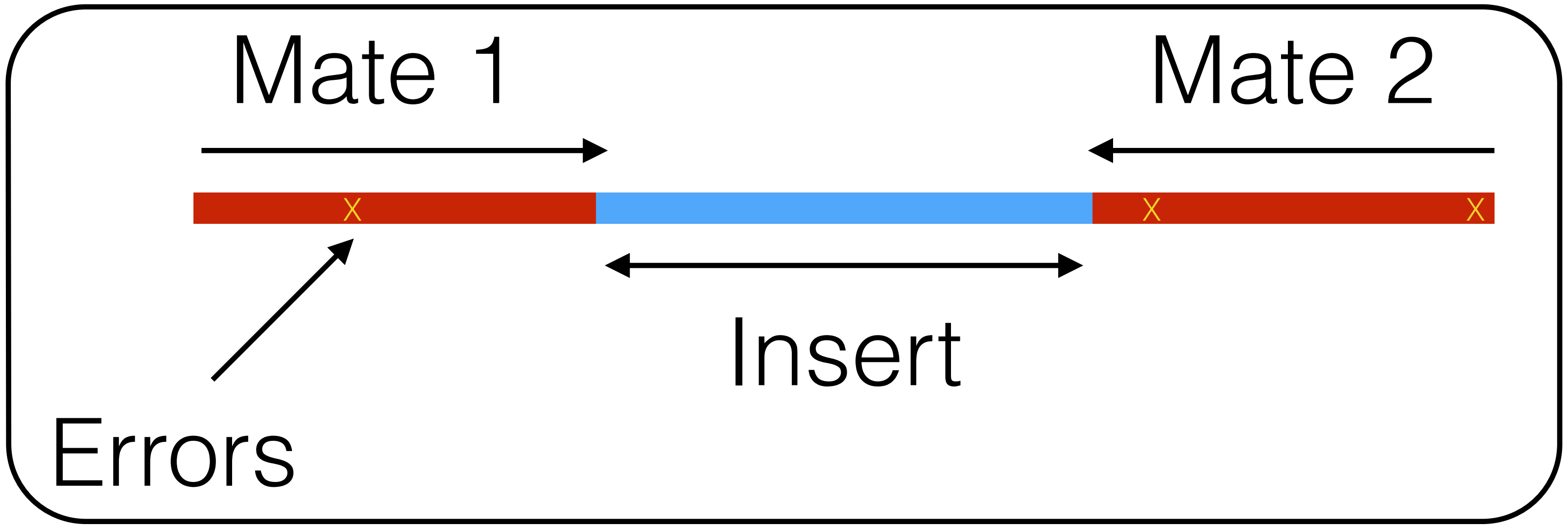
# Sequence mapping and assembly

Alistair Ward - Boston College

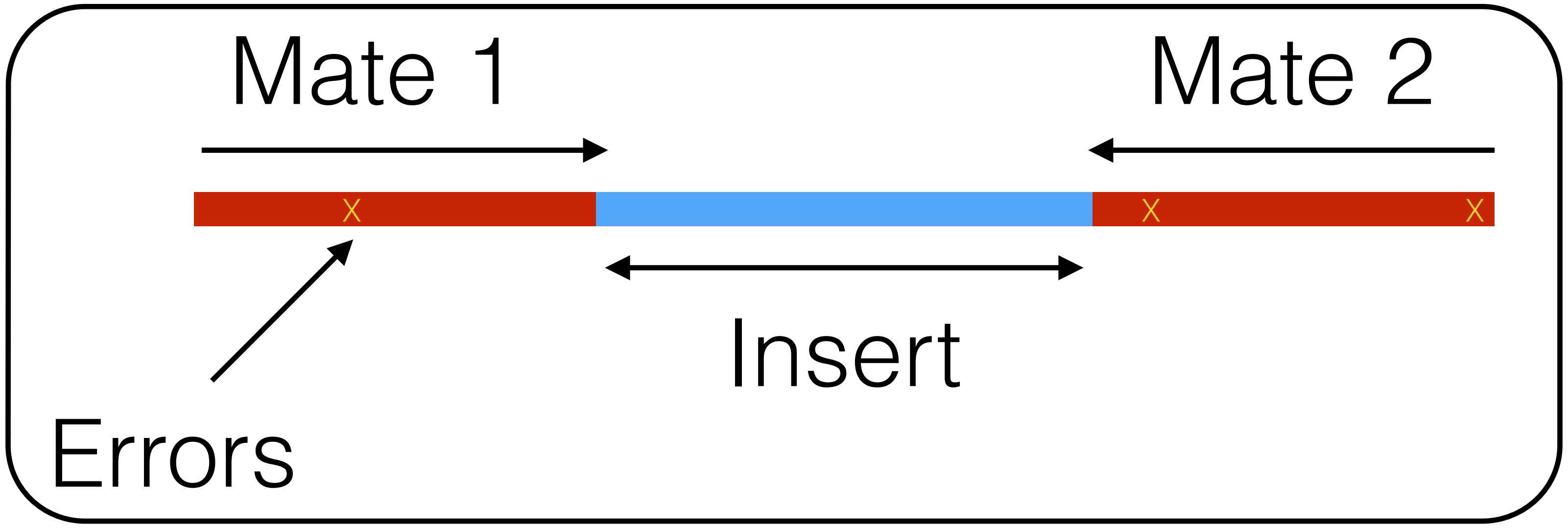
# Sequenced a genome?

- Fragmented a genome -> DNA library
- PCR amplification
- Sequence reads (ends of DNA fragment for mate pairs)
- We no longer have any positional information or relational information between fragments




We have millions/billions of sequenced DNA fragments

# Sequenced a genome?




Stored in a ***fastq*** file

```
@READ_NAME/1
GCACTGTGTGTGCTA
+
IIHIBABIIIBI@BI
```

Unique read name - /1 indicates first mate

Read sequence

Base qualities

# What we will cover

- Multiple strategies for making sense of the DNA sequences
  - Mapping to a reference (resequencing):
    - Traditional mapping (detail) — Mosaik, Bwa, Bowtie, Stampy
    - Split-read mapping — Scissors, Pindel
    - Graph alignment — glia
  - Assembly methods — Cortex, Velvet, sga

# Mapping to a reference genome

- This is like a jigsaw puzzle
- Compare reads to a reference genome, accounting for genetic differences
- Two major approaches:
  - Hashing the reference
  - Burrows-Wheeler transform

# Hash based approach

- Find all k-mers in the reference genome

- Store all positions in a hash table

# Break up reads

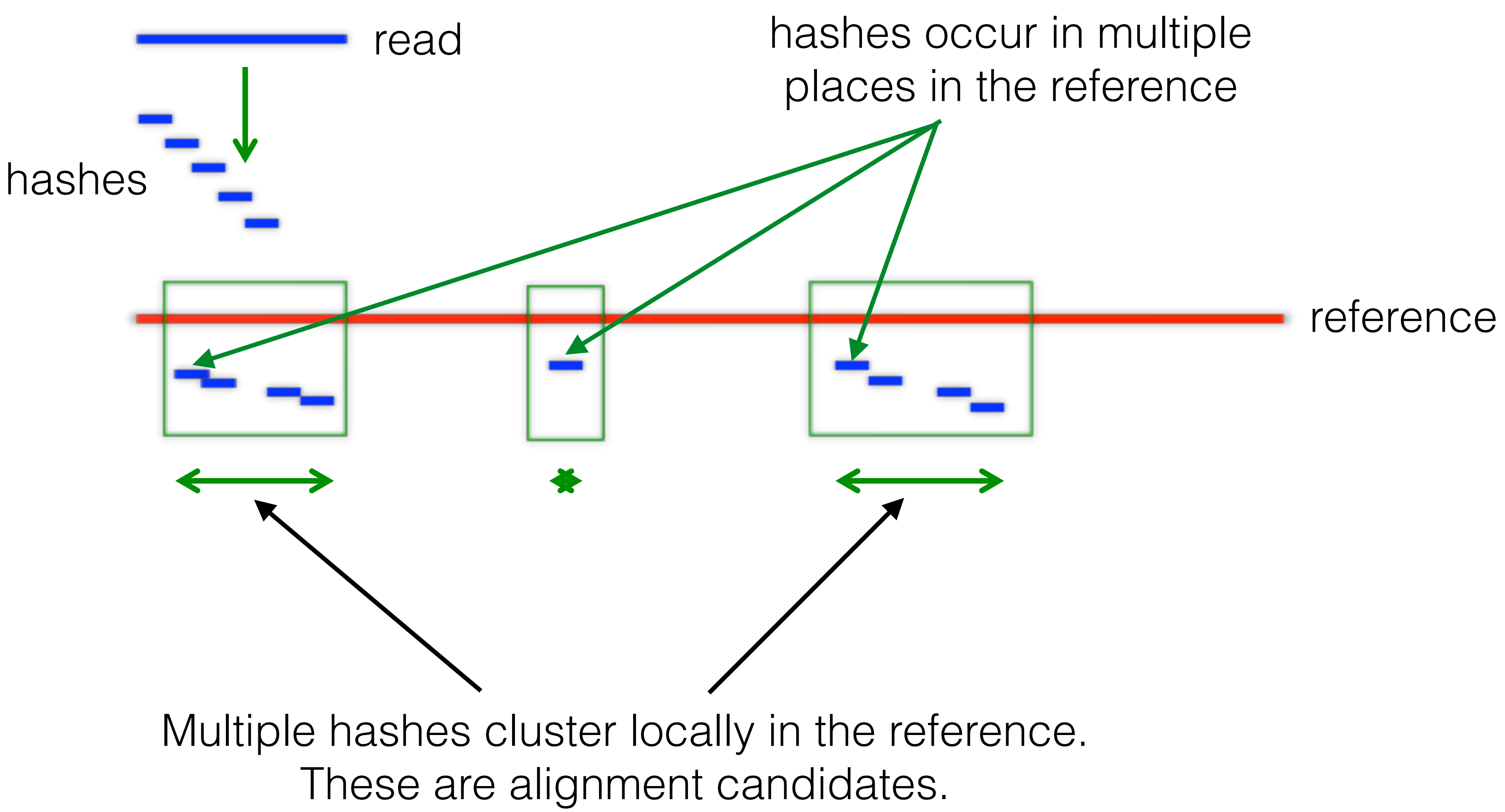
- Determine where a read can fit accounting for:
  - Sequencing errors,
  - True genetic differences with the reference
- Break read into hashes

```
ACACATGTACGTAGTCGTAGTGCTAGTCAGCT - read length n

ACACATGTACGTAGT                  - hash 1
 CACATGTACGTAGTC                 - hash 2
  ACATGTACGTAGTCG                - hash 3
                                 …
                GTAGTGCTAGTCAGC  - hash n-2
                 TAGTGCTAGTCAGCT - hash n-1
```

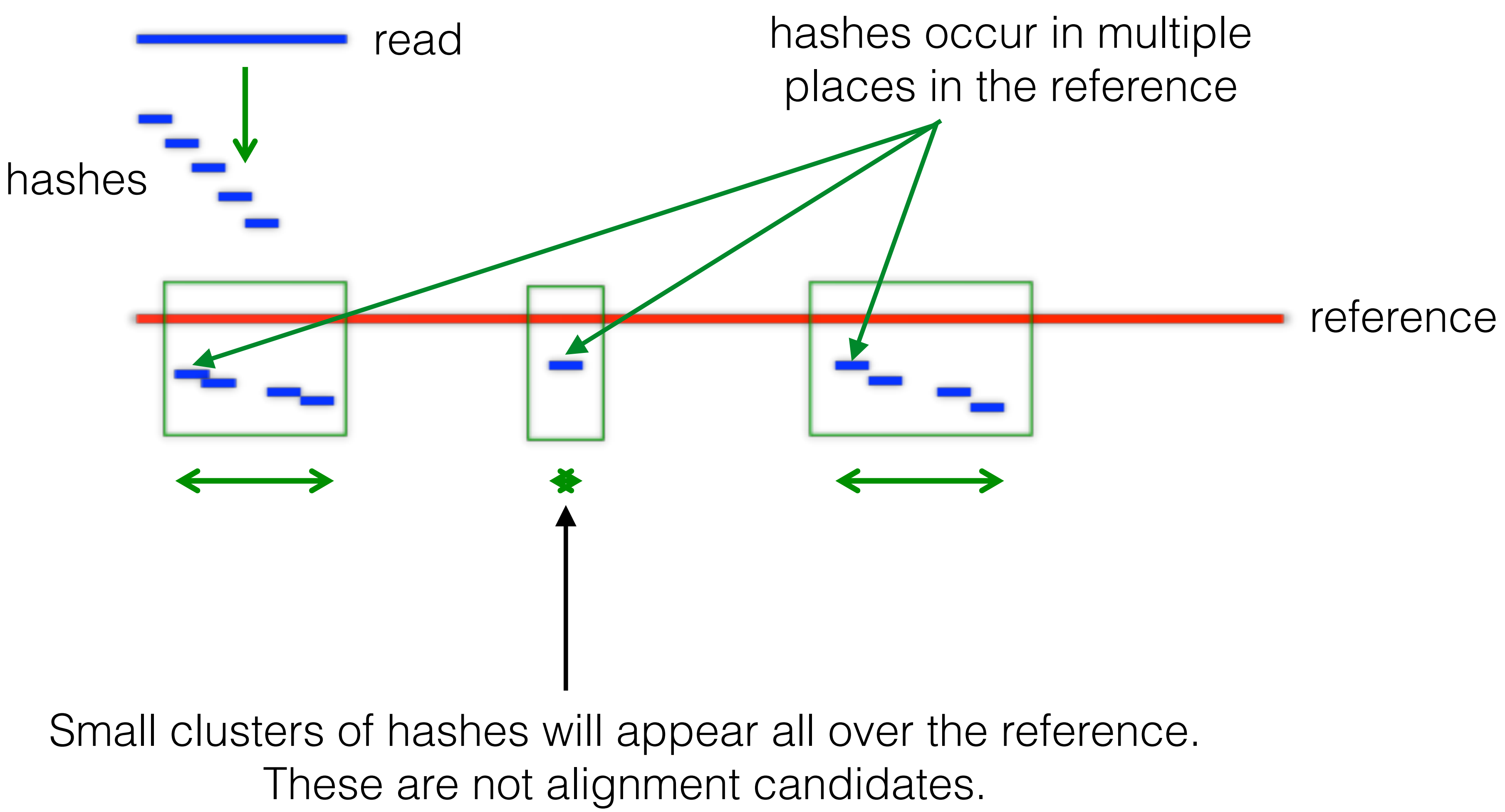
# Compare read to reference

- Find where each hash lands in the reference:



