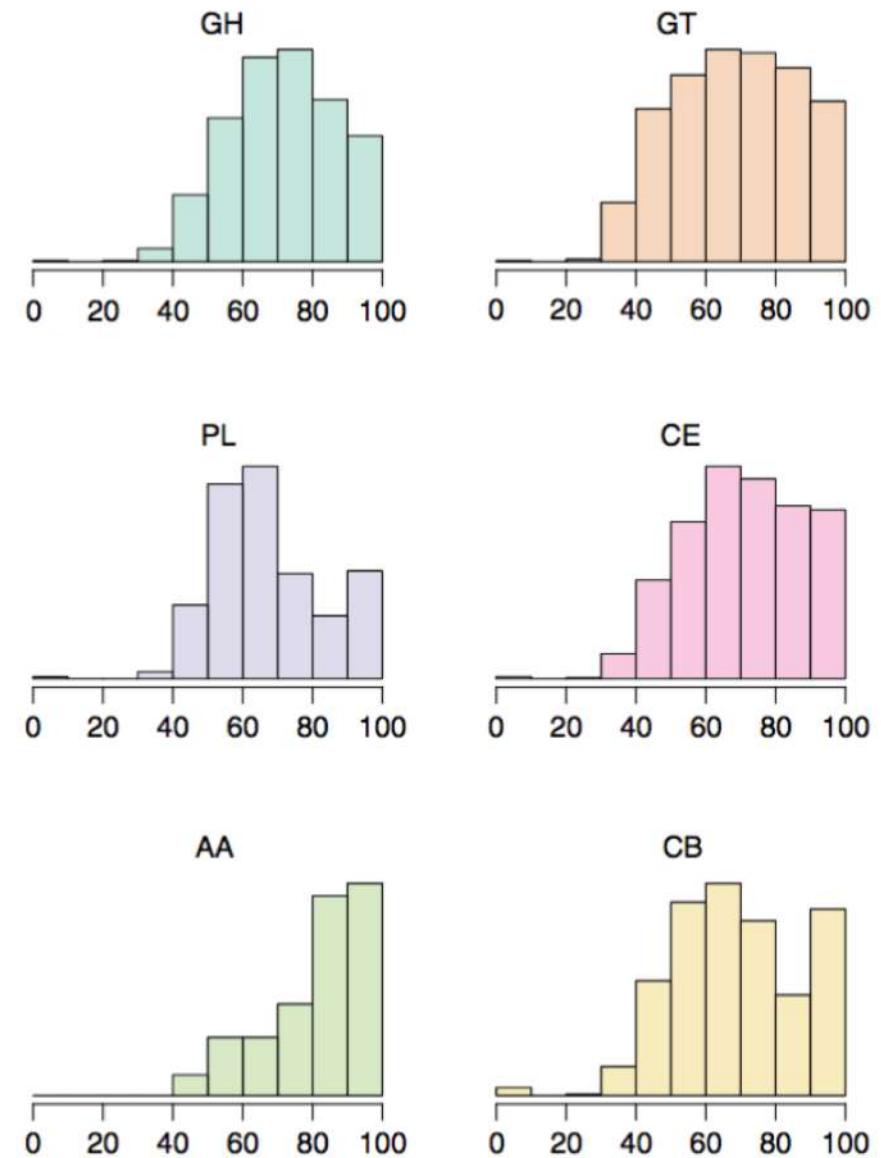
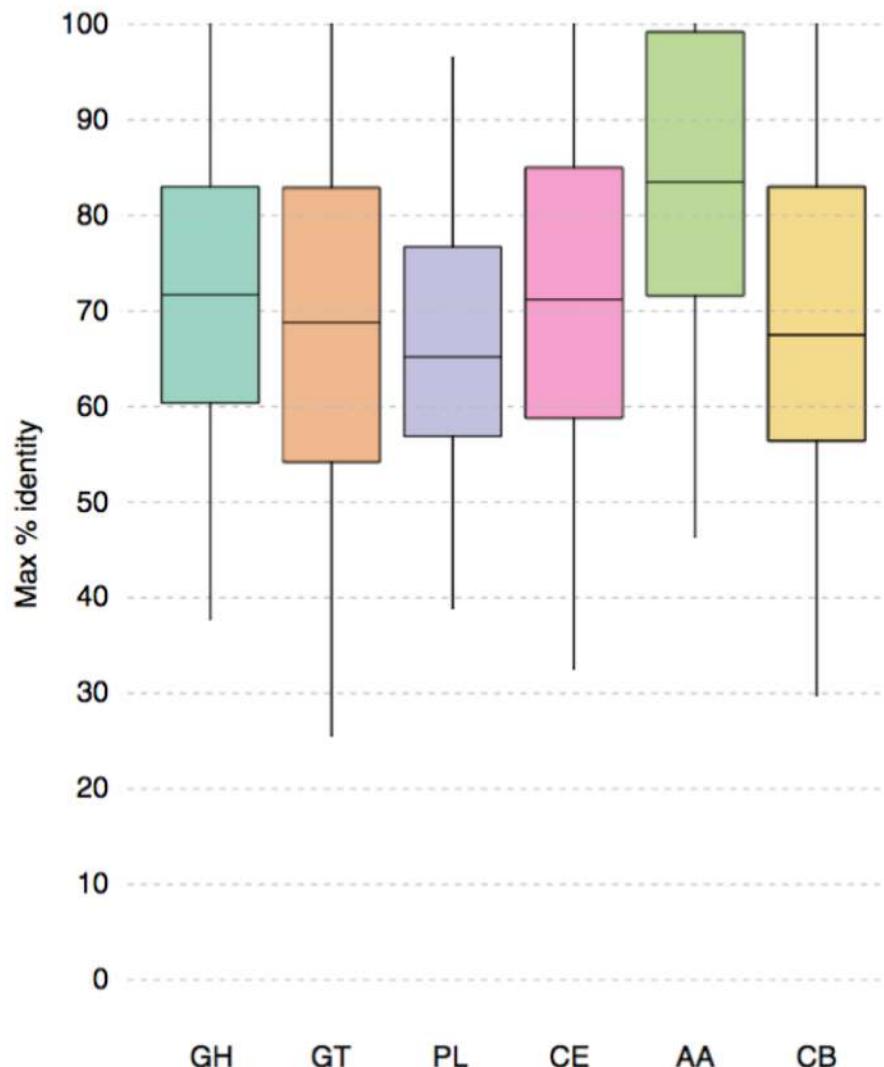
# Assembly of 913 microbial genomes from metagenomic sequencing of the cow rumen

