Cesky Krumlov
Short read alignment:
An introduction

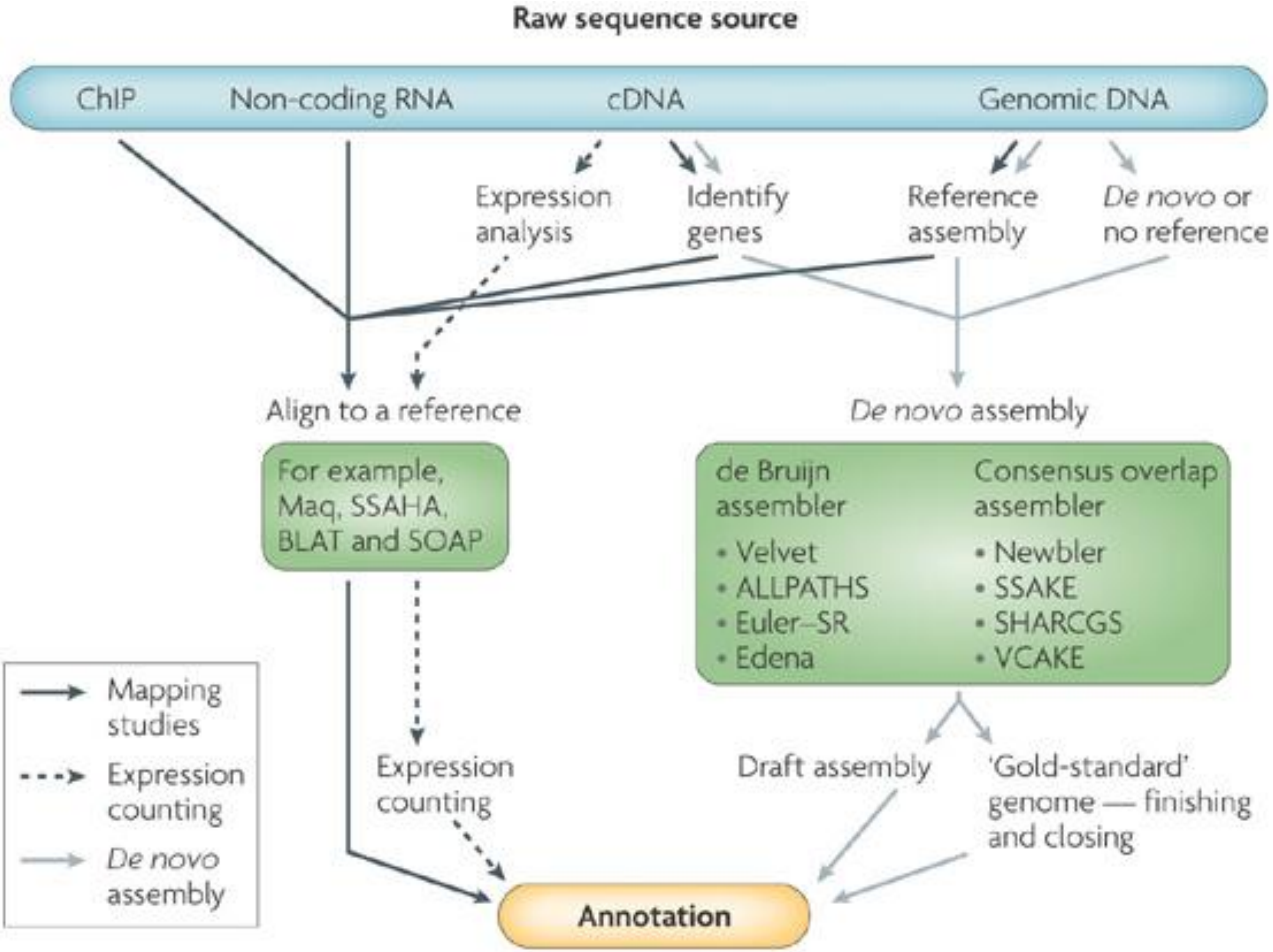
Dr Konrad Paszkiewicz

University of Exeter, UK

k.h.paszkiewicz@exeter.ac.uk

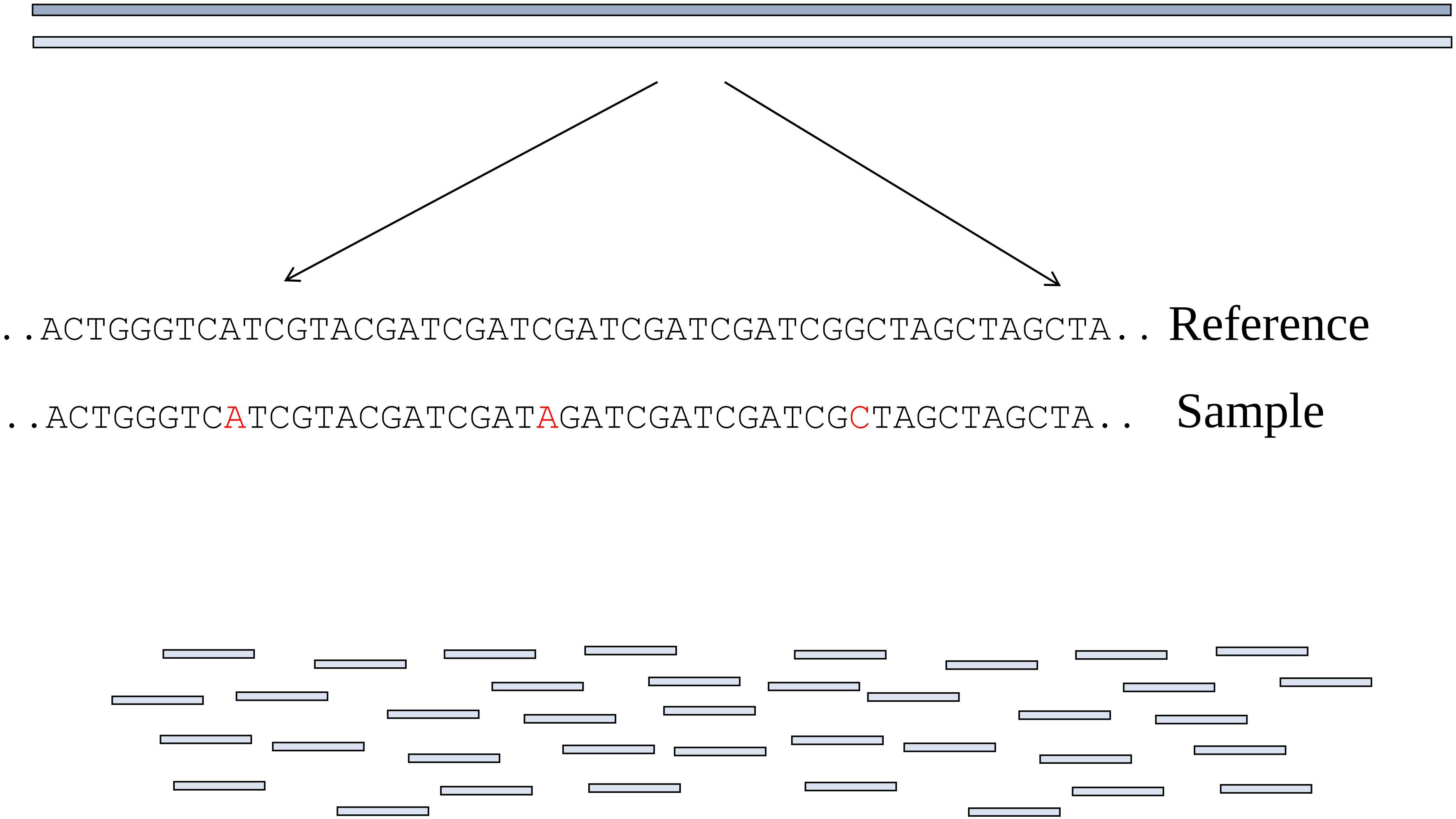
Contents

- **Alignment algorithms for short-reads**
  - Background – Blast (why can't we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Suffix/Prefix Tries
  - Indels
  - Other alignment considerations
  - Typical alignment pipeline
  - New methods of SNP calling



Nature Reviews | Microbiology

Alignment of reads to a reference



..ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA.. Reference

..ACTGGGTCATCGTACGATCGATAGATCGATCGATCGCTAGCTAGCTA.. Sample

Why is short read alignment hard?

The shorter a read, the less likely it is to have a unique match to a reference sequence

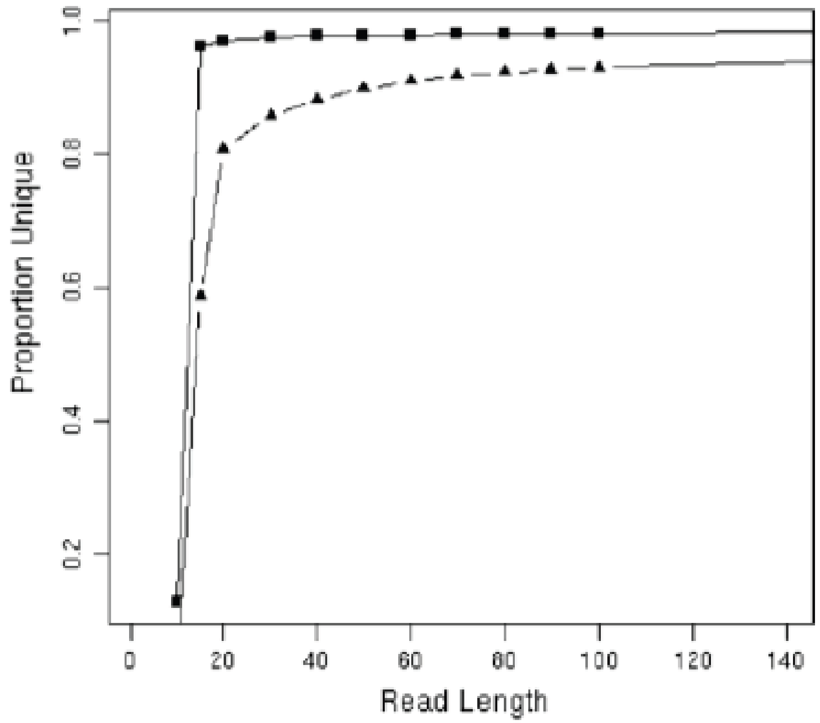


Fig. 1 The proportion of unique sequence in the *Streptococcus suis* (squares) and *Mus musculus* (triangles) genomes for varying read lengths. This graph indicates that read length has a critical affect on the ability to place reads uniquely to the genome

Why do we generate short reads?

- Sanger reads lengths ~ 800-2000bp

- Generally we define short reads as anything below 200bp
    - Illumina (100bp – 250bp)
    - SoLID (80bp max)
    - Ion Torrent (200-400bp max...)
    - Roche 454 – 400-800bp

- Even with these platforms it is cheaper to produce short reads (e.g. 50bp) rather than 100 or 200bp reads

- Diminishing returns:
    - For some applications 50bp is more than sufficient
        - Resequencing of smaller organisms
        - Bacterial de-novo assembly
        - ChIP-Seq
        - Digital Gene Expression profiling
        - Bacterial RNA-seq

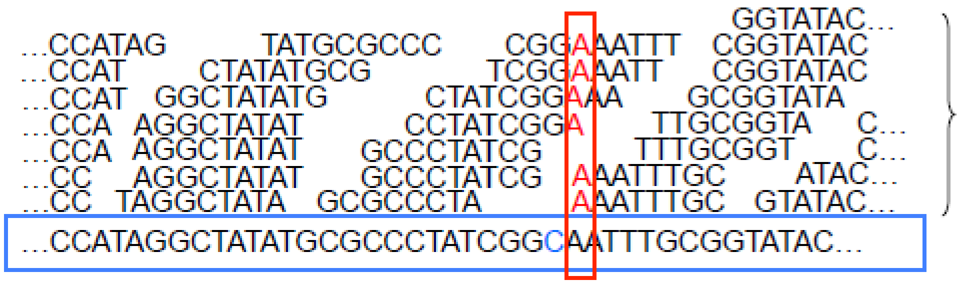
Short read alignment applications

Genotyping:

Methylation

SNPs

Indels

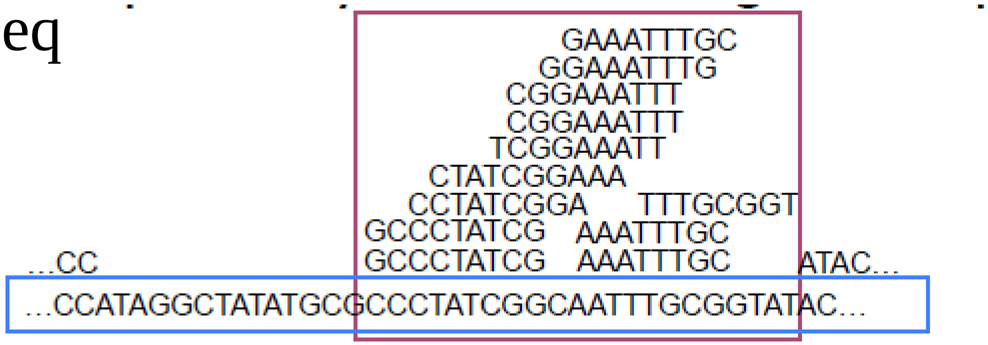


```
                                         GGTATAC...
...CCATAG     TATGCGCCC     CGGAAATTT  CGGTATAC
...CCAT   CTATATGCG        TCGGAAATT   CGGTATAC
...CCAT GGCTATATG      CTATCGGAAA     GCGGTATA
...CCA AGGCTATAT      CCTATCGGA     TTGCGGTA    C...
...CCA AGGCTATAT   GCCCTATCG       TTTGCGGT     C...
...CC  AGGCTATAT   GCCCTATCG   AAATTTGC     ATAC...
...CC TAGGCTATA  GCGCCCTA      AAATTTGC   GTATAC...
...CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC...
```

Classify and measure peaks:

ChIP-Seq

RNA-Seq

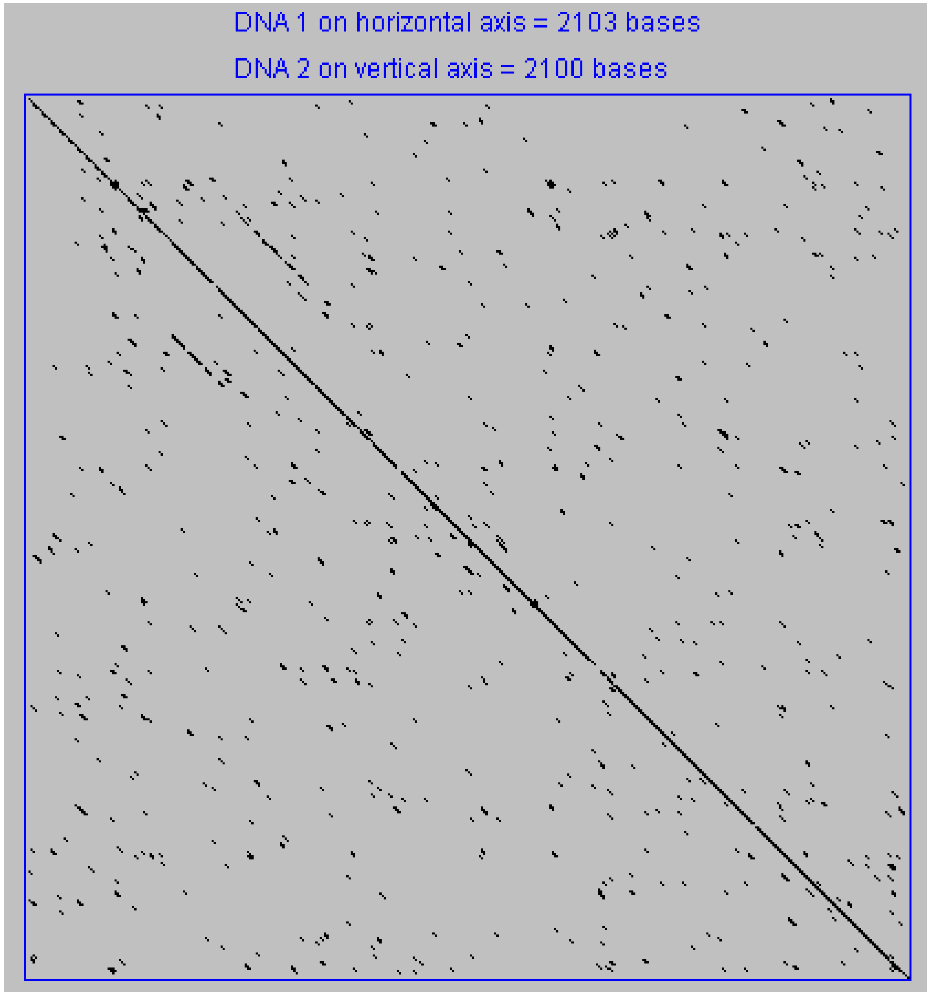


```
                               GAAATTTGC
                             GGAAATTTG
                           CGGAAATTT
                           CGGAAATTT
                          TCGGAAATT
                       CTATCGGAAA
                      CCTATCGGA    TTTGCGGT
                   GCCCTATCG   AAATTTGC
...CC              GCCCTATCG   AAATTTGC     ATAC...
...CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC...
```

Contents

- **Alignment algorithms for short-reads**
  - **Background – Blast (why can’t we use it?)**
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Indel detection
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
  - New methods of SNP calling

Dot Matrix Method
- Aligning by eye



Sequence Alignment

```
ATCGAT-ACG
ATGGATTACG
```

3 possibilities

Match	Mismatch	Indel
...A... | ...A...	...C... ...G...	...-... ...T...

Alignment cost

Points for a matching letter:	1
Points for a non-matching letter:	-1
Points for inserting a gap:	-2

Global Pair-wise Alignment

ATCGATACG, ATGGATTACG

```
ATCGAT-ACG
|| ||| |||
ATGGATTACG
```

Matches:	+1	+1		+1	+1	+1		+1	+1	+1	= +8
Mismatches:			-1								= -1
Gaps:							-2				= -2

Total score = +5

Dynamic Programming

Global alignment (Needleman-Wunsch) algorithm

Example – align GATC to GAC

0	-	G	A	T	C
-	0				
G					
A					
C					

Dynamic Programming

Global alignment (Needleman-Wunsch) algorithm

■	-	G	A	T	C
-	0 →	-2 →	-4 →	-6 →	-8
G					
A					
C					

Points for match = +1
Points for mismatch = -1
Points for a gap insertion = -2

Dynamic Programming

Global alignment (Needleman-Wunsch) algorithm

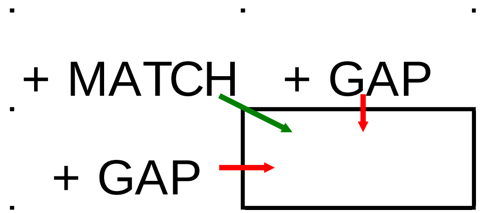
[redacted]	-	G	A	T	C
-	0 →	-2 →	-4 →	-6 →	-8
G	↓ -2				
A	↓ -4				
C	↓ -6				

Points for match = +1
Points for mismatch = -1
Points for a gap insertion = -2

Dynamic Programming

Global alignment (Needleman-Wunsch) algorithm

	-	G	A	T	C
-	0 +1	-2	-4	-6	-8
G	-2	Max= 1			
A	-4				
C	-6				



+ MATCH

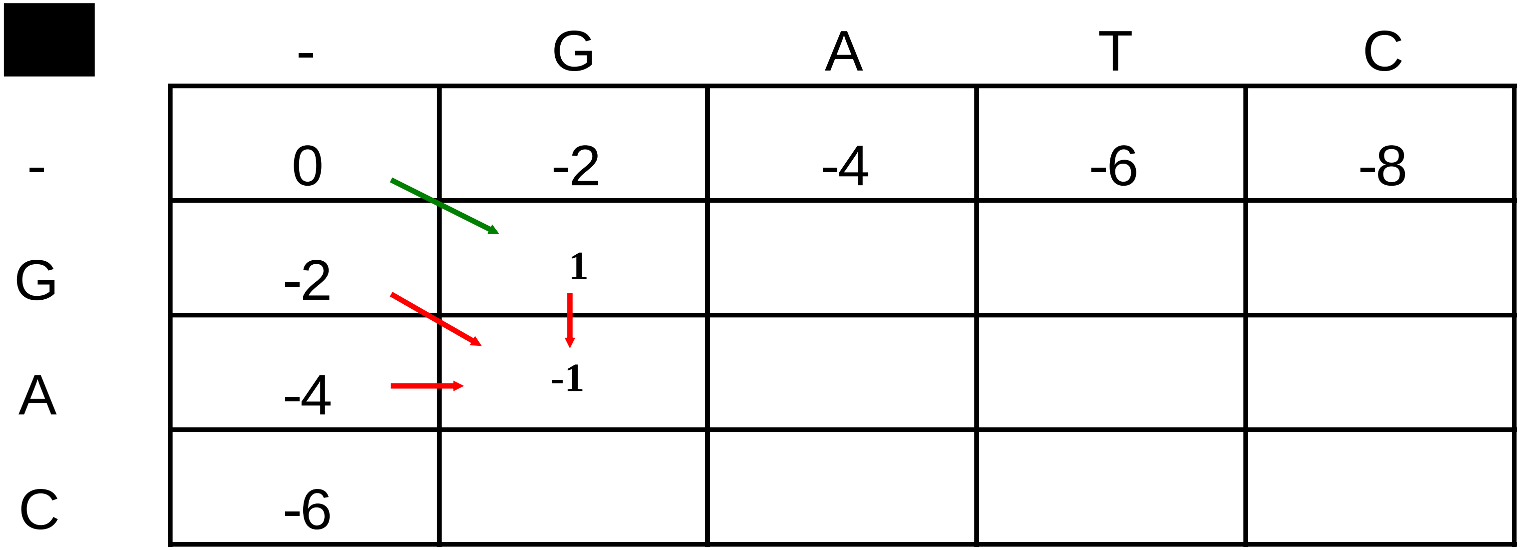
+ GAP

+ GAP

Points for match = +1
Points for mismatch = -1
Points for a gap insertion = -2

Dynamic Programming

Global alignment (Needleman-Wunsch) algorithm

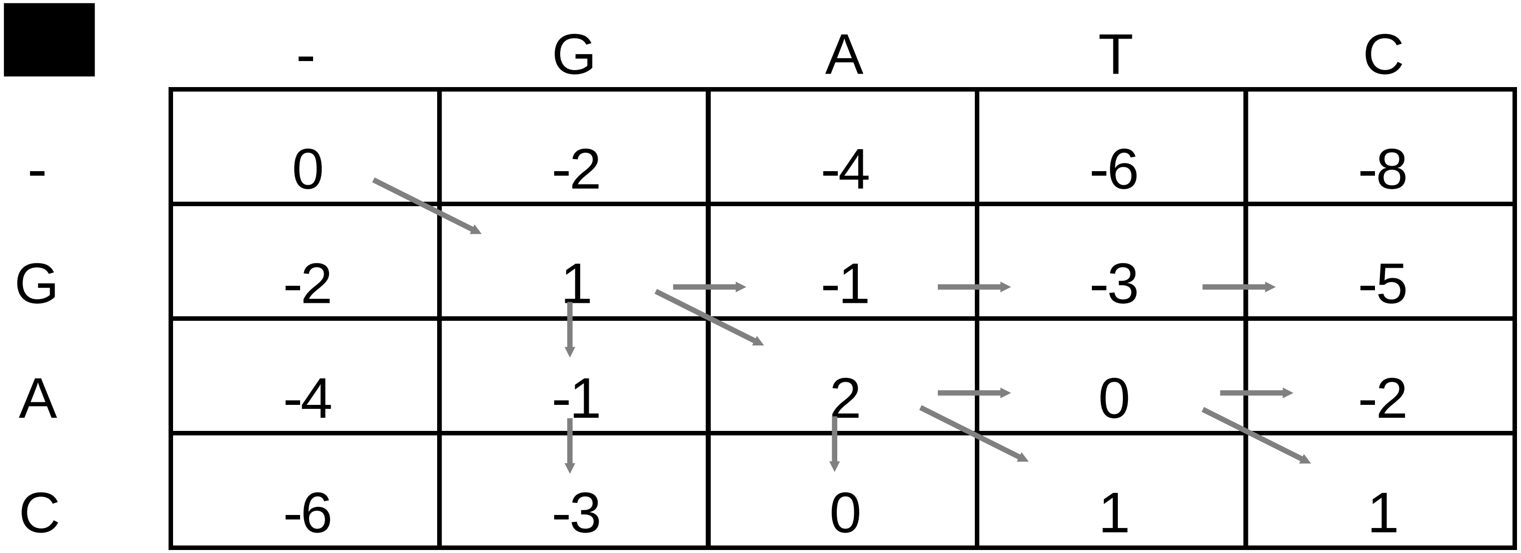


	-	G	A	T	C
-	0	-2	-4	-6	-8
G	-2	1			
A	-4	-1			
C	-6				

Points for match = +1
Points for mismatch = -1
Points for a gap insertion = -2

Dynamic Programming

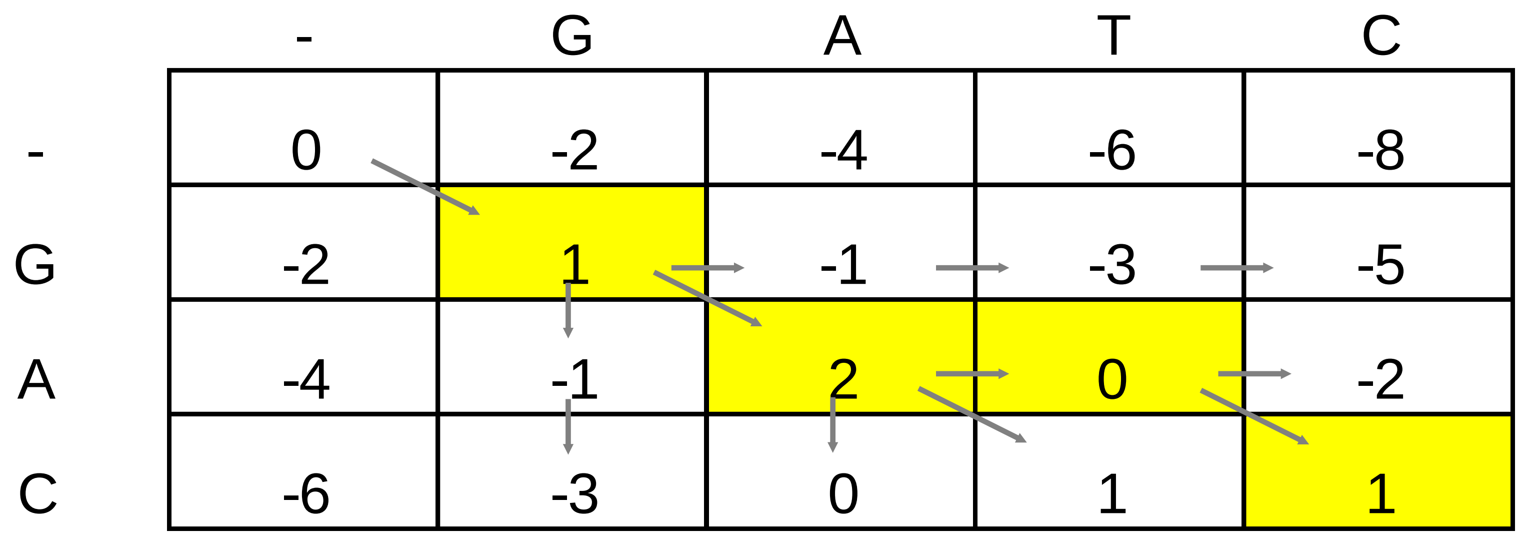
Global alignment (Needleman-Wunsch) algorithm



[redacted]	-	G	A	T	C
-	0	-2	-4	-6	-8
G	-2	1	-1	-3	-5
A	-4	-1	2	0	-2
C	-6	-3	0	1	1

Points for match = +1
Points for mismatch = -1
Points for a gap insertion = -2

Backtracking and final alignment



	-	G	A	T	C
-	0	-2	-4	-6	-8
G	-2	1	-1	-3	-5
A	-4	-1	2	0	-2
C	-6	-3	0	1	1

```
GATC
|| |
GA-C
```

Dynamic programming

- Guaranteed to give you the best possible alignment

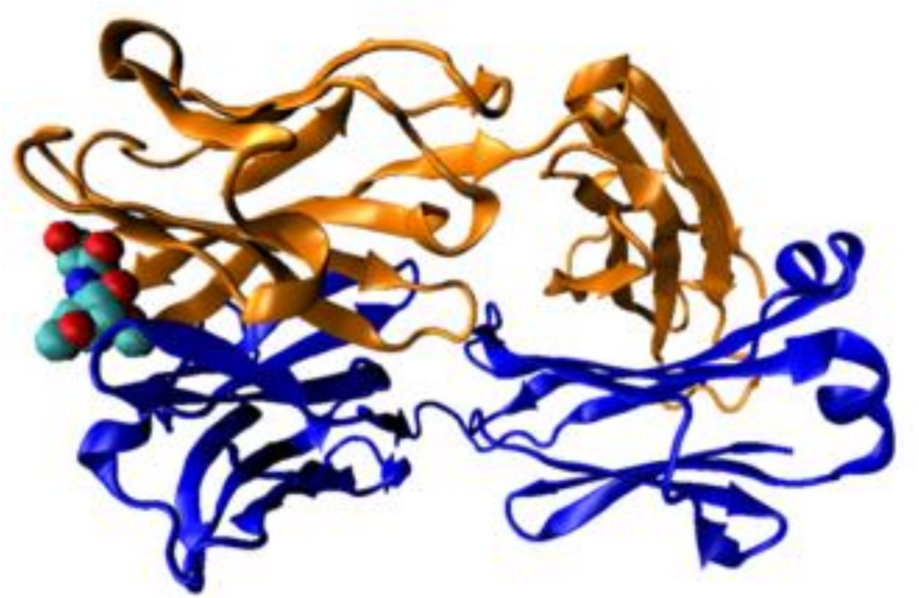
- In biology, this algorithm is very inefficient because most sequences will not align to each other

- Takes a long time to run

BLAST –
Basic Local Alignment Search Tool

Background – BLAST

- Primarily designed to identify homologous sequences
  - Blast is a hashed seed-extend algorithm
  - Functional conservation
  - Only some parts of a sequence are usually constrained



BLAST - Original version

Example:

Seed size = 4,

No mismatches in seed

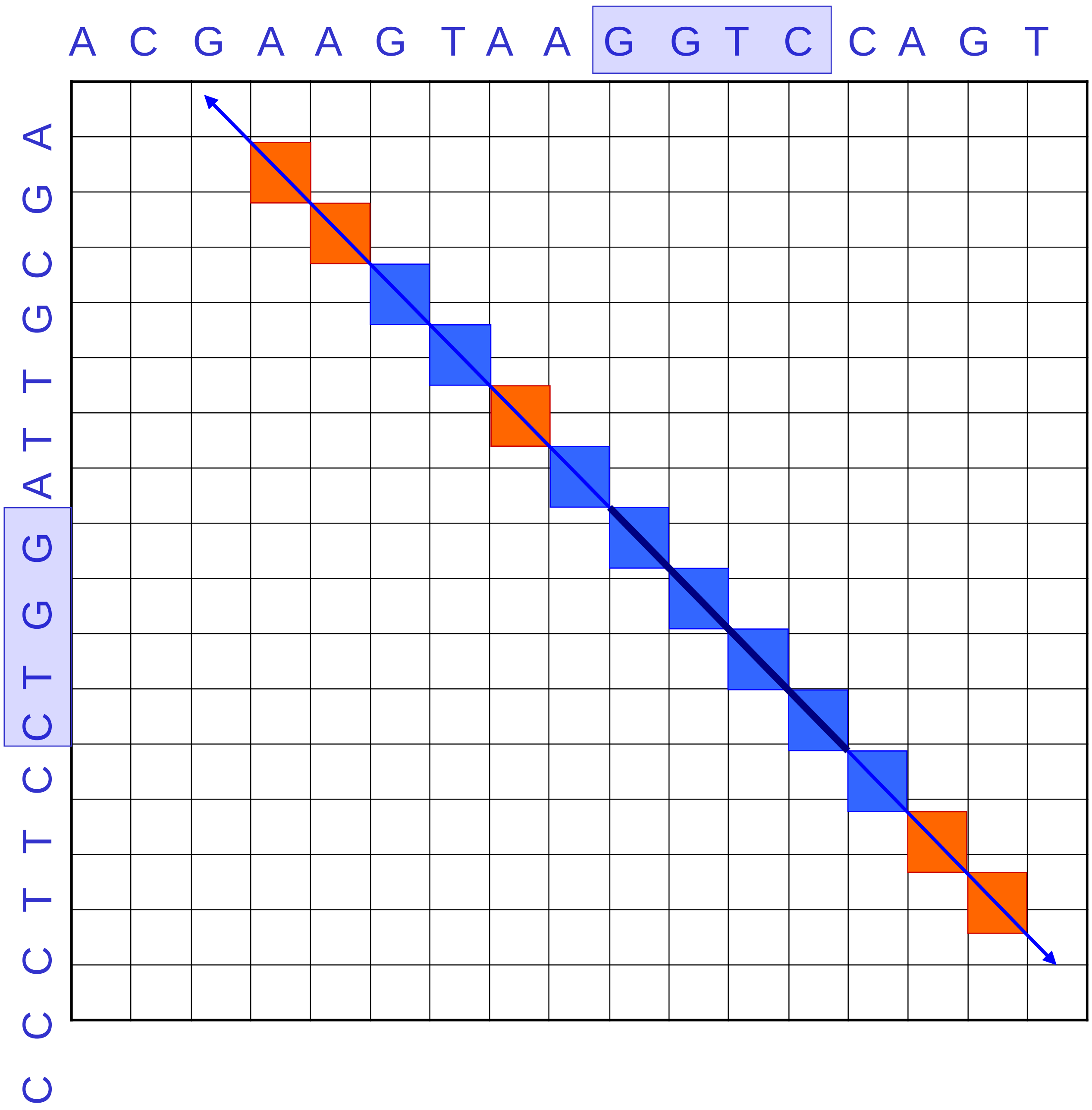
The matching word GGTC initiates an alignment

Extension to the left and right with no gaps until alignment score falls below 50%

Output:

GTAAGGTCC

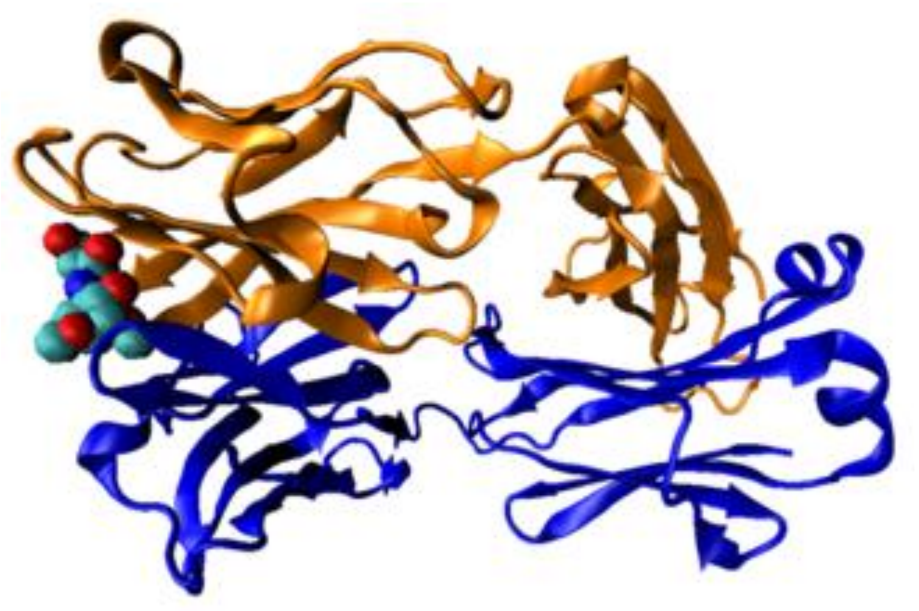
GTTAGGTCC


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

BLAST - Original algorithm

- Finding seeds significantly increases the speed of BLAST compared to doing a full local alignment over a whole sequence

- BLAST first finds highly conserved or identical sequences which are then extended with a local alignment.