# Compare read to reference

- Find where each hash lands in the reference:



# Smith Waterman algorithm

- Find the optimal alignment for each candidate.
- Maximise similarity measure between two sequences

# Smith-Waterman example

- Generate a matrix with the sequences to compare
- Populate matrix with scores

$$M(i,0) = 0 \text{ for } 0 \leq i \leq m$$

$$M(0,j) = 0 \text{ for } 0 \leq j \leq n$$

$$M(i,j) = \max \begin{bmatrix} 0 \\ M(i-1, j-1) + s(a_i, b_j) \\ \max_{k\geq1}\{M(i-k, j) + W_k\} \\ \max_{l\geq1}\{M(i, j-l) + W_l\} \end{bmatrix}$$

| | - | A | ... | A |
|---|---|---|---|---|
| - | M(0,0) | M(1,0) | ... | M(i,0) |
| A | M(0,1) | M(1,1) | ... | M(i,1) |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| A | M(0,j) | M(1,j) | ... | M(i,j) |

# Smith-Waterman example

$M(i,0) = 0$ for $0 \leq i \leq m$

$M(0,j) = 0$ for $0 \leq j \leq n$

| | - | A | C | A | C | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | | | | | | | | |
| G | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |

# Smith-Waterman example

$$M(i,j) = \max \begin{bmatrix} 0 \\ M(i-1, j-1) + s(a_i, b_j) \\ \max_{k \geq 1}\{M(i-k, j) + W_k\} \\ \max_{l \geq 1}\{M(i, j-l) + W_l\} \end{bmatrix}$$

$$M(i-1, j-1) + s(a_i, b_j)$$

$s(a_i, b_j) = +2$ if $a = b$ Match

$s(a_i, b_j) = -1$ if $a \neq b$ Mismatch

$$M(1, 1) = +2$$

| | - | A | C | A | C | A |
|---|---|---|---|---|---|---|
| - | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | M(1,1) | | | | |
| G | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |

# Smith-Waterman example

$$M(i,j) = \max \begin{bmatrix} 0 \\ M(i-1, j-1) + s(a_i, b_j) \\ \max_{k \geq 1}\{M(i-k, j) + W_k\} \\ \max_{l \geq 1}\{M(i, j-l) + W_l\} \end{bmatrix}$$

$$M(i-1, j-1) + s(a_i, b_j)$$

$s(a_i, b_j) = +2$ if $a = b$ Match

$s(a_i, b_j) = -1$ if $a \neq b$ Mismatch

$M(1, 1) = +2$

| | - | A | C | A | C | A |
|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** | 0 | 2 | | | | |
| **G** | 0 | | | | | |
| **C** | 0 | | | | | |
| **A** | 0 | | | | | |
| **C** | 0 | | | | | |
| **A** | 0 | | | | | |
| **C** | 0 | | | | | |
| **A** | 0 | | | | | |

# Smith-Waterman example

$$M(i,j) = \max \begin{bmatrix} 0 \\ M(i-1, j-1) + s(a_i, b_j) \\ \max_{k \geq 1}\{M(i-k, j) + W_k\} \\ \max_{l \geq 1}\{M(i, j-l) + W_l\} \end{bmatrix}$$

Insertion or deletion scoring

$$W_i = -1$$

| | - | A | C | A | C | A |
|---|---|---|---|---|---|---|
| - | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 2 | M(2,1) | | | |
| G | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |
| C | 0 | | | | | |
| A | 0 | | | | | |

# Smith-Waterman example

| | - | A | C | A | C | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** | 0 | 2 | 1 | | | | | | |
| **G** | 0 | | | | | | | | |
| **C** | 0 | | | | | | | | |
| **A** | 0 | | | | | | | | |
| **C** | 0 | | | | | | | | |
| **A** | 0 | | | | | | | | |
| **C** | 0 | | | | | | | | |
| **A** | 0 | | | | | | | | |

# Smith-Waterman example

| | - | A | C | A | C | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 2 | 1 | 2 | | | | | |
| G | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |

# Smith-Waterman example

| | - | A | C | A | C | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 2 | 1 | 2 | 1 | 2 | 1 | 0 | 2 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| C | 0 | 0 | 3 | 2 | 3 | 2 | 3 | 2 | 1 |
| A | 0 | 2 | 2 | 5 | 4 | 5 | 4 | 3 | 4 |
| C | 0 | 1 | 4 | 4 | 7 | 6 | 7 | 6 | 5 |
| A | 0 | 2 | 3 | 6 | 6 | 9 | 8 | 7 | 8 |
| C | 0 | 1 | 4 | 5 | 8 | 8 | 11 | 10 | 9 |
| A | 0 | 2 | 3 | 6 | 7 | 10 | 10 | 10 | 12 |

# Traceback

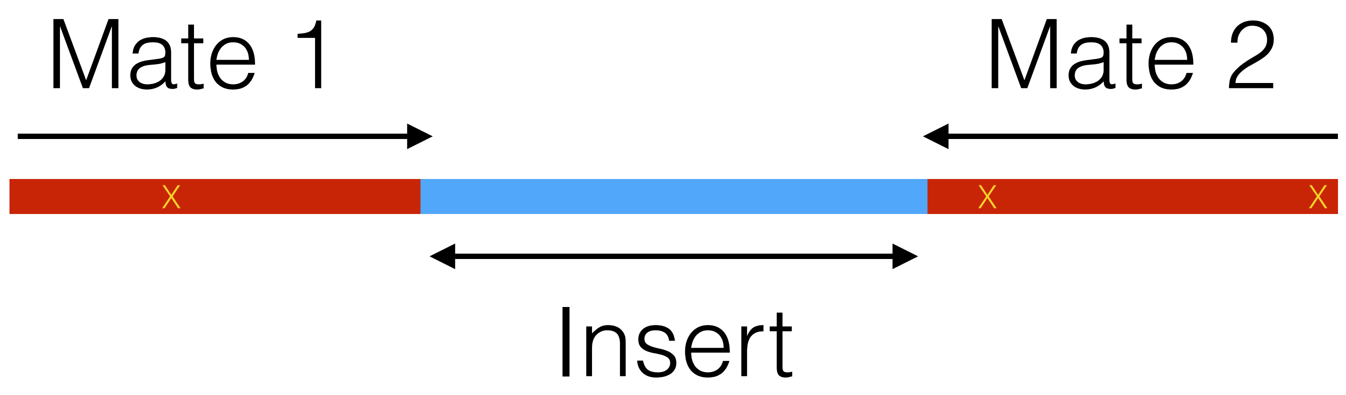
- Start at highest value
- Diagonal line is a match/mismatch
- Up/down or left/right are indels

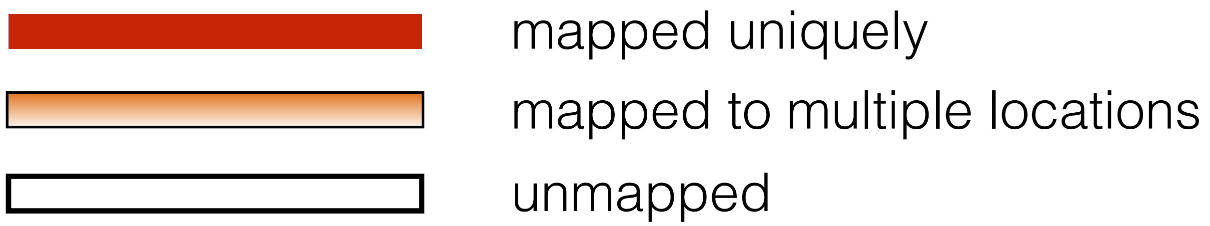
Sequence 1
A-CACACTA

Sequence 2
AGCACAC-A

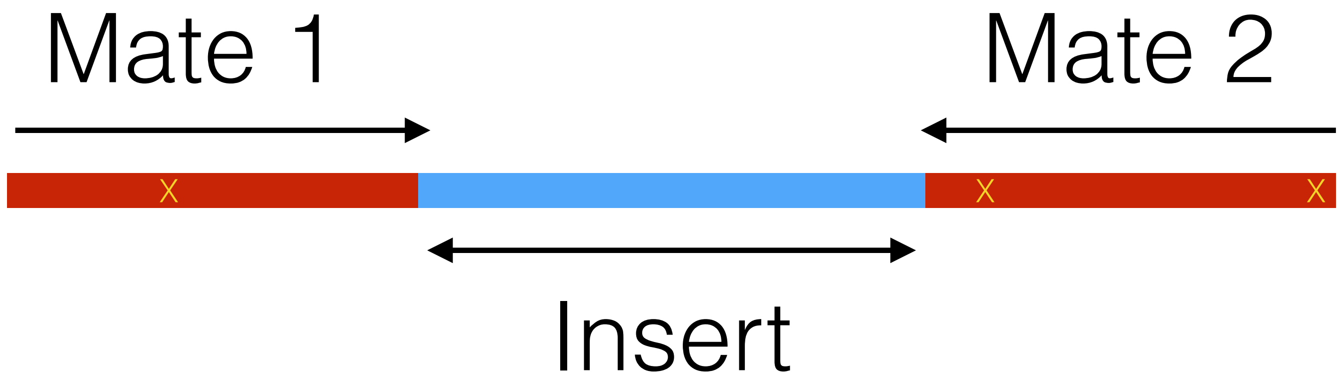
| | - | A | C | A | C | A | C | T | A |
|---|---|---|---|---|---|---|---|---|---|
| **-** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** | 0 | 2 | 1 | 2 | 1 | 2 | 1 | 0 | 2 |
| **G** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| **C** | 0 | 0 | 3 | 2 | 3 | 2 | 3 | 2 | 1 |
| **A** | 0 | 2 | 2 | 5 | 4 | 5 | 4 | 3 | 4 |
| **C** | 0 | 1 | 4 | 4 | 7 | 6 | 7 | 6 | 5 |
| **A** | 0 | 2 | 3 | 6 | 6 | 9 | 8 | 7 | 8 |
| **C** | 0 | 1 | 4 | 5 | 8 | 8 | 11 | 10 | 9 |
| **A** | 0 | 2 | 3 | 6 | 7 | 10 | 10 | 10 | 12 |