Robert D. Stewart<sup>1</sup>, Marc D. Auffret<sup>2</sup>, Amanda Warr<sup>1</sup>, Andrew H. Wiser<sup>3</sup>, Maximilian O. Press<sup>1</sup><sup>3</sup>,  
Kyle W. Langford<sup>3</sup>, Ivan Liachko<sup>3</sup>, Timothy J. Snelling<sup>4</sup>, Richard J. Dewhurst<sup>1</sup><sup>2</sup>, Alan W. Walker<sup>1</sup><sup>4</sup>,  
Rainer Roehe<sup>2</sup> & Mick Watson<sup>1</sup><sup>3</sup>

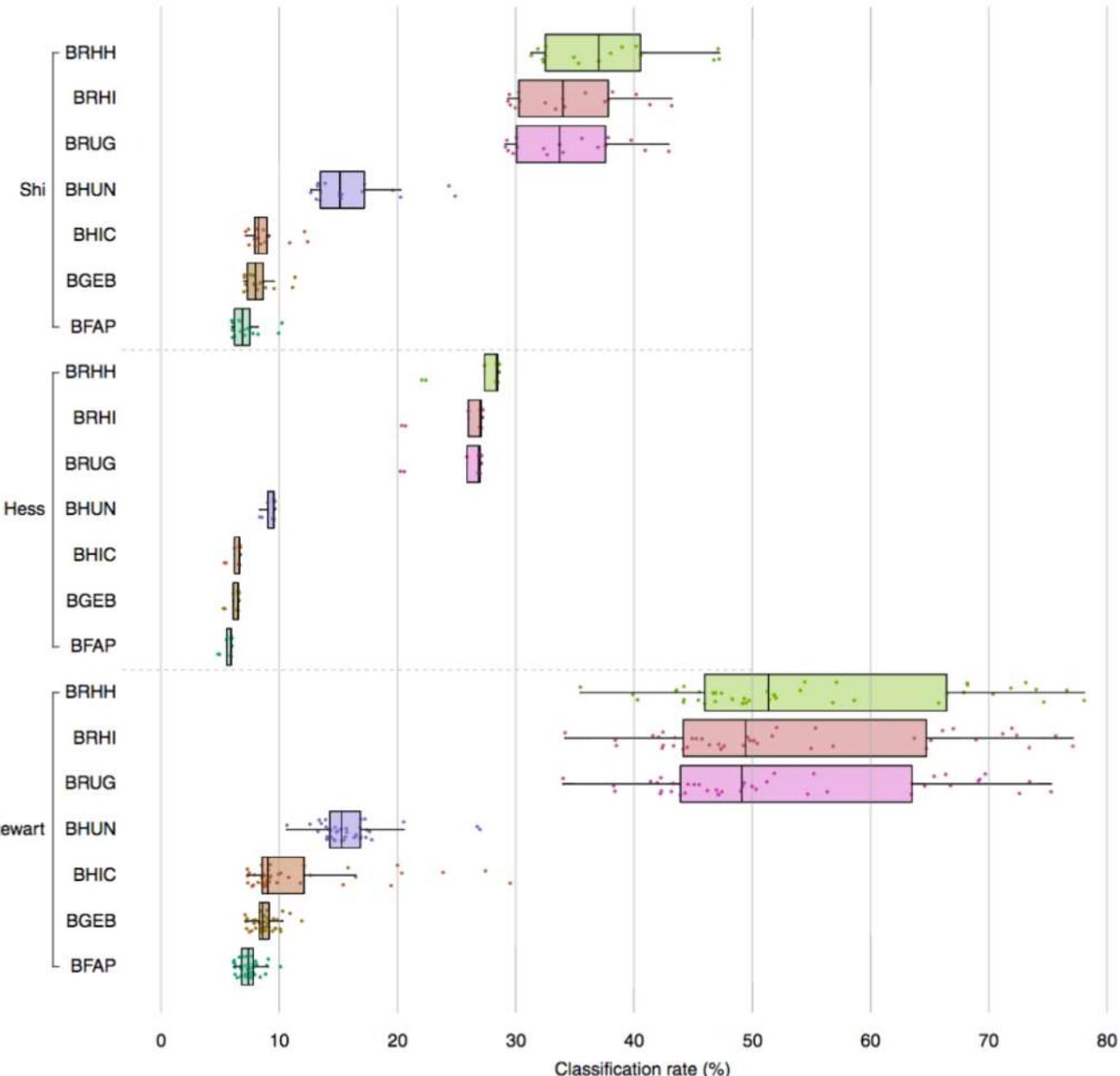
- 800 Gb of sequence data derived from 43 Scottish cattle
- Using metagenomic binning and Hi-C techniques
- Assembly of 913 draft bacterial and archaeal genomes
- Most of these genomes represent previously unsequenced strains and species.
- The draft genomes contain over 69,000 proteins predicted to be involved in carbohydrate metabolism, over 90% of which do not have a good match in public databases.



- The draft genomes contain over 69,000 proteins predicted to be involved in carbohydrate metabolism, over 90% of which do not have a good match in public databases.



- Inclusion of the 913 genomes presented here improves metagenomic read classification by sevenfold against the study's own data, and by fivefold against other publicly available rumen datasets.
- dataset substantially improves the coverage of rumen microbial genomes in the public databases and represents a valuable resource for biomass-degrading enzyme discovery and studies of the rumen microbiome