BLAST – Speed (or lack thereof)

- Typically BLAST will take approximately 0.1 – 1 second to search 1 sequence against a database
- Depends on size of database, e-value cutoff and number of hits to report selected
- 60 million reads equates to 70 CPU days!
- Even on multi-core systems this is too long!
- Especially if you have multiple samples!
- This is still true of FPGA and SIMD (vectorised) implementations of BLAST

When NOT to use *BLAST*

- A typical situation: you have lots DNA sequences and want to extend it or find where on a genome it maps.
- In other words, you want an **exact** or **near-exact** match to a sequence that is part of an **assembled genome**.
- Short reads require very fast algorithms for finding near-exact matches in genomic sequences:
  - BLAT
    - Highly recommended: the BLAT paper (Kent WJ (2003) *Genome Res* 12:656-64) - it is famous for its unorthodox writing style
  - SOAP
  - Bowtie/Bowtie 2
  - MAQ
  - BWA
  - Shrimp2

Contents

- **Alignment algorithms for short-reads**
  - Background – Blast (why can’t we use it?)
  - **Adapting hashed seed-extend algorithms to work with shorter reads**
  - Indel detection
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
  - New methods of SNP calling

Adapting hashed seed-extend algorithms to work with shorter reads

- Improve seed matching sensitivity
  - Allow mismatches within seed
    - BLAST
  - Allow mismatches + Adopt spaced-seed approach
    - ELAND, SOAP, MAQ, RMAP, ZOOM
  - Allow mismatches + Spaced-seeds + Multi-seeds
    - SSAHA2, BLAT, ELAND2
- Above and/or Improve speed of local alignment for seed extension
  - Single Instruction Multiple Data
    - Shrimp2, CLCBio
  - Reduce search space to region around seed

Hashed seed-extend algorithms

- These are most similar to BLAST
- Are not designed to work with large databases

- **2 step process**
  - Identify a match to the seed sequence in the reference
  - Extend match using sensitive (but slow) Smith-Waterman algorithm (dynamic programming)

Seed-extend algorithm

Reference sequence:

...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...

Short read:

GTCATCGTACGATCGATAGATCGATCGATCGGCTA

Note that the short read has 1 difference wrt to reference

Seed-extend algorithm

Reference sequence:

...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...

Short read:

GTCATCGTACG	ATCGATAGATCG	ATCGATCGGCTA
11bp word	11bp word	11bp word

The algorithm will try to match each word to the reference. If there is a match at with any single word it will perform a local alignment to extend the match

Seed-extend algorithm

Reference sequence:

Seed

Extend with Smith Waterman

...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...

GTCATCGTACGATCGAACGATCGATCGATCGGCTA

Short read:

GTCATCGTACG ATCGATAGATCG ATCGATCGGCTA

Here the algorithm is able to match the short read with a word length of 11bp

Seed-extend algorithm

Reference sequence:

...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...

Short read:

GTCATCGTACGATCGATCGATCGATCGATCGGCAA

Note that the short read has 3 differences
Possibly sequencing errors, possibly SNPs

Seed-extend algorithm

Reference sequence:

```
...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...
```

Short read:

GTCATCGTACG	ATCGATCGATCG	ATCGATCGGCAA
11bp word	11bp word	11bp word

Note that the short read has 3 differences

Seed-extend algorithm

Reference sequence:

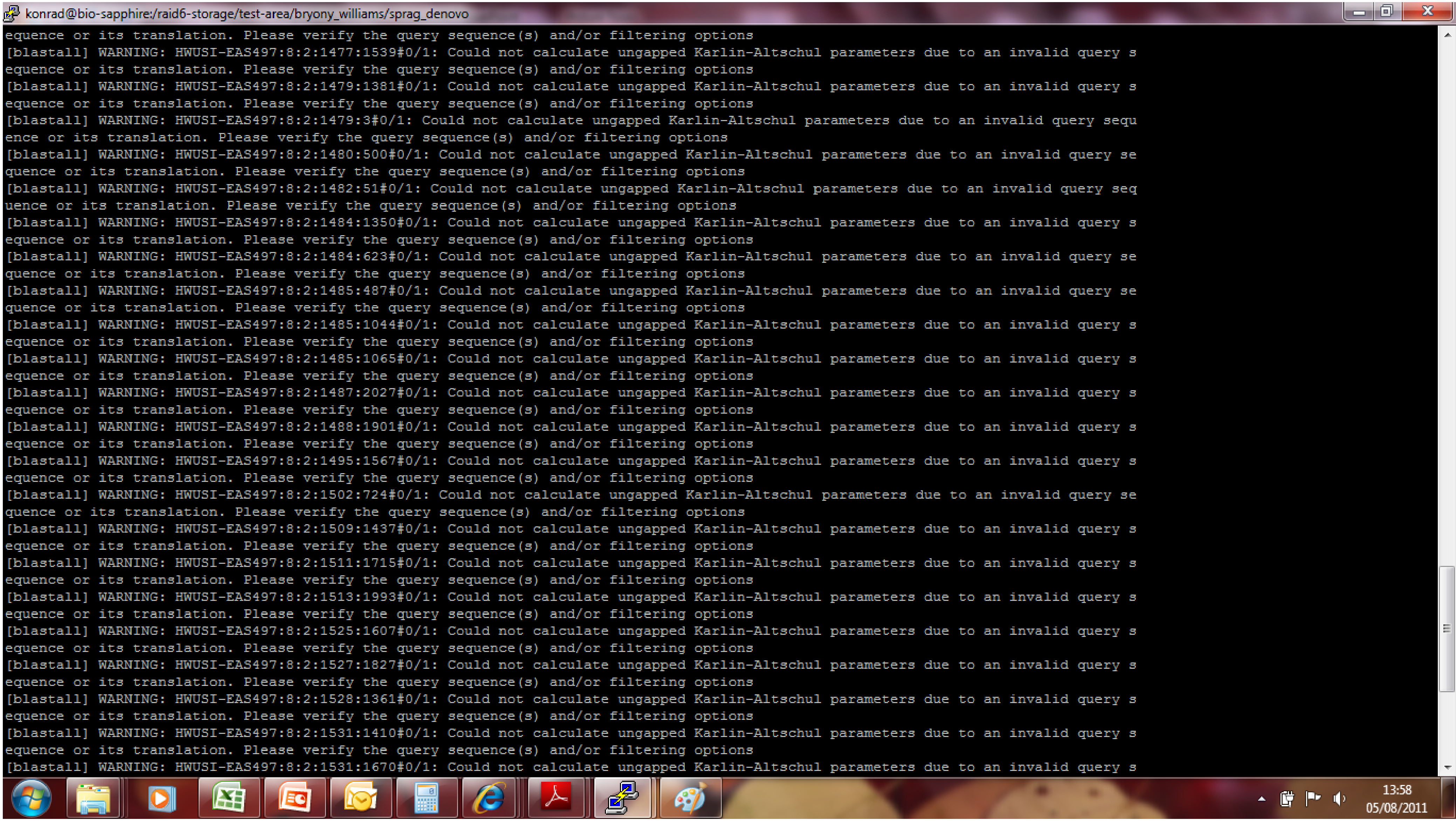
...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...

Short read:

GTCATCGTACG ATCGATCGATCG ATCGATCGGCAA

No seeds match

Therefore the algorithm would find no hits at all!



konrad@bio-sapphire:/raid6-storage/test-area/bryony_williams/sprag_denovo

```
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1477:1539#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1479:1381#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1479:3#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query sequ
ence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1480:500#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query se
quence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1482:51#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query seq
uence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1484:1350#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1484:623#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query se
quence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1485:487#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query se
quence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1485:1044#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1485:1065#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1487:2027#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1488:1901#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1495:1567#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1502:724#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query se
quence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1509:1437#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1511:1715#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1513:1993#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1525:1607#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1527:1827#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1528:1361#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1531:1410#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
equence or its translation. Please verify the query sequence(s) and/or filtering options
[blastall] WARNING: HWUSI-EAS497:8:2:1531:1670#0/1: Could not calculate ungapped Karlin-Altschul parameters due to an invalid query s
```

Adapting hashed seed-extend algorithms to work with shorter reads

- Improve seed matching sensitivity
  - **Allow mismatches within seed**
    - **BLAST**
  - Allow mismatches + Adopt spaced-seed approach
    - ELAND, SOAP, MAQ, RMAP, ZOOM
  - Allow mismatches + Spaced-seeds + Multi-seeds
    - SSAHA2, BLAT, ELAND2
- Above and/or Improve speed of local alignment for seed extension
  - Single Instruction Multiple Data
    - Shrimp2, CLCBio
  - Reduce search space to region around seed

Adapting hashed seed-extend algorithms to work with shorter reads

- Improve seed matching sensitivity
  - **Allow mismatches within seed**
    - **BLAST**
  - **Allow mismatches + Adopt spaced-seed approach**
    - **ELAND, MAQ, RMAP, ZOOM**
  - Allow mismatches + Spaced-seeds + Multi-seeds
    - SSAHA2, BLAT, ELAND2
- Above and/or Improve speed of local alignment for seed extension
  - Single Instruction Multiple Data
    - Shrimp2, CLCBio
  - Reduce search space to region around seed

Consecutive seed

Consecutive seed 9bp with no mismatches:

ACTCCCATCGTCATCGTACTAGGGATCGTAACA   Reference sequence

CCACTGTCCTCCTACATAGGAACGA   SNP ‘heavy’ read

TC**A**TC**G**TAC

TC**C**TC**C**TAC

Cannot find seed match due to A->C SNP and G->C SNP

Even allowing for 2 mismatches in the seed - no seeds match.
No hits!

Spaced seeds

To increase sensitivity we can used spaced-seeds:

11111111111	Consecutive seed template with *length* 9bp
ACTATCATCGTACACAT	Reference
TCATCGTAC	Query

11001100110011001	Spaced-seed template with *weight* 9bp
ACTATCATCGTACACAT	Reference
ACTCTCACCGTACACAT	Query

Spaced seeds

Spaced seed with weight 9bp and no mismatches:

```
ACTCCCATTGTCATCGTACTTGGGATCGTAACA
    CCACTGTAATCGTACATGGGAACGA
```

Reference sequence

SNP ‘heavy’ read

```
CCATTGTCATCGTACAT
CCXXTGXXATXXTAXXT
```

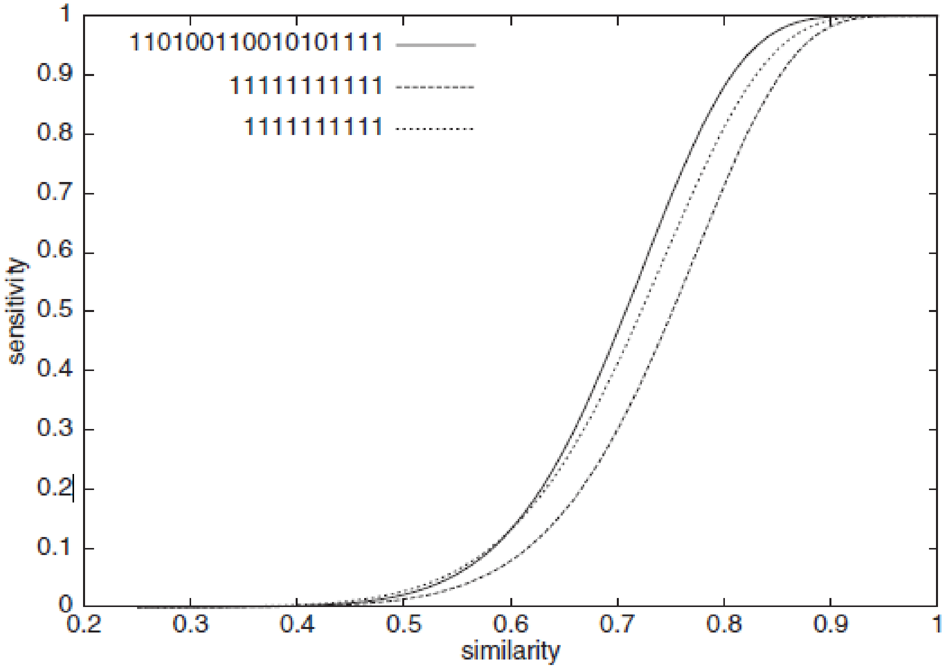
Despite SNPs – seed matched with 0 mismatches

Can now extend with Smith-Waterman or other local alignment

Spaced seeds

Spaced seeds:

- A seed template ‘111010010100110111’ is 55% more sensitive than BLAST’s default template ‘11111111111’ for two sequences of 70% similarity
- Typically seeds of length ~30bp and allow up to 2 mismatches in short read datasets



Ma, B. et al. PatternHunter. Bioinformatics Vol 18, No 3, 2002

Contents

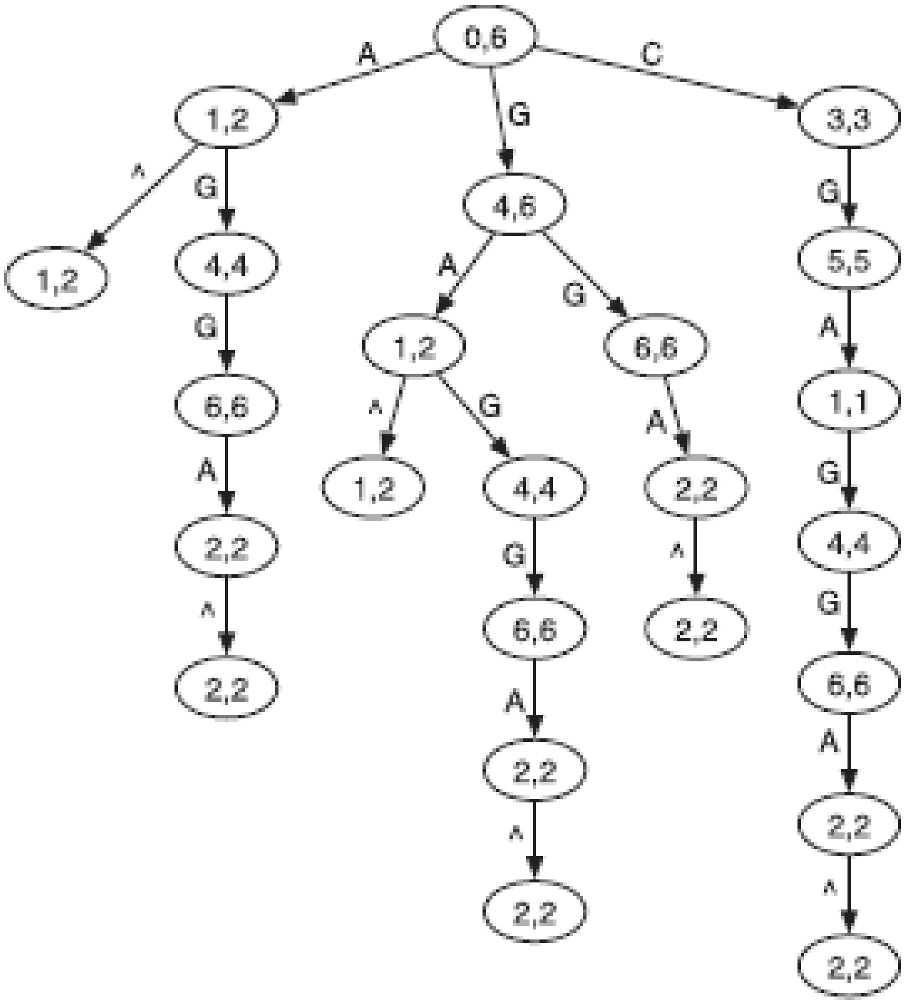
- **Alignment algorithms for short-reads**
  - Background – Blast (why can’t we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - **Suffix/Prefix Tries**
  - Indel detection
  - Other alignment considerations
  - Typical alignment pipeline
  - New methods of SNP calling

Suffix-Prefix Trie

- A family of methods which uses a Trie structure to search a reference sequence
  - Bowtie
  - BWA
  - SOAP version 2
- Trie – data structure which stores the suffixes (i.e. ends of a sequence)
- Key advantage over hashed algorithms:
  - Alignment of multiple copies of an identical sequence in the reference only needs to be done once
  - Use of an FM-Index to store Trie can drastically reduce memory requirements (e.g. Human genome can be stored in 2Gb of RAM)
  - Burrows Wheeler Transform to perform fast lookups

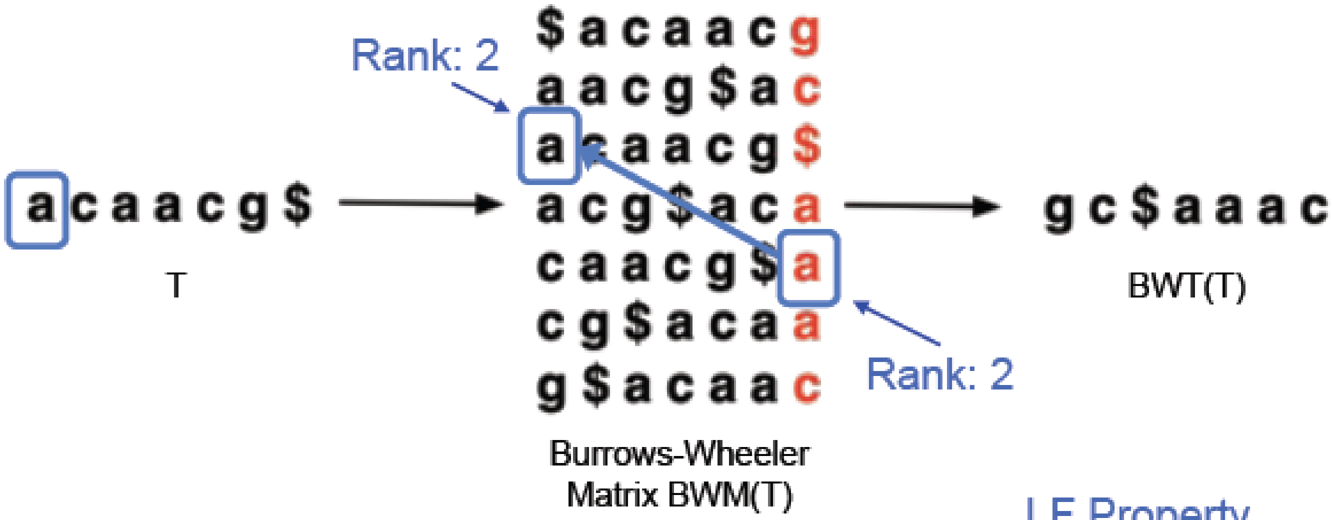
Suffix Trie

AGGAGC



Heng Li & Nils Homer. Sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics. Vol 11. No 5. 473 483, 2010

Suffix Trie



acaacg$

T

```
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac
```

Burrows-Wheeler
Matrix BWM(T)

gc$aaac

BWT(T)

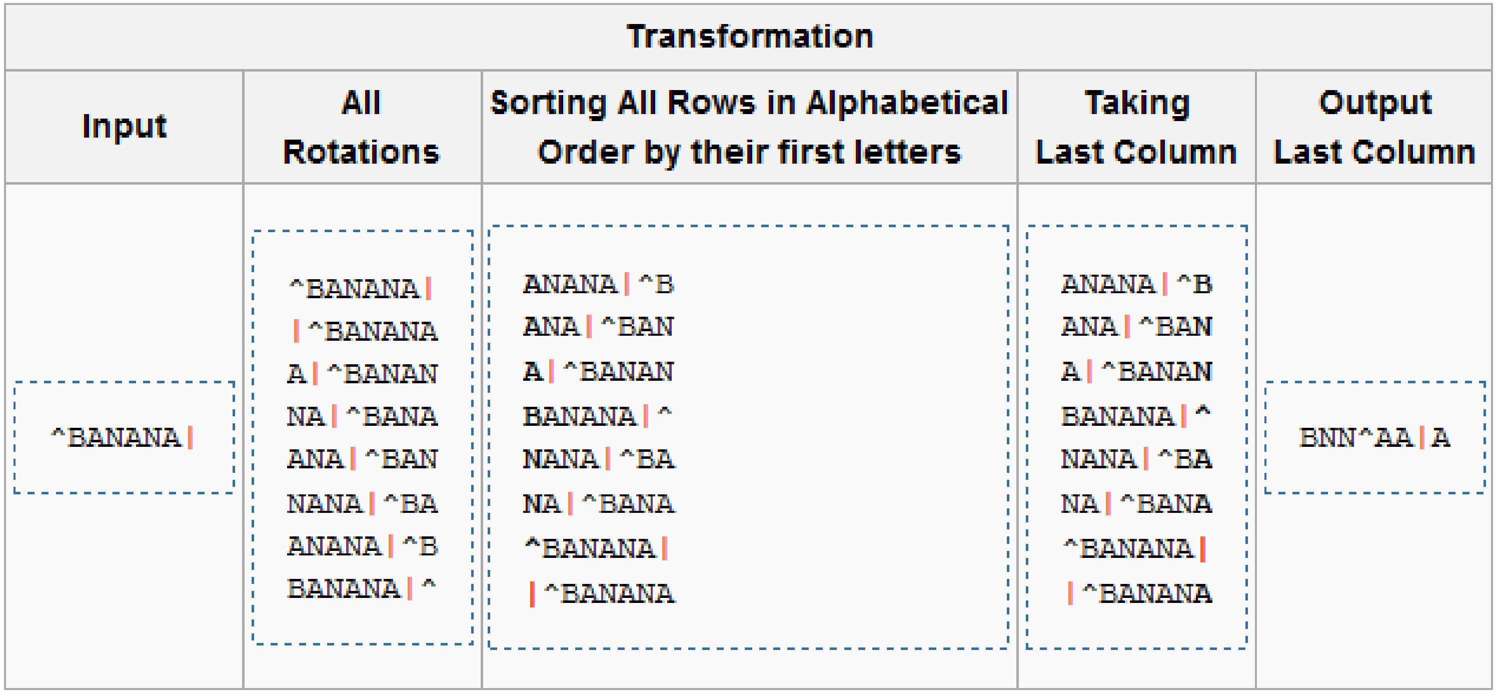
LF Property implicitly encodes Suffix Array

- BWT(T) is the index for T

A block sorting lossless data compression algorithm.
Burrows M, Wheeler DJ (1994) *Digital Equipment Corporation.* Technical Report 124

Burrows-Wheeler Algorithm

- Encodes data so that it is easier to compress
- Burrows-Wheeler transform of the word BANANA
- Can later be reversed to recover the original word



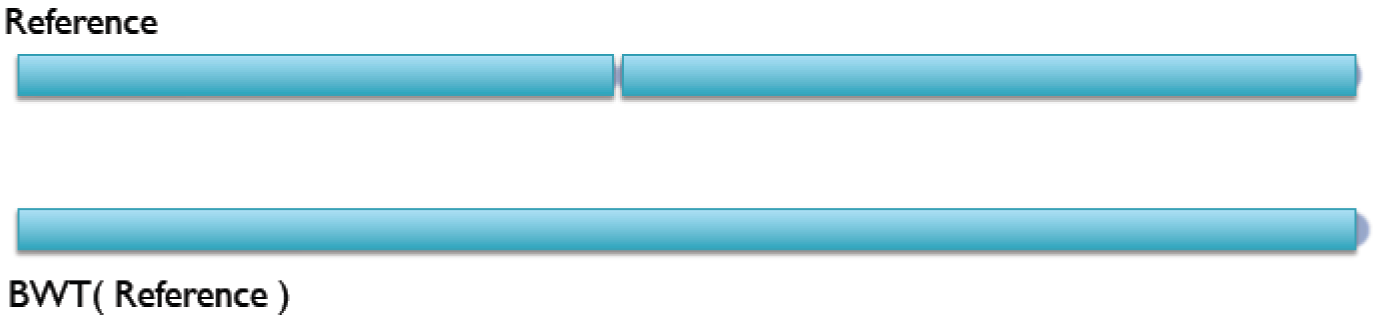
<table>
<tr><th colspan="5">Transformation</th></tr>
<tr><th>Input</th><th>All Rotations</th><th>Sorting All Rows in Alphabetical Order by their first letters</th><th>Taking Last Column</th><th>Output Last Column</th></tr>
<tr><td>^BANANA|</td><td>^BANANA|
|^BANANA
A|^BANAN
NA|^BANA
ANA|^BAN
NANA|^BA
ANANA|^B
BANANA|^</td><td>ANANA|^B
ANA|^BAN
A|^BANAN
BANANA|^
NANA|^BA
NA|^BANA
^BANANA|
|^BANANA</td><td>ANANA|^B
ANA|^BAN
A|^BANAN
BANANA|^
NANA|^BA
NA|^BANA
^BANANA|
|^BANANA</td><td>BNN^AA|A</td></tr>
</table>

More Burrows-Wheeler

Input	SIX.MIXED.PIXIES.SIFT.SIXTY.PIXIE.DUST.BOXES
Burrows-Wheeler Output	TEXYDST.E.IXIXIXXSSMPPS.B..E.S.EUSFXDIIOIIIT

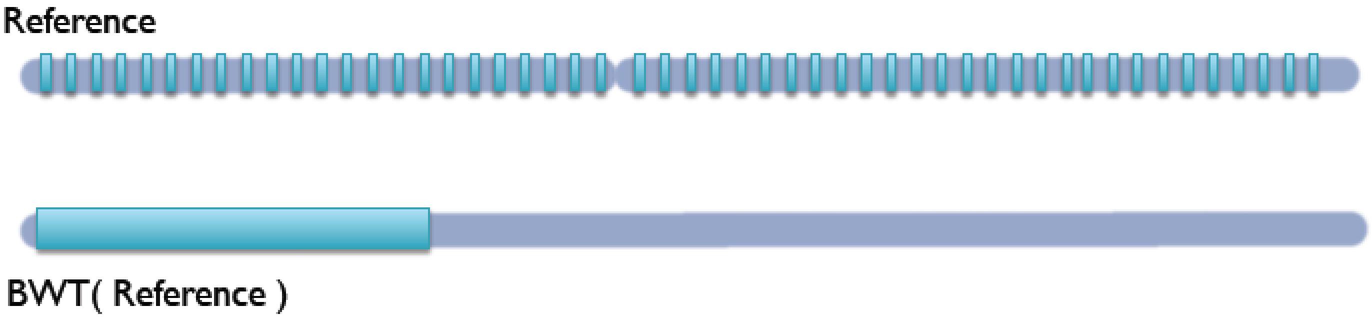
Repeated characters mean that it is easier to compress

Bowtie/Soap2 example



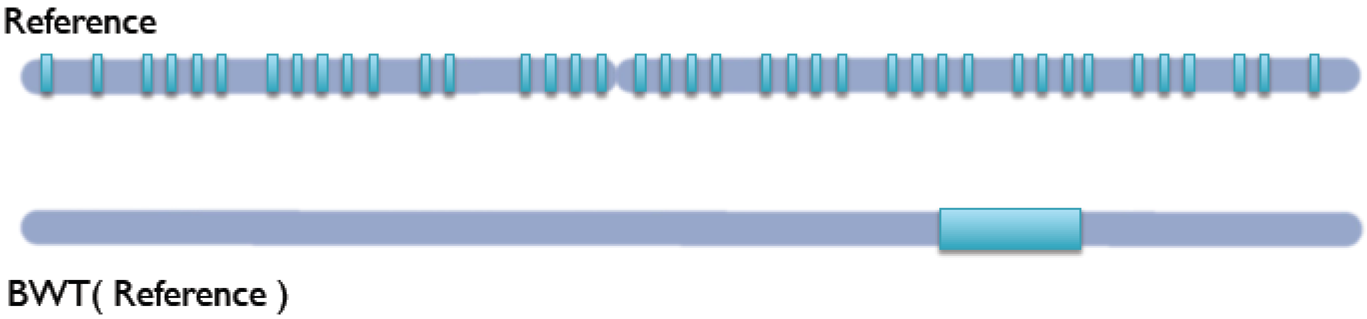
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