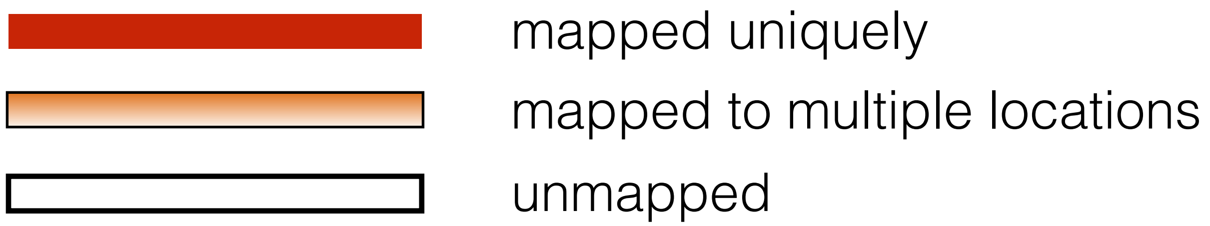
# Paired end reads



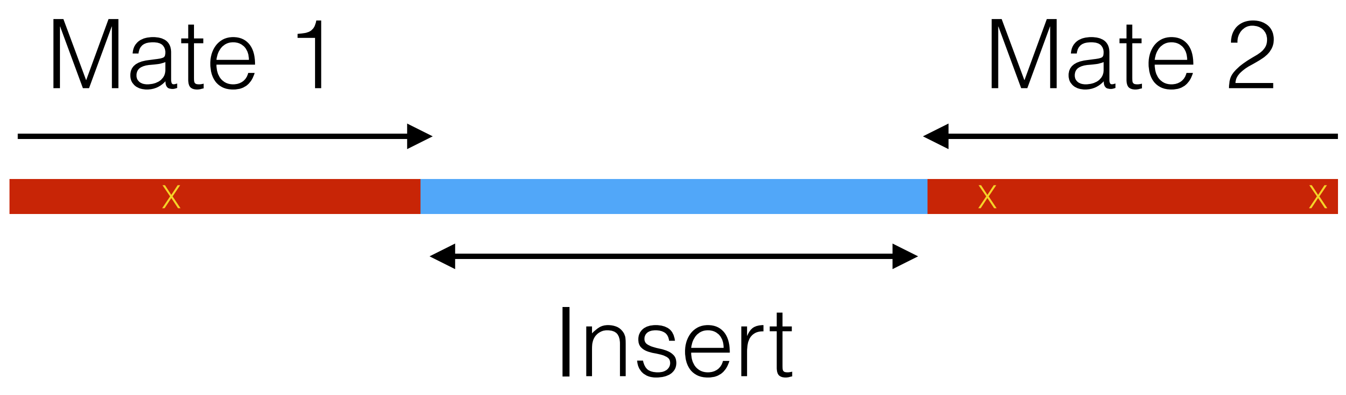
# Paired end reads

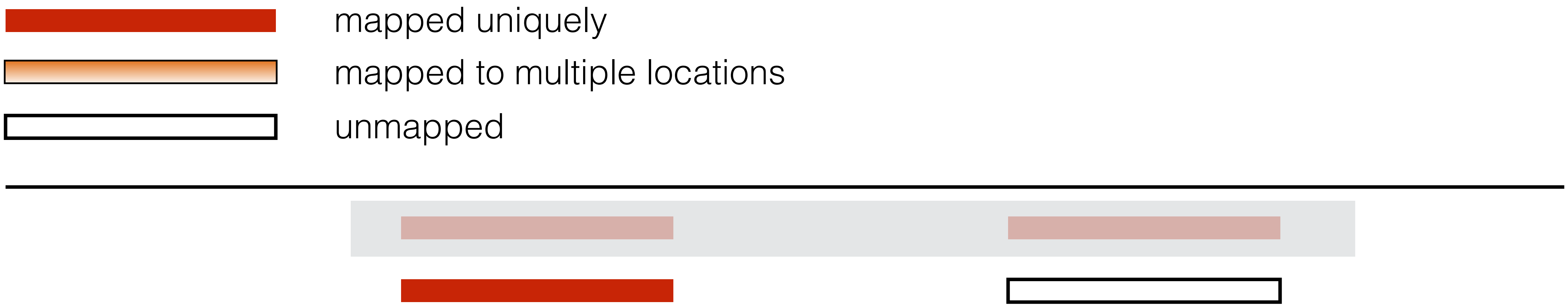


Both mates map uniquely

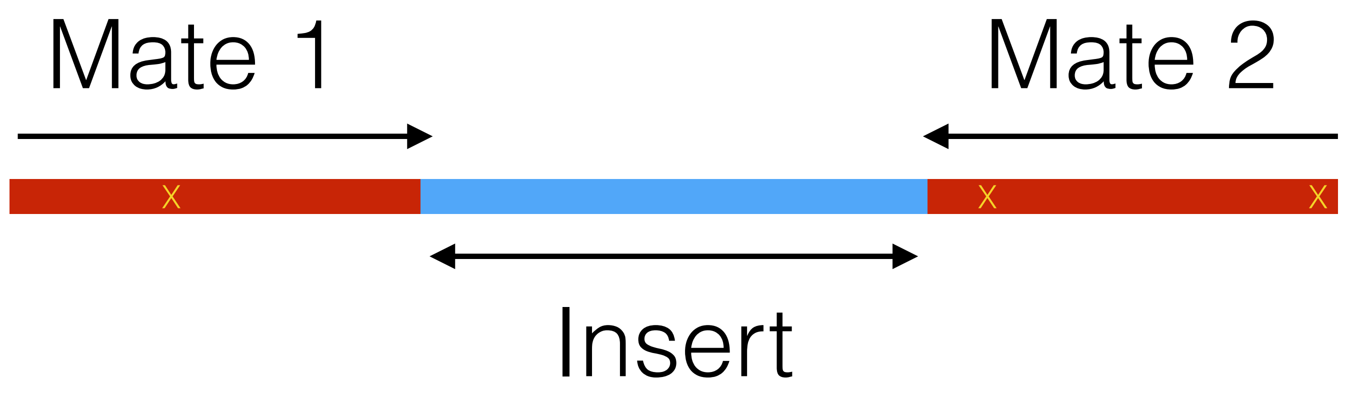
# Paired end reads

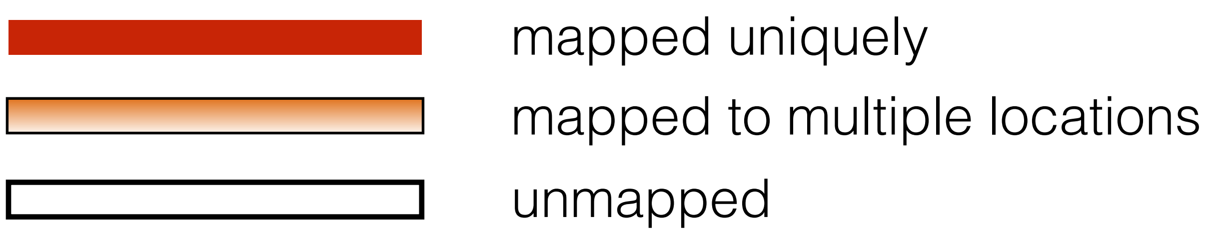


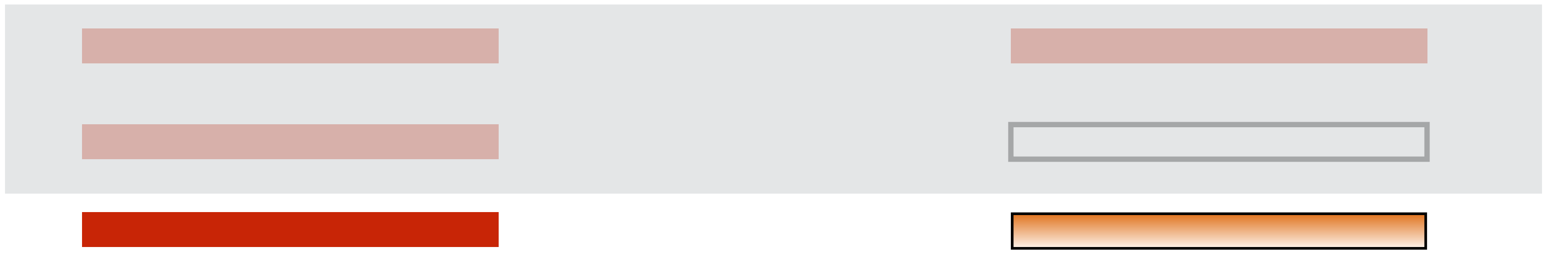




One mate maps uniquely, the other is unmapped

# Paired end reads

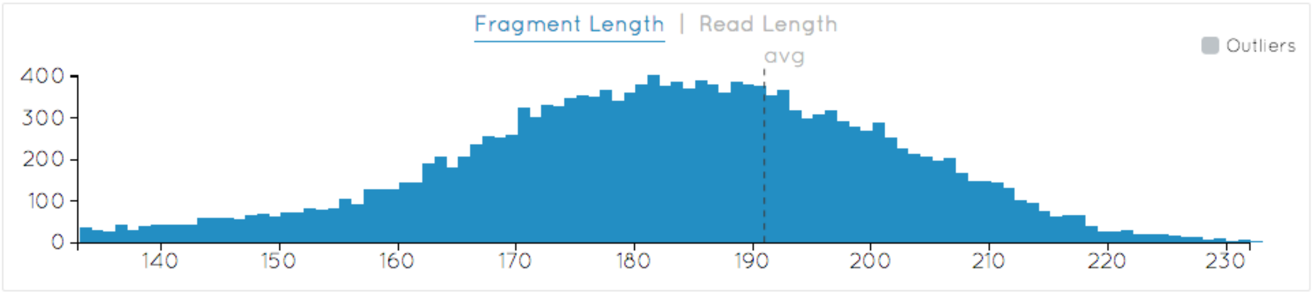


One mate maps uniquely, the other maps to multiple locations

Use fragment length distribution to determine most likely location




bam.iobio.io

# Alignment output

The result of most modern aligners is a BAM file, the binary form of SAM (Sequence Alignment/Map) file

Header section

@HD - Header line

SO = sort order

Can take the values:

- unknown
- unsorted
- coordinate
- queryname

```
@HD	VN:1.0	SO:coordinate
@SQ	SN:1	LN:249250621	M5:1b22b98cdeb4a9304cb5d48026a85128	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:2	LN:243199373	M5:a0d9851da00400dec1098a9255ac712e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:3	LN:198022430	M5:fdfd811849cc2fadebc929bb925902e5	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:4	LN:191154276	M5:23dccd106897542ad87d2765d28a19a1	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:5	LN:180915260	M5:0740173db9ffd264d728f32784845cd7	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:6	LN:171115067	M5:1d3a93a248d92a729ee764823acbbc6b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:7	LN:159138663	M5:618366e953d6aaad97dbe4777c29375e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:8	LN:146364022	M5:96f514a9929e410c6651697bded59aec	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:9	LN:141213431	M5:3e273117f15e0a400f01055d9f393768	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:10	LN:135534747	M5:988c28e000e84c26d552359af1ea2e1d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:11	LN:135006516	M5:98c59049a2df285c76ffb1c6db8f8b96	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:12	LN:133851895	M5:51851ac0e1a115847ad36449b0015864	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:13	LN:115169878	M5:283f8d7892baa81b510a015719ca7b0b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:14	LN:107349540	M5:98f3cae32b2a2e9524bc19813927542e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:15	LN:102531392	M5:e5645a794a8238215b2cd77acb95a078	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:16	LN:90354753	M5:fc9b1a7b42b97a864f56b348b06095e6	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:17	LN:81195210	M5:351f64d4f4f9ddd45b35336ad97aa6de	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:18	LN:78077248	M5:b15d4b2d29dde9d3e4f93d1d0f2cbc9c	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:19	LN:59128983	M5:1aacd71f30db8e561810913e0b72636d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:20	LN:63025520	M5:0dec9660ec1efaaf33281c0d5ea2560f	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:21	LN:48129895	M5:2979a6085bfe28e3ad6f552f361ed74d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:22	LN:51304566	M5:a718acaa6135fdca8357d5bfe94211dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:X	LN:155270560	M5:7e0e2e580297b7764e31dbc80c2540dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:Y	LN:59373566	M5:1fa3474750af0948bdf97d5a0ee52e51	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:MT	LN:16569	M5:c68f52674c9fb33aef52dcf399755519	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@RG	ID:SRR062634	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062635	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062641	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@PG	ID:bwa_index	PN:bwa	CL:bwa index -a bwtsw $reference_fasta	VN:0.5.9-r16
@PG	ID:bwa_aln_fastq	PN:bwa	CL:bwa aln -q 15 -f $sai_file $reference_fasta $fastq_file	PP:bwa_index	VN:0.5
@PG	ID:bwa_sam	PN:bwa	CL:bwa sampe -a 618 -r $rg_line -f $sam_file $reference_fasta $sai_file(s) $fastq_file(s)
@PG	ID:sam_to_fixed_bam	PN:samtools	CL:samtools view -bSu $sam_file | samtools sort -n -o - samtools_nsort_tmp | sa
@PG	ID:gatk_target_interval_creator PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T RealignerTarget
@PG	ID:bam_realignment_around_known_indels	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T IndelRea
@PG	ID:bam_count_covariates PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T CountCovariates -R $refe
@PG	ID:bam_recalibrate_quality_scores	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T TableRec
@PG	ID:bam_calculate_bq	PN:samtools	CL:samtools calmd -Erb $bam_file $reference_fasta > $bq_bam_file	PP:bam_
@PG	ID:bam_merge	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
@PG	ID:bam_mark_duplicates	PN:picard	CL:java $jvm_args -jar MarkDuplicates.jar INPUT=$bam_file OUTPUT=$markdup_bam_
@PG	ID:bam_merge.1	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
```

# Alignment output

The result of most modern aligners is a BAM file, the binary form of SAM (Sequence Alignment/Map) file