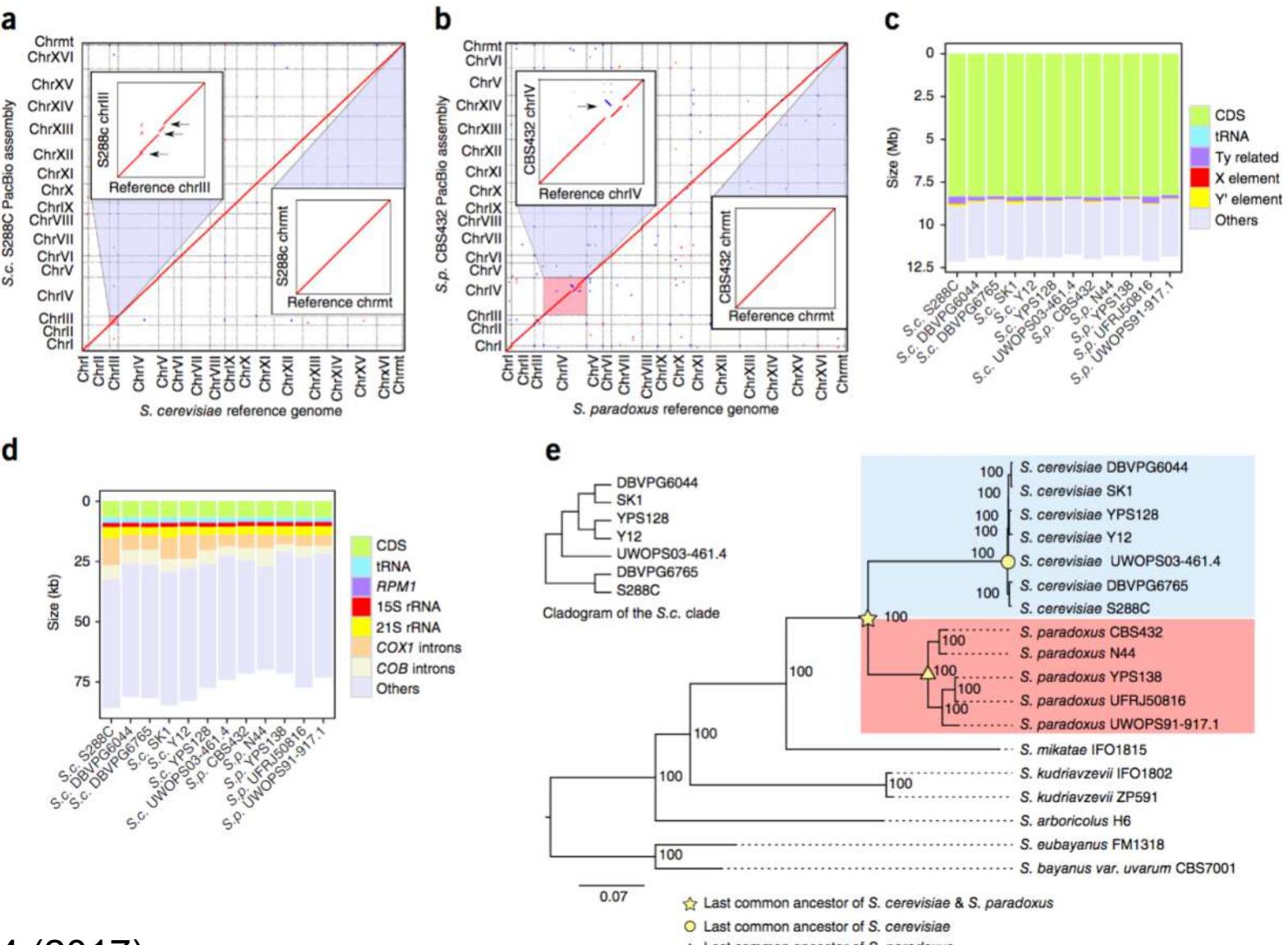


**Fig. 4** Classification rate for three datasets against various Kraken databases. BFAP bacterial, archaeal, fungal and protozoan genomes from RefSeq, BGEB BFAP + 1003 GEBA genomes, BHIC BFAP + 63 hRUG genomes, BHUN BFAP + 410 genomes from the Hungate 1000 project, BRUG BFAP + 850 RUG MAGs, BRHI BFAP + all 913 genomes from this study, BRHH BFAP + 913 RUGs + 410 Hungate 1000 genomes. Addition of rumen-specific RUGs or Hungate 1000 genomes has the most dramatic effect

# Contrasting evolutionary genome dynamics between domesticated and wild yeasts

Jia-Xing Yue<sup>1</sup> , Jing Li<sup>1</sup>, Louise Aigrain<sup>2</sup>, Johan Hallin<sup>1</sup> , Karl Persson<sup>3</sup> , Karen Oliver<sup>2</sup>, Anders Bergström<sup>2</sup>, Paul Coupland<sup>2,5</sup>, Jonas Warringer<sup>3</sup> , Marco Cosentino Lagomarsino<sup>4</sup>, Gilles Fischer<sup>4</sup>, Richard Durbin<sup>2</sup> & Gianni Liti<sup>1</sup>

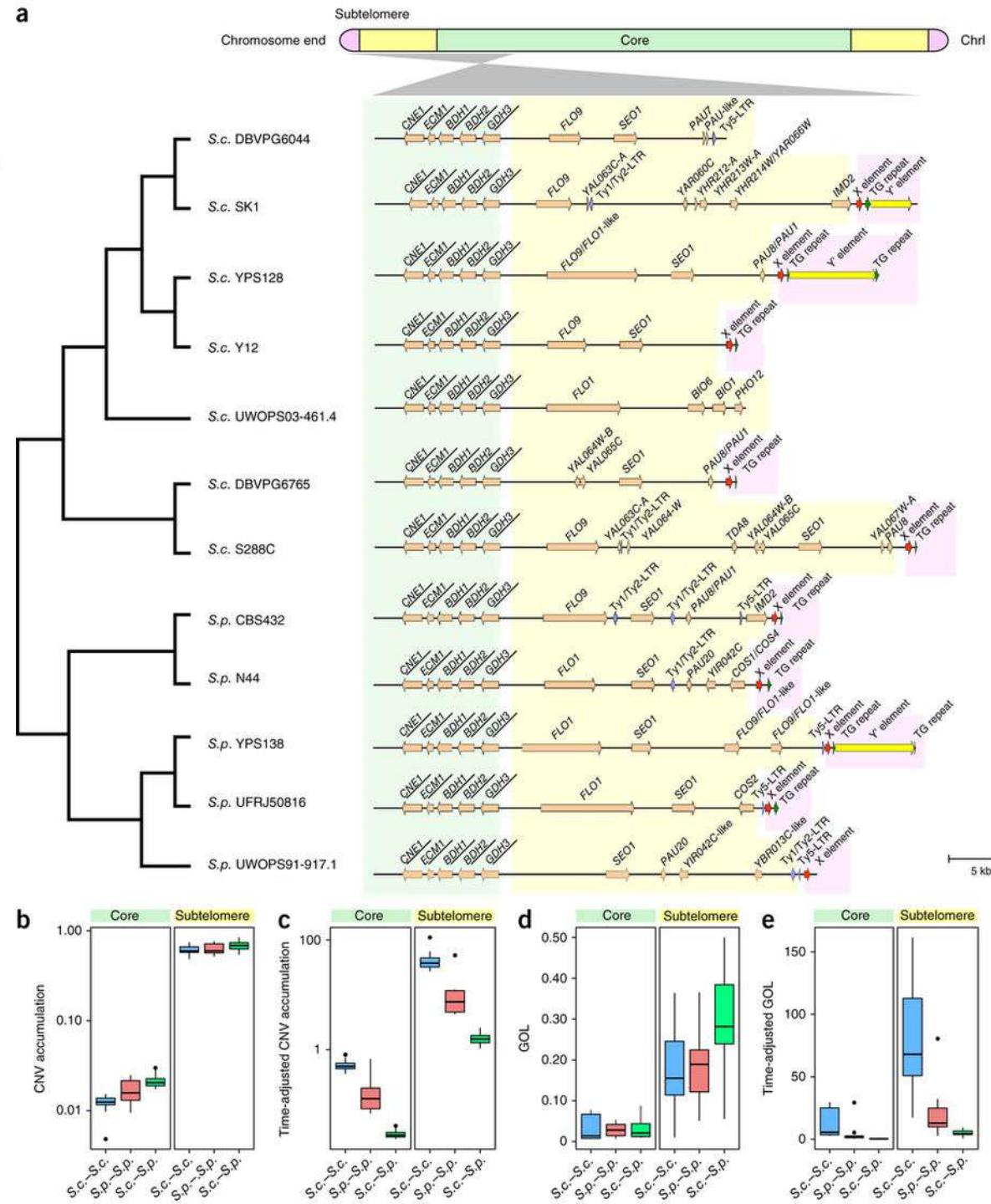
- long-read sequencing to generate **end-to-end genome assemblies** for **12 strains** representing major subpopulations of the partially domesticated yeast *Saccharomyces cerevisiae* and its wild relative *S. paradoxus*.