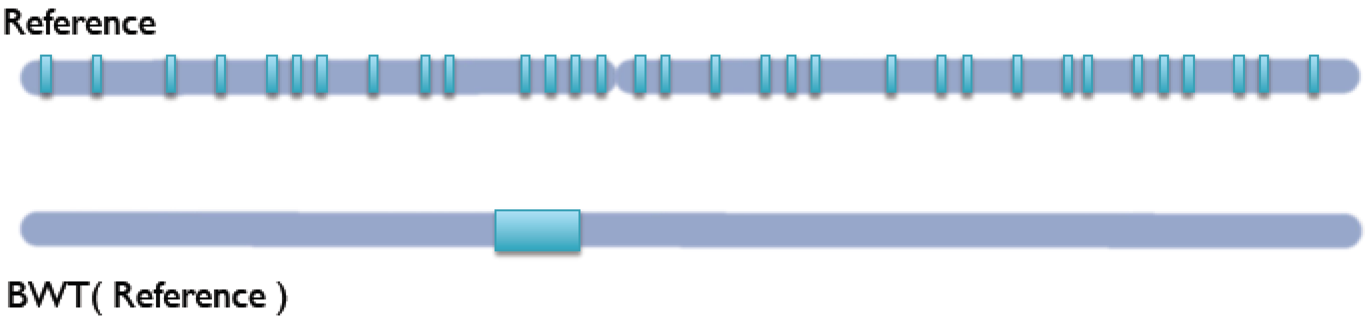
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



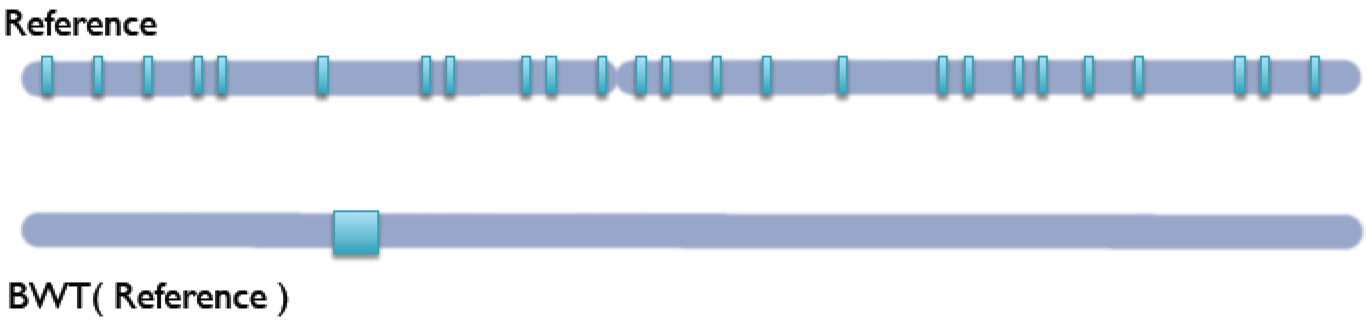
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



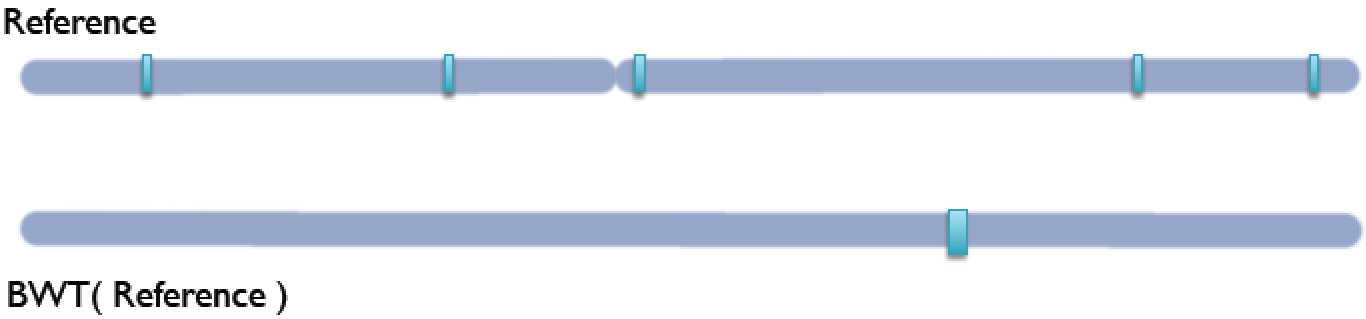
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



Reference

BWT(Reference)

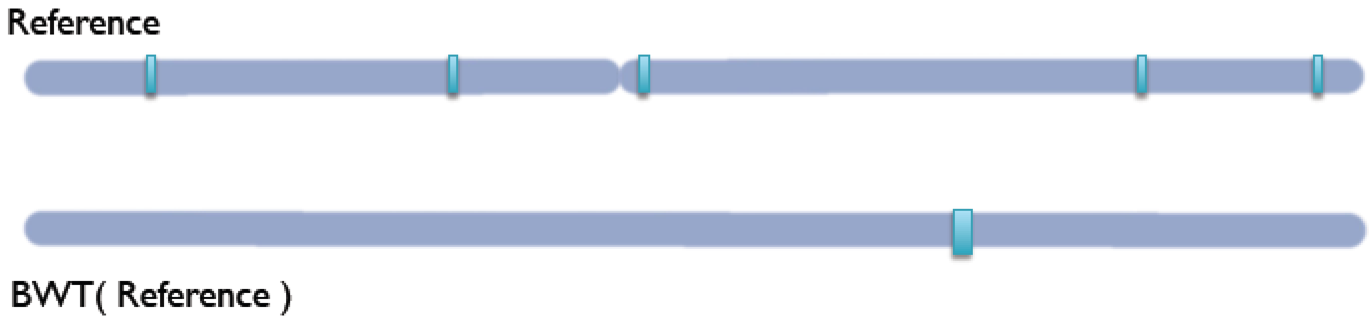
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example

Reference
BWT(Reference)

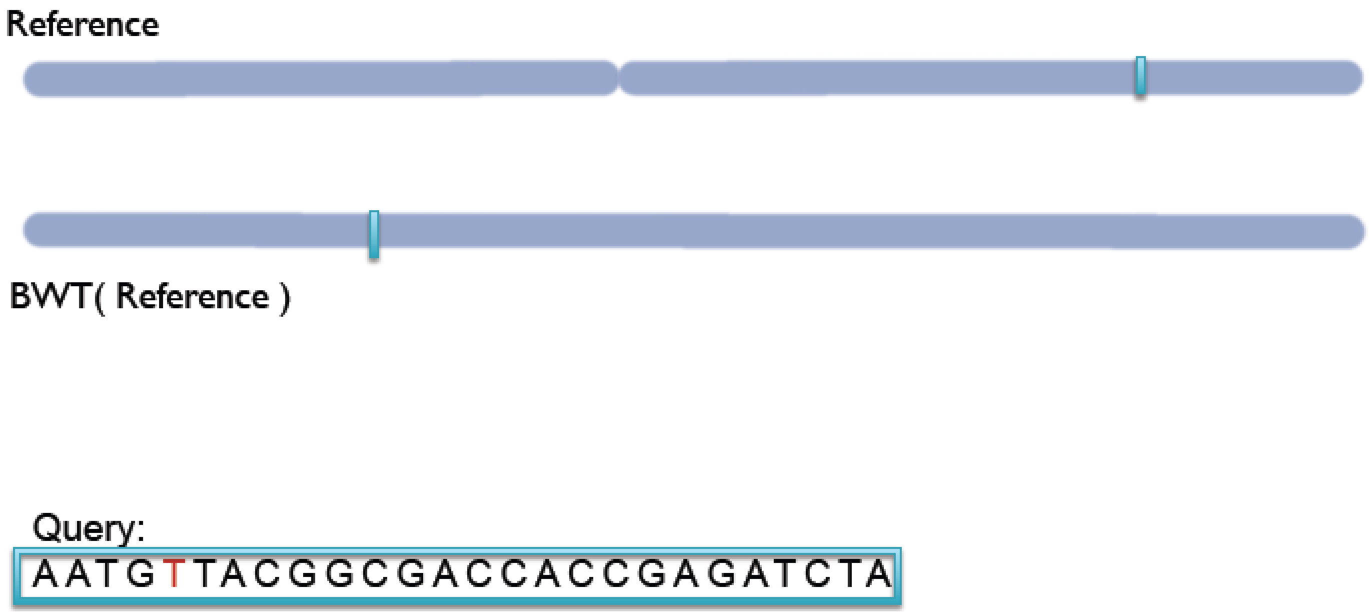
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



Query:
AATGTTACGGCGACCACCGAGATCTA

Bowtie/Soap2 example



Bowtie/Soap2 vs. BWA

- Bowtie and Soap2 cannot handle gapped alignments
  - No indel detection => Many false SNP calls

Bowtie/Soap2:

```
ACTCCCATTGTCATCGTACTTGGGATCGTAACA   Reference
    CCATTGTCATCGTACTTGGGATCTA
          TCATCGTACTTGGGATCTA
                   TTGGGATCTA
```

False SNPs

N.B. Bowtie2 can handle gapped alignments

Bowtie/Soap2 vs. BWA

- Bowtie and Soap2 cannot handle gapped alignments
  - No indel detection => Many false SNP calls

BWA:

```
ACTCCCATTGTCATCGTACTTGGGATCGTAACA   Reference
    CCATTGTCATCGTACTTGGGATC-TA
          TCATCGTACTTGGGATC-TA
                   TTGGGATC-TA
```

N.B. Bowtie2 can handle gapped alignments

Comparison

Hash referenced spaced seeds

- Requires ~50Gb of memory
- Runs 30-fold slower
- Is much simpler to program
- Most sensitive

Suffix/Prefix Trie

- Requires <2Gb of memory
- Runs 30-fold faster
- Is much more complicated to program
- Least sensitive

There are limits however

With 100bp reads, indels or variable regions longer than 3-4bp are likely to be missed entirely because reads will not map to the reference

```
ACTCCCATTGTCATCGTACTTGGGATCGTAACA  Reference
    CCATTGTCAACCATCTAGTAGCT-TA
          TCAACCATCTAGTAGCT-TA
                   ACCATCTA-TA
```

You only find what you are looking for

- What happens if there are SNPs and Indels in the same region?

Let’s assume that the SNP caller made this call of a single SNP:

ATGTATGTA
ATGTGTGTA

and the indel caller produced this call of a 3 base deletion:

ATGTATGTA
ATGT- - - TA

Should we assume this is a heterozygous SNP opposite a heterozygous Indel or a more complex locus?

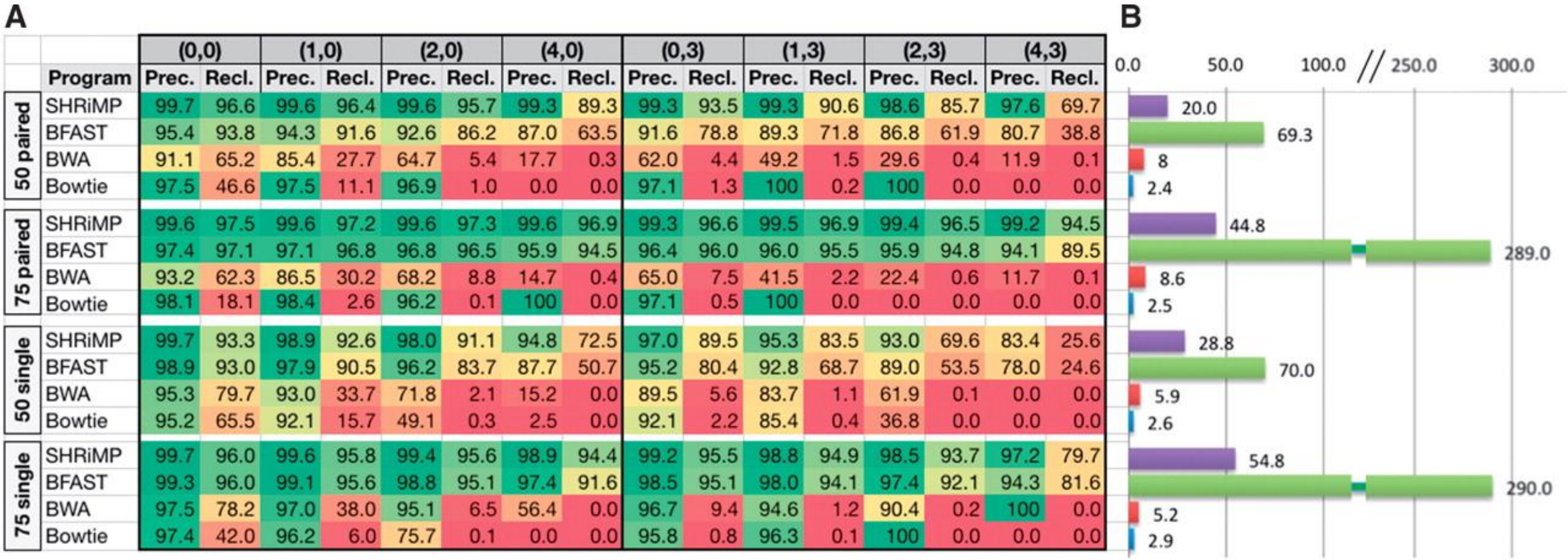
Comparison

- Bowtie's reported 30-fold speed increase over hash-based MAQ with small loss in sensitivity
- Limitations to Trie-based approaches:
    - Only able to find alignments within a certain 'edit distance’
    - Bowtie does not do gapped alignments – no indels!
    - Important to quality clip reads (-q in BWA)
    - Non-A/C/G/T bases on reads are often treated as mismatches
    - Make sure Ns are removed!

Hash based approaches are more suitable for divergent alignments

- Rule of thumb:
    - <2% divergence -> Trie-based
        - E.g. human alignments
    - >2% divergence -> seed-extend based approach
        - E.g. wild mouse strain alignments

Precision and recall by amount of variation for 4 datasets, by polymorphism:
(number of SNPs, Indel size)



A

		(0,0)		(1,0)		(2,0)		(4,0)		(0,3)		(1,3)		(2,3)		(4,3)	
	Program	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.
50 paired	SHRiMP	99.7	96.6	99.6	96.4	99.6	95.7	99.3	89.3	99.3	93.5	99.3	90.6	98.6	85.7	97.6	69.7
	BFAST	95.4	93.8	94.3	91.6	92.6	86.2	87.0	63.5	91.6	78.8	89.3	71.8	86.8	61.9	80.7	38.8
	BWA	91.1	65.2	85.4	27.7	64.7	5.4	17.7	0.3	62.0	4.4	49.2	1.5	29.6	0.4	11.9	0.1
	Bowtie	97.5	46.6	97.5	11.1	96.9	1.0	0.0	0.0	97.1	1.3	100	0.2	100	0.0	0.0	0.0
75 paired	SHRiMP	99.6	97.5	99.6	97.2	99.6	97.3	99.6	96.9	99.3	96.6	99.5	96.9	99.4	96.5	99.2	94.5
	BFAST	97.4	97.1	97.1	96.8	96.8	96.5	95.9	94.5	96.4	96.0	96.0	95.5	95.9	94.8	94.1	89.5
	BWA	93.2	62.3	86.5	30.2	68.2	8.8	14.7	0.4	65.0	7.5	41.5	2.2	22.4	0.6	11.7	0.1
	Bowtie	98.1	18.1	98.4	2.6	96.2	0.1	100	0.0	97.1	0.5	100	0.0	0.0	0.0	0.0	0.0
50 single	SHRiMP	99.7	93.3	98.9	92.6	98.0	91.1	94.8	72.5	97.0	89.5	95.3	83.5	93.0	69.6	83.4	25.6
	BFAST	98.9	93.0	97.9	90.5	96.2	83.7	87.7	50.7	95.2	80.4	92.8	68.7	89.0	53.5	78.0	24.6
	BWA	95.3	79.7	93.0	33.7	71.8	2.1	15.2	0.0	89.5	5.6	83.7	1.1	61.9	0.1	0.0	0.0
	Bowtie	95.2	65.5	92.1	15.7	49.1	0.3	2.5	0.0	92.1	2.2	85.4	0.4	36.8	0.0	0.0	0.0
75 single	SHRiMP	99.7	96.0	99.6	95.8	99.4	95.6	98.9	94.4	99.2	95.5	98.8	94.9	98.5	93.7	97.2	79.7
	BFAST	99.3	96.0	99.1	95.6	98.8	95.1	97.4	91.6	98.5	95.1	98.0	94.1	97.4	92.1	94.3	81.6
	BWA	97.5	78.2	97.0	38.0	95.1	6.5	56.4	0.0	96.7	9.4	94.6	1.2	90.4	0.2	100	0.0
	Bowtie	97.4	42.0	96.2	6.0	75.7	0.1	0.0	0.0	95.8	0.8	96.3	0.1	100	0.0	0.0	0.0

David M et al. Bioinformatics 2011;27:1011-1012

Summary of open-source short read alignment programs

Program	Algorithm	SoLID	Long reads	Gapped alignment	Paired-end	Quality scores used?
Bfast	Hashing ref	Yes	No	Yes	Yes	No
Bowtie*	FM-Index	Yes	No	No*	Yes	Yes
Blat	Hashing ref	No	Yes	Yes	No	No
BWA	FM-Index	Yes	Yes	Yes	Yes	No
MAQ	Hashing reads	Yes	No	Yes	Yes	Yes
Mosaik	Hashing ref	Yes	Yes	Yes	Yes	No
Novoalign	Hashing ref	No	No	Yes	Yes	Yes
Shrimp2	Hashing ref	Yes	Yes	Yes	Yes	Yes
SOAP2	FM-Index	No	No	No	Yes	Yes
SSAHA2	Hashing ref.	No	No	No	Yes	Yes

Heng Li & Nils Homer. Sequence alignment algorithms for next-generation sequencing.
Briefings in Bioinformatics. Vol 11. No 5. 473 483, 2010

* Bowtie2 (just released) does support gapped alignments

Contents

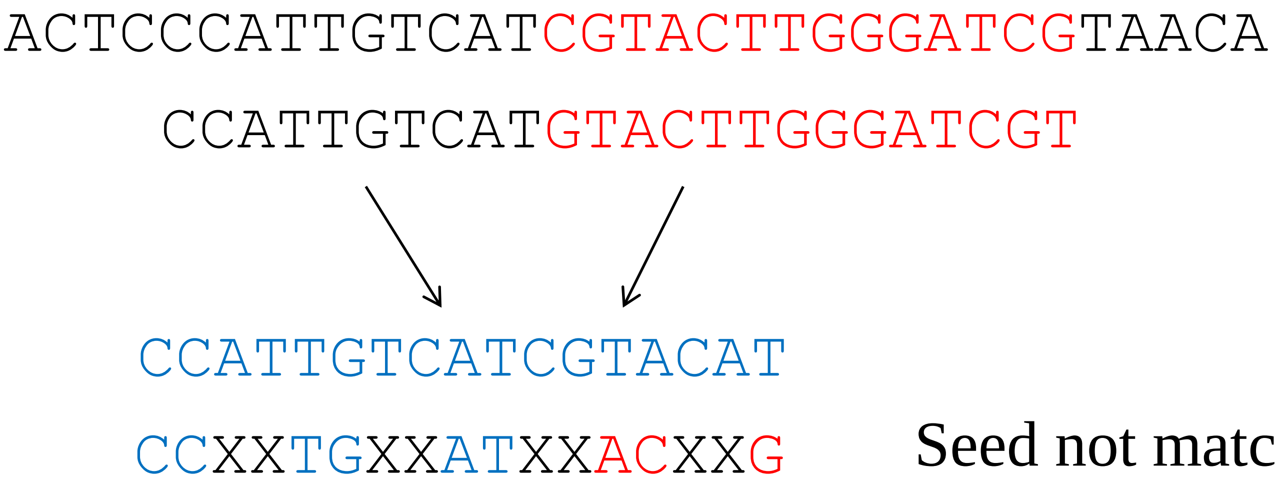
- **Alignment algorithms for short-reads**
  - Background – Blast (why can’t we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Indel detection
  - Suffix/Prefix Tries
  - **Other alignment considerations**
  - Typical alignment pipeline
  - New methods of SNP calling

Other alignment considerations

- Indel detection
- Effect of paired-end alignments
- Using base quality to inform alignments
- PCR duplicates
- Methylation experiments – bisulfite treated reads
- Multi-mapping reads
- Aligning spliced-reads from RNA-seq experiments
- Local realignment to improve SNP/Indel detection
- Platform specific errors
- Unmapped reads

Indel detection

Spaced seed with weight 9bp and no mismatches:



ACTCCCATTGTCATCGTACTTGGGATCGTAACA

CCATTGTCATGTACTTGGGATCGT

CCATTGTCATCGTACAT

CCXXTGXXATXXACXXG

Reference sequence

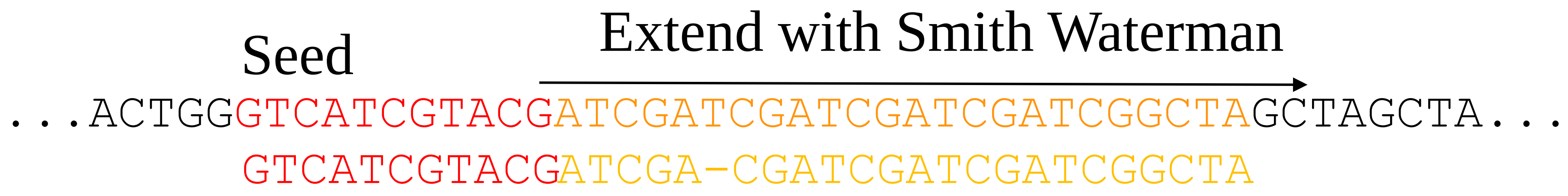
Read containing a deletion

Seed not matched due to frame shift caused by gap

No seed match. No alignment!

Indel detection

Reference sequence:



Seed

Extend with Smith Waterman

```
...ACTGGGTCATCGTACGATCGATCGATCGATCGATCGGCTAGCTAGCTA...
        GTCATCGTACGATCGA-CGATCGATCGATCGGCTA
```

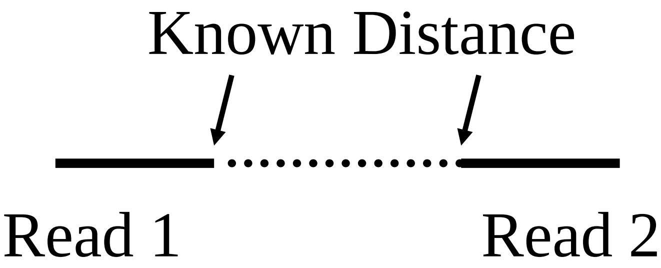
Most alignment programs can only detect gaps in Smith-Waterman phase
once a seed has been identified. Some algorithms (e.g. Bowtie) do not even allow gaps at this stage

This reduces sensitivity especially with multiple insertions in a small region

Indel detection

- Some algorithms do allow gaps within seed
  - Indel seeds for homology search *Bioinformatics (2006) 22(14): e341-e349 doi:10.1093/bioinformatics/btl263*
  - Weese D, Emde AK, Rausch T, et al. RazerS–fast read mapping with sensitivity control. Genome Res 2009;19:1646–54
  - Rumble SM, Lacroute P, Dalca AV, et al. SHRiMP: accurate mapping of short color-space reads. PLoS Comput Biol 2009;5:e1000386
- Use of multiple seeds
  - Especially useful for longer reads (>36bp)
  - Li R, Li Y, Kristiansen K, et al. SOAP: short oligonucleotide alignment program. Bioinformatics 2008;24:713–4
  - Jiang H, Wong WH. SeqMap: mapping massive amount of oligonucleotides to the genome. Bioinformatics 2008;24: 2395–6

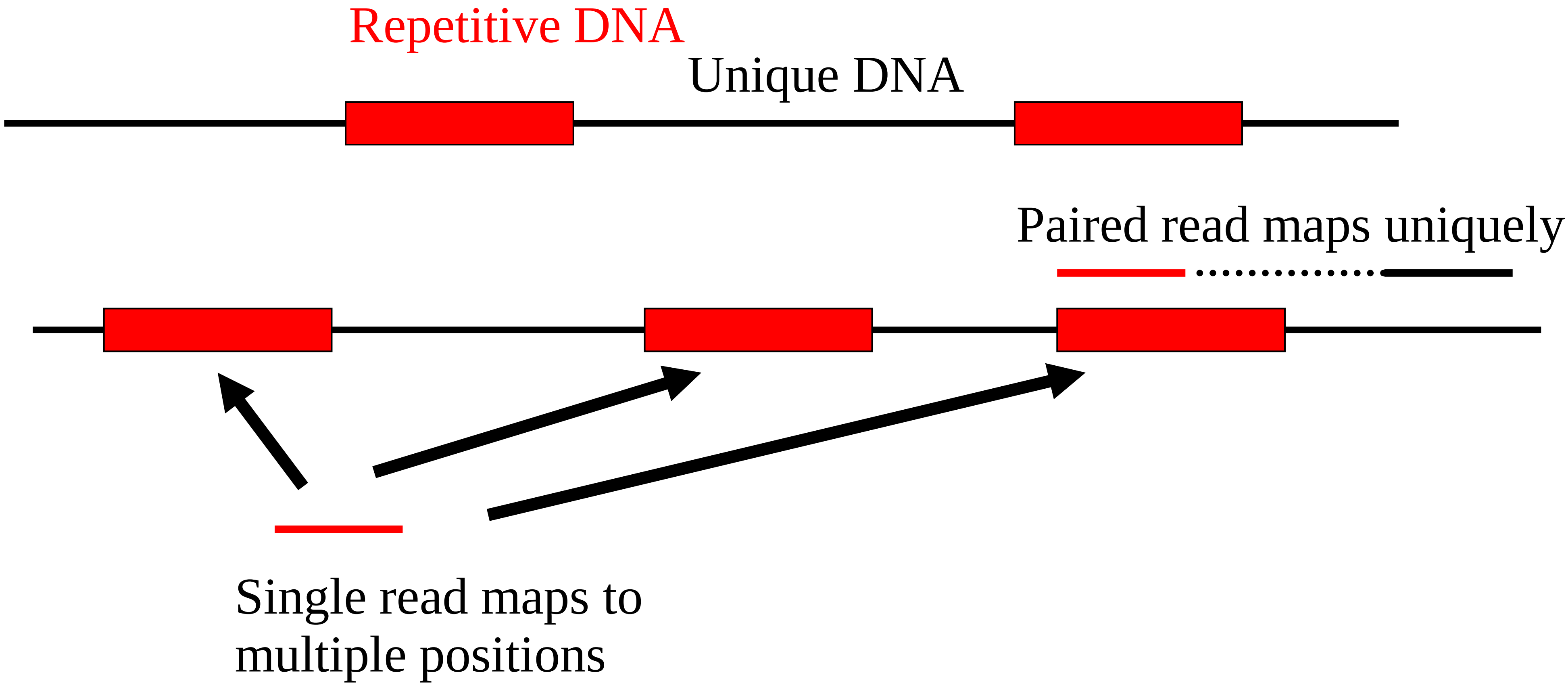
Paired-end reads are important



Known Distance

Read 1

Read 2



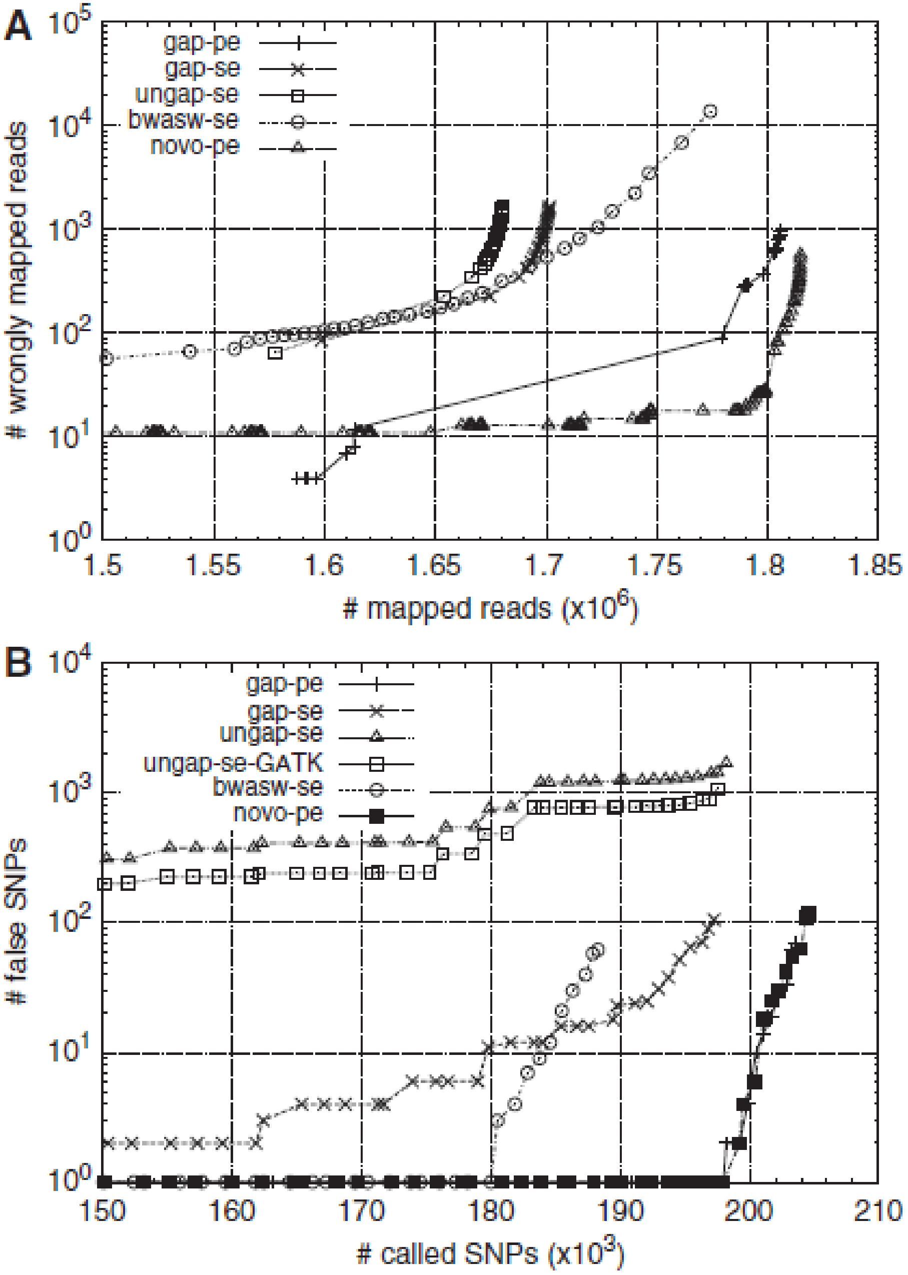
Repetitive DNA

Unique DNA

Paired read maps uniquely

Single read maps to
multiple positions

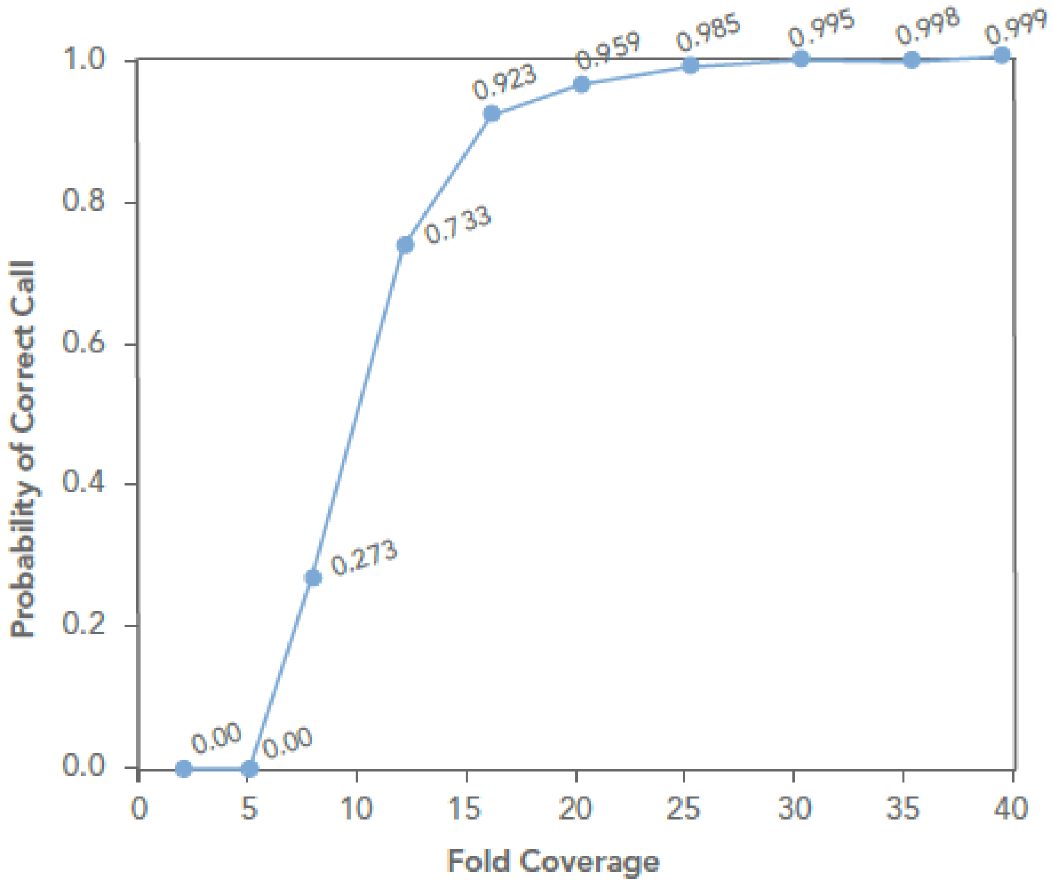
Effect of paired-end alignments



Heng Li & Nils Homer. Sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics. Vol 11. No 5. 473 483, 2010