Header section

@SQ - Reference sequences

```
@HD	VN:1.0	SO:coordinate
@SQ	SN:1	LN:249250621	M5:1b22b98cdeb4a9304cb5d48026a85128	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:2	LN:243199373	M5:a0d9851da00400dec1098a9255ac712e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:3	LN:198022430	M5:fdfd811849cc2fadebc929bb925902e5	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:4	LN:191154276	M5:23dccd106897542ad87d2765d28a19a1	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:5	LN:180915260	M5:0740173db9ffd264d728f32784845cd7	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:6	LN:171115067	M5:1d3a93a248d92a729ee764823acbbc6b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:7	LN:159138663	M5:618366e953d6aaad97dbe4777c29375e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:8	LN:146364022	M5:96f514a9929e410c6651697bded59aec	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:9	LN:141213431	M5:3e273117f15e0a400f01055d9f393768	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:10	LN:135534747	M5:988c28e000e84c26d552359af1ea2e1d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:11	LN:135006516	M5:98c59049a2df285c76ffb1c6db8f8b96	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:12	LN:133851895	M5:51851ac0e1a115847ad36449b0015864	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:13	LN:115169878	M5:283f8d7892baa81b510a015719ca7b0b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:14	LN:107349540	M5:98f3cae32b2a2e9524bc19813927542e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:15	LN:102531392	M5:e5645a794a8238215b2cd77acb95a078	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:16	LN:90354753	M5:fc9b1a7b42b97a864f56b348b06095e6	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:17	LN:81195210	M5:351f64d4f4f9ddd45b35336ad97aa6de	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:18	LN:78077248	M5:b15d4b2d29dde9d3e4f93d1d0f2cbc9c	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:19	LN:59128983	M5:1aacd71f30db8e561810913e0b72636d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:20	LN:63025520	M5:0dec9660ec1efaaf33281c0d5ea2560f	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:21	LN:48129895	M5:2979a6085bfe28e3ad6f552f361ed74d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:22	LN:51304566	M5:a718acaa6135fdca8357d5bfe94211dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:X	LN:155270560	M5:7e0e2e580297b7764e31dbc80c2540dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:Y	LN:59373566	M5:1fa3474750af0948bdf97d5a0ee52e51	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:MT	LN:16569	M5:c68f52674c9fb33aef52dcf399755519	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@RG	ID:SRR062634	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062635	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062641	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@PG	ID:bwa_index	PN:bwa	CL:bwa index -a bwtsw $reference_fasta	VN:0.5.9-r16
@PG	ID:bwa_aln_fastq	PN:bwa	CL:bwa aln -q 15 -f $sai_file $reference_fasta $fastq_file	PP:bwa_index	VN:0.5
@PG	ID:bwa_sam	PN:bwa	CL:bwa sampe -a 618 -r $rg_line -f $sam_file $reference_fasta $sai_file(s) $fastq_file(s)
@PG	ID:sam_to_fixed_bam	PN:samtools	CL:samtools view -bSu $sam_file | samtools sort -n -o - samtools_nsort_tmp | sa
@PG	ID:gatk_target_interval_creator PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T RealignerTarget
@PG	ID:bam_realignment_around_known_indels	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T IndelRea
@PG	ID:bam_count_covariates PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T CountCovariates -R $refe
@PG	ID:bam_recalibrate_quality_scores	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T TableRec
@PG	ID:bam_calculate_bq	PN:samtools	CL:samtools calmd -Erb $bam_file $reference_fasta > $bq_bam_file	PP:bam_
@PG	ID:bam_merge	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
@PG	ID:bam_mark_duplicates	PN:picard	CL:java $jvm_args -jar MarkDuplicates.jar INPUT=$bam_file OUTPUT=$markdup_bam_
@PG	ID:bam_merge.1	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
```

# Alignment output

The result of most modern aligners is a BAM file, the binary form of SAM (Sequence Alignment/Map) file

Header section

@RG - Read groups

```
@HD	VN:1.0	SO:coordinate
@SQ	SN:1	LN:249250621	M5:1b22b98cdeb4a9304cb5d48026a85128	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:2	LN:243199373	M5:a0d9851da00400dec1098a9255ac712e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:3	LN:198022430	M5:fdfd811849cc2fadebc929bb925902e5	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:4	LN:191154276	M5:23dccd106897542ad87d2765d28a19a1	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:5	LN:180915260	M5:0740173db9ffd264d728f32784845cd7	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:6	LN:171115067	M5:1d3a93a248d92a729ee764823acbbc6b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:7	LN:159138663	M5:618366e953d6aaad97dbe4777c29375e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:8	LN:146364022	M5:96f514a9929e410c6651697bded59aec	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:9	LN:141213431	M5:3e273117f15e0a400f01055d9f393768	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:10	LN:135534747	M5:988c28e000e84c26d552359af1ea2e1d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:11	LN:135006516	M5:98c59049a2df285c76ffb1c6db8f8b96	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:12	LN:133851895	M5:51851ac0e1a115847ad36449b0015864	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:13	LN:115169878	M5:283f8d7892baa81b510a015719ca7b0b	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:14	LN:107349540	M5:98f3cae32b2a2e9524bc19813927542e	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:15	LN:102531392	M5:e5645a794a8238215b2cd77acb95a078	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:16	LN:90354753	M5:fc9b1a7b42b97a864f56b348b06095e6	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:17	LN:81195210	M5:351f64d4f4f9ddd45b35336ad97aa6de	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:18	LN:78077248	M5:b15d4b2d29dde9d3e4f93d1d0f2cbc9c	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:19	LN:59128983	M5:1aacd71f30db8e561810913e0b72636d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:20	LN:63025520	M5:0dec9660ec1efaaf33281c0d5ea2560f	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:21	LN:48129895	M5:2979a6085bfe28e3ad6f552f361ed74d	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:22	LN:51304566	M5:a718acaa6135fdca8357d5bfe94211dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:X	LN:155270560	M5:7e0e2e580297b7764e31dbc80c2540dd	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:Y	LN:59373566	M5:1fa3474750af0948bdf97d5a0ee52e51	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@SQ	SN:MT	LN:16569	M5:c68f52674c9fb33aef52dcf399755519	UR:ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/
@RG	ID:SRR062634	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062635	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@RG	ID:SRR062641	CN:WUGSC	DS:SRP001294	LB:2845856850	PI:206	PL:ILLUMINA	SM:HG00096
@PG	ID:bwa_index	PN:bwa	CL:bwa index -a bwtsw $reference_fasta	VN:0.5.9-r16
@PG	ID:bwa_aln_fastq	PN:bwa	CL:bwa aln -q 15 -f $sai_file $reference_fasta $fastq_file	PP:bwa_index	VN:0.5
@PG	ID:bwa_sam	PN:bwa	CL:bwa sampe -a 618 -r $rg_line -f $sam_file $reference_fasta $sai_file(s) $fastq_file(s)
@PG	ID:sam_to_fixed_bam	PN:samtools	CL:samtools view -bSu $sam_file | samtools sort -n -o - samtools_nsort_tmp | sa
@PG	ID:gatk_target_interval_creator PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T RealignerTarget
@PG	ID:bam_realignment_around_known_indels	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T IndelRea
@PG	ID:bam_count_covariates PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T CountCovariates -R $refe
@PG	ID:bam_recalibrate_quality_scores	PN:GenomeAnalysisTK	CL:java $jvm_args -jar GenomeAnalysisTK.jar -T TableRec
@PG	ID:bam_calculate_bq	PN:samtools	CL:samtools calmd -Erb $bam_file $reference_fasta > $bq_bam_file	PP:bam_
@PG	ID:bam_merge	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
@PG	ID:bam_mark_duplicates	PN:picard	CL:java $jvm_args -jar MarkDuplicates.jar INPUT=$bam_file OUTPUT=$markdup_bam_
@PG	ID:bam_merge.1	PN:picard	CL:java $jvm_args -jar MergeSamFiles.jar INPUT=$bam_file(s) OUTPUT=$merged_bam VALIDAT
```

# Alignment output

The result of most modern aligners is a BAM file, the binary form of SAM (Sequence Alignment/Map) file

```
SRR062634.9882510	163	1	10001	15	13M1D30M57S	=	10069	166	TAACCCTAACCCTACCCTAACCCTAACCCTAACCCTAACCCTATCCCCACCCTAACCCCAACCCT
SRR062641.21956756	163	1	10001	9	11S36M1I13M39S	=	10051	120	TAACCCTACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTTAACCCTAACCCTAACC
SRR062641.13613107	99	1	10002	0	8M1I6M1I6M1I42M1I9M25S	=	10110	162	AACCCTAACCCCTAACCCCTAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062641.13792993	99	1	10002	1	56M44S	=	10058	113	AACCCTAACCCTAACCCTAACCCGAACCCGAACCCTAACCCGAACCCGAACCCGAACCCGAACCCTAACCCGA
SRR062634.14718531	99	1	10003	15	2S53M45S	=	10188	250	CTACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTCAAC
SRR062634.10759422	163	1	10004	10	1S39M1I9M1I9M40S	=	10099	166	CCCTTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTTAACCCTAACCCCTAA
SRR062635.17851444	163	1	10004	10	4M1I41M54S	=	10116	194	CCCTAAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTCATCCTAACCCTAACCC
SRR062641.13809271	163	1	10004	10	2S48M50S	=	10052	119	ACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCAACCCCAAACC
SRR062634.20009599	161	1	10005	0	80M20S	7	10040	0	CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062641.19689524	163	1	10005	0	95M5S	=	10040	112	CCTAACCCTAACCCTACCCCTAACCCTACCCCTAACCCTACCCCTAACCCTACCCCTAACCCTAACCCTAACC
SRR062635.23422300	163	1	10006	0	96M4S	=	10069	142	CTAACTCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACTCTAACCCTAACCCTAACCCTAACCC
SRR062635.24326466	65	1	10007	0	79M21S	=	249240050	249230043	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062641.8816787	99	1	10013	22	15M1I40M44S	=	10200	266	TAACCCTAACCCTAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC
SRR062641.17764198	99	1	10014	15	100M	=	10140	203	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTTACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062634.18483372	99	1	10015	0	86M14S	=	10040	105	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062641.23619492	163	1	10015	0	85M15S	=	10087	126	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062635.23900153	16	1	10016	0	39S61M	*	0	0	TCACCCTACCCCTCACCCTAACCCCACCCCTAACCCTACCCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062635.22825119	99	1	10017	0	90M10S	=	10180	208	CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062641.14990562	163	1	10017	9	100M	=	10117	199	TCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062634.1462817	1123	1	10019	0	85M15S	=	10183	214	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTACGCCT
SRR062635.1802246	99	1	10019	0	85M15S	=	10183	214	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062635.3717195	99	1	10020	0	86M14S	=	10178	217	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062635.7942302	163	1	10020	0	86M14S	=	10180	209	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062634.379459	163	1	10021	9	81M19S	=	10052	102	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062634.15015146	99	1	10023	29	100M	=	10183	253	CCTAACCCTAACCCTAACCCTAACCCTAACCCTTAACCTTACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062635.20402363	99	1	10023	0	75M25S	=	10118	150	CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062635.23537339	99	1	10023	0	86M14S	=	10183	197	CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062641.23895847	163	1	10024	1	85M15S	=	10159	202	CTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCC
SRR062634.23408959	99	1	10025	0	81M19S	=	10051	119	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062634.3270083	99	1	10025	0	81M19S	=	10178	187	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062634.9822035	99	1	10025	0	81M19S	=	10179	194	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062634.13032701	163	1	10026	0	76M24S	=	10064	111	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062634.287737	163	1	10027	2	75M25S	=	10061	109	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062634.6456728	99	1	10027	0	80M20S	=	10183	196	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062634.24424096	99	1	10029	0	79M21S	=	10178	201	TCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
SRR062634.22877871	99	1	10030	0	78M22S	=	10178	195	CTAACCCTAACCCAAACCCTAACCCTAACCCTAACCCTAACCCTAACACTAACCCTAACCCTAACCCTAACCA
SRR062641.10204190	99	1	10030	0	76M24S	=	10165	192	CTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCC
SRR062641.19443706	99	1	10030	2	76M24S	=	10201	250	CTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCC
SRR062634.245607	163	1	10031	0	78M22S	=	10178	190	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062635.15443355	99	1	10031	0	77M23S	=	10178	198	TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062635.24207489	99	1	10032	0	72M28S	=	10178	192	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062641.14544497	99	1	10032	0	70M30S	=	10148	173	AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
SRR062635.13859433	129	1	10033	0	74M26S	19	59118895	0	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT
SRR062641.20641826	99	1	10033	0	76M24S	=	10182	188	ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA
SRR062634.4100543	163	1	10034	9	66M34S	=	10069	104	CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCAAAC
SRR062634.9699295	1187	1	10034	0	70M30S	=	10182	199	CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC
SRR062635.18256445	163	1	10034	0	75M25S	=	10183	199	CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC
SRR062641.2061	163	1	10034	0	74M26S	=	10182	198	CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTACCCC
SRR062634.19100891	163	1	10035	0	64M36S	=	10177	184	CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCAAACCCTAACCCTAACCCAAACT
SRR062634.20493097	99	1	10035	3	80M20S	=	10117	144	CTTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
```

# Mapping quality

An important quantity attached to each mapped read:

The probability that a read in **incorrectly** placed

$$Q = -\log_{10} P$$

Q is the Phred score

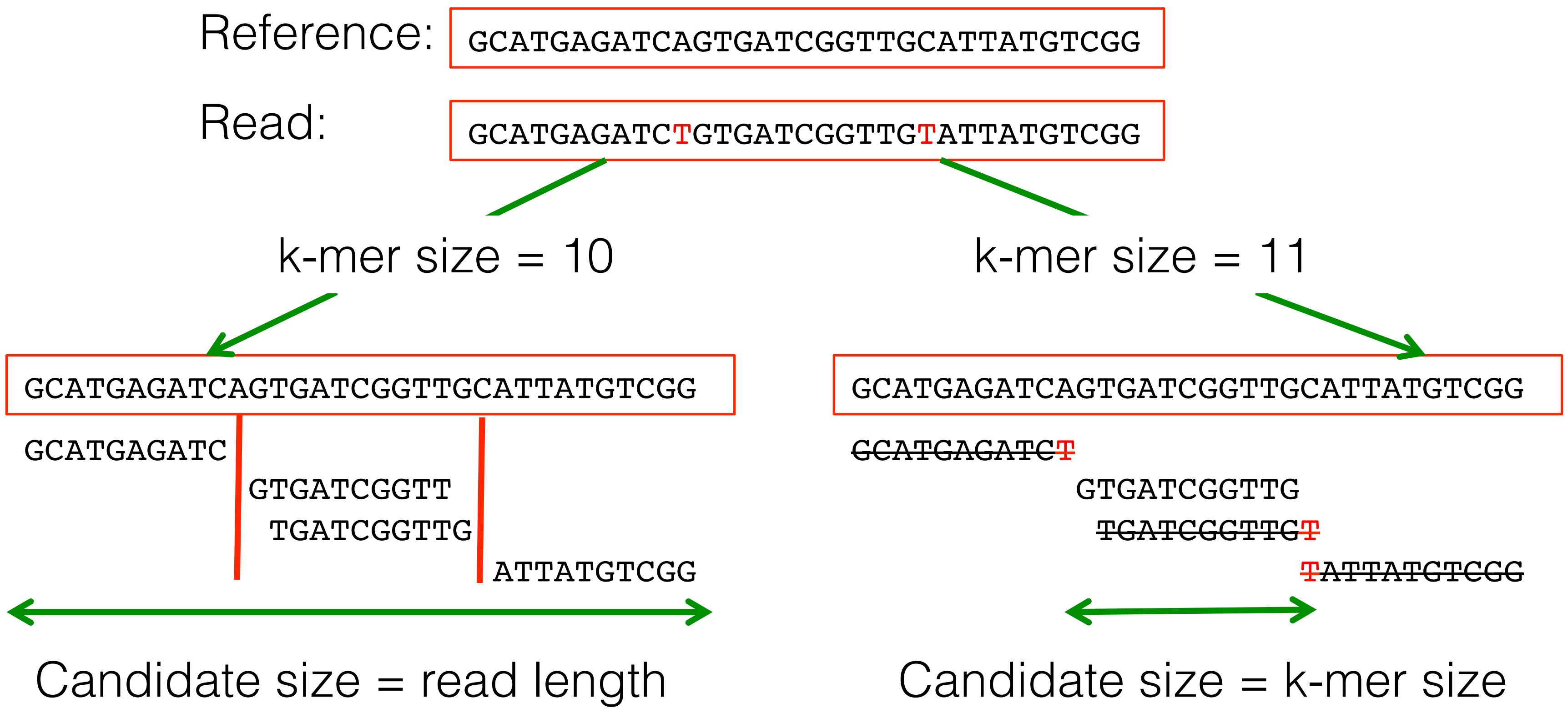
Q = 30 means there is a 1 in 1000 chance
that the read is misaligned

# Parameters

- Can you just use an aligner out of the box?
- Yes, but it is wise to understand what parameters are doing
- What are you looking for?

# Parameters - k-mer size

Short reads - choice of k-mer size is important



# Burrows-Wheeler

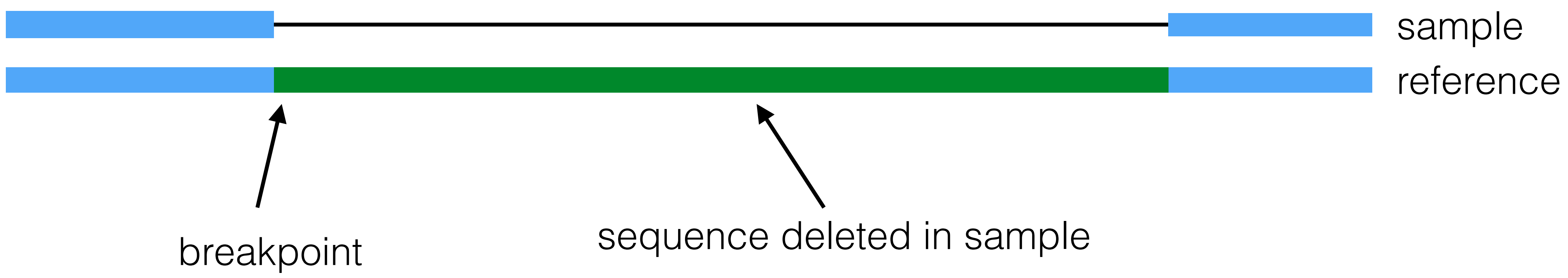
- Align the query sequence against the suffix tree of the reference
- Represent the suffix tree with an FM-index using the Burrows-Wheeler transform
  - Reduces the memory footprint

# Mapping pros and cons

- Vast majority of sample sequence can be accurately placed
- Problems with:
    - Large scale differences - structural variation
    - Reference bias
    - Repetitive DNA
- How can we address these shortcomings?

# Mapping across a deletion

- A read straddles a deletion



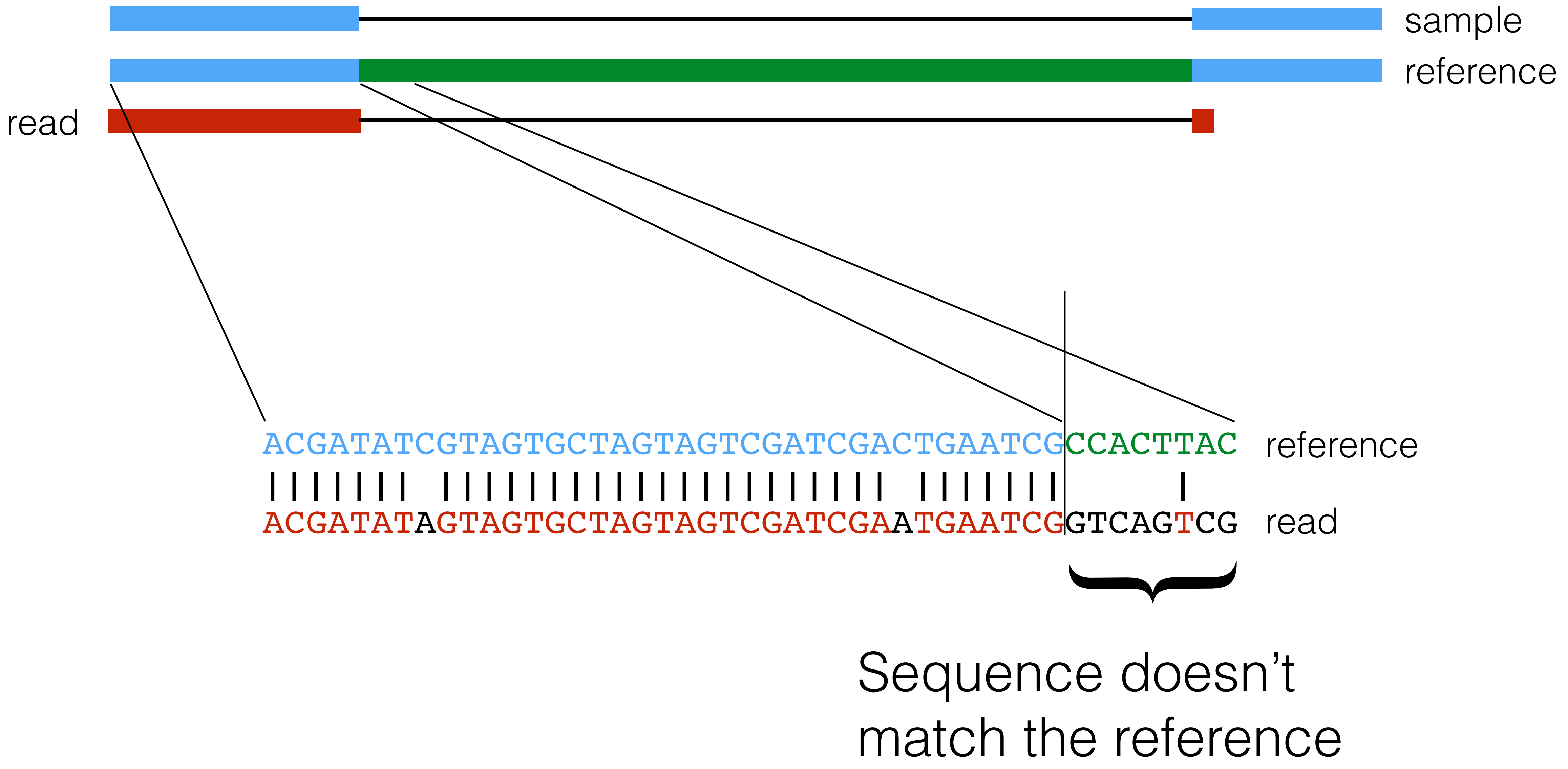
# Mapping across a deletion

- A read straddles a deletion

What happens if we map this read to the reference?