# Contrasting evolutionary genome dynamics between domesticated and wild yeasts

Jia-Xing Yue<sup>1</sup>, Jing Li<sup>1</sup>, Louise Aigrain<sup>2</sup>, Johan Hallin<sup>1</sup>, Karl Persson<sup>3</sup>, Karen Oliver<sup>2</sup>, Anders Bergström<sup>2</sup>, Paul Coupland<sup>2,5</sup>, Jonas Warringer<sup>3</sup>, Marco Cosentino Lagomarsino<sup>4</sup>, Gilles Fischer<sup>4</sup>, Richard Durbin<sup>2</sup> & Gianni Liti<sup>1</sup>

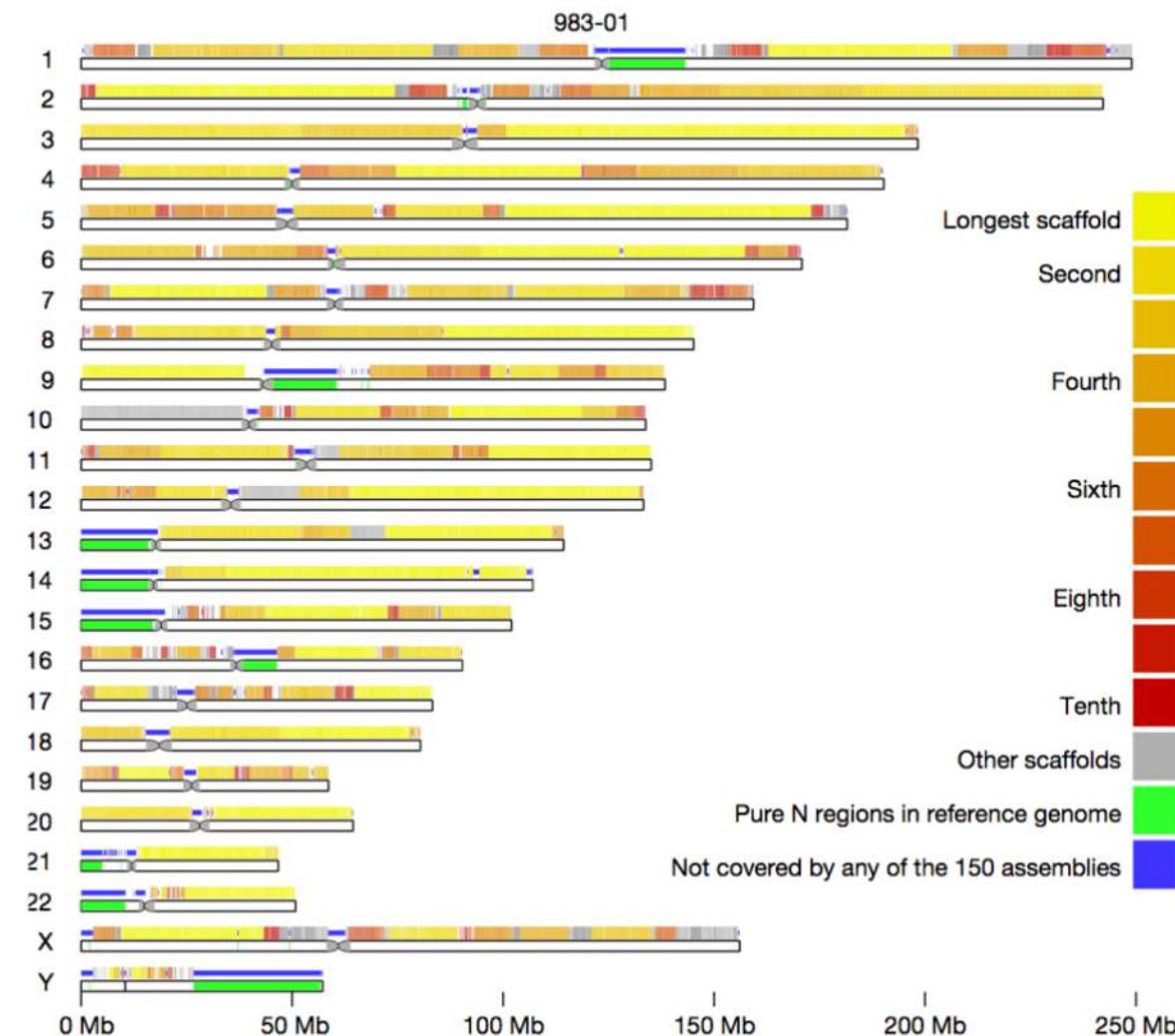
- enable precise definition of chromosomal boundaries between cores and subtelomeres
- S. paradoxus* shows faster accumulation of balanced rearrangements (inversions, reciprocal translocations and transpositions), *S. cerevisiae* accumulates unbalanced rearrangements (novel insertions, deletions and duplications) more rapidly.
- Such striking contrasts between wild and domesticated yeasts are likely to reflect the influence of human activities on structural genome evolution.