Effect of coverage on SNP call accuracy

- Depends crucially on ploidy
- Bacterial genomes can get away with 10-20x
- For human genomes and other diploids 20-30x is regarded as standard
- Poly-ploids (e.g wheat) may need much higher coverage



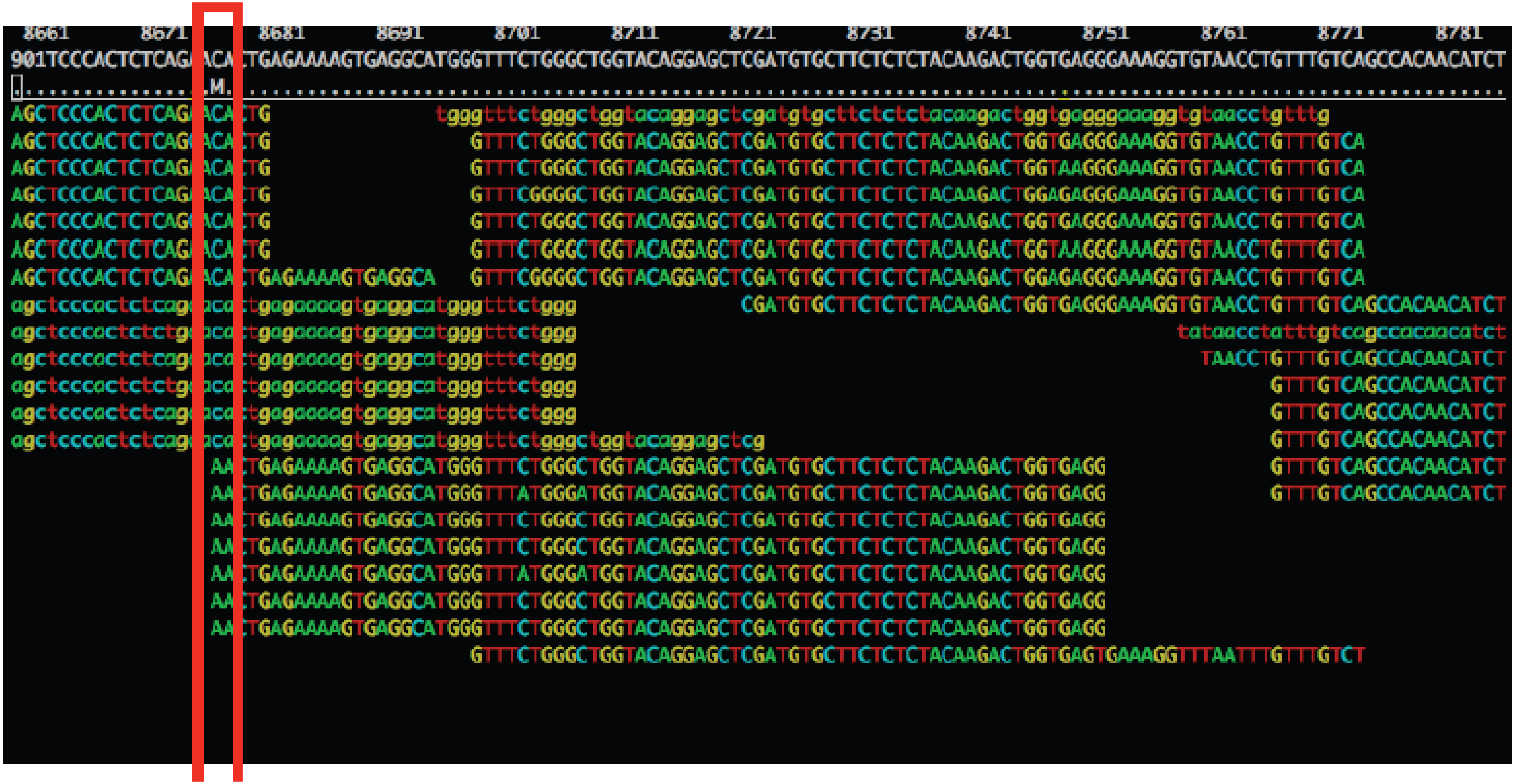
Source – Illumina Tech Note
Human diploid sample

PCR duplicates

- **2nd generation sequencers are not single-molecule sequencers**
  - All have at least one PCR amplification step
  - Can result in duplicate DNA fragments
  - This can bias SNP calls or introduce false SNPs

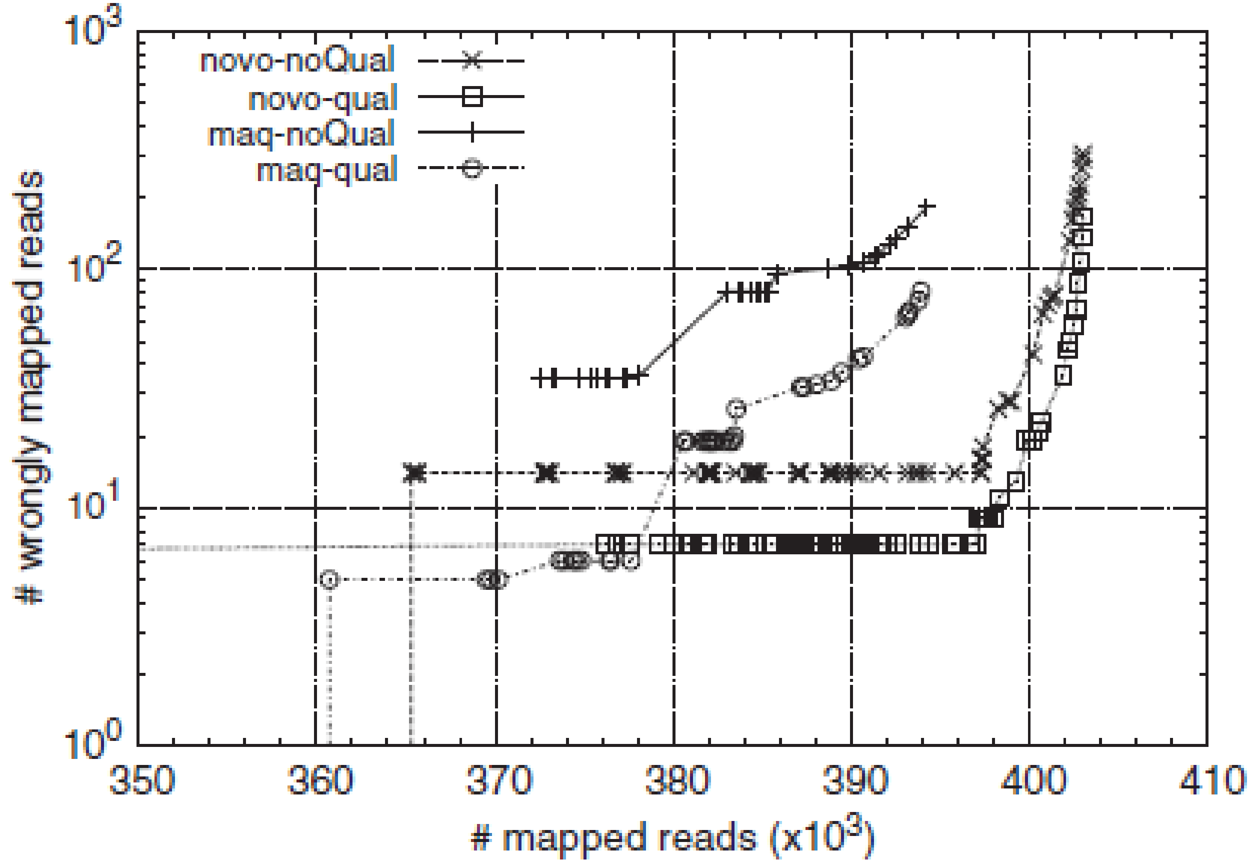
- **Generally duplicates only make up a small fraction of the results**
  - Good libraries have < 2-3% of duplicates
  - SAMtools and Picard can identify and remove these when aligned against a reference genome
  - Do NOT do this for RNA and ChIP-seq data!

PCR duplicates



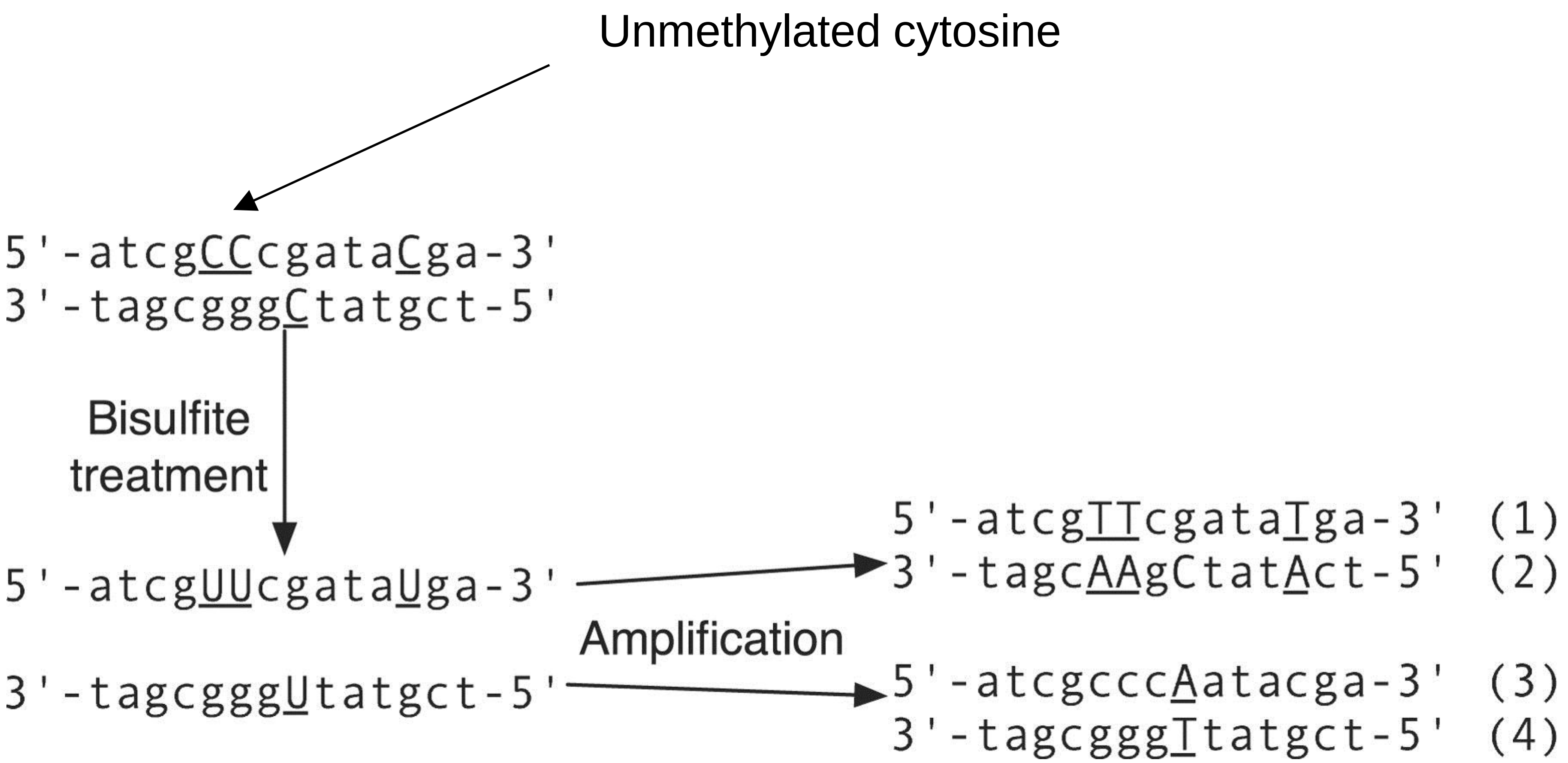
```
 8661      8671      8681      8691      8701      8711      8721      8731      8741      8751      8761      8771      8781
901TCCCACTCTCAG ACA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
............... .M. ...........................................................................................................
AGCTCCCACTCTCAG ACA TG              tgggtttctgggctggtacaggagctcgatgtgcttctctctacaagactggtgagggaaaggtgtaacctgtttg
AGCTCCCACTCTCAG ACA TG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAG ACA TG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAG ACA TG                 GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAG ACA TG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAG ACA TG                 GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTAAGGGAAAGGTGTAACCTGTTTGTCA
AGCTCCCACTCTCAG ACA TGAGAAAAGTGAGGCA   GTTTCGGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGAGAGGGAAAGGTGTAACCTGTTTGTCA
agctcccactctcag aca tgagaaaagtgaggcatgggtttctggg              CGATGTGCTTCTCTCTACAAGACTGGTGAGGGAAAGGTGTAACCTGTTTGTCAGCCACAACATCT
agctcccactctctg aca tgagaaaagtgaggcatgggtttctggg                                                   tataacctatttgtcagccacaacatct
agctcccactctcag aca tgagaaaagtgaggcatgggtttctggg                                                     TAACCTGTTTGTCAGCCACAACATCT
agctcccactctctg aca tgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcag aca tgagaaaagtgaggcatgggtttctggg                                                           GTTTGTCAGCCACAACATCT
agctcccactctcag aca tgagaaaagtgaggcatgggtttctgggctggtacaggagctcg                                           GTTTGTCAGCCACAACATCT
                 AA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AA TGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG              GTTTGTCAGCCACAACATCT
                 AA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AA TGAGAAAAGTGAGGCATGGGTTTATGGGATGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                 AA TGAGAAAAGTGAGGCATGGGTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGG
                                       GTTTCTGGGCTGGTACAGGAGCTCGATGTGCTTCTCTCTACAAGACTGGTGAGTGAAAGGTTTAATTTGTTTGTCT
```

Base quality impacts on read mapping



Heng Li & Nils Homer. Sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics. Vol 11. No 5. 473 483, 2010

Methylation experiments



Unmethylated cytosine

5'-atcgCCcgataCga-3'
3'-tagcgggCtatgct-5'

Bisulfite treatment

5'-atcgUUcgataUga-3'

3'-tagcgggUtatgct-5'

Amplification

5'-atcgTTcgataTga-3' (1)
3'-tagcAAgCtatAct-5' (2)

5'-atcgcccAatacga-3' (3)
3'-tagcgggTtatgct-5' (4)

Methylation experiments

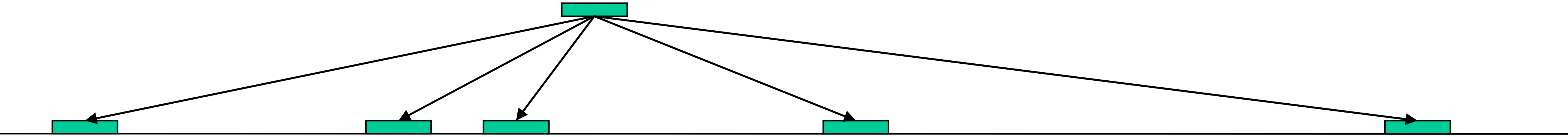
- Directly aligning reads against a reference will fail due to excessive mismatches in non-methylated regions

- Most aligners deal with this by creating 2 reference sequences
  - One has all Cs converted to Ts
  - One has all Gs converted to As

- Convert Cs to Ts in all reads aligned against C-T reference
- Convert Gs to As in all reads aligned against G-A reference

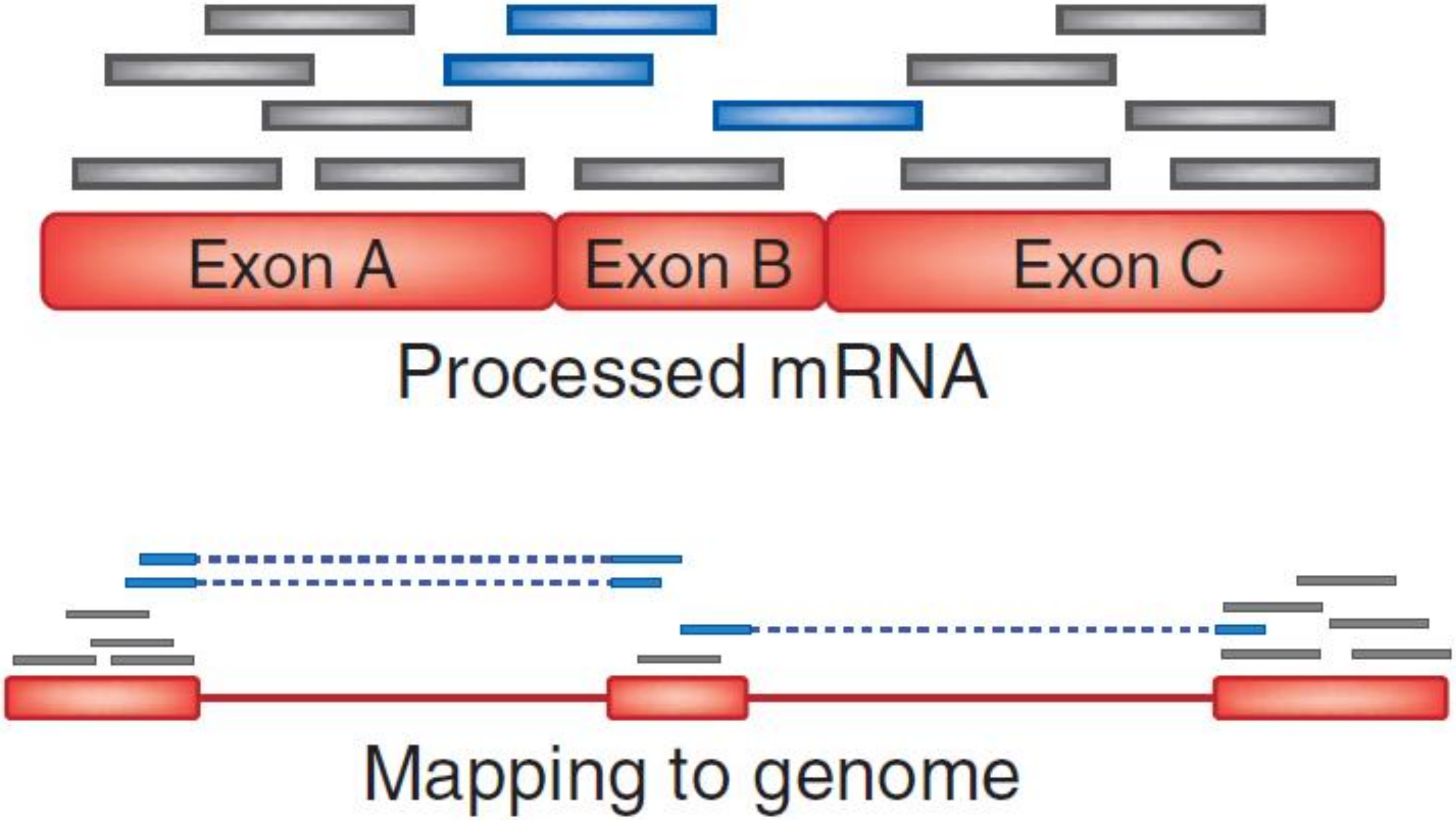
- If there are no mutations or sequencing errors the reads will always map to one of the two references

Multiple mapping reads



- A single read may occur more than once in the reference genome.
- This may be due to gene or whole chromosome duplication or repetitive sequences
- Aligners generally allow you to chose how these are dealt with
- Some aligners automatically assign a multi-mapping read to one of the locations at random (e.g. MAQ)

Spliced-read mapping



- Need packages which can account for splice variants
- Examples: TopHat, ERANGE

Local realignment to improve SNP/Indel detection

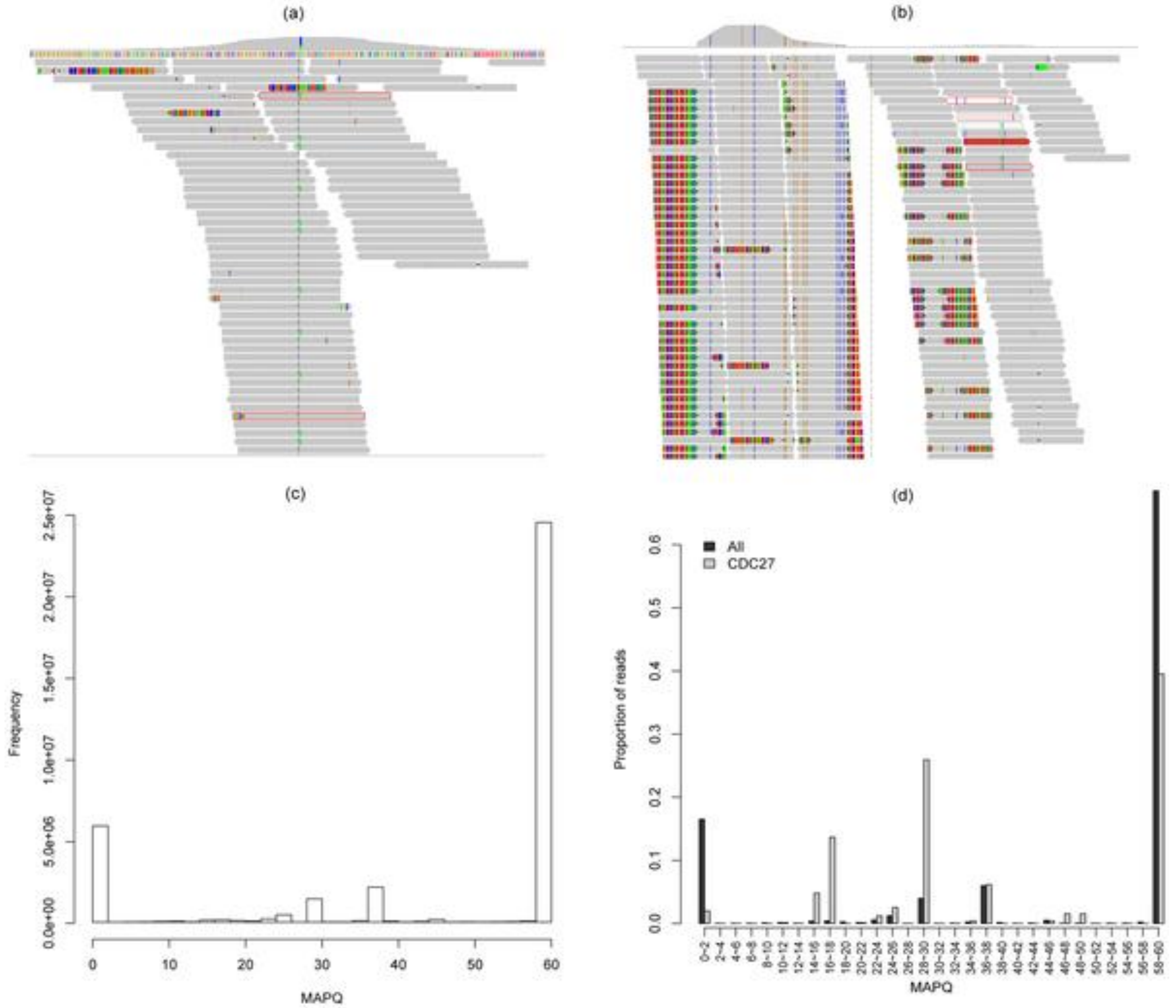
- Read aligners map each read (or read pair) independently of all other reads
- Around indels and other variants it can be helpful to make use of other metrics

  e.g. Global median coverage for multi-mapping reads
- Tools such as GATK, SAMtools, Pindel and Breakdancer realign reads in the vicinity of variants to improve calls

http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit

Chen, K. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation *Nature Methods* 6, 677 - 681 (2009)
Li H.*, Handsaker B.*, Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. and 1000 Genome Project Data Processing Subgroup (2009) The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9

Figure 6. A visual examination of a spurious gene (CDC27).



http://www.plosone.org/article/info:doi/10.1371/journal.pone.0038470

All platforms have errors and artefacts

Illumina

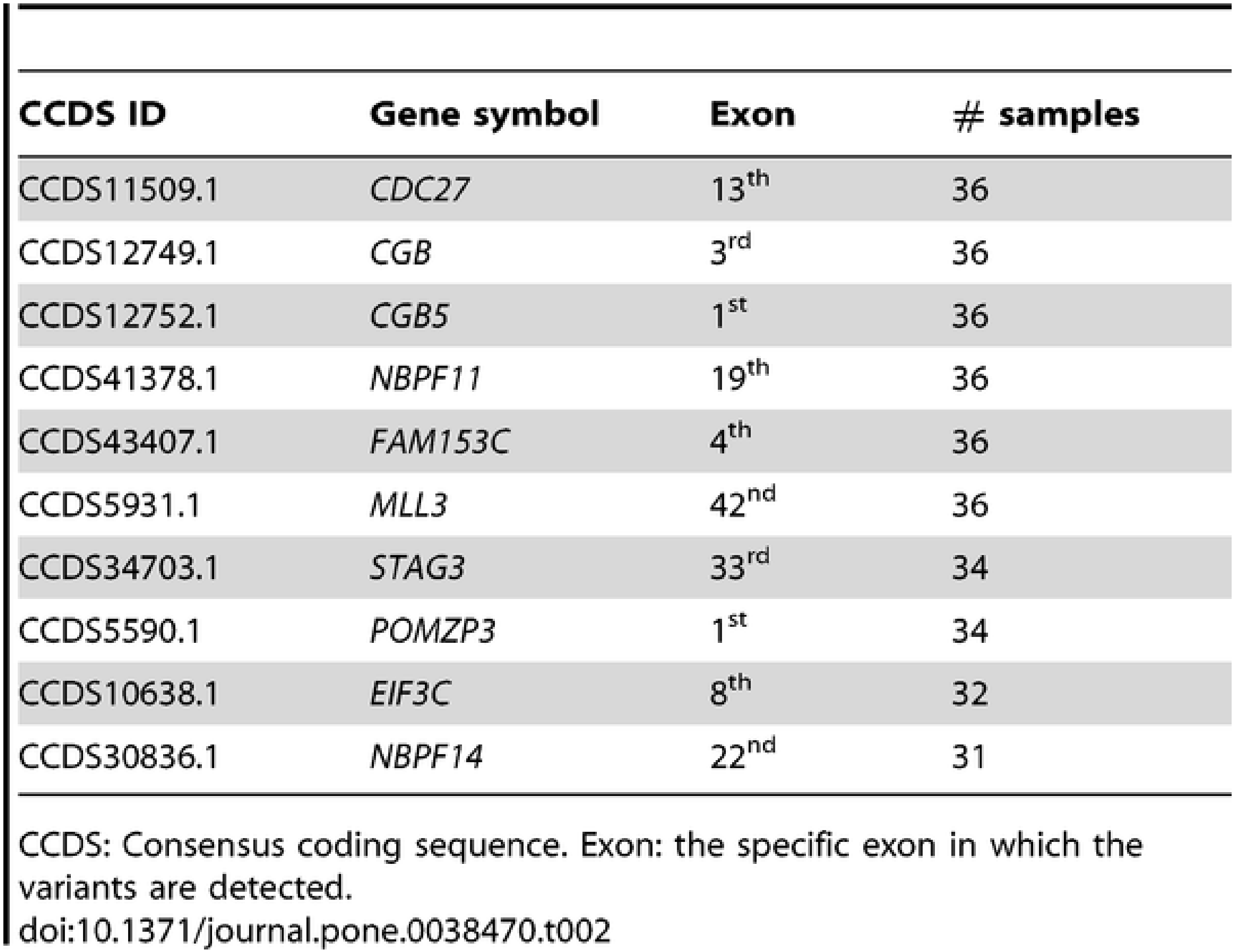
SoLID/ABI-Life

Roche 454

Ion Torrent

1. Removal of low quality bases
2. Removal of adaptor sequences
3. Platform specific artefacts (e.g homopolymers)

Table 2. Spurious genes having mutations detected in 30 samples.