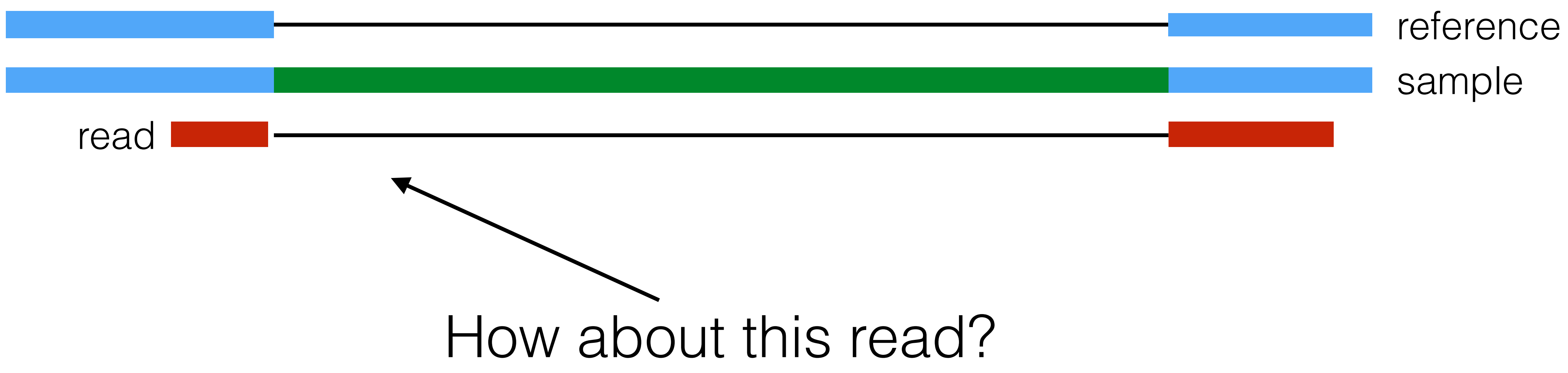
# Successful mapping

- A read straddles a deletion



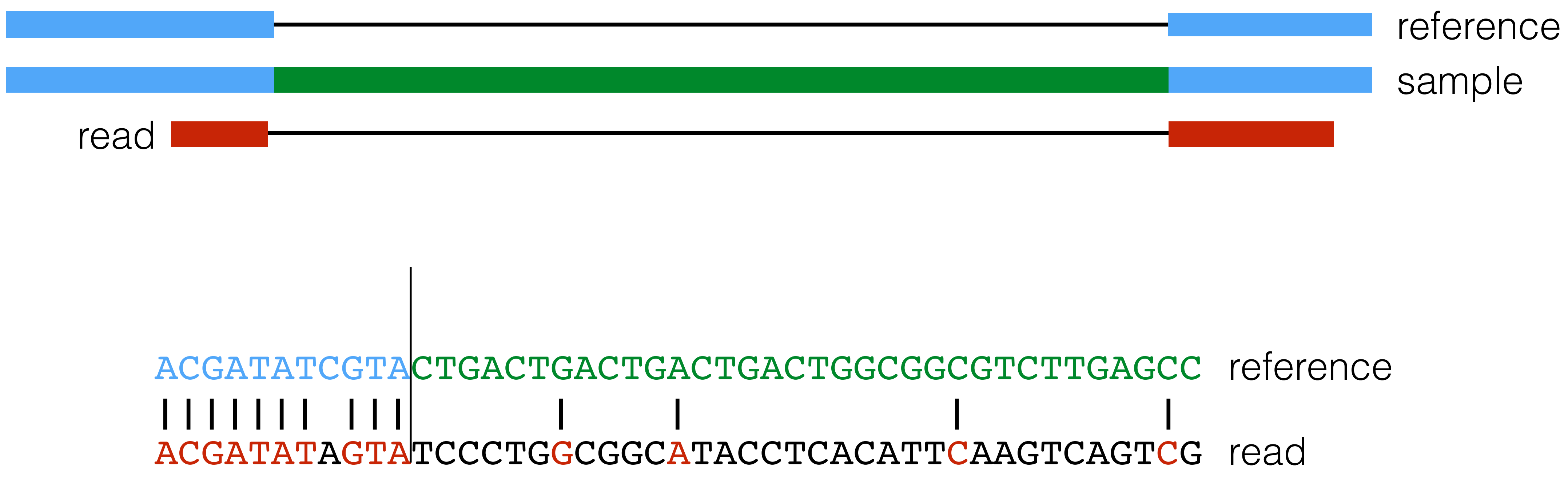
# Mapping across a deletion

- A read straddles a deletion



# Failed mapping

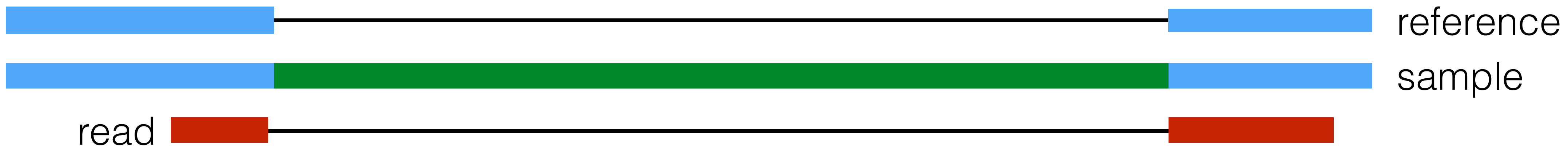
- A read straddles a deletion




This read cannot be mapped

# Failed mapping

- A read straddles a deletion




```
ACGATATCGTACTGACTGACTGACTGACTGGCGGCGTCTTGAGCC  reference
||||||| |||      |    |           |        |
ACGATATAGTATCCCTGGCGGCATACCTCACATTCAAGTCAGTCG  read
```

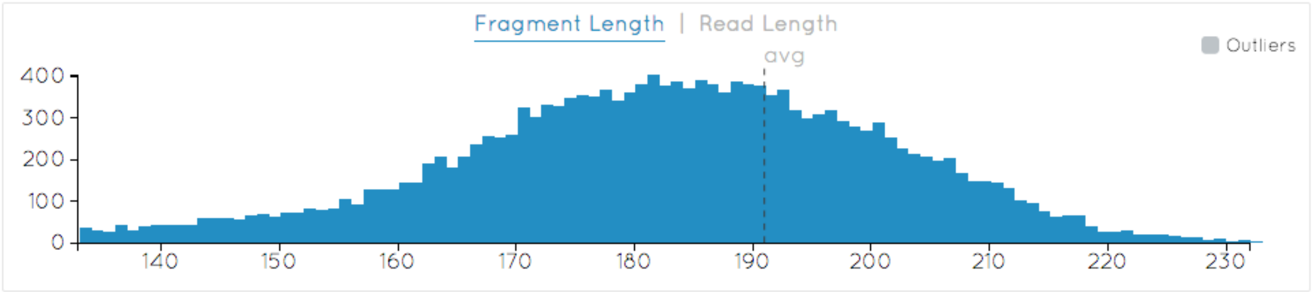
This read cannot be mapped

Or can it?

# Split read mapping

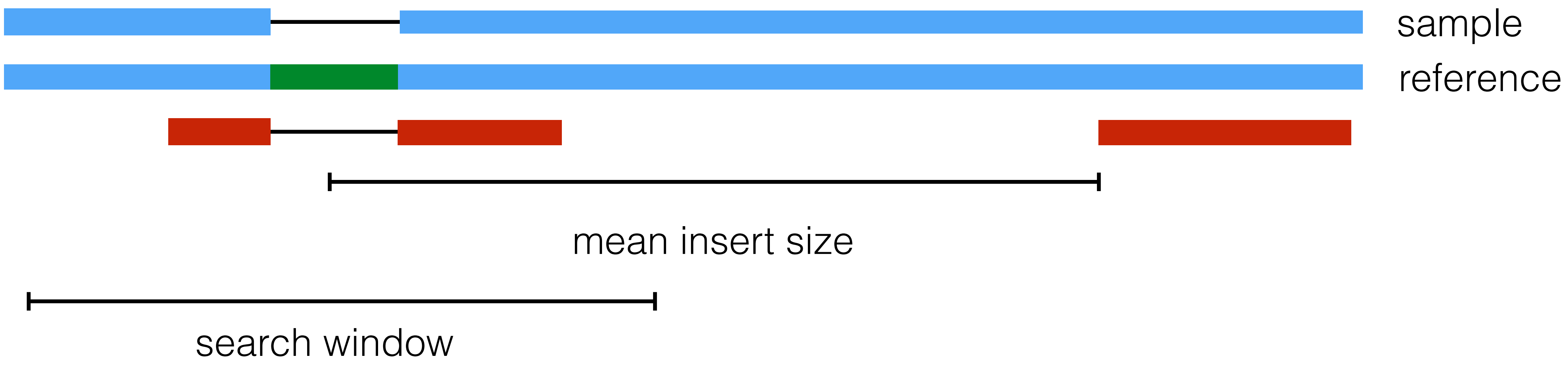
Unmapped

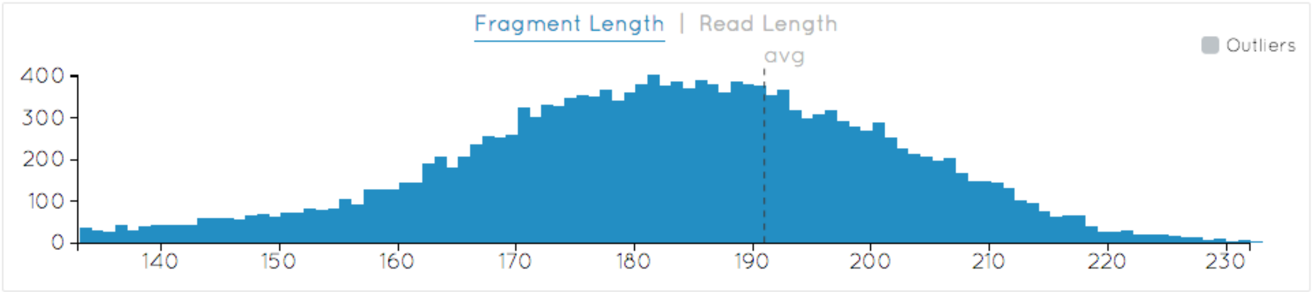
Anchor - uniquely mapped read




Estimate of fragment length -
we have an idea of where to look

# Split read mapping




Estimate of fragment length -
we have an idea of where to look

# Mapping across a deletion

Use Smith-Waterman algorithm across a window

Match: 30 (10)
Mismatch: -60 (-9)
Open gap: -60 (-15)
Extend gap: -1 (-1)

Opening a gap is not penalized more than a mismatch

# Mapping strategies

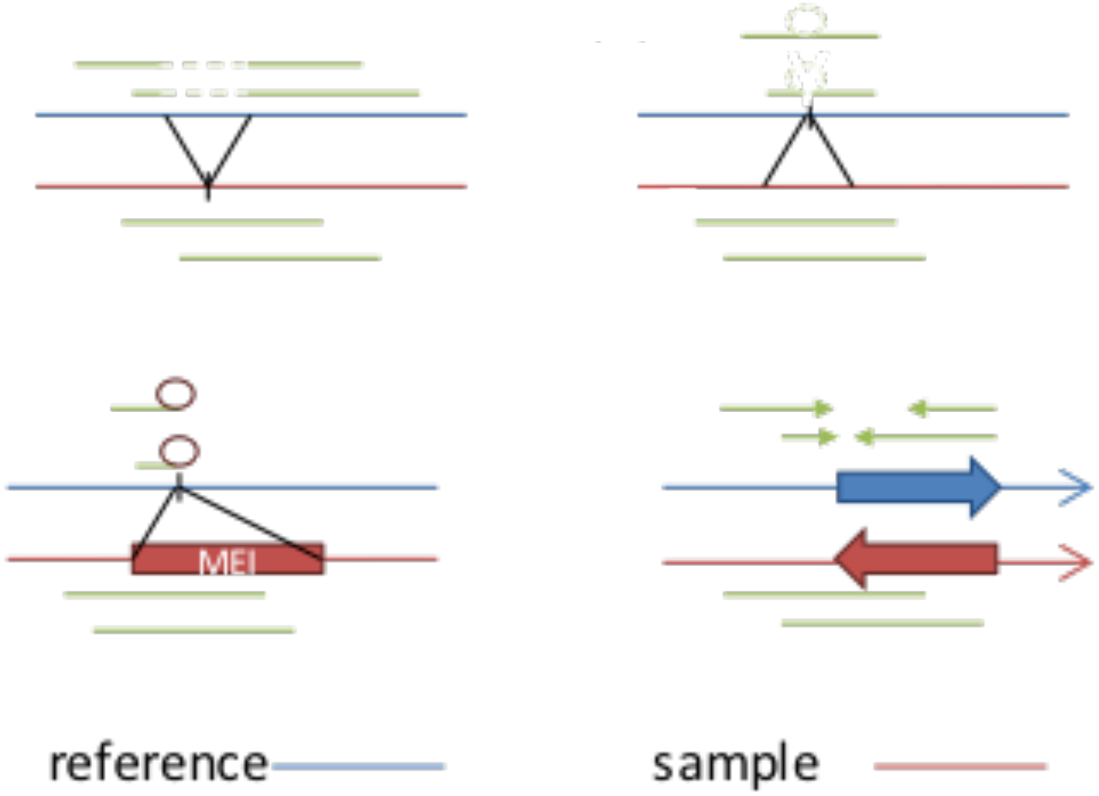
Try to map the read assuming that the sample contains one of the following structural variants