## Sequencing and *de novo* assembly of 150 genomes from Denmark as a population reference

Lasse Maretty<sup>1,\*</sup>, Jacob Malte Jensen<sup>2,3,\*</sup>, Bent Petersen<sup>4\*</sup>, Jonas Andreas Sibbesen<sup>1\*</sup>, Siyang Liu<sup>1,5\*</sup>, Palle Villesen<sup>2,3,6\*</sup>, Laurits Skov<sup>2,3\*</sup>, Kirstine Belling<sup>4\*</sup>, Christian Theil Have<sup>7</sup>, Jose M. G. Izazugaza<sup>4</sup>, Marie Grosjean<sup>4</sup>, Jette Bork-Jensen<sup>7</sup>, Jakob Grove<sup>3,8,9</sup>, Thomas D. Als<sup>3,8,9</sup>, Shujia Huang<sup>10,11</sup>, Yuqi Chang<sup>10</sup>, Ruiqi Xu<sup>5</sup>, Weijian Ye<sup>5</sup>, Junhua Rao<sup>5</sup>, Xiaosen Guo<sup>10,12</sup>, Jihua Sun<sup>5,7</sup>, Hongzhi Cao<sup>10</sup>, Chen Ye<sup>10</sup>, Johan van Beusekom<sup>4</sup>, Thomas Espeseth<sup>13,14</sup>, Esben Flindt<sup>12</sup>, Rune M. Friberg<sup>2,3</sup>, Anders E. Halager<sup>2,3</sup>, Stephanie Le Hellard<sup>14,15</sup>, Christina M. Hultman<sup>16</sup>, Francesco Lescat<sup>3,8,9</sup>, Shengting Li<sup>3,8,9</sup>, Ole Lund<sup>4</sup>, Peter Løngren<sup>4</sup>, Thomas Mailund<sup>2,3</sup>, Maria Luisa Matey-Hernandez<sup>4</sup>, Ole Mors<sup>3,6,9</sup>, Christian N. S. Pedersen<sup>2,3</sup>, Thomas Sicheritz-Pontén<sup>4</sup>, Patrick Sullivan<sup>16,17</sup>, Ali Syed<sup>4</sup>, David Westergaard<sup>4</sup>, Rachita Yadav<sup>4</sup>, Ning Li<sup>5</sup>, Xun Xu<sup>10</sup>, Torben Hansen<sup>7</sup>, Anders Krogh<sup>1</sup>, Lars Bolund<sup>8,10</sup>, Thorkild I. A. Sørensen<sup>7,18,19</sup>, Oluf Pedersen<sup>7</sup>, Ramneek Gupta<sup>4</sup>, Simon Rasmussen<sup>4</sup>§, Søren Besenbacher<sup>2,6</sup>§, Anders D. Børglum<sup>3,8,9</sup>§, Jun Wang<sup>3,10,12</sup>§, Hans Eiberg<sup>20</sup>§, Karsten Kristiansen<sup>10,12</sup>§, Søren Brunak<sup>4,21</sup>§ & Mikkel Heide Schierup<sup>2,3,22</sup>§

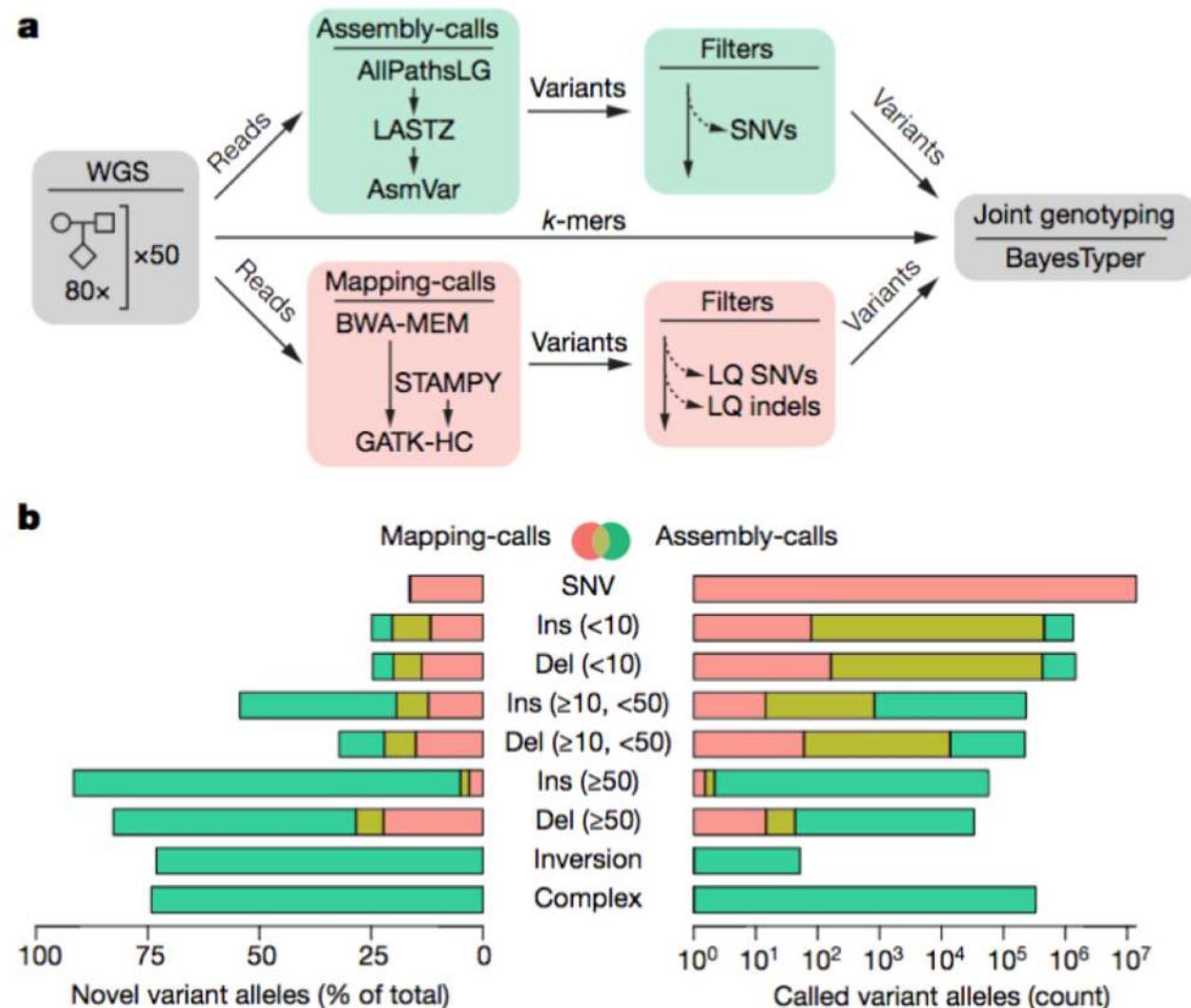
- Hundreds of thousands of human genomes are now being sequenced to characterize genetic variation and use this information to augment association mapping studies of complex disorders and other phenotypic traits.
- Genetic variation is identified mainly by mapping short reads to the reference genome. However, these approaches are biased against discovery of structural variants and variation in the more complex parts of the genome.
- report *de novo* assemblies of 150 individuals (50 trios) from the GenomeDenmark project.



We found that 16.4% of the called SNVs were novel (not in the Single Nucleotide Polymorphism database 142 (dbSNP142) or 1000 Genomes Project phase 3 structural variant call-set), **whereas as many as 91.6% of insertions  $\geq 50$  bp were novel** ([Fig. 2b](#)).