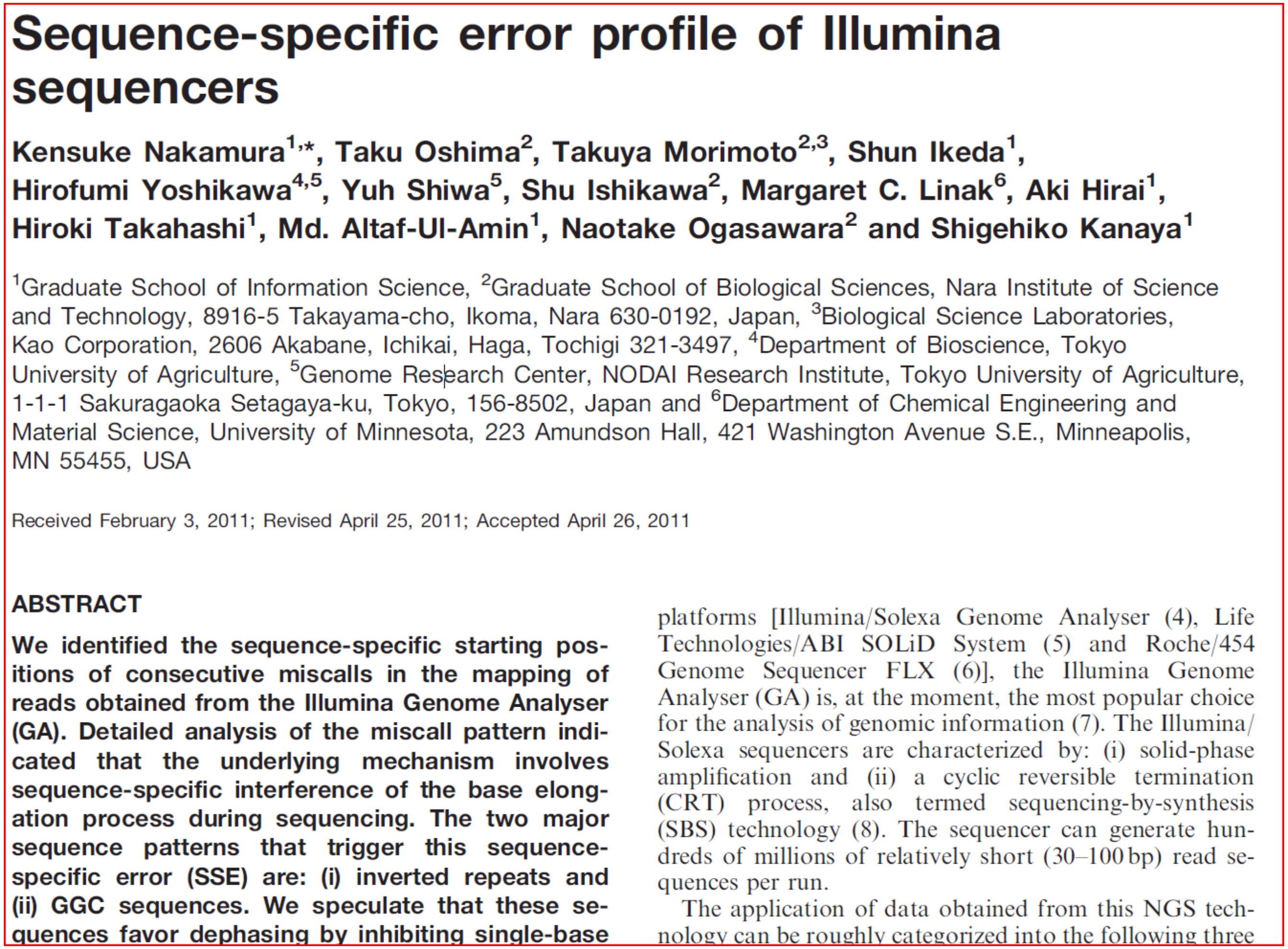
CCDS ID	Gene symbol	Exon	# samples
CCDS11509.1	*CDC27*	13th	36
CCDS12749.1	*CGB*	3rd	36
CCDS12752.1	*CGB5*	1st	36
CCDS41378.1	*NBPF11*	19th	36
CCDS43407.1	*FAM153C*	4th	36
CCDS5931.1	*MLL3*	42nd	36
CCDS34703.1	*STAG3*	33rd	34
CCDS5590.1	*POMZP3*	1st	34
CCDS10638.1	*EIF3C*	8th	32
CCDS30836.1	*NBPF14*	22nd	31

CCDS: Consensus coding sequence. Exon: the specific exon in which the variants are detected.

doi:10.1371/journal.pone.0038470.t002

http://www.plosone.org/article/info:doi/10.1371/journal.pone.0038470

Illumina artefacts



Sequence-specific error profile of Illumina sequencers

Kensuke Nakamura[1],*, Taku Oshima[2], Takuya Morimoto[2,3], Shun Ikeda[1], Hirofumi Yoshikawa[4,5], Yuh Shiwa[5], Shu Ishikawa[2], Margaret C. Linak[6], Aki Hirai[1], Hiroki Takahashi[1], Md. Altaf-Ul-Amin[1], Naotake Ogasawara[2] and Shigehiko Kanaya[1]

[1]Graduate School of Information Science, [2]Graduate School of Biological Sciences, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan, [3]Biological Science Laboratories, Kao Corporation, 2606 Akabane, Ichikai, Haga, Tochigi 321-3497, [4]Department of Bioscience, Tokyo University of Agriculture, [5]Genome Research Center, NODAI Research Institute, Tokyo University of Agriculture, 1-1-1 Sakuragaoka Setagaya-ku, Tokyo, 156-8502, Japan and [6]Department of Chemical Engineering and Material Science, University of Minnesota, 223 Amundson Hall, 421 Washington Avenue S.E., Minneapolis, MN 55455, USA

Received February 3, 2011; Revised April 25, 2011; Accepted April 26, 2011

ABSTRACT

We identified the sequence-specific starting positions of consecutive miscalls in the mapping of reads obtained from the Illumina Genome Analyser (GA). Detailed analysis of the miscall pattern indicated that the underlying mechanism involves sequence-specific interference of the base elongation process during sequencing. The two major sequence patterns that trigger this sequence-specific error (SSE) are: (i) inverted repeats and (ii) GGC sequences. We speculate that these sequences favor dephasing by inhibiting single-base

platforms [Illumina/Solexa Genome Analyser (4), Life Technologies/ABI SOLiD System (5) and Roche/454 Genome Sequencer FLX (6)], the Illumina Genome Analyser (GA) is, at the moment, the most popular choice for the analysis of genomic information (7). The Illumina/Solexa sequencers are characterized by: (i) solid-phase amplification and (ii) a cyclic reversible termination (CRT) process, also termed sequencing-by-synthesis (SBS) technology (8). The sequencer can generate hundreds of millions of relatively short (30–100 bp) read sequences per run.

The application of data obtained from this NGS technology can be roughly categorized into the following three

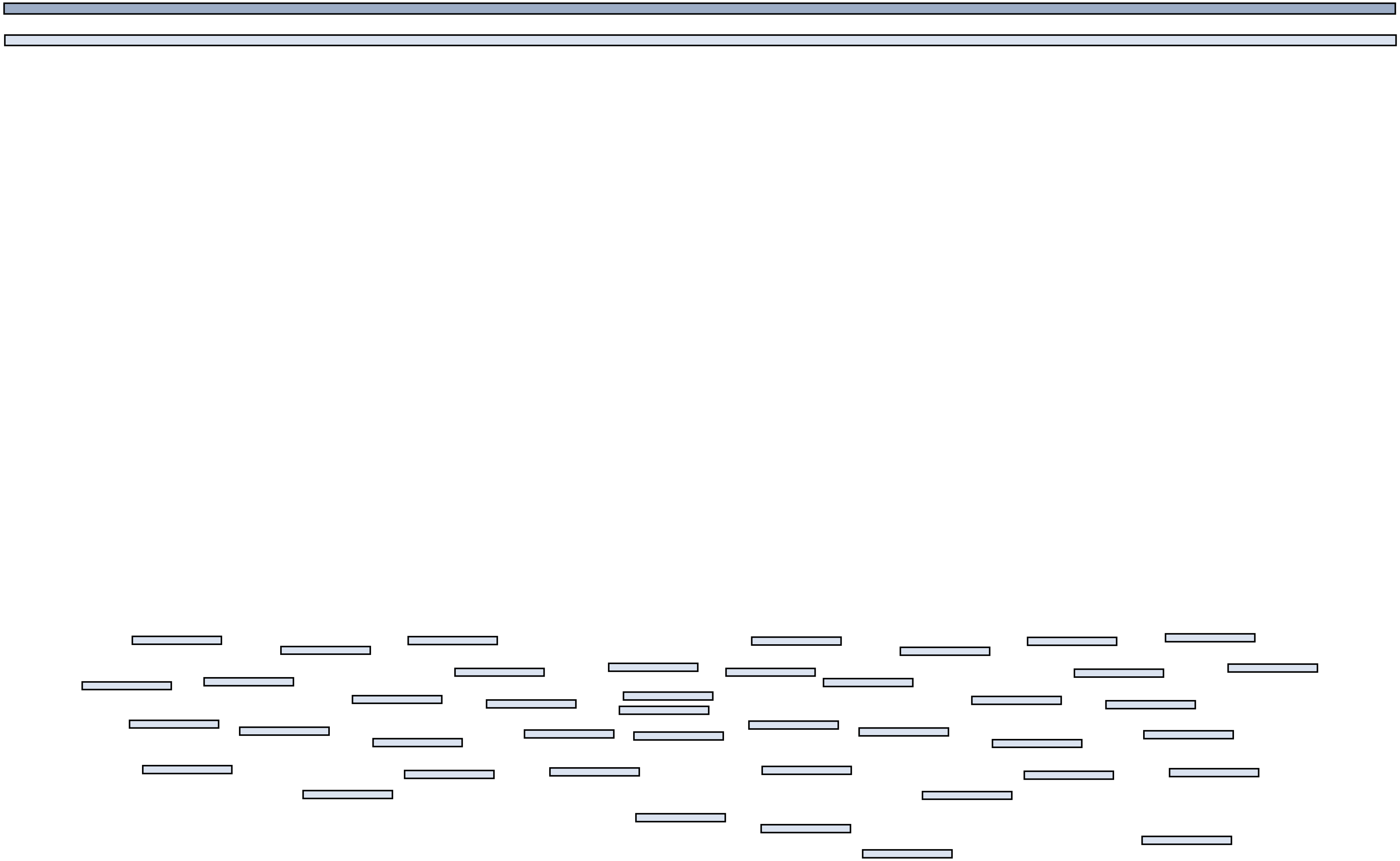
Nakamura, K. et al. Sequence-specific error profile of Illumina sequencers
Nucl. Acids Res. (2011) May 16, 2011

Illumina artefacts

1. GC rich regions are under represented
    a. PCR
    b. Sequencing
2. Substitutions more common than insertions
3. GGC/GCC motif is associated with low quality and mismatches
4. Filtering low quality reads exacerbates low coverage of GC regions

Software should ideally account for this technology specific bias but doesn't (yet)

Unmapped reads

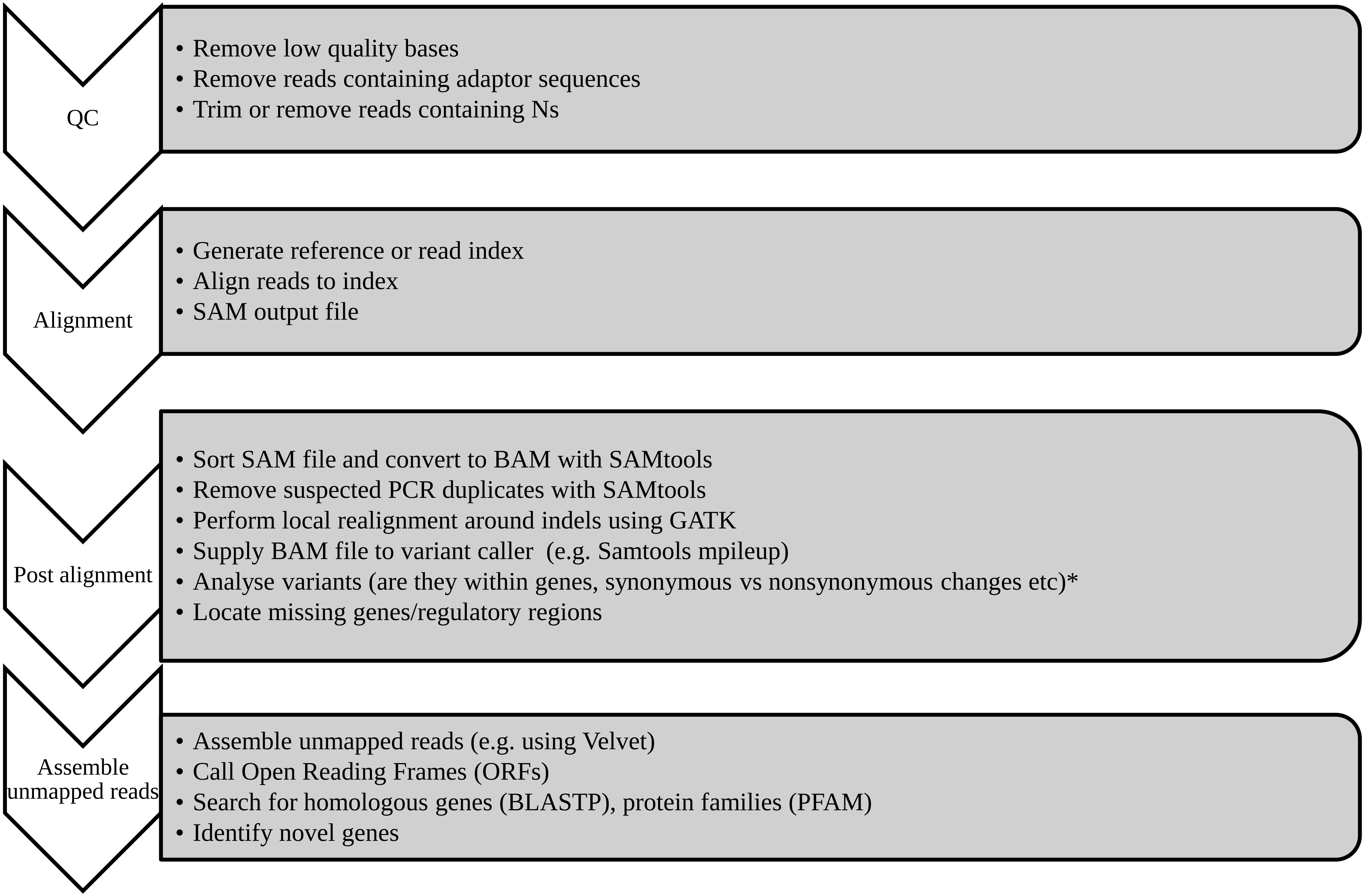


Unmapped reads

- Can be the result of:
  - Sequencing errors (should be small fraction if quality filtering applied before mapping)
  - Contamination
  - Excessive matches to repeats
  - Highly divergent regions between samples
  - Novel genetic material not present in reference
  - Plasmids

- Should be assembled de-novo with paired-end information if possible
- Resulting contigs run through MegaBlast against NCBI NT to check species
- Check against RepBase to remove repetitive contigs
- Call ORFs
- Blast ORFs using BlastP against NCBI NR or Swissprot and Blast2GO
- Run through PFAM

Typical alignment pipeline



QC

- Remove low quality bases
- Remove reads containing adaptor sequences
- Trim or remove reads containing Ns

Alignment

- Generate reference or read index
- Align reads to index
- SAM output file

Post alignment

- Sort SAM file and convert to BAM with SAMtools
- Remove suspected PCR duplicates with SAMtools
- Perform local realignment around indels using GATK
- Supply BAM file to variant caller (e.g. Samtools mpileup)
- Analyse variants (are they within genes, synonymous vs nonsynonymous changes etc)*
- Locate missing genes/regulatory regions

Assemble unmapped reads

- Assemble unmapped reads (e.g. using Velvet)
- Call Open Reading Frames (ORFs)
- Search for homologous genes (BLASTP), protein families (PFAM)
- Identify novel genes

* http://bioinformatics.net.au/software.nesoni.shtml

Contents

- **Alignment algorithms for short-reads**
  - Background – Blast (why can't we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Indel detection
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
  - **New methods of SNP calling**

New methods of SNP calling

- FreeBayes (http://arxiv.org/pdf/1207.3907v2.pdf)
- Warning - unpublished
  - Haplotype calling

ACA Reference Genome

Assume a SNP at both 5’ A->T and 3’ A->G

Do we have a homozygous?

TCG

Or a heterozygous?

ACG

TCA

New methods of SNP calling

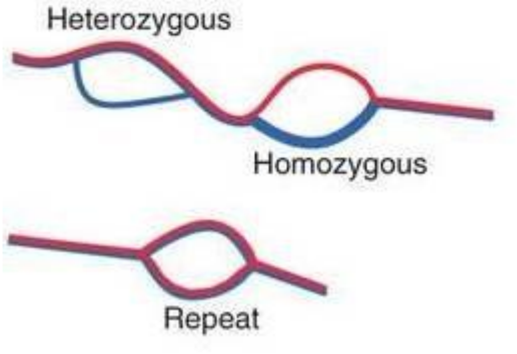
- FreeBayes (http://arxiv.org/pdf/1207.3907v2.pdf)
  - Haplotype calling

```
ACTCCGTTTGTCATCGTACTTGGGATCGTAACA    Strand 1
ACTCCCATTGTCATCGTACTTGGGATCGTAACA    Strand 2 (rev-comp)
```

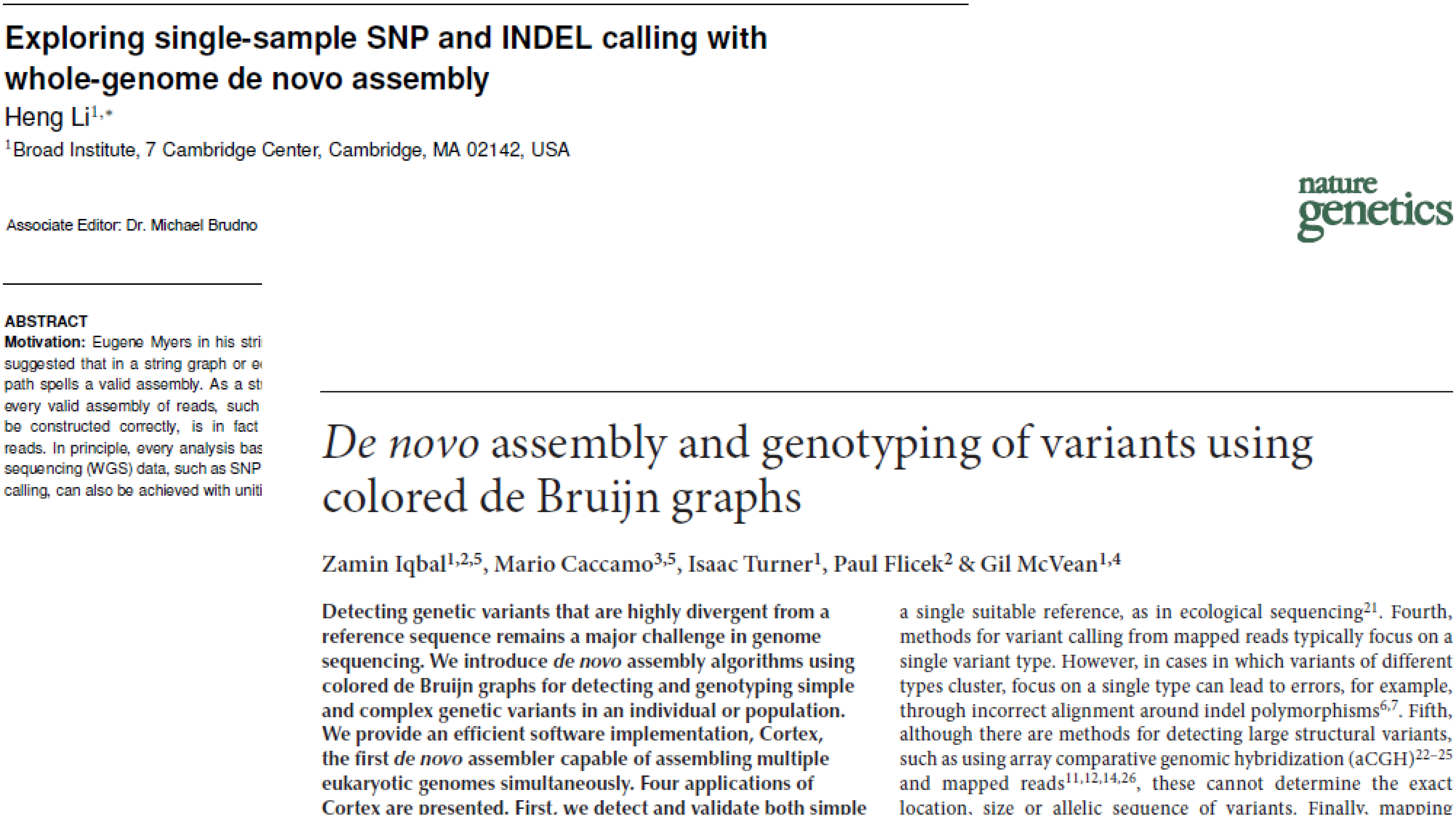
New methods of SNP calling

- Why align at all?
  - We only do this because of computational constraints
  - Ideally we want to assemble denovo and then align to reference genome

- Cortex is a step in this direction:
  - Denovo genome assembler, but keeps track of differences which could be due to SNPs/Indels



Variant calling with de-novo assembly



Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly

Heng Li[1,*]

[1] Broad Institute, 7 Cambridge Center, Cambridge, MA 02142, USA

Associate Editor: Dr. Michael Brudno

ABSTRACT

Motivation: Eugene Myers in his stri
suggested that in a string graph or e
path spells a valid assembly. As a st
every valid assembly of reads, such
be constructed correctly, is in fact
reads. In principle, every analysis bas
sequencing (WGS) data, such as SNP
calling, can also be achieved with uniti

nature genetics

De novo assembly and genotyping of variants using colored de Bruijn graphs

Zamin Iqbal[1,2,5], Mario Caccamo[3,5], Isaac Turner[1], Paul Flicek[2] & Gil McVean[1,4]

Detecting genetic variants that are highly divergent from a reference sequence remains a major challenge in genome sequencing. We introduce *de novo* assembly algorithms using colored de Bruijn graphs for detecting and genotyping simple and complex genetic variants in an individual or population. We provide an efficient software implementation, Cortex, the first *de novo* assembler capable of assembling multiple eukaryotic genomes simultaneously. Four applications of Cortex are presented. First, we detect and validate both simple

a single suitable reference, as in ecological sequencing[21]. Fourth, methods for variant calling from mapped reads typically focus on a single variant type. However, in cases in which variants of different types cluster, focus on a single type can lead to errors, for example, through incorrect alignment around indel polymorphisms[6,7]. Fifth, although there are methods for detecting large structural variants, such as using array comparative genomic hybridization (aCGH)[22–25] and mapped reads[11,12,14,26], these cannot determine the exact location, size or allelic sequence of variants. Finally, mapping

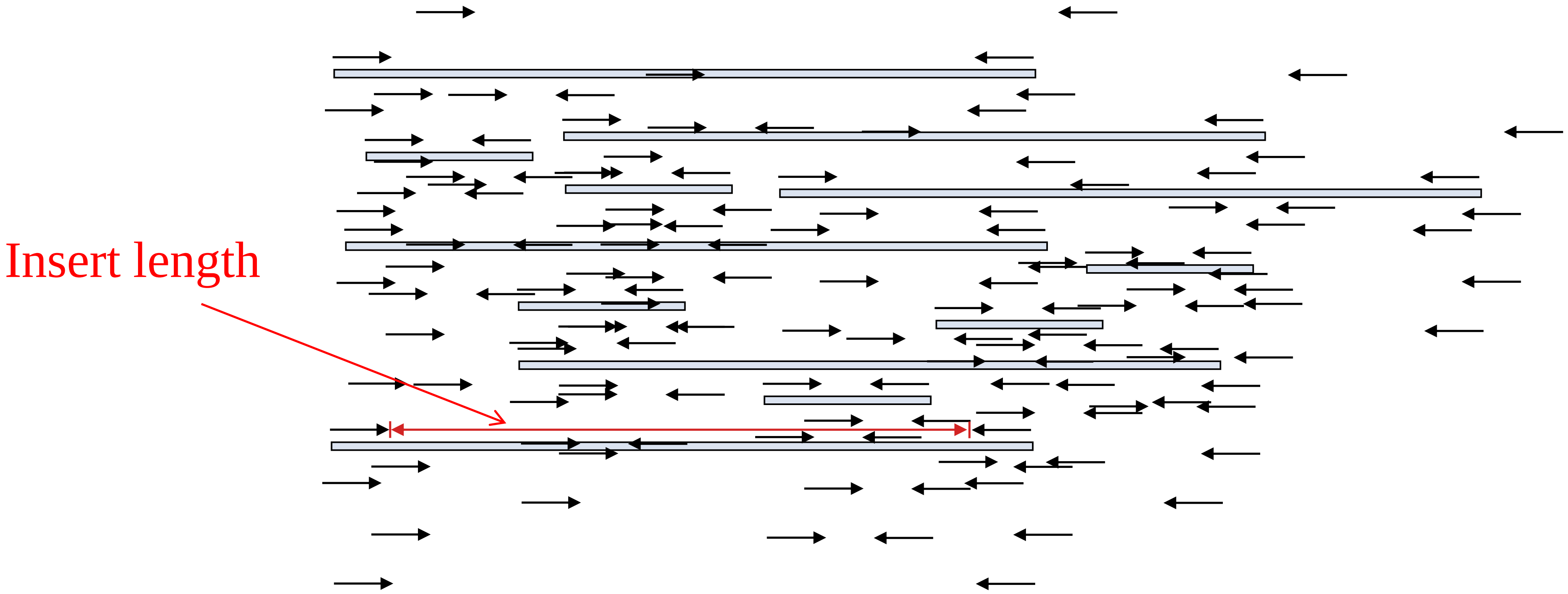
Questions!

biosciences.exeter.ac.uk/facilities/sequencing/usefulresources/

Assembly algorithms for short reads

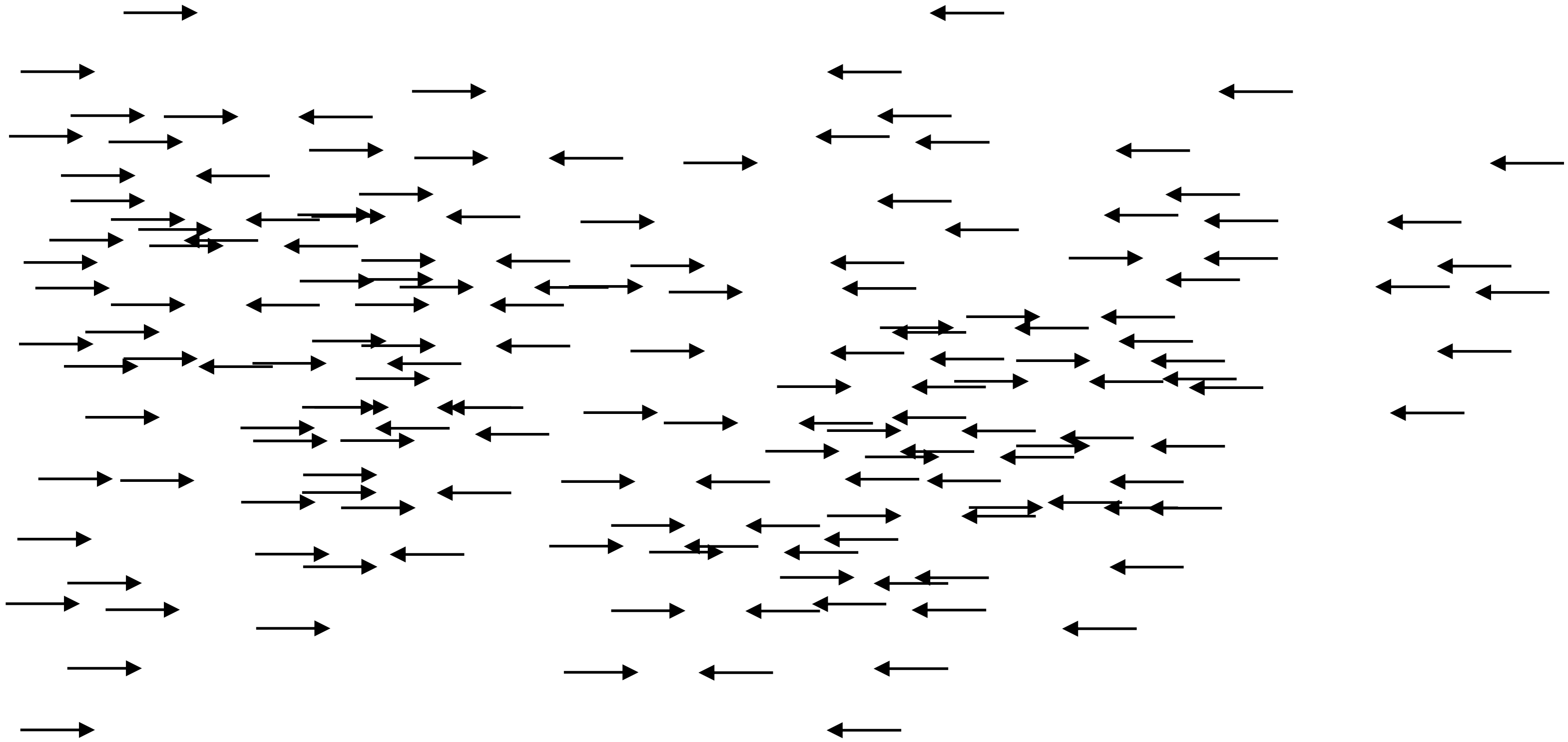
De-novo sequence assembly

1. Sequence DNA fragments from each end



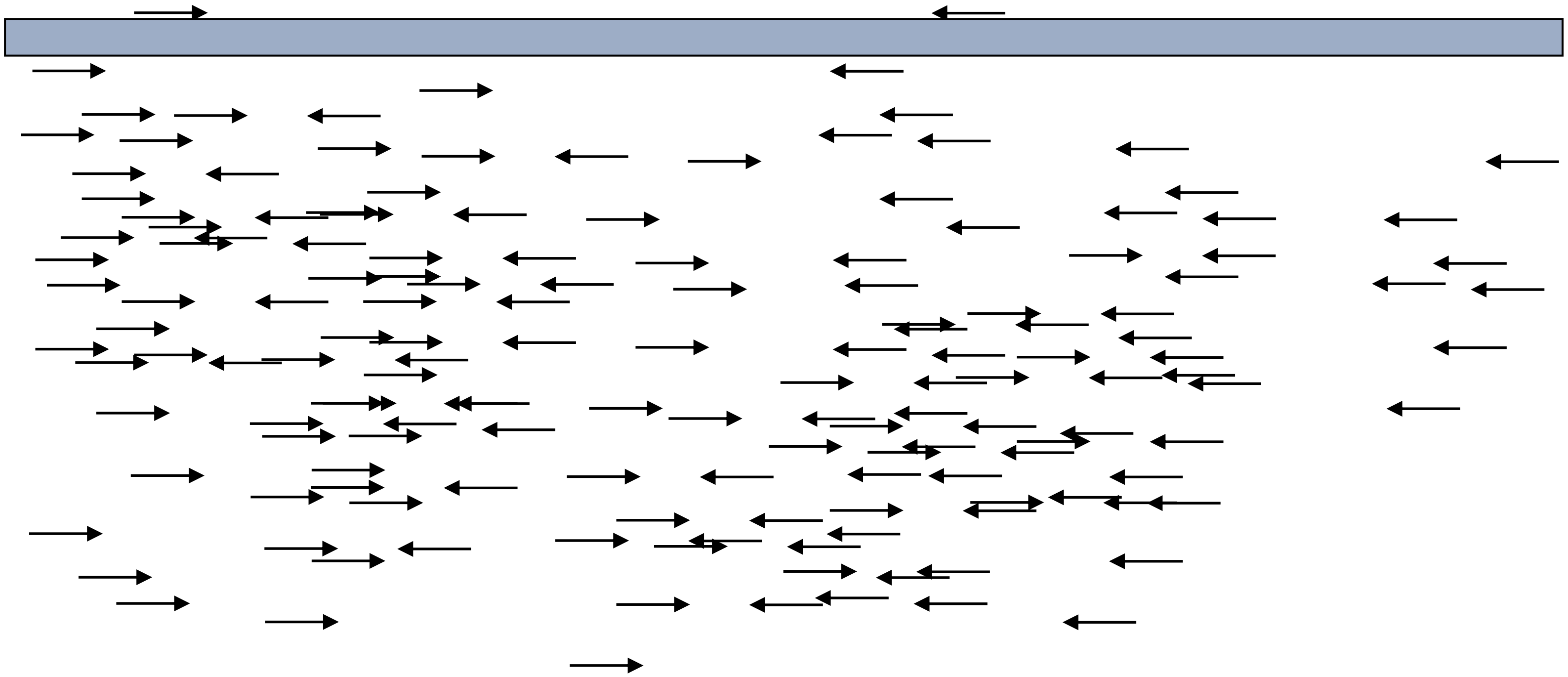
De-novo sequence assembly

1. Sequence DNA fragments from each end
2. Reads aligned to generate contigs



De-novo sequence assembly

1. Sequence DNA fragments from each end
2. Reads aligned to generate contigs

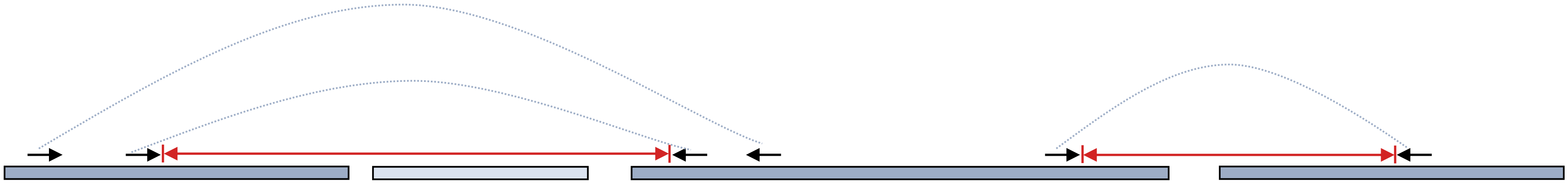


De-novo sequence assembly

1. Sequence DNA fragments from each end
2. Reads aligned to generate contigs
3. Supercontigs derived from paired reads on different contigs

De-novo sequence assembly

1. Sequence DNA fragments from each end
2. Reads aligned to generate contigs
3. Supercontigs derived from paired reads on different contigs



4. Ordering of contigs is determined
5. Different insert lengths and read lengths can resolve ambiguities
6. Insert size can be increased to 2-20kb by using mate-pair libraries (helps to span repetitive regions)

Mate-pair vs paired-end

- Often causes confusion
- Paired-end usually refers to libraries prepared for the Illumina platform with insert sizes 50-500bp.
- Mate-pair is a different library preparation protocol and usually produces insert sizes 2kb-20kb.