Sample contains a deletion

Sample contains inserted sequence

Sample contains mobile element

Sample contains an inversion

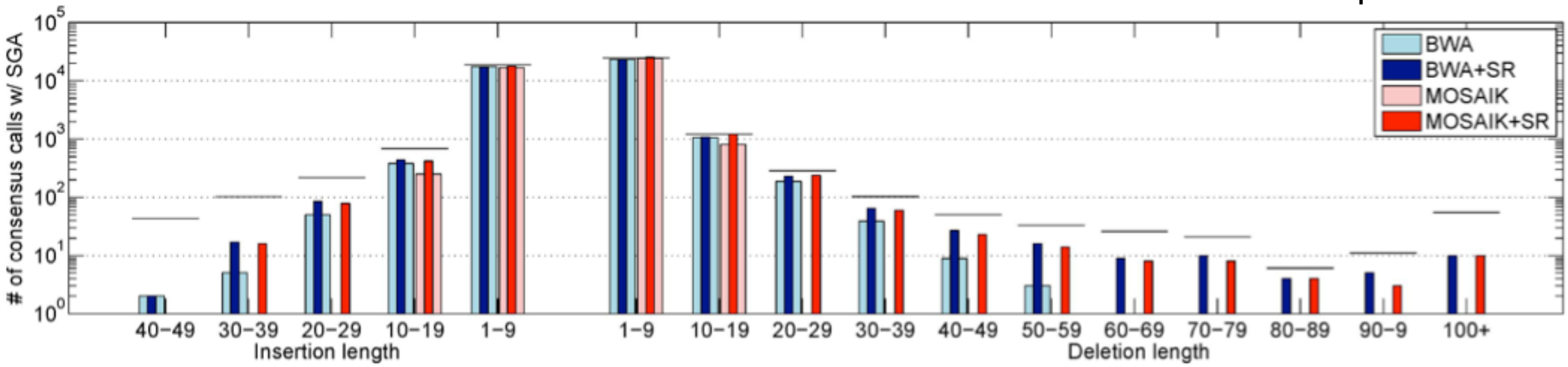



*Scissors methodology

# Does it work

Call indels in AFR samples from 1000 Genomes Project

SR = split read

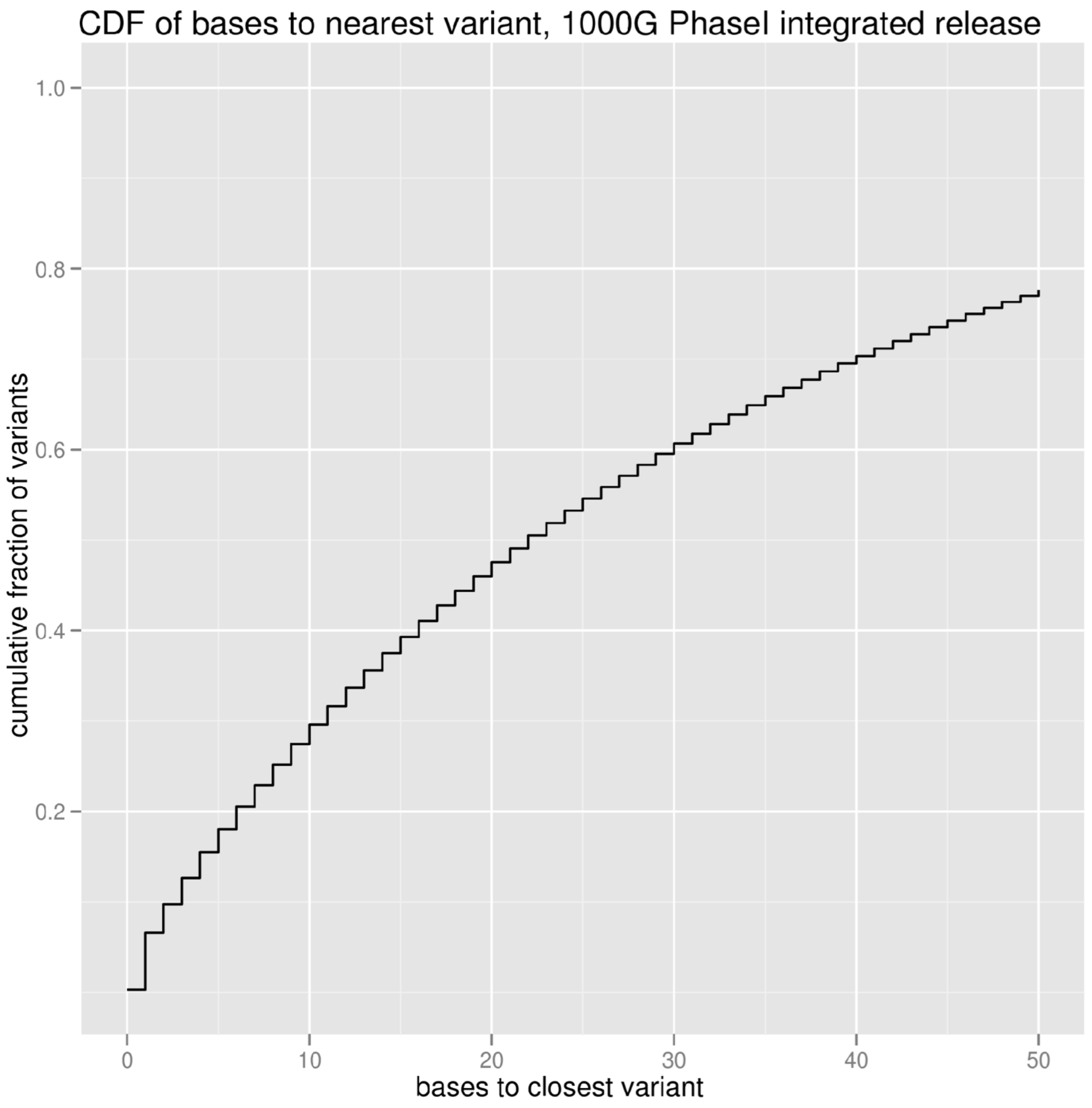



Scissors - unpublished

# Clusters of variants

What happens if multiple variants are in close proximity?

Are we leveraging all of our current knowledge when mapping?




Erik Garrison - 1000 Genomes Phase I

# Toy clustered variants example

Sample contains an insertion, a deletion and a SNP, all in close proximity

# Clustered variants toy example

Mapping will not be able to accurately place this read

Split read mapping will not save us!

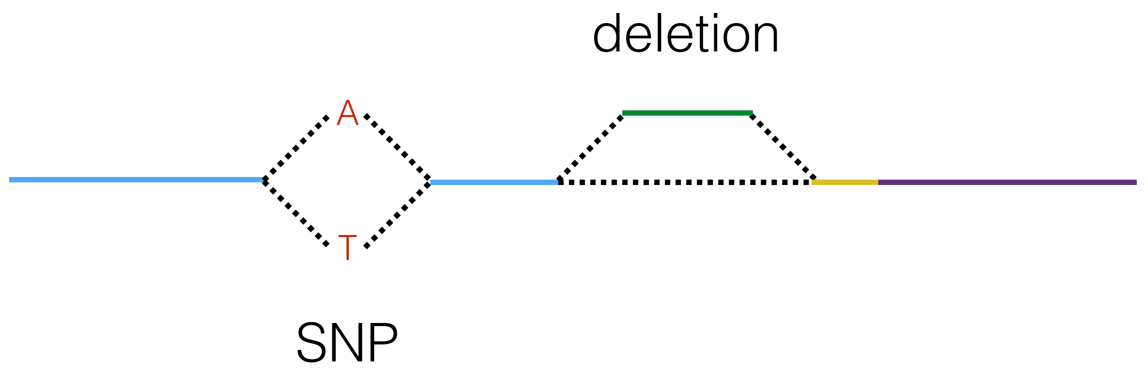
# Known variation

1000 Genomes Project Phase I - 1,092 individuals

- 38 million SNPs
- 1.4 million bi-allelic indels
- 14,000 large deletions

What if we already know that the SNP and the deletion exist in the human population?

Build a graph of the local region

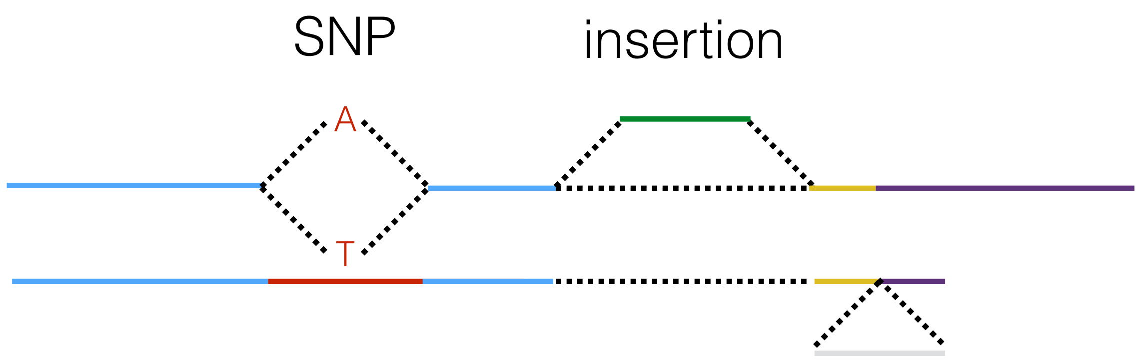



An integrated map of genetic variation from 1,092 human genomes - Nature **491**, 56-65 (2012)

# Map against the local graph

Take our previous un-mappable read

Map to the local graph



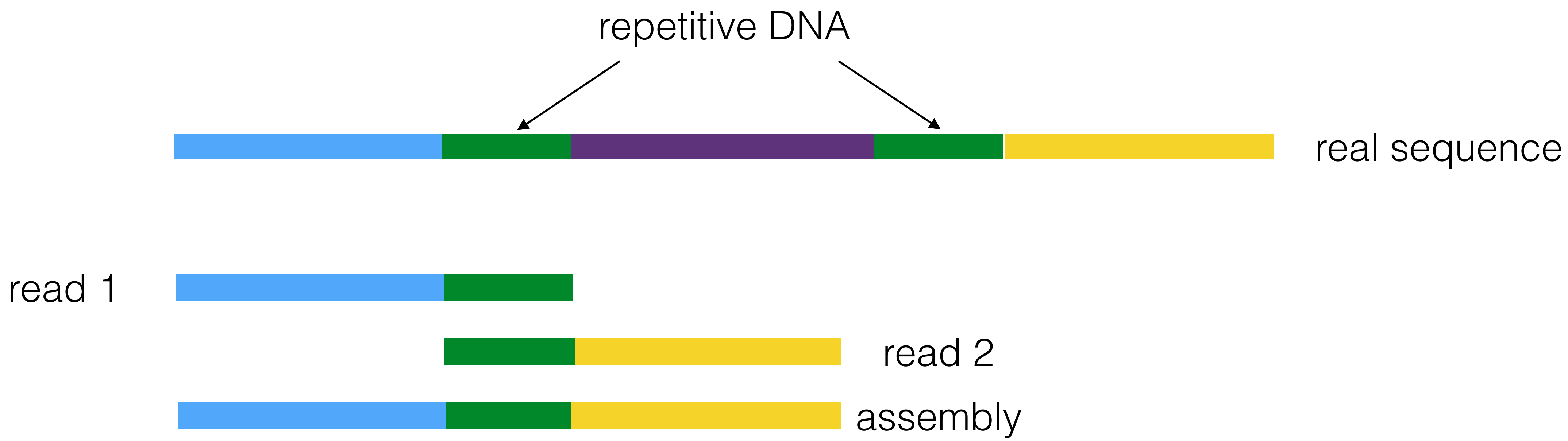

The only difference to the graph is the final insertion
- we can easily place this read now

# Unresolved problems

- Mapping tries to match the reference, so inherently introduces a bias towards the reference
- We have to modify parameters based on the read content (e.g. deletions)
- Mapping to repetitive DNA is still problematic
- What if there is no or an incomplete reference for the sequenced organism?

# Assembly

Can we just overlap the reads to create an assembly?



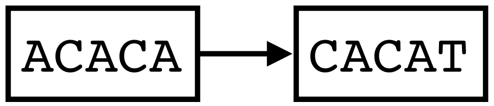

We can, if there isn't too much repetitive DNA
BUT, >50% is repetitive

# Clean de Bruijn graph

Break reads into k-mers (of length 5)
Each k-mer is a node in the graph

```
read  ACACATGTACGTAC
      ACACA
       CACAT
        ACATG
         CATGT
          ATGTA
           TGTAC
            GTACG
             TACGT
              ACGTA
               CGTAC
```

→ assemble de Bruijn graph

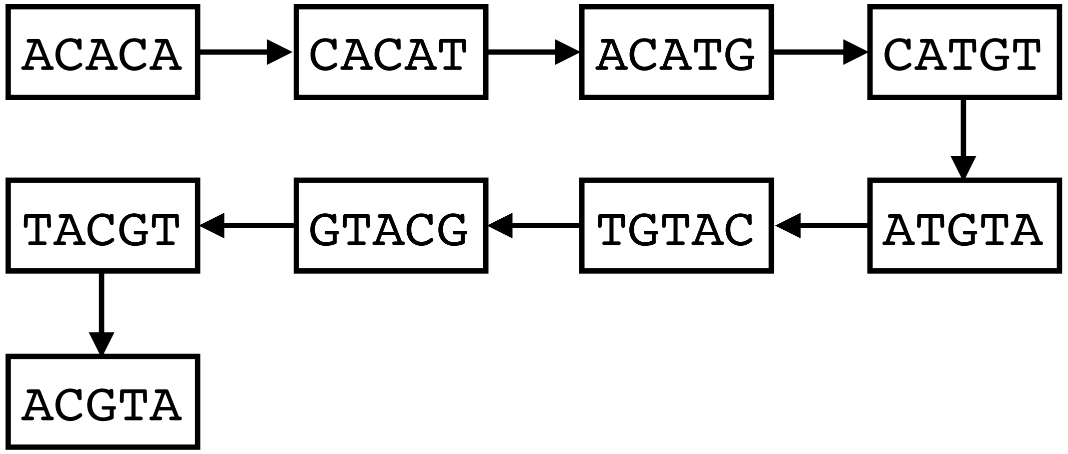


# Clean de Bruijn graph

Break reads into k-mers (of length 5)
Each k-mer is a node in the graph

```
read  ACACATGTACGTAC
      ACACA
       CACAT
        ACATG
         CATGT
          ATGTA
           TGTAC
            GTACG
             TACGT
              ACGTA
               CGTAC
```

assemble de Bruijn graph




This is the de Bruijn graph representation of the read

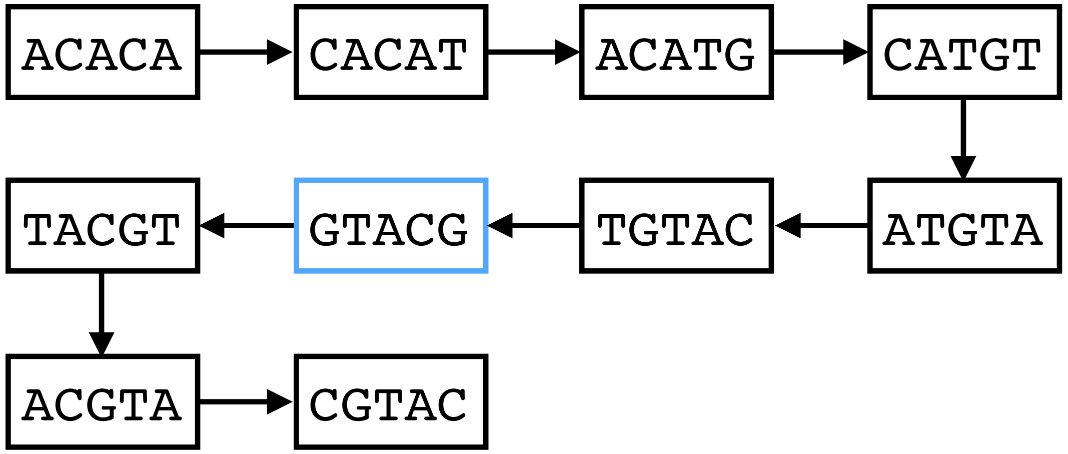
# De Bruijn graph

Let’s add one more base (‘G’) to the read

read ACACATGTACGTACG

```
ACACA
 CACAT
  ACATG
   CATGT
    ATGTA
     TGTAC
      GTACG
       TACGT
        ACGTA
         CGTAC
          GTACG
```