The fraction of novel variants increased rapidly with variant length, especially for insertions ([Fig. 2d](#)), with most longer variants contributed by the assembly-based approach...

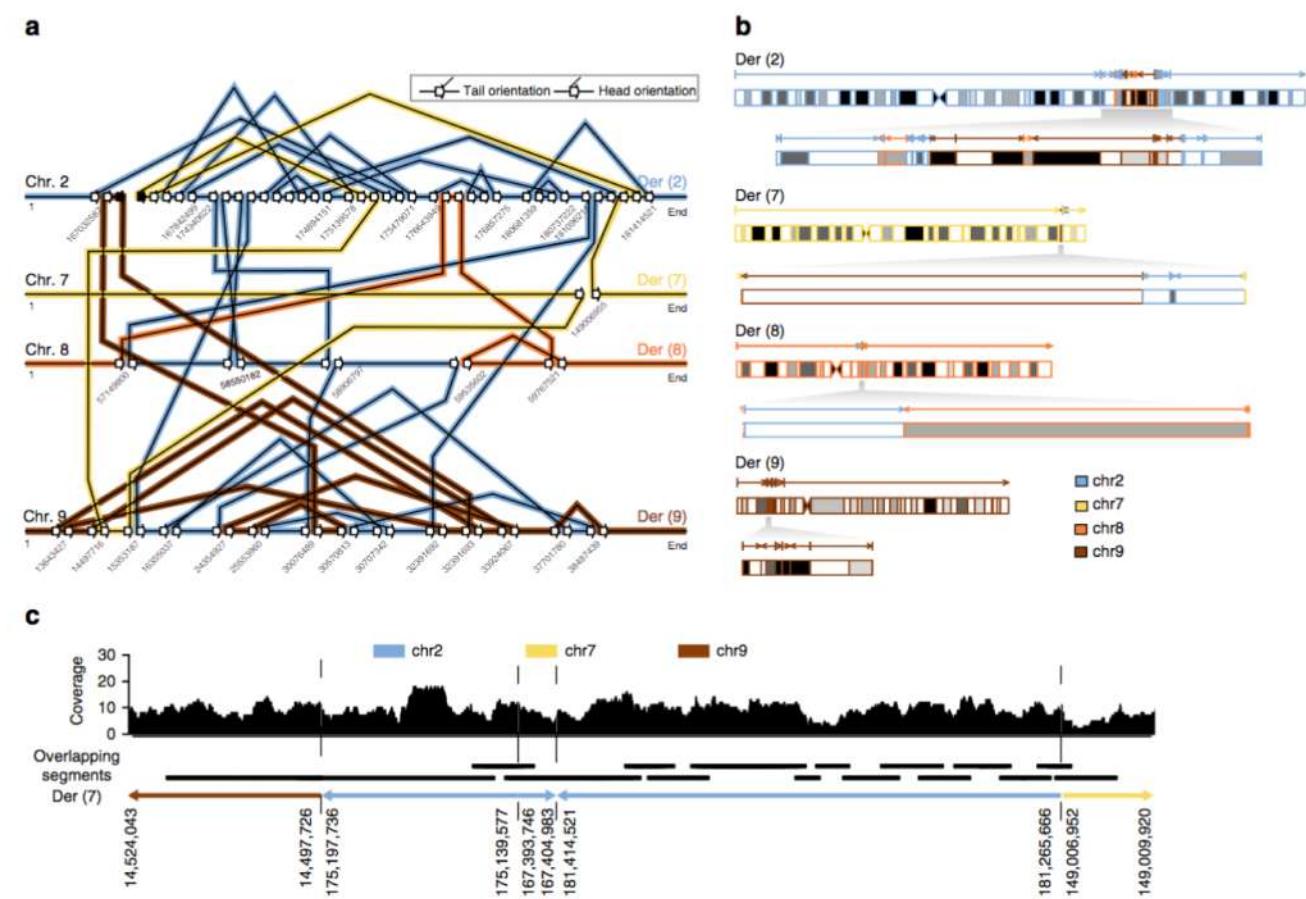
For instance, ...we called 33,653 deletions  $\geq 50$  bp, whereas the 1000 Genomes Project identified 42,279 such variants in 25 times more individuals who were more diverse than our study population.



# Mapping and phasing of structural variation in patient genomes using nanopore sequencing

Mircea Cretu Stancu<sup>1</sup>, Markus J. van Roosmalen<sup>1</sup>, Ivo Renkens<sup>1</sup>, Marleen M. Nieboer<sup>1</sup>, Sjors Middelkamp<sup>1</sup>, Joep de Ligt<sup>1</sup>, Giulia Pregno<sup>2</sup>, Daniela Giachino<sup>1</sup>, Giorgia Mandrile<sup>2</sup>, Jose Espejo Valle-Inclan<sup>1</sup>, Jerome Korzelius<sup>1</sup>, Ewart de Bruijn<sup>1</sup>, Edwin Cuppen<sup>3</sup>, Michael E. Talkowski<sup>4,5,6</sup>, Tobias Marschall<sup>1</sup>, Jeroen de Ridder<sup>1</sup> & Wigard P. Kloosterman<sup>1</sup>