Contents

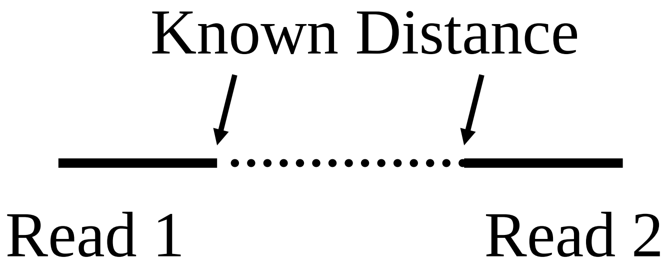
- **Alignment algorithms for short-reads**
  - Background – Blast (why can't we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
- **Assembly algorithms for short reads**
  - **Effect of repeats**
  - Overlap-Consensus
  - de Bruijn graphs
  - Assembly evaluation metrics
  - Typical assembly pipeline

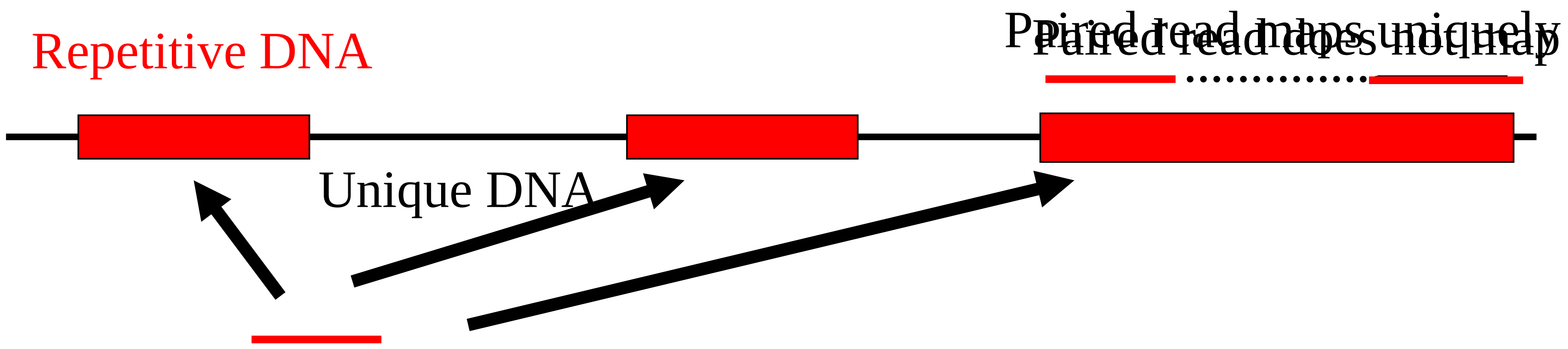
Repetitive sequence

- Main reason for fragmented genome assemblies
- Additional sequencing depth will not help overcome repeat limited assemblies

Whiteford N, Haslam N, Weber G, et al. An analysis of the feasibility of short read sequencing. Nucleic Acids Res 2005;33:e171

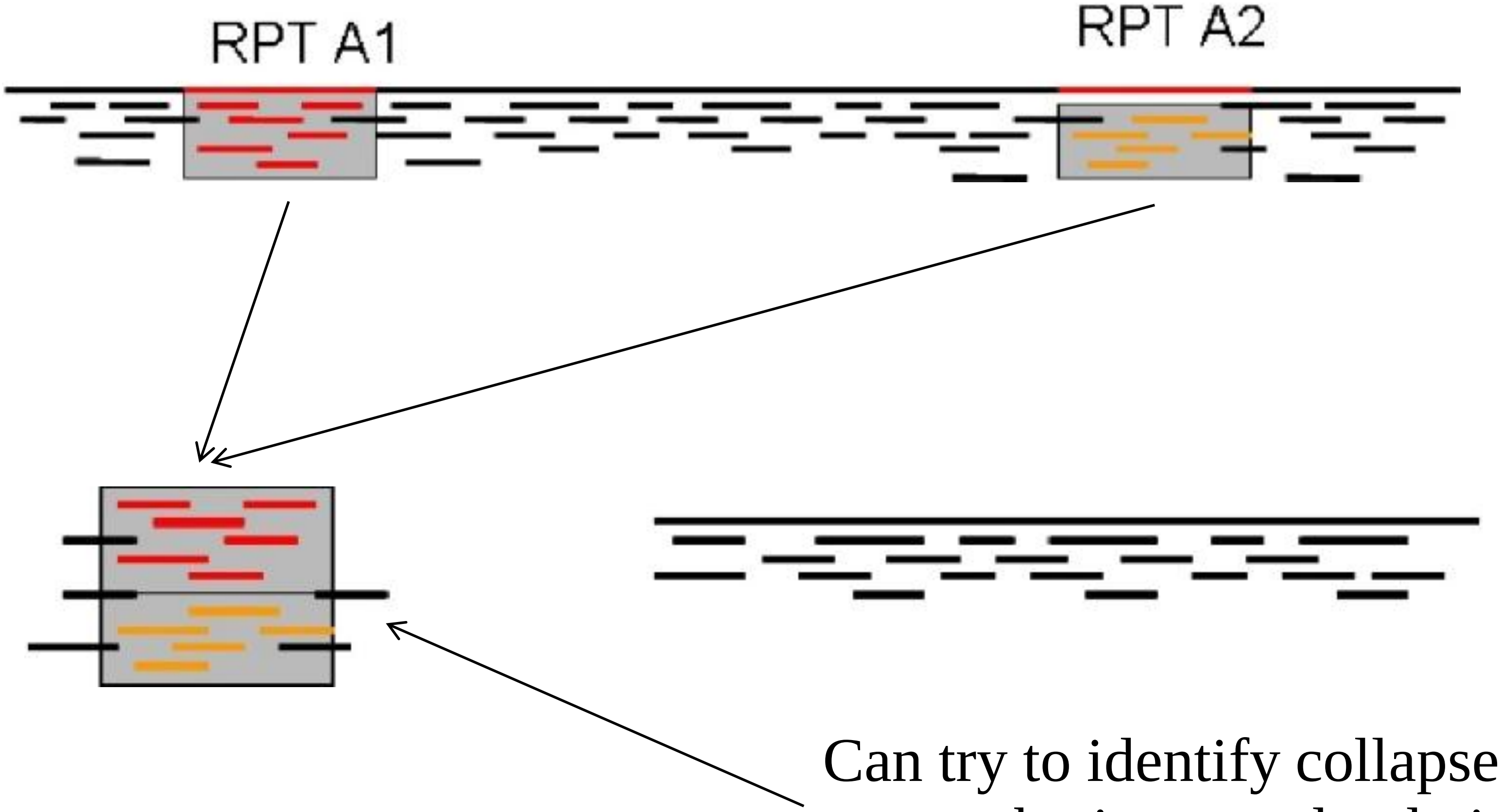
Repetitive sequence





Single read maps to multiple positions

Repetitive sequence



Can try to identify collapsed repeats by increased relative coverage

http://www.cbcb.umd.edu/research/assembly_primer.shtml

Repetitive sequence

- Main reason for fragmented genome assemblies
- Additional sequencing depth will not help overcome repeat limited assemblies

- Can estimate the number of repetitive regions, based on relative coverage
- Only longer reads or paired-end/mate-pair reads can overcome this
- PacBio reads can extend up to 10-20kb but expensive and impractical for most labs
- Large mate pair insert sizes ~20kb are possible, but library preparation is inefficient (2-3 days of trial and error). Also a significant fraction will be error-prone and/or chimeric

Whiteford N, Haslam N, Weber G, et al. An analysis of the feasibility of short read sequencing. Nucleic Acids Res 2005;33:e171

Assumptions made by de-novo assemblers

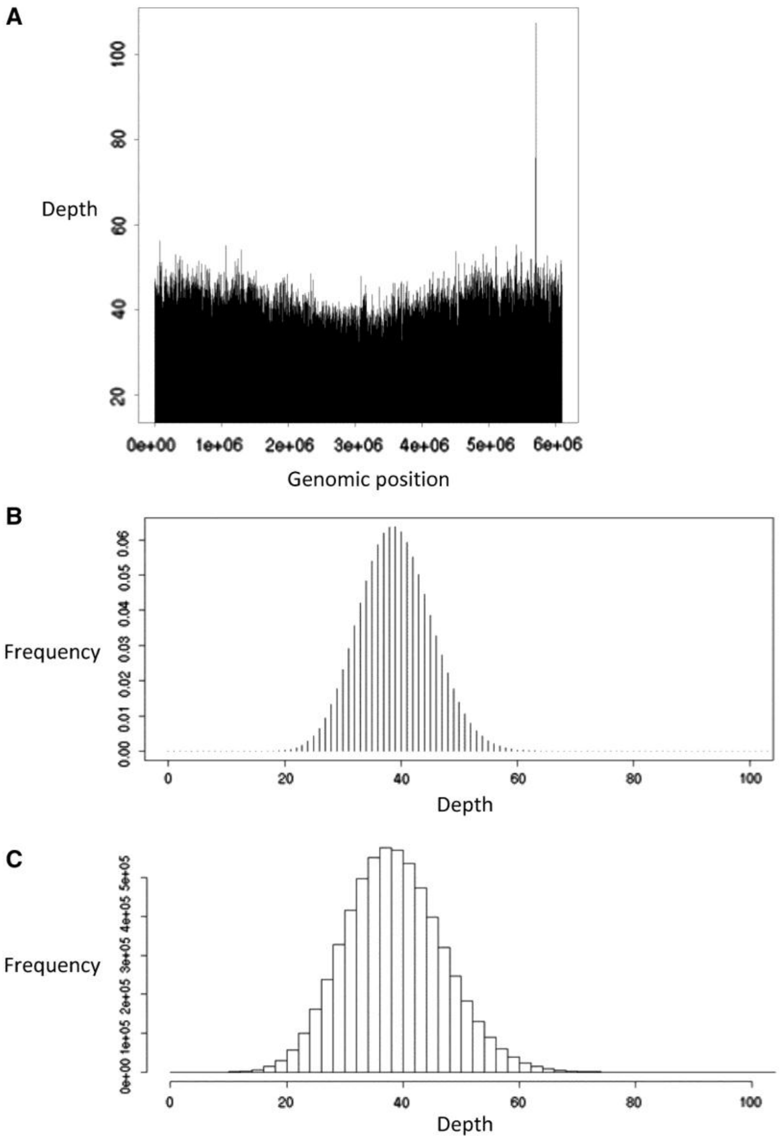
- Based on Lander-Waterman model
  - Number of times a base is sequenced follows a Poisson distribution
  - Reads are randomly distributed throughout a genome
  - The ability to detect an overlap between two reads is not dependent on the base-composition of the read

$$P = 1 - \left[1 - \frac{L}{G}\right]^N .$$

L = Read length
N = Number of reads
G = Genome size
P = Probability base is sequenced

Lander, E.S. and Waterman, M.S. (1988). "Genomic Mapping by Fingerprinting Random Clones: A Mathematical Analysis". *Genomics* **2** (3): 231–239

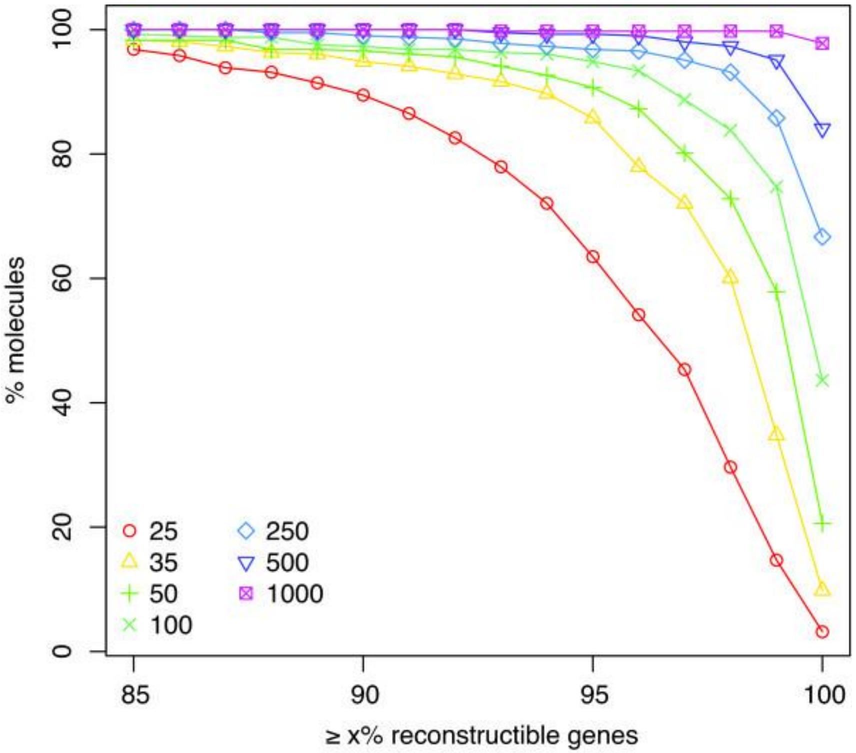
Assumptions are not true



Paszkiewicz K , Studholme D J Brief Bioinform 2010;11:457-472

NGS de-novo assemblies are draft quality at best

- 500 contigs covering most of a bacterial genome can be obtained in 1 week from genomic DNA to Genbank submission
- To get 1 contigs covering **all** genomic sequence could take many months
- Is the extra effort worth it?
- Short answer: Usually not.



Assembly complexity of prokaryotic genomes using short reads
Carl Kingsford , **Michael C Schatz** and **Mihai Pop**
BMC Bioinformatics 2010, **11:**21

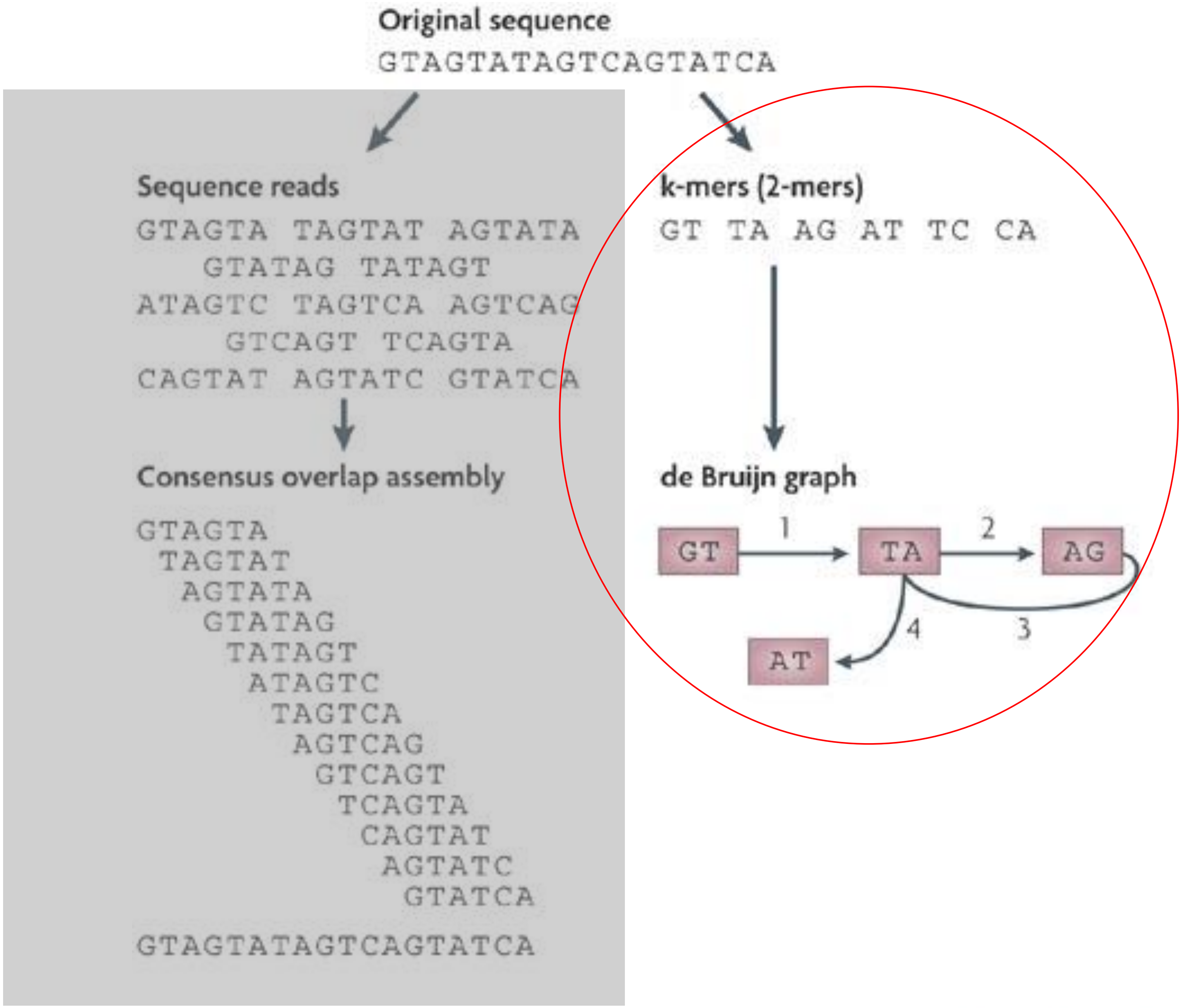
Contents

- **Alignment algorithms for short-reads**
  - Background – Blast (why can’t we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
- **Assembly algorithms for short reads**
  - Effect of repeats
  - **Overlap-Consensus**
  - **de Bruijn graphs**
  - Assembly evaluation metrics
  - Typical assembly pipeline

Overlap consensus vs. de Bruijn

- **2 main categories of assembly algorithms**
  - Overlap Consensus (OLC) and de Bruijn graph assemblers

- OLC
  - Primarily used for Sanger and hybrid assemblies
  - Memory constraints prevent its use beyond 1 million reads or so
- de Bruijn
  - Primarily used for NGS assemblies
  - Still memory hungry but possible



Original sequence

GTAGTATAGTCAGTATCA

Sequence reads

```
GTAGTA TAGTAT AGTATA
   GTATAG TATAGT
ATAGTC TAGTCA AGTCAG
    GTCAGT TCAGTA
CAGTAT AGTATC GTATCA
```

Consensus overlap assembly

```
GTAGTA
 TAGTAT
  AGTATA
   GTATAG
    TATAGT
     ATAGTC
      TAGTCA
       AGTCAG
        GTCAGT
         TCAGTA
          CAGTAT
           AGTATC
            GTATCA
GTAGTATAGTCAGTATCA
```

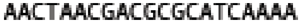
k-mers (2-mers)

GT TA AG AT TC CA

de Bruijn graph

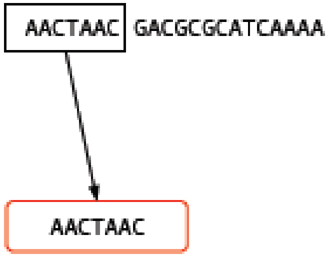
Nature Reviews | Microbiology

de Bruijn graph assembly



AACTAACGACGCGCATCAAAA

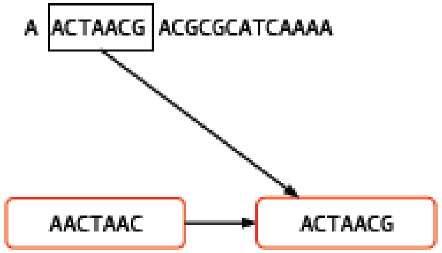
de Bruijn graph assembly



AACTAAC GACGCGCATCAAAA

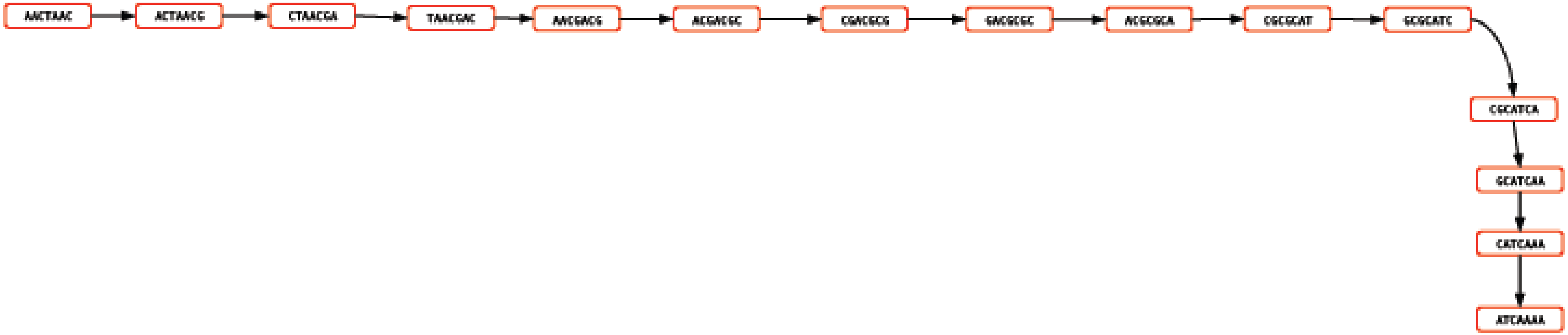
AACTAAC

de Bruijn graph assembly



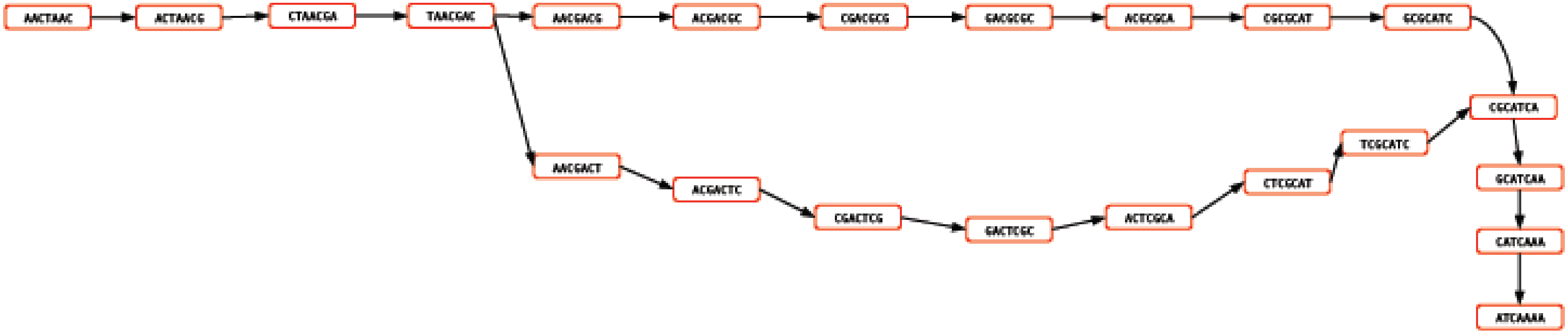
de Bruijn graph assembly

AACTAACGACGCGCATCAAAA

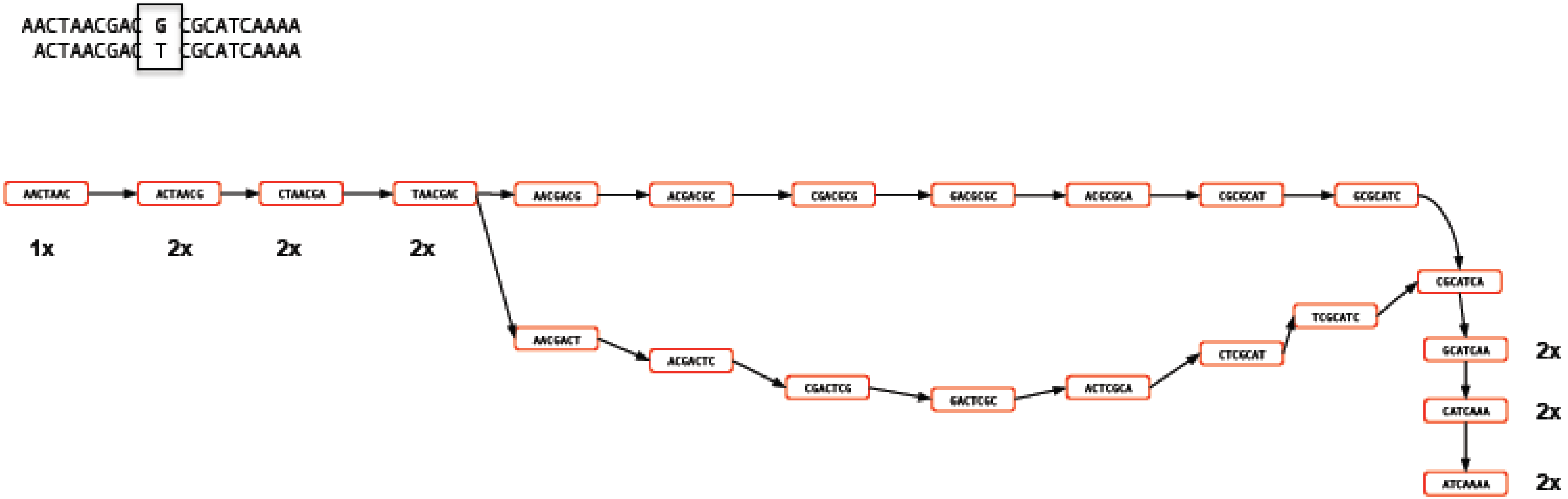


de Bruijn graph assembly

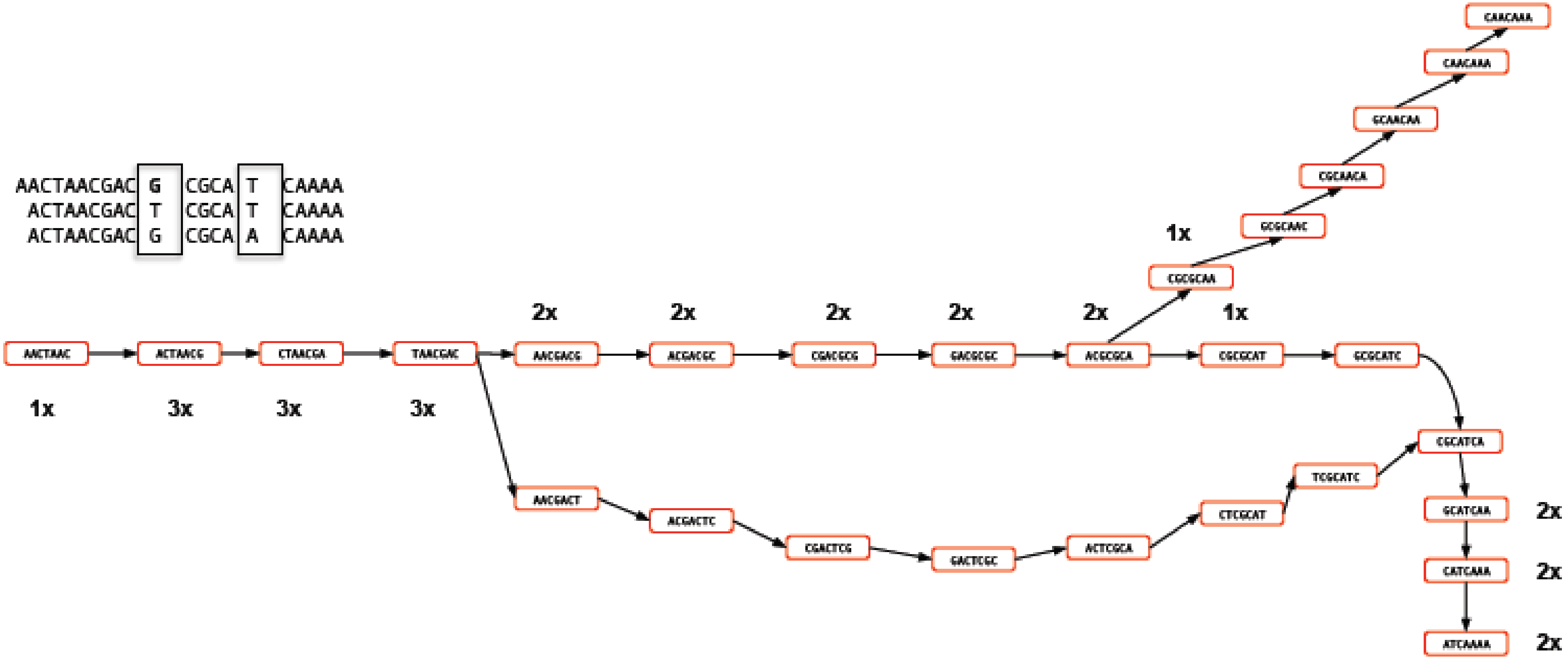
```
AACTAACGAC G CGCATCAAAA
 ACTAACGAC T CGCATCAAAA
```



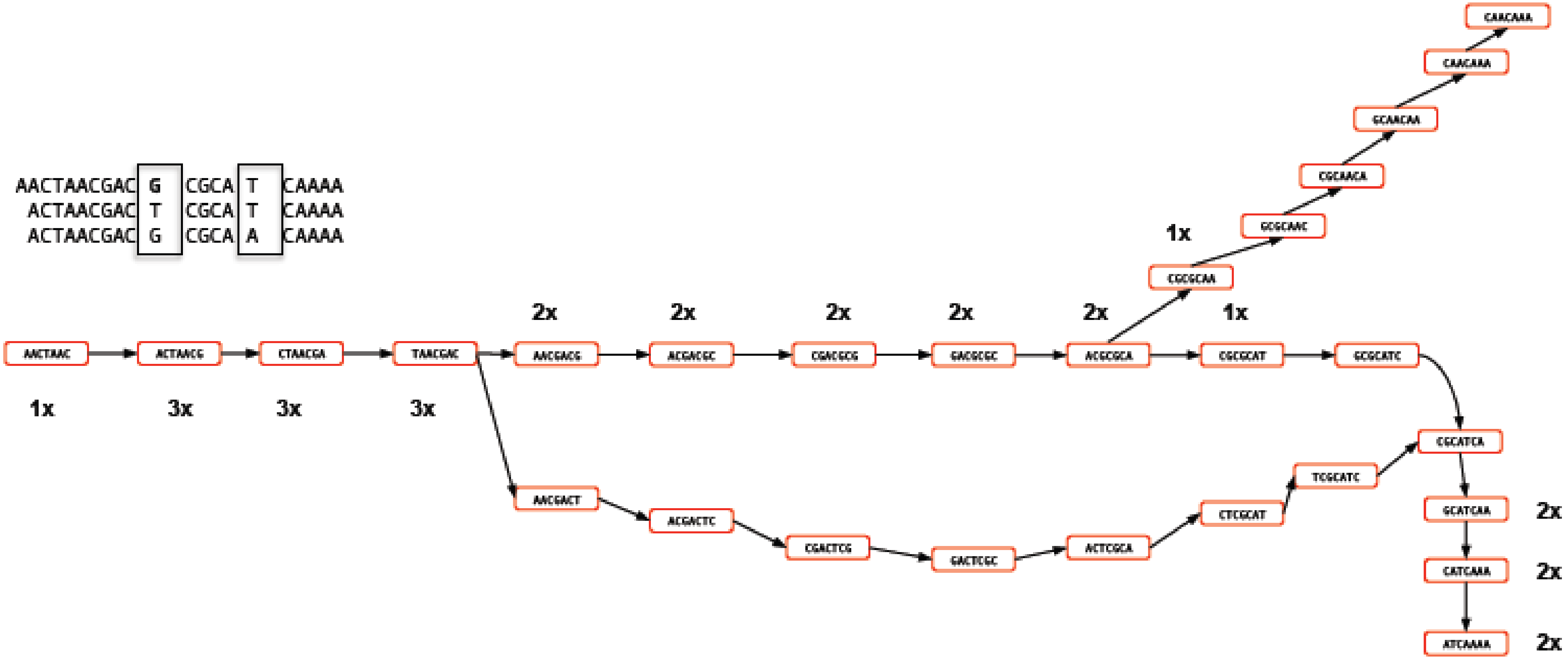
de Bruijn graph assembly

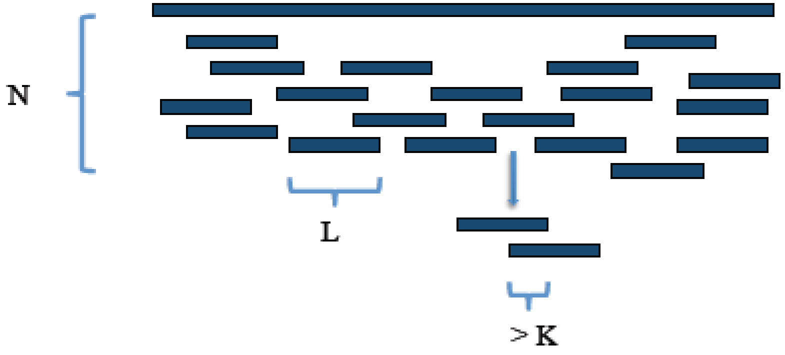


de Bruijn graph assembly



de Bruijn graph assembly

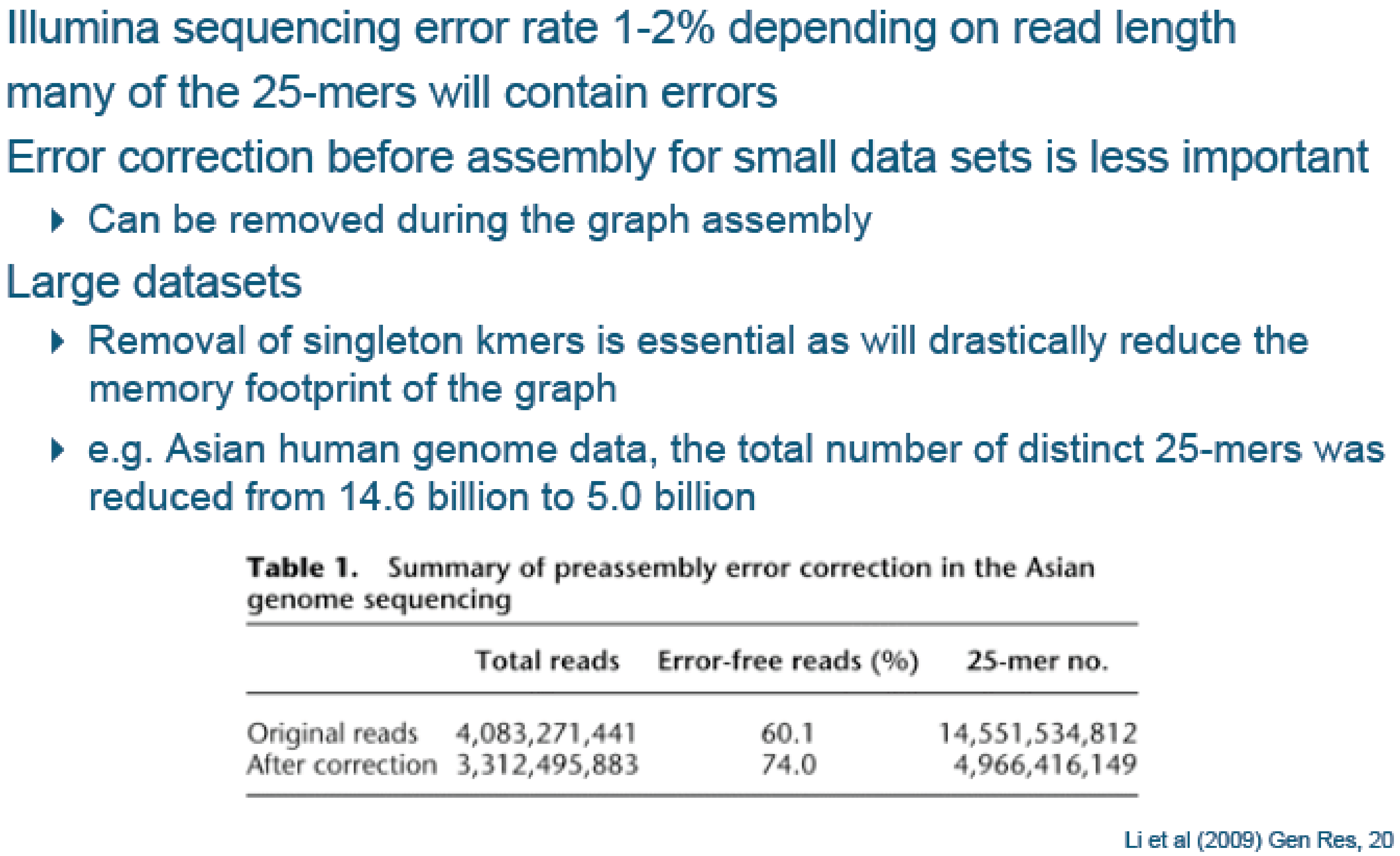




$$P(d > 0) = 1 - e^{-(N(L-K)/G)}$$

Diagrams courtesy M. Caccamo, TGAC

Dealing with errors



Illumina sequencing error rate 1-2% depending on read length

many of the 25-mers will contain errors

Error correction before assembly for small data sets is less important

- Can be removed during the graph assembly

Large datasets

- Removal of singleton kmers is essential as will drastically reduce the memory footprint of the graph
- e.g. Asian human genome data, the total number of distinct 25-mers was reduced from 14.6 billion to 5.0 billion

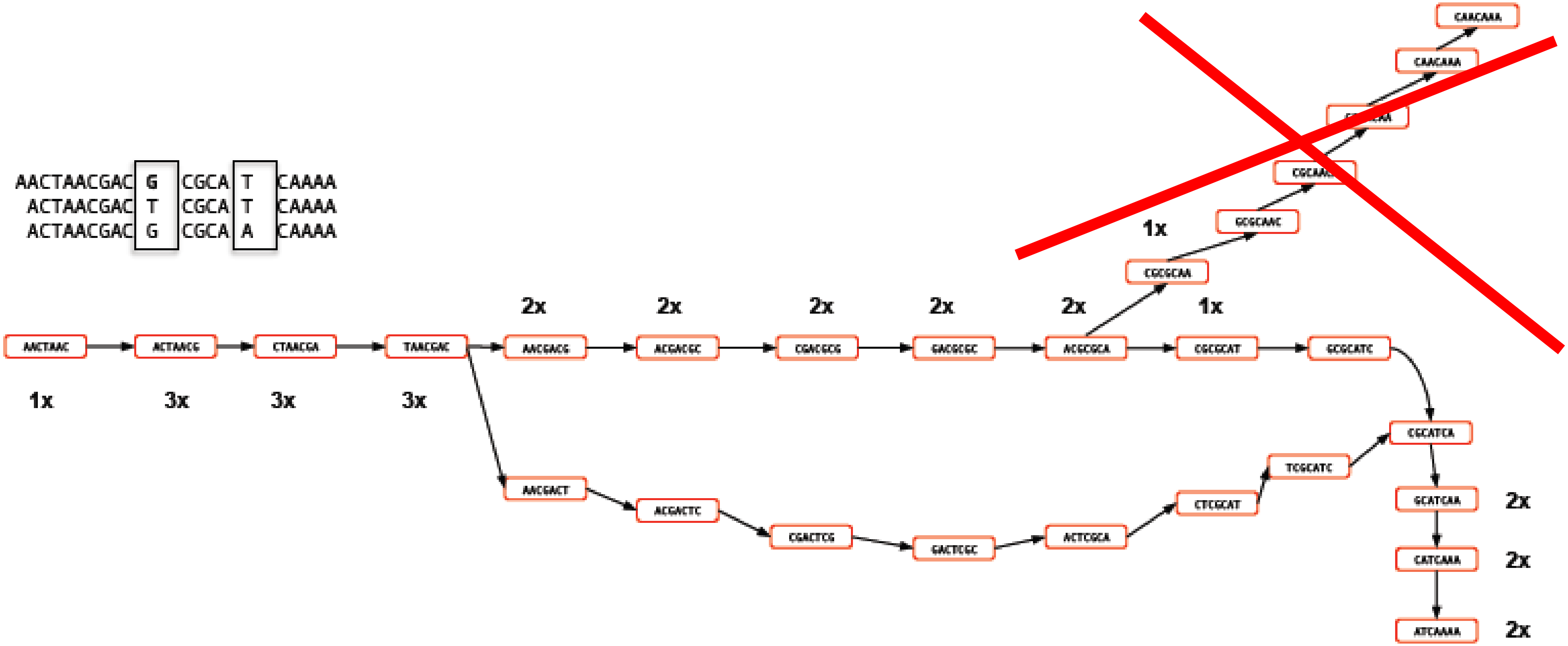
Table 1. Summary of preassembly error correction in the Asian genome sequencing

	Total reads	Error-free reads (%)	25-mer no.
Original reads	4,083,271,441	60.1	14,551,534,812
After correction	3,312,495,883	74.0	4,966,416,149

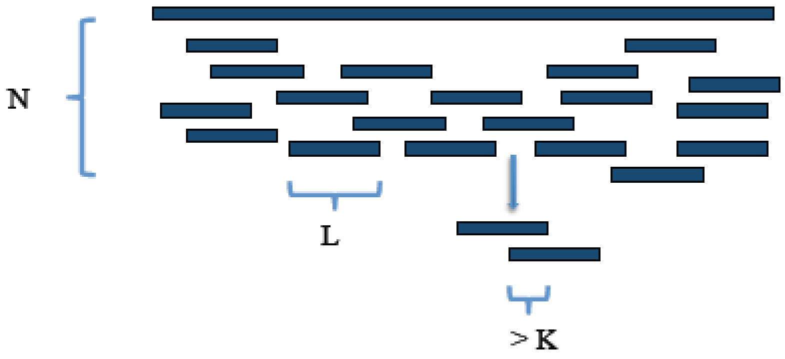
Li et al (2009) Gen Res, 20

Thomas Keane and Jan Aerts, Wellcome Trust Sanger

de Bruijn graph assembly error correction



```
AACTAACGAC G CGCA T CAAAA
 ACTAACGAC T CGCA T CAAAA
 ACTAACGAC G CGCA A CAAAA
```



$$P(d > 0) = 1 - e^{-(N(L-K)/G)}$$

Diagrams courtesy M. Caccamo, TGAC

Errors or rare sequence?

- Depends on the type of data:
  - Assumptions are probably true for single haploid genome data
  - Diploid and polyploid expect any branches to have equal coverage
  - Less clear for RNA-seq due to splicing
  - Completely false assumption for metagenomic and metatranscriptomic data!

Short read assemblers

- First de Bruijn based assembler was Newbler
  - Adapted to handle main 454 error – indels in homopolymers

- Several other de Bruijn assemblers developed subsequently
  - Velvet, Euler-SR, ABySS, ALLPATHS2
  - Most can use paired-end and mate-pair information

- Most cannot deal with mammalian sized genomes
  - ABySS – distributed genome assembly via MPI
  - SOAPde-novo (BGI) Cortex (TGAC)
    - Early removal of spurious errors

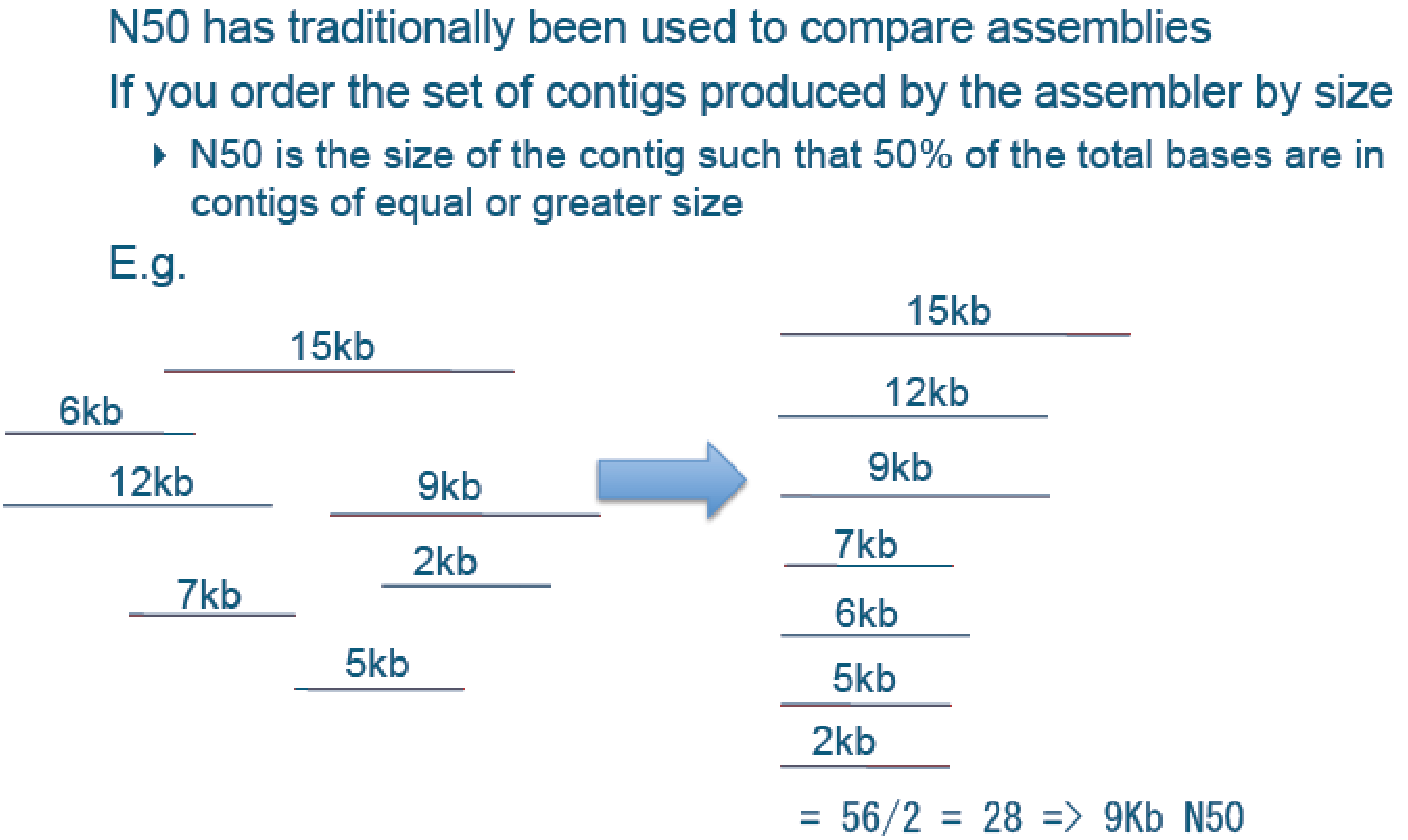
- Hybrid assemblers
  - MIRA – capable of assembling 454, Sanger and short reads
  - Memory hungry

- Other approaches
  - String graph assemblers
  - Fermi, SGA
  - Correcting PacBio reads with Illumina

Contents

- **Alignment algorithms for short-reads**
  - Background – Blast (why can’t we use it?)
  - Adapting hashed seed-extend algorithms to work with shorter reads
  - Suffix/Prefix Tries
  - Other alignment considerations
  - Typical alignment pipeline
- **Assembly algorithms for short reads**
  - Effect of repeats
  - Overlap-Consensus
  - de Bruijn graphs
  - **Assembly evaluation metrics**
  - Typical assembly pipeline

Assembly evaluation – N50



N50 has traditionally been used to compare assemblies

If you order the set of contigs produced by the assembler by size

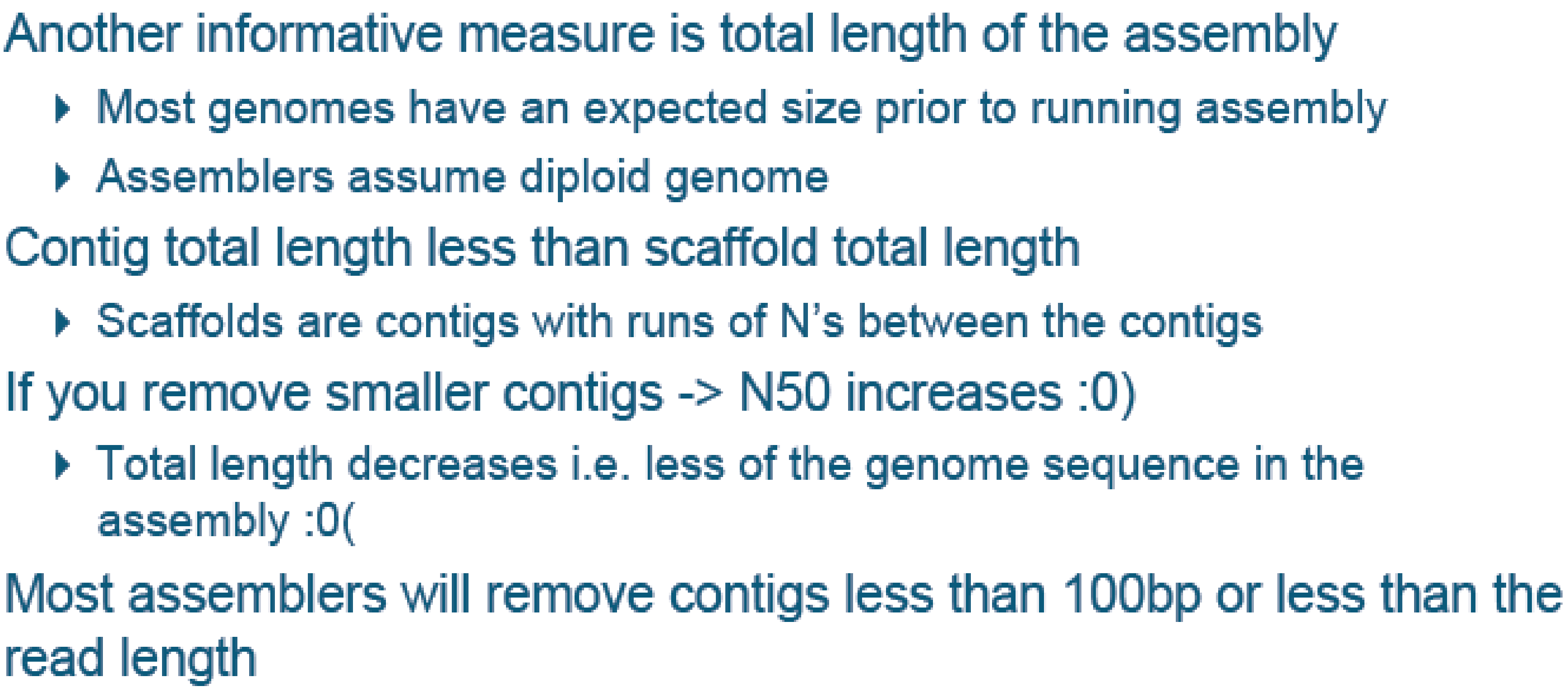
- N50 is the size of the contig such that 50% of the total bases are in contigs of equal or greater size

E.g.

= 56/2 = 28 => 9Kb N50

Thomas Keane and Jan Aerts, Wellcome Trust Sanger

Assembly length vs. N50



Another informative measure is total length of the assembly

- Most genomes have an expected size prior to running assembly
- Assemblers assume diploid genome

Contig total length less than scaffold total length

- Scaffolds are contigs with runs of N’s between the contigs

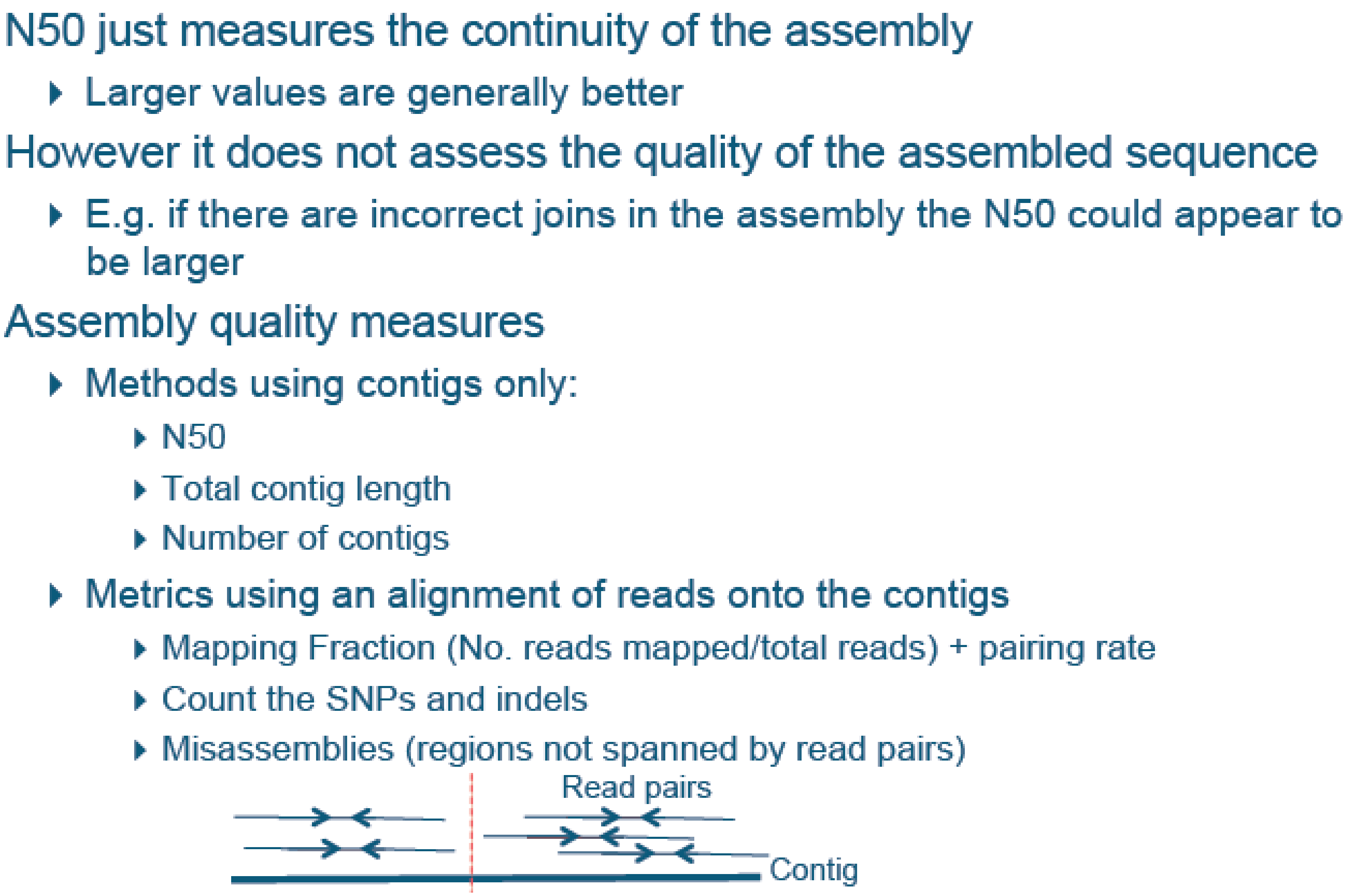
If you remove smaller contigs -> N50 increases :0)

- Total length decreases i.e. less of the genome sequence in the assembly :0(

Most assemblers will remove contigs less than 100bp or less than the read length

Thomas Keane and Jan Aerts, Wellcome Trust Sanger

Assembly evaluation metrics



N50 just measures the continuity of the assembly

- Larger values are generally better

However it does not assess the quality of the assembled sequence

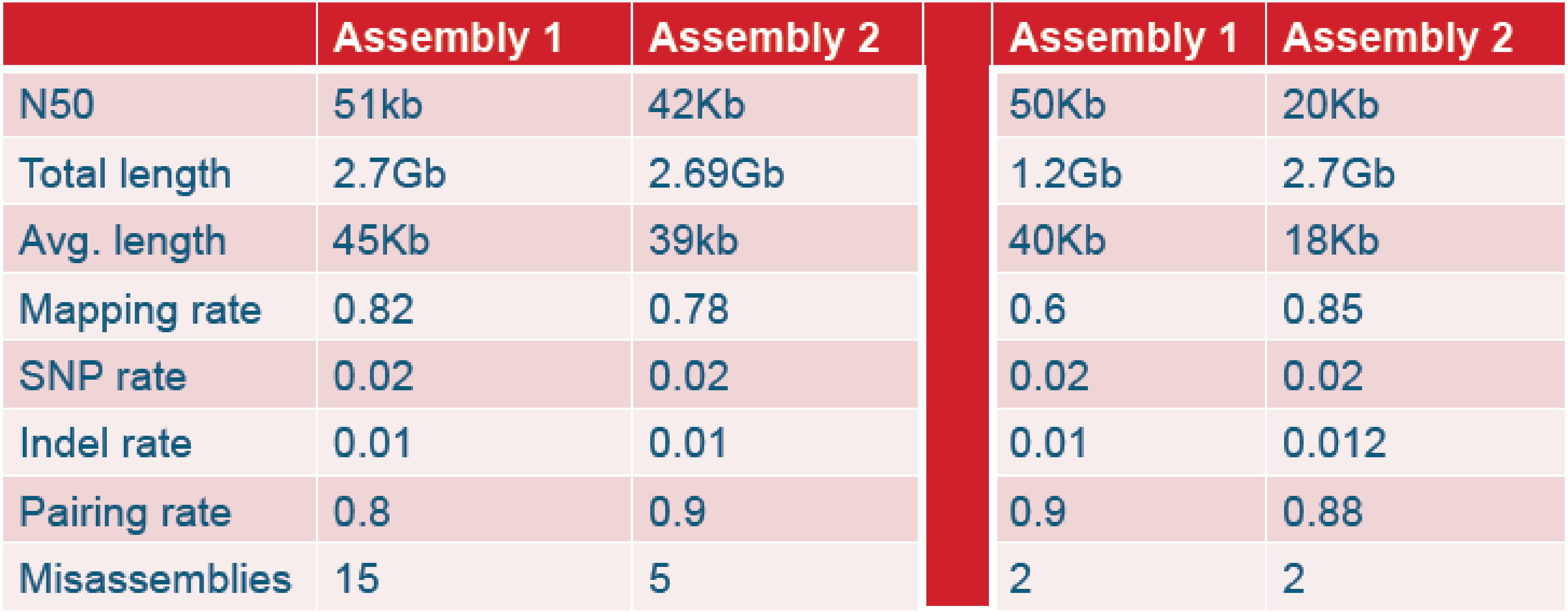
- E.g. if there are incorrect joins in the assembly the N50 could appear to be larger

Assembly quality measures

- Methods using contigs only:
  - N50
  - Total contig length
  - Number of contigs
- Metrics using an alignment of reads onto the contigs
  - Mapping Fraction (No. reads mapped/total reads) + pairing rate
  - Count the SNPs and indels
  - Misassemblies (regions not spanned by read pairs)

Thomas Keane and Jan Aerts, Wellcome Trust Sanger

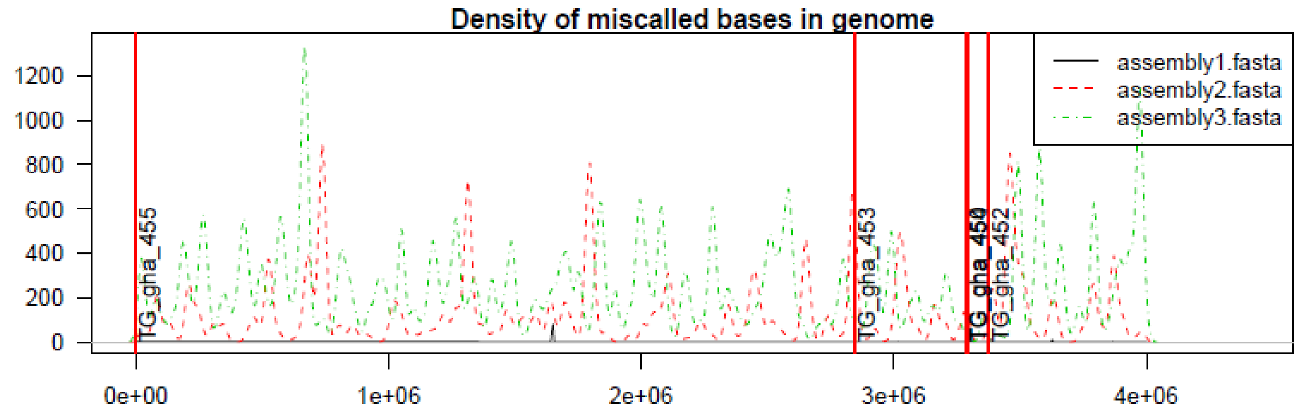
Which human assembly is better? Why?