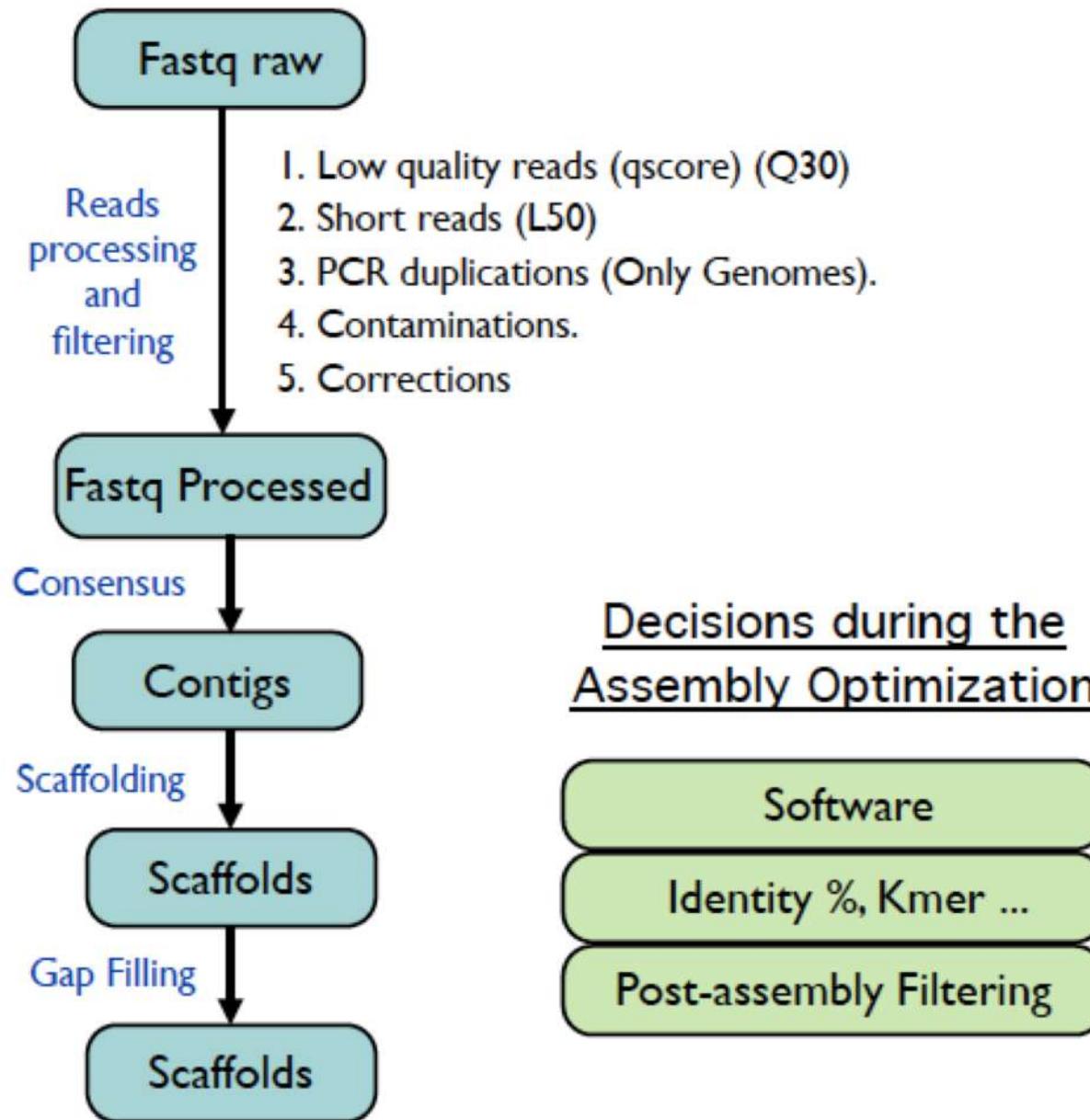
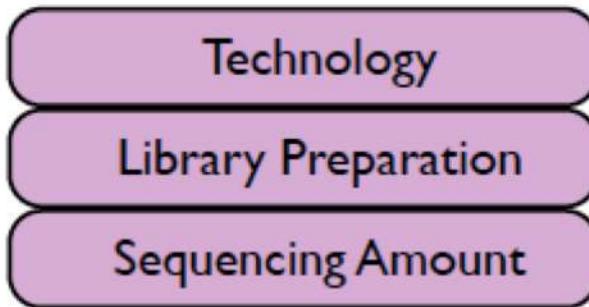
- long reads are superior to short reads with regard to detection of de novo chromothripsis rearrangements.
- long reads also enable efficient phasing of genetic variations, which we leveraged to determine the parental origin of all de novo chromothripsis breakpoints and to resolve the structure of these complex rearrangements.



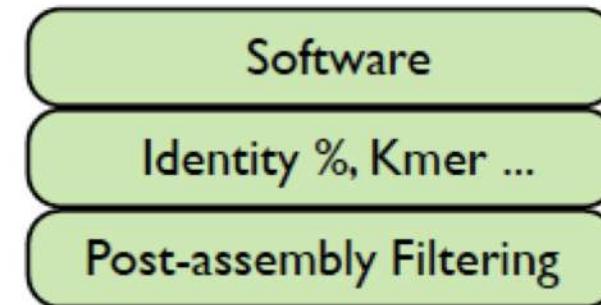
# Summary

# Overview of a sequencing project

## Decisions during the Experimental Design



## Decisions during the Assembly Optimization



# Things to consider

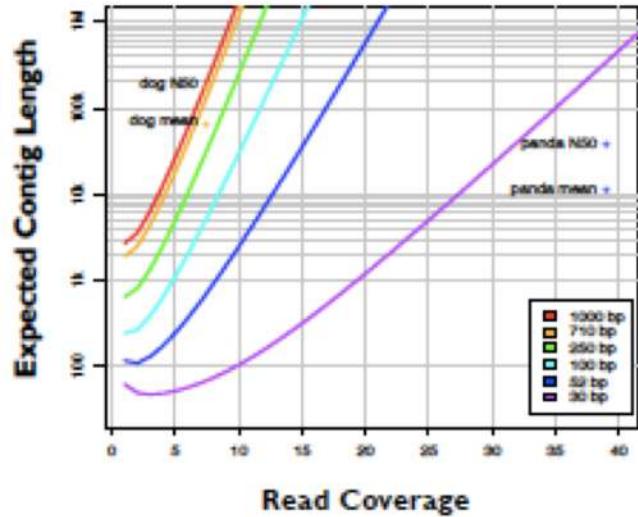
- \$\$\$\$\$\$\$\$\$
- **Project type**
  - virus, bacteria, eukaryote, meta-genome
- **Goals**
  - Just an assembly to showcase the world?
  - Sequence pandemic species = conservation? (No right or wrong answer)
  - Any biological question?
  - Why *de novo* sequence a species?
- **Hardware**
  - You need CPUs, but RAM is more important
    - Imagine storing all the hashes or kmers
  - This may change depending on nature of data (all long reads within n years?)

# Consider technologies and experiments

- Use multiple techniques to answer your questions
  - Long read only
  - Long read + Hi-C
  - Long read + optical maps + Hi-C
  - 10X linked reads
  - 10X + Hi-C
  - Single cell?
  - Experimental advancement?
- Sometimes limitations becomes experiment preparation rather than the technological one

# Ingredients for a good assembly

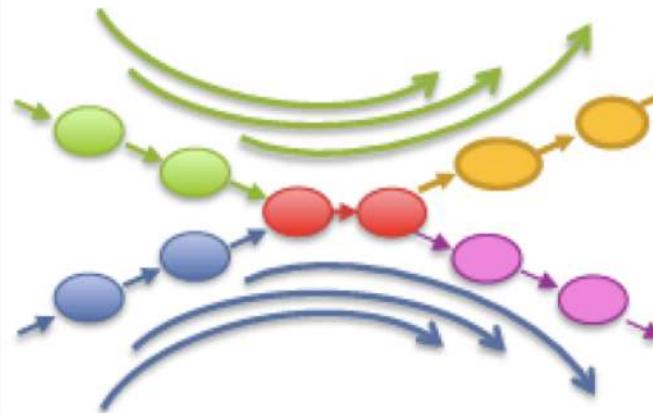
## Coverage



### High coverage is required

- Oversample the genome to ensure every base is sequenced with long overlaps between reads
- Biased coverage will also fragment assembly