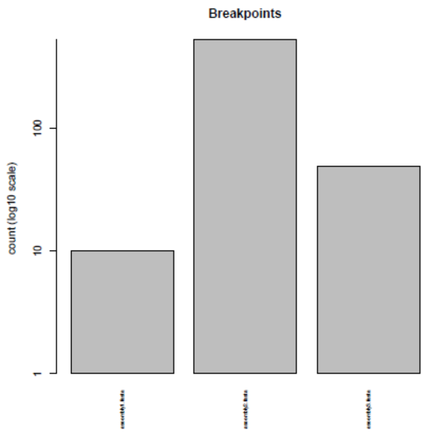


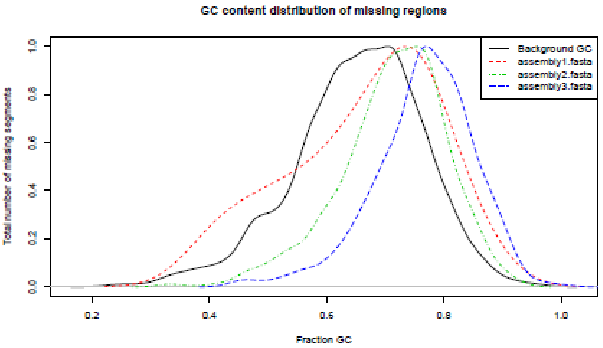
	Assembly 1	Assembly 2		Assembly 1	Assembly 2
N50	51kb	42Kb		50Kb	20Kb
Total length	2.7Gb	2.69Gb		1.2Gb	2.7Gb
Avg. length	45Kb	39kb		40Kb	18Kb
Mapping rate	0.82	0.78		0.6	0.85
SNP rate	0.02	0.02		0.02	0.02
Indel rate	0.01	0.01		0.01	0.012
Pairing rate	0.8	0.9		0.9	0.88
Misassemblies	15	5		2	2

Thomas Keane and Jan Aerts, Wellcome Trust Sanger

Assembly benchmarking software







Darling et al Mauve Assembly Metrics *Bioinformatics (2011) btr451 first published online August 2, 2011*

http://t.co/BbpbTPz

Types of assemblers

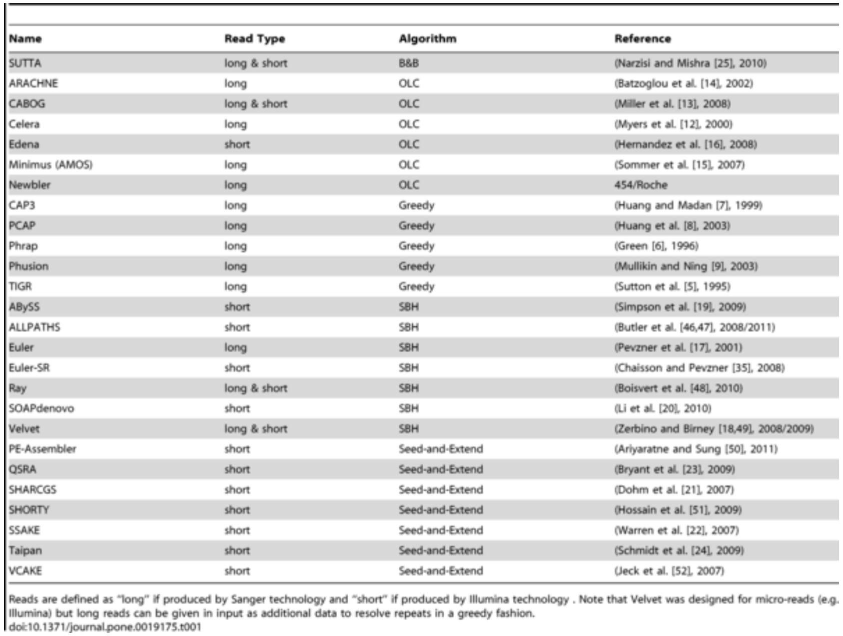
2 main categories, many variations

Each tends to have its own niche

Memory and hardware requirements can differ substantially

Typically a parameter scan is need to get the ‘best’ assembly

This means many assemblies need to be generated



Name	Read Type	Algorithm	Reference
SUTTA	long & short	B&B	(Narzisi and Mishra [25], 2010)
ARACHNE	long	OLC	(Batzoglou et al. [14], 2002)
CABOG	long & short	OLC	(Miller et al. [13], 2008)
Celera	long	OLC	(Myers et al. [12], 2000)
Edena	short	OLC	(Hernandez et al. [16], 2008)
Minimus (AMOS)	long	OLC	(Sommer et al. [15], 2007)
Newbler	long	OLC	454/Roche
CAP3	long	Greedy	(Huang and Madan [7], 1999)
PCAP	long	Greedy	(Huang et al. [8], 2003)
Phrap	long	Greedy	(Green [6], 1996)
Phusion	long	Greedy	(Mullikin and Ning [9], 2003)
TIGR	long	Greedy	(Sutton et al. [5], 1995)
ABySS	short	SBH	(Simpson et al. [19], 2009)
ALLPATHS	short	SBH	(Butler et al. [46,47], 2008/2011)
Euler	long	SBH	(Pevzner et al. [17], 2001)
Euler-SR	short	SBH	(Chaisson and Pevzner [35], 2008)
Ray	long & short	SBH	(Boisvert et al. [48], 2010)
SOAPdenovo	short	SBH	(Li et al. [20], 2010)
Velvet	long & short	SBH	(Zerbino and Birney [18,49], 2008/2009)
PE-Assembler	short	Seed-and-Extend	(Ariyaratne and Sung [50], 2011)
QSRA	short	Seed-and-Extend	(Bryant et al. [23], 2009)
SHARCGS	short	Seed-and-Extend	(Dohm et al. [21], 2007)
SHORTY	short	Seed-and-Extend	(Hossain et al. [51], 2009)
SSAKE	short	Seed-and-Extend	(Warren et al. [22], 2007)
Taipan	short	Seed-and-Extend	(Schmidt et al. [24], 2009)
VCAKE	short	Seed-and-Extend	(Jeck et al. [52], 2007)

Reads are defined as “long” if produced by Sanger technology and “short” if produced by Illumina technology . Note that Velvet was designed for micro-reads (e.g. Illumina) but long reads can be given in input as additional data to resolve repeats in a greedy fashion.
doi:10.1371/journal.pone.0019175.t001

Narzisi G, Mishra B, Comparing De Novo Genome Assembly:
The Long and Short of It. 2011 PLoS ONE 6(4):

De novo assembly of short sequence reads
Paszkiewicz, K. Studholme, D.
Briefings in Bioinformatics
August 2010 11(5): 457-472

Which assembler is best?

- Depends on:
  - Type of reads (Illumina, SoLID, 454, Ion Torrent, PacBio, Sanger etc)
  - Paired/mate-pair data?
  - Genome
  - Repeat content
  - Available hardware

- Prokaryote genomes – Velvet
- Larger genomes ABySS or Soapdenovo

Narzisi G, Mishra B, Comparing De Novo Genome Assembly:
The Long and Short of It. 2011 PLoS ONE 6(4):

Merging assemblies

- Often assemblies are produced from 454 or Sanger data and need to be merged with Illumina data
- In order of preference:
    1. Attempt to assemble 454/Sanger reads with Illumina reads using MIRA
    2. Merge assemblies separately using minimus2 or SSPACE
    3. Input 454/Sanger contigs as part of a reference guided assembly (e.g. Velvet/Columbus)

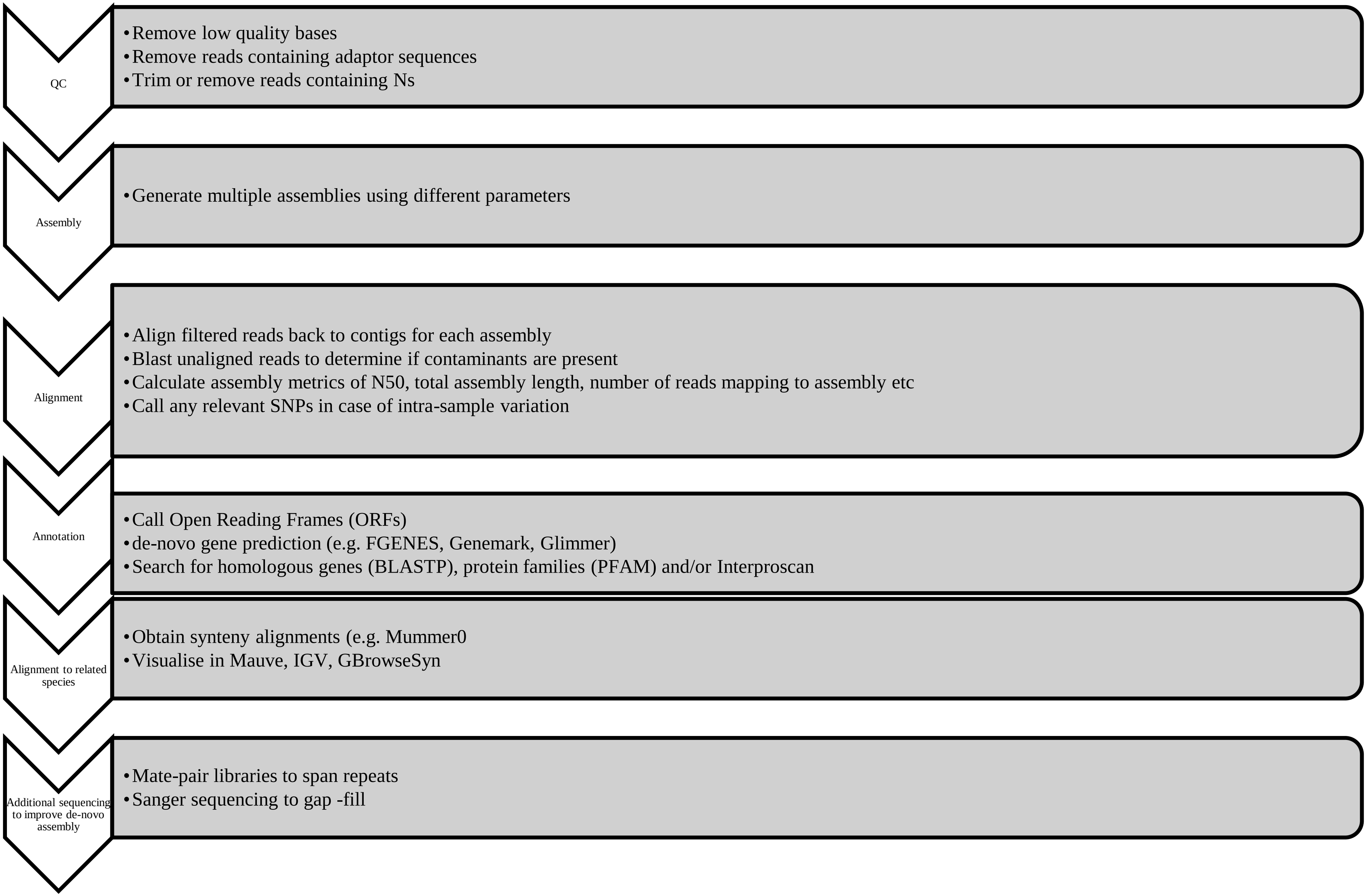
Transcriptome assembly

- de-novo transcriptome assembly is also possible
- RNA-seq reads can be assembled and isoform abundance estimated
- Much harder as Lander-Waterman assumptions of randomly distributed reads are not true
- Also complicated by splice-variants and the need to statistically model isoform abundance based on read distributions

- Oases/Velvet
- Trans-ABySS
- SOAPde-novo
- Trinity

Good experimental option for vertebrates and other non-model organisms where a reference genome is not available

Typical assembly pipeline



QC

- Remove low quality bases
- Remove reads containing adaptor sequences
- Trim or remove reads containing Ns

Assembly

- Generate multiple assemblies using different parameters

Alignment

- Align filtered reads back to contigs for each assembly
- Blast unaligned reads to determine if contaminants are present
- Calculate assembly metrics of N50, total assembly length, number of reads mapping to assembly etc
- Call any relevant SNPs in case of intra-sample variation

Annotation

- Call Open Reading Frames (ORFs)
- de-novo gene prediction (e.g. FGENES, Genemark, Glimmer)
- Search for homologous genes (BLASTP), protein families (PFAM) and/or Interproscan

Alignment to related species

- Obtain synteny alignments (e.g. Mummer0
- Visualise in Mauve, IGV, GBrowseSyn

Additional sequencing to improve de-novo assembly

- Mate-pair libraries to span repeats
- Sanger sequencing to gap -fill

Optimal de-novo sequencing strategy and review papers

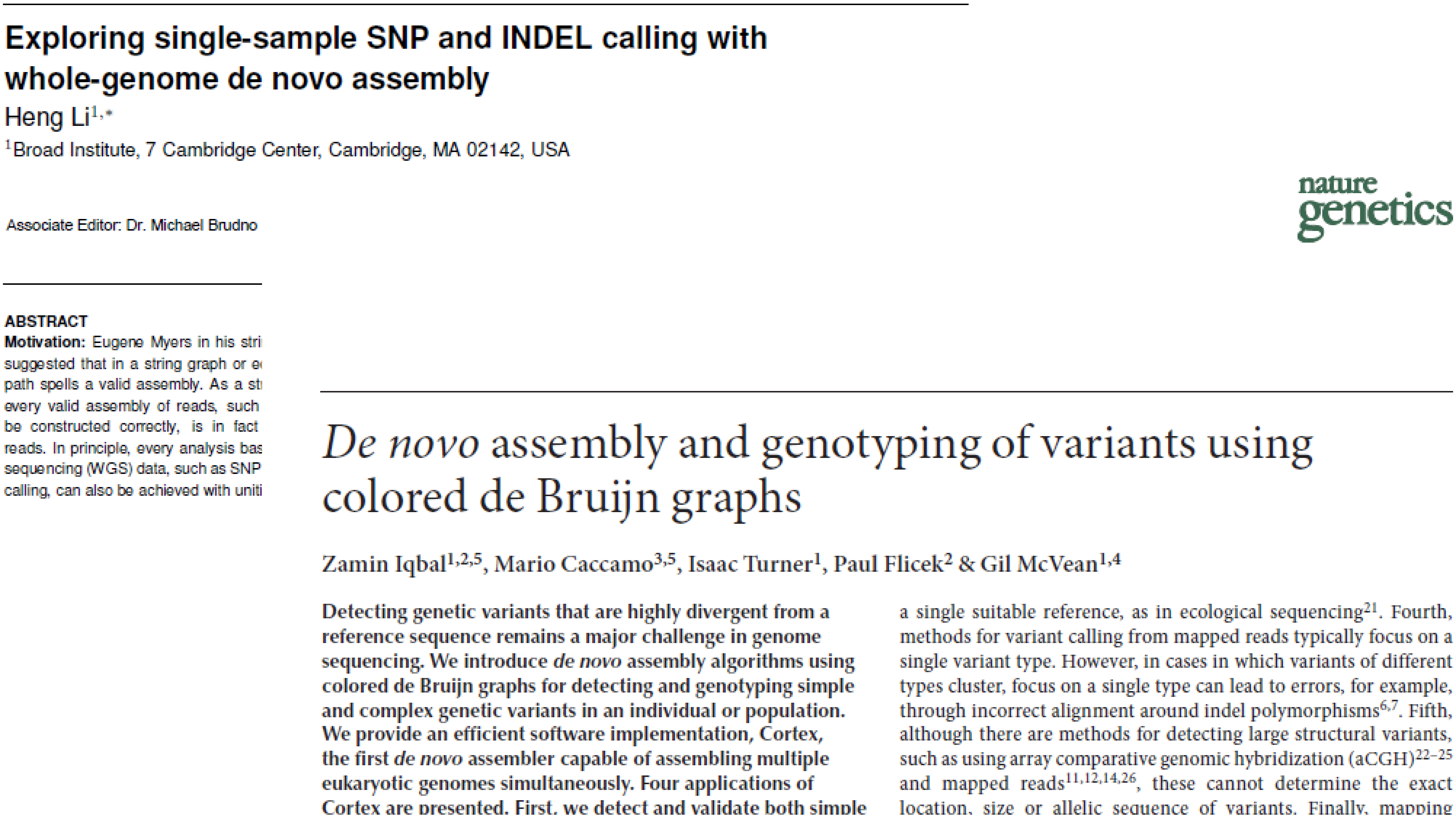
Assessing the benefits of using mate-pairs to resolve repeats in de novo short-read prokaryotic assemblies
Joshua Wetzel , Carl Kingsford and Mihai Pop
BMC Bioinformatics 2011, **12:**95

**Comparing De Novo Genome Assembly:
The Long and Short of It.**
Narzisi, G. Mishra B.
2011 PLoS ONE 6(4)

De novo assembly of short sequence reads
Paszkiewicz, K. Studholme, D.
Briefings in Bioinformatics
August 2010 11(5): 457-472

A new strategy for genome assembly using short sequence reads and reduced representation libraries
Young A.L., Abaan H.O., Zerbino D, et al.
Genome Research 2010;20:249–56.

Variant calling with de-novo assembly



Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly

Heng Li[1,*]

[1] Broad Institute, 7 Cambridge Center, Cambridge, MA 02142, USA

Associate Editor: Dr. Michael Brudno

ABSTRACT

Motivation: Eugene Myers in his stri
suggested that in a string graph or e
path spells a valid assembly. As a st
every valid assembly of reads, such
be constructed correctly, is in fact
reads. In principle, every analysis bas
sequencing (WGS) data, such as SNP
calling, can also be achieved with uniti

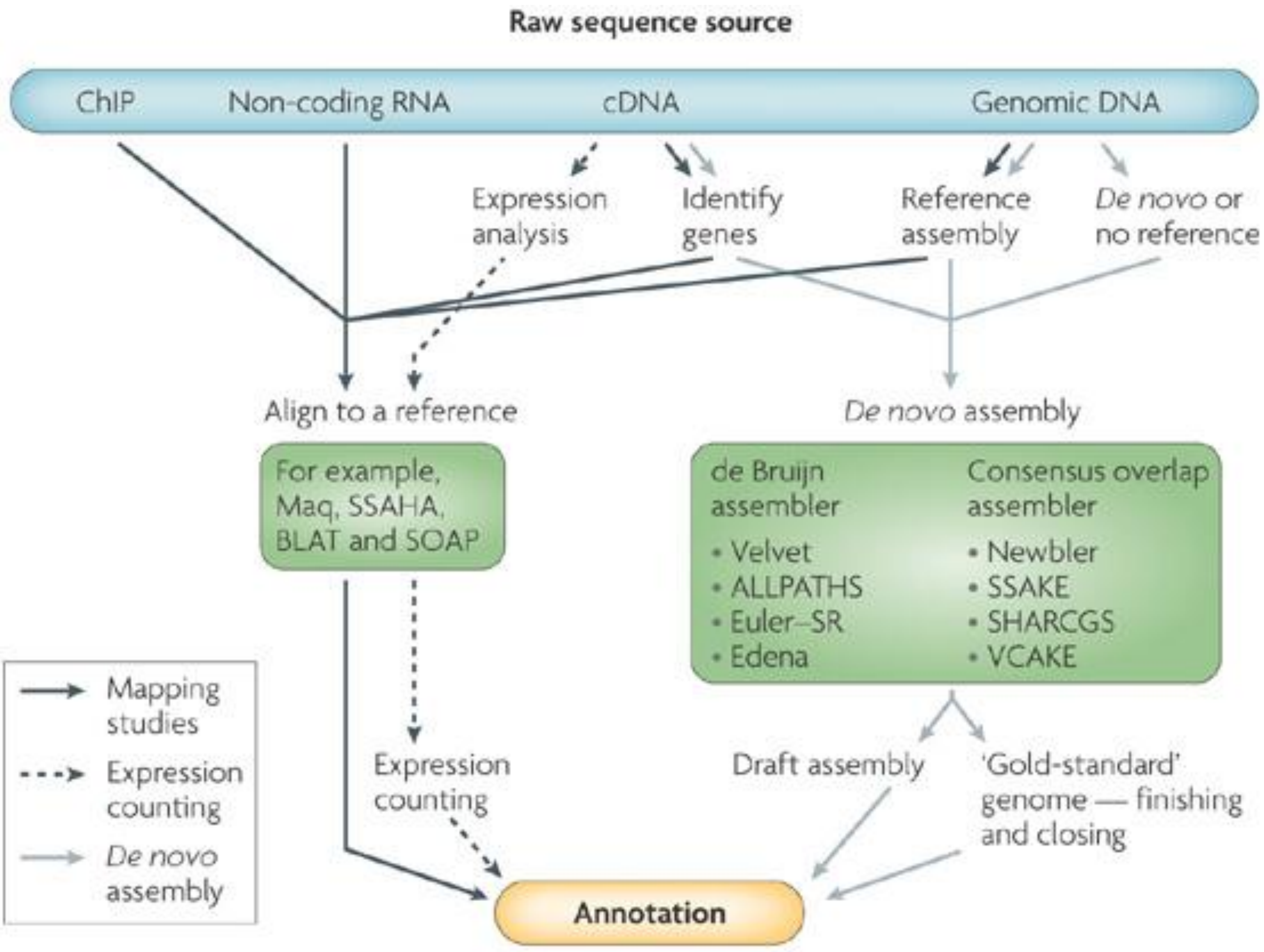
nature genetics

De novo assembly and genotyping of variants using colored de Bruijn graphs

Zamin Iqbal[1,2,5], Mario Caccamo[3,5], Isaac Turner[1], Paul Flicek[2] & Gil McVean[1,4]

Detecting genetic variants that are highly divergent from a reference sequence remains a major challenge in genome sequencing. We introduce *de novo* assembly algorithms using colored de Bruijn graphs for detecting and genotyping simple and complex genetic variants in an individual or population. We provide an efficient software implementation, Cortex, the first *de novo* assembler capable of assembling multiple eukaryotic genomes simultaneously. Four applications of Cortex are presented. First, we detect and validate both simple

a single suitable reference, as in ecological sequencing[21]. Fourth, methods for variant calling from mapped reads typically focus on a single variant type. However, in cases in which variants of different types cluster, focus on a single type can lead to errors, for example, through incorrect alignment around indel polymorphisms[6,7]. Fifth, although there are methods for detecting large structural variants, such as using array comparative genomic hybridization (aCGH)[22–25] and mapped reads[11,12,14,26], these cannot determine the exact location, size or allelic sequence of variants. Finally, mapping



Raw sequence source

ChIP
Non-coding RNA
cDNA
Genomic DNA

Expression analysis
Identify genes
Reference assembly
De novo or no reference

Align to a reference

For example, Maq, SSAHA, BLAT and SOAP

De novo assembly

de Bruijn assembler

- Velvet
- ALLPATHS
- Euler–SR
- Edena

Consensus overlap assembler

- Newbler
- SSAKE
- SHARCGS
- VCAKE

Expression counting

Draft assembly

'Gold-standard' genome — finishing and closing

Annotation

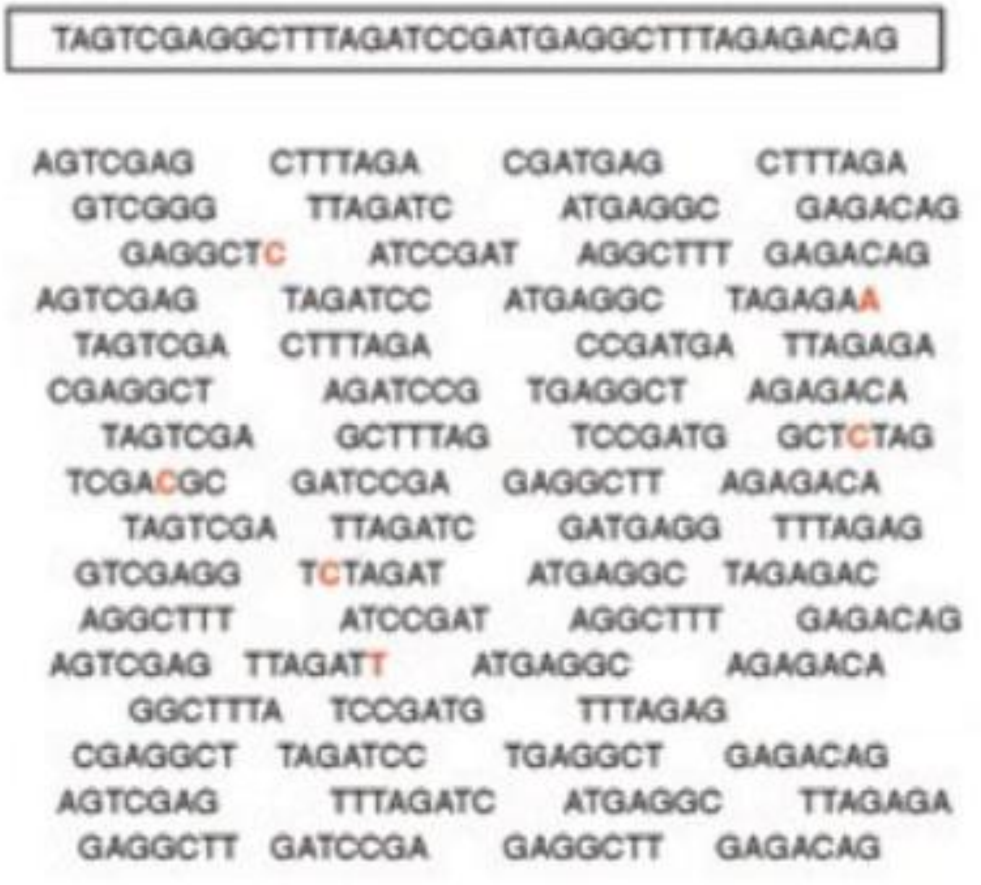
Mapping studies
Expression counting
De novo assembly

Nature Reviews | Microbiology

Questions!

biosciences.exeter.ac.uk/facilities/sequencing/usefulresources/

de-Bruijn graph assembly 1



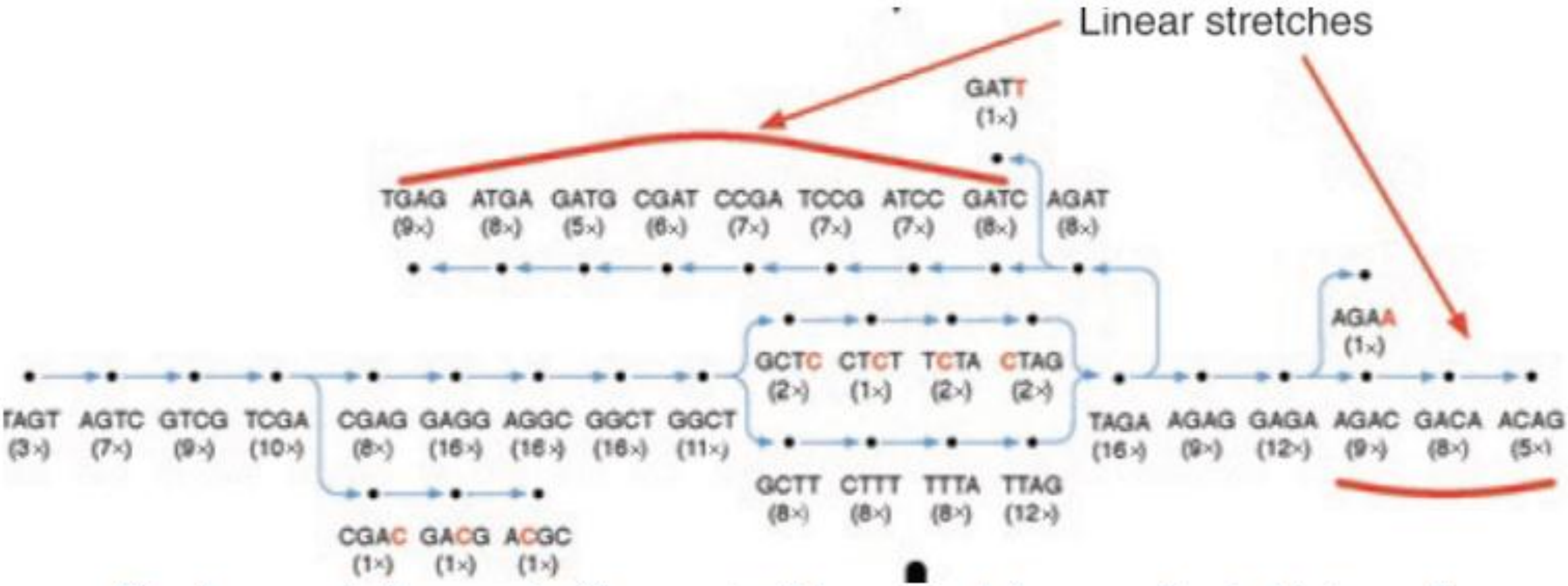
```
TAGTCGAGGCTTTAGATCCGATGAGGCTTTAGAGACAG

AGTCGAG     CTTTAGA     CGATGAG      CTTTAGA
  GTCGGG      TTAGATC      ATGAGGC      GAGACAG
    GAGGCTC      ATCCGAT     AGGCTTT  GAGACAG
AGTCGAG      TAGATCC     ATGAGGC     TAGAGAA
  TAGTCGA   CTTTAGA          CCGATGA   TTAGAGA
 CGAGGCT        AGATCCG   TGAGGCT     AGAGACA
   TAGTCGA      GCTTTAG      TCCGATG   GCTCTAG
 TCGACGC     GATCCGA    GAGGCTT    AGAGACA
    TAGTCGA    TTAGATC       GATGAGG    TTTAGAG
  GTCGAGG     TCTAGAT       ATGAGGC   TAGAGAC
  AGGCTTT         ATCCGAT      AGGCTTT      GAGACAG
 AGTCGAG  TTAGATT       ATGAGGC        AGAGACA
    GGCTTTA   TCCGATG        TTTAGAG
  CGAGGCT   TAGATCC      TGAGGCT     GAGACAG
 AGTCGAG         TTTAGATC     ATGAGGC       TTAGAGA
  GAGGCTT  GATCCGA     GAGGCTT    GAGACAG
```

Genome is sampled with random sequencing 7bp reads (e.g. Illumina or 454)
Note errors in the reads are represented in red

Flicek & Birney (2009) Nat Meth, 6

de-Bruijn graph assembly 2

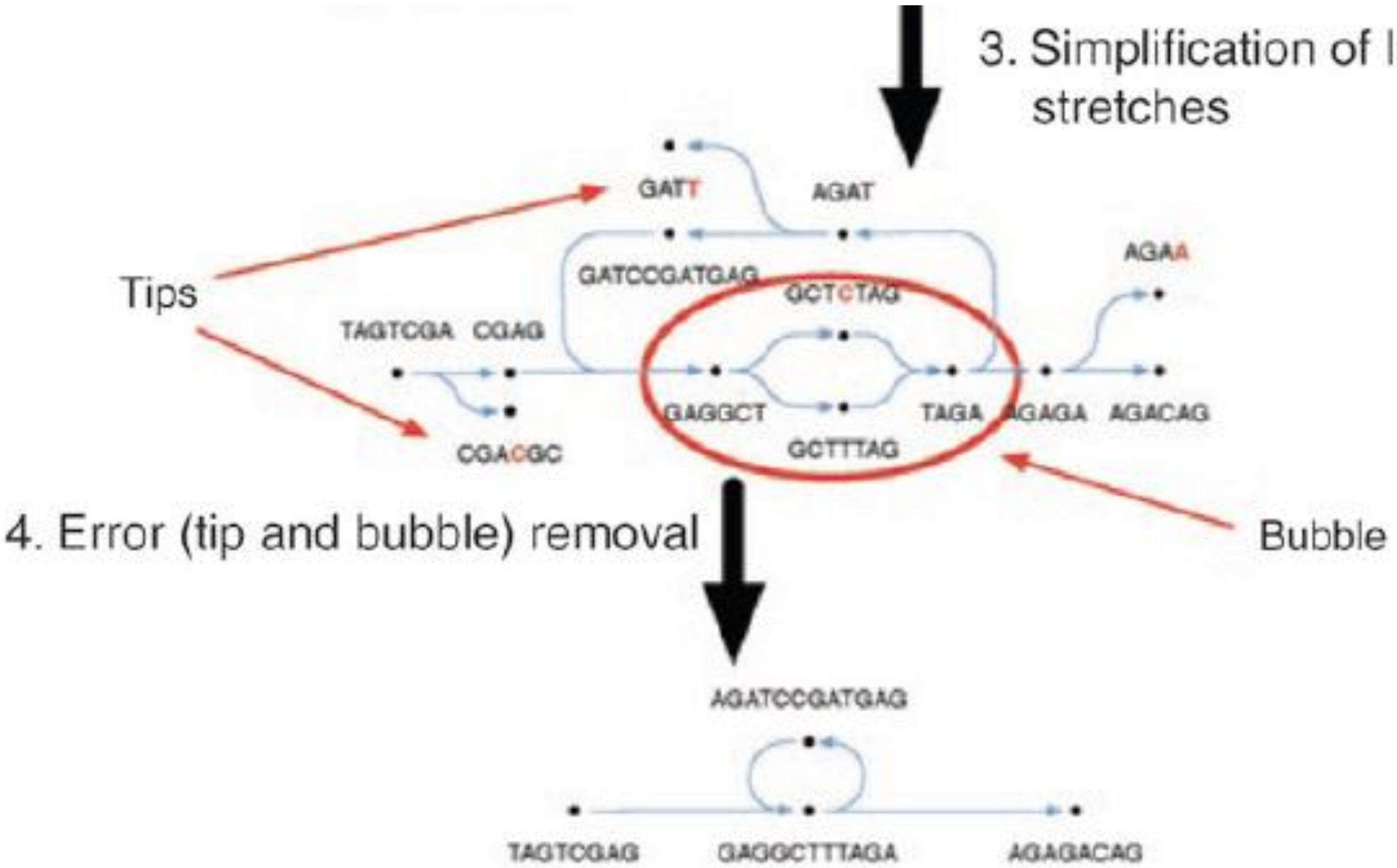


The *k*-mers in the reads (4-mers in this example) are collected into nodes and the coverage at each node is recorded (numbers at nodes)

Features

- continuous linear stretches within the graph
- Sequencing errors are low frequency tips in the graph

de-Bruijn graph assembly 3



Graph is simplified to combine nodes that are associated with the continuous linear stretches into single, larger nodes of various *k*-mer sizes

Error correction removes the tips and bubbles that result from sequencing errors

Final graph structure that accurately and completely describes in the original genome sequence