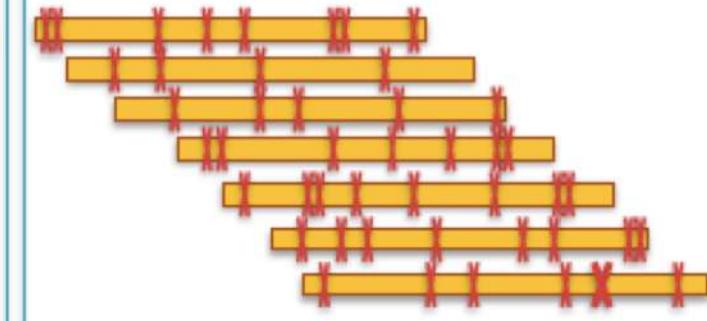
## Read Length



### Reads & mates must be longer than the repeats

- Short reads will have *false overlaps* forming hairball assembly graphs
- With long enough reads, assemble entire chromosomes into contigs

## Quality



### Errors obscure overlaps

- Reads are assembled by finding kmers shared in pair of reads
- High error rate requires very short seeds, increasing complexity and forming assembly hairballs

# Assembly competitions

Downloaded from [genome.cshlp.org](http://genome.cshlp.org) on March 8, 2016 - Published by Cold Spring Harbor Laboratory Press

## Resource

---

### Assemblathon 1: A competitive assessment of *de novo* short read assembly methods

Bradnam *et al.* *GigaScience* 2013, **2**:10  
<http://www.gigasciencejournal.com/content/2/1/10>



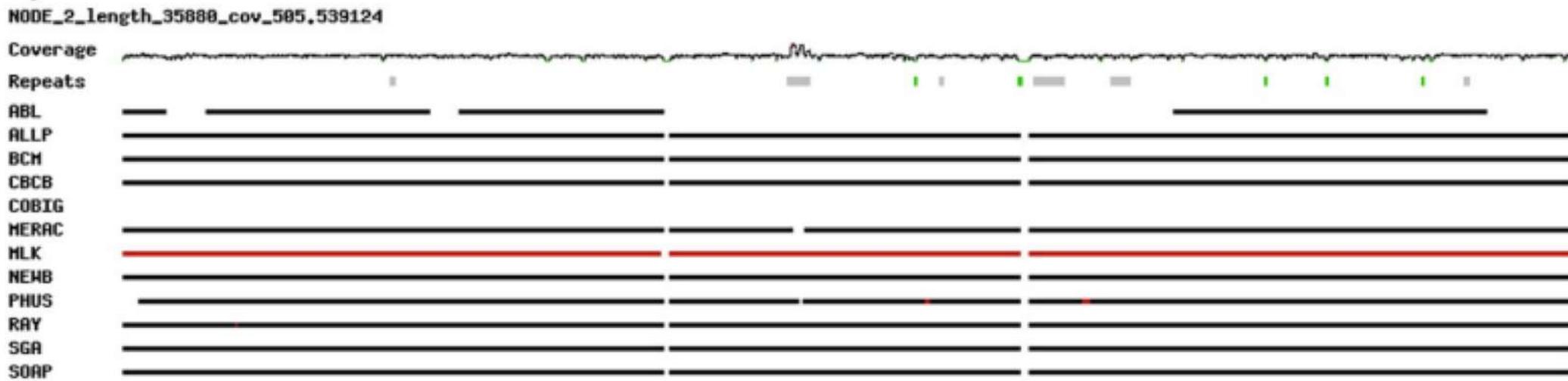
RESEARCH

Open Access

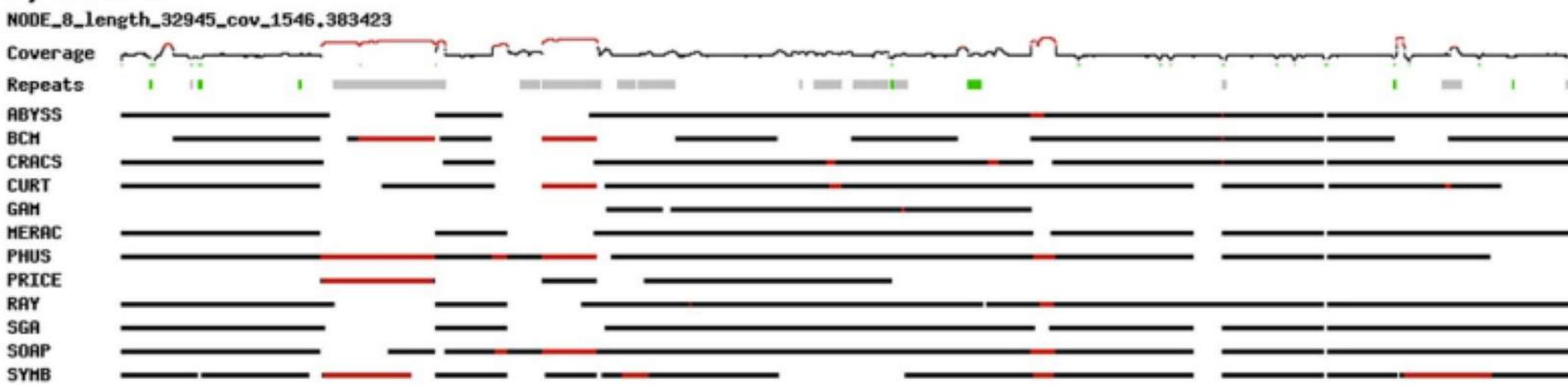
### Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species

# No assembly is best..

## A) Bird



## B) Snake



# Things to consider

Problem	Description
Cost	In 2014 Illumina claimed the \$1,000 genome barrier had been broken (if you first spend ~\$10 million on hardware).
Library prep	A critical, and often overlooked, step in the process.
Sequence diversity	Illumina, 454, Ion Torrent, PacBio, Oxford Nanopore: which mix of sequence data will you be using?
Hardware	Some genome assemblers have very high CPU/RAM requirements. Might need specialized cluster.
Expertise	Not always easy to even get assembly software installed, let alone understand how to run it properly.
Software	There is a <u>lot</u> of choice out there.

Credit: Keith Bradnam

# Not covered but should be

Alignment method in overlap graph

String graph (reduced overlap graph)

Shortest Superstring Problem (SSP)

Hamiltonian path

Choice of kmers in DBG

Bloom filter