→ assemble de Bruijn graph




Final node

GTACG

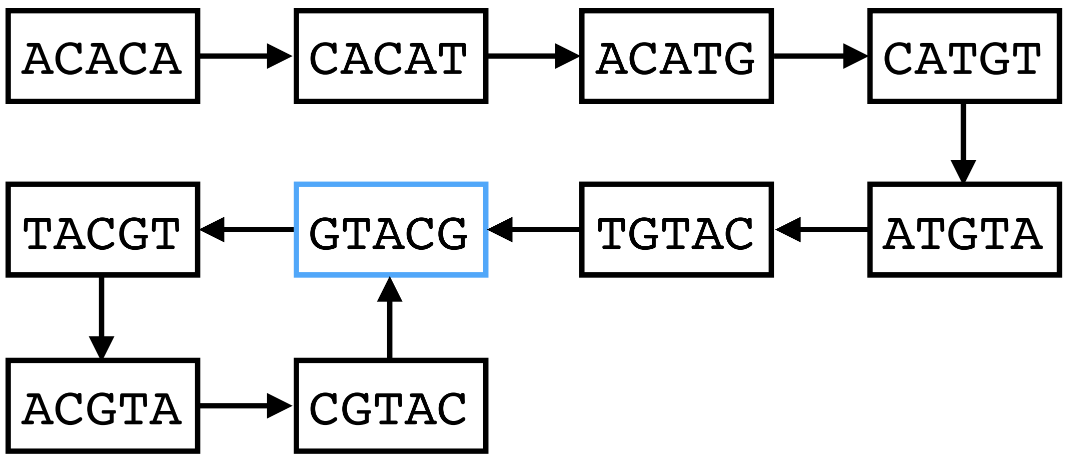
# De Bruijn graph

Let’s add one more base (‘G’) to the read

read ACACATGTACGTACG

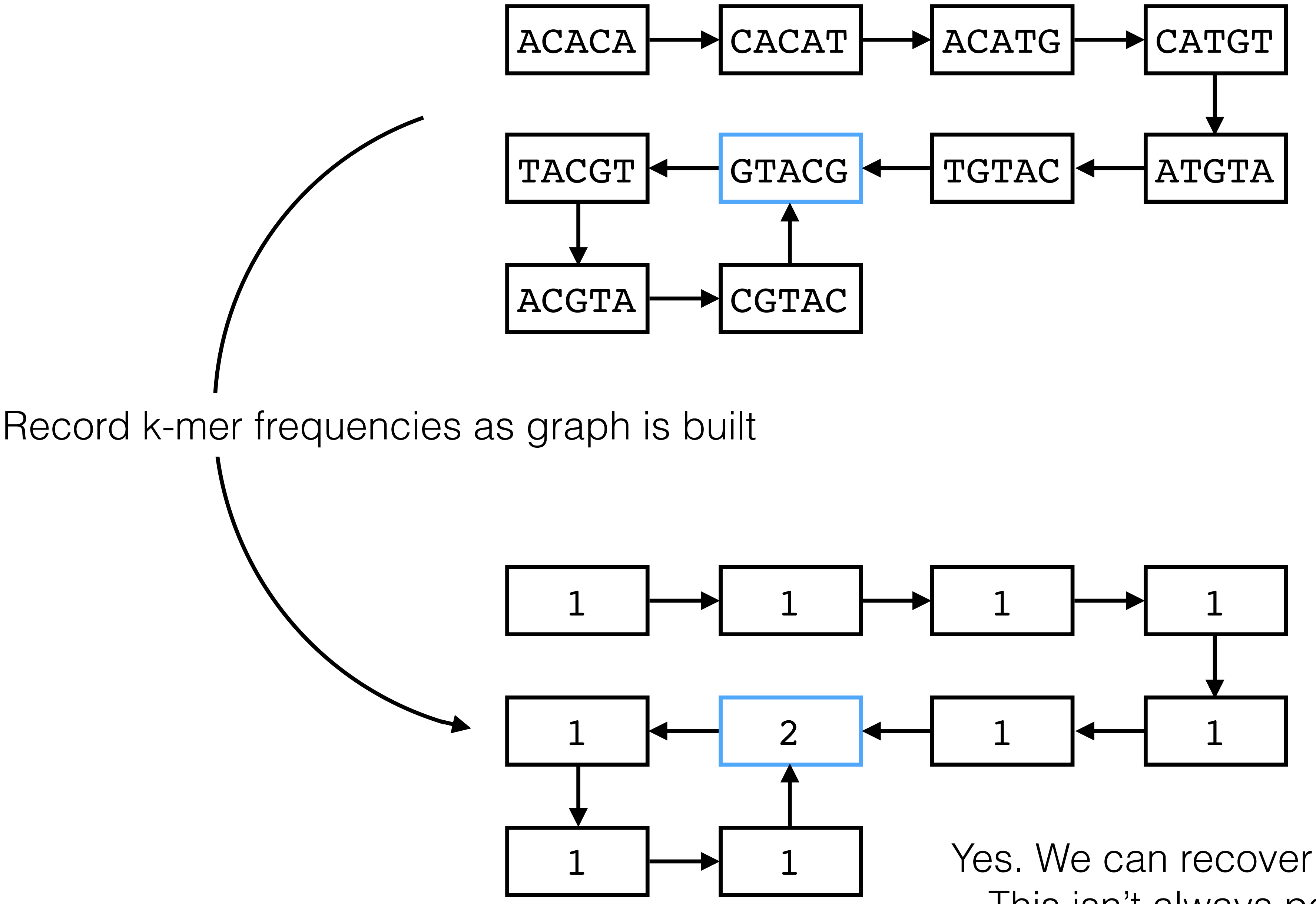
```
ACACA
 CACAT
  ACATG
   CATGT
    ATGTA
     TGTAC
      GTACG
       TACGT
        ACGTA
         CGTAC
          GTACG
```

→ assemble de Bruijn graph




Our graph has a loop!
Can we retrieve our read from the graph?

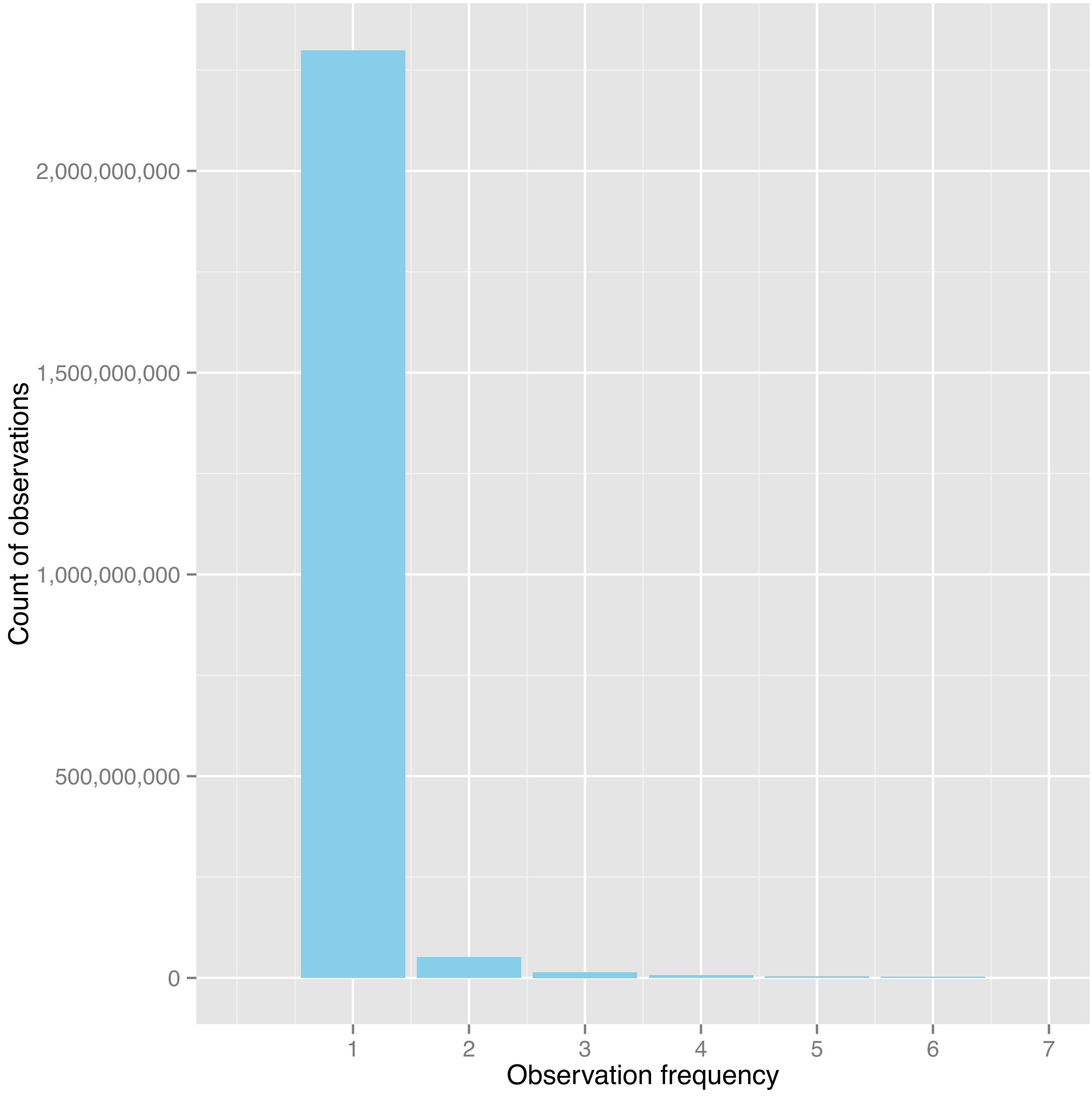
# Graph back to read




Yes. We can recover the read.
This isn't always possible.
(Sanger sequencing)

# Distribution of k-mers

- Consider using a k-mer length of 23
- There are $4^{23} = 7 \times 10^{13}$ possible mers of length 23, (70,000,000,000,000 k-mers)
- The human genome only has $3 \times 10^9$bp
- Most k-mers are unique

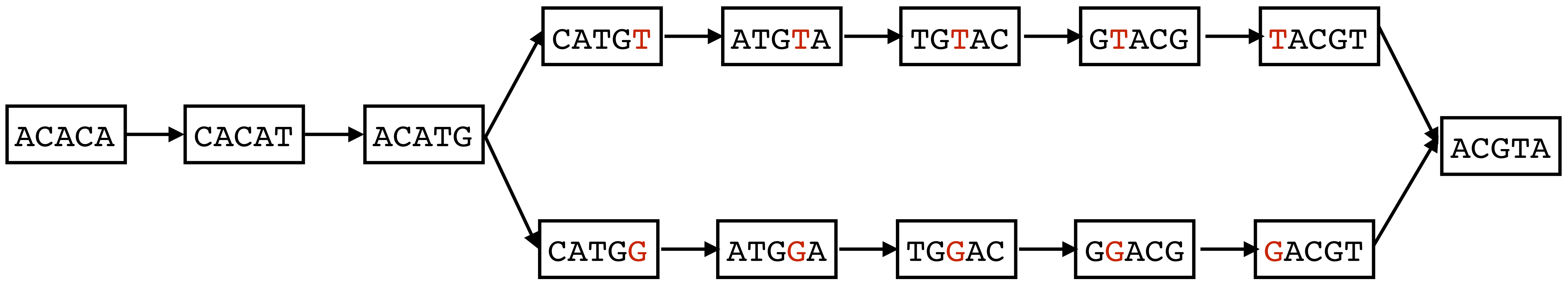


# Bubbles

Sample has a heterozygous SNP

ACACATGTACGTAC

ACACATGGACGTAC

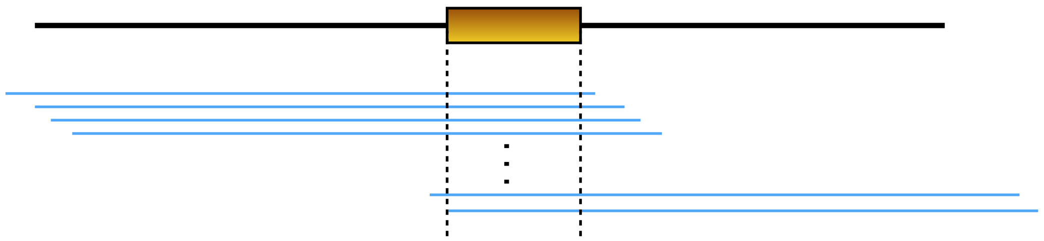



Is this a SNP or an error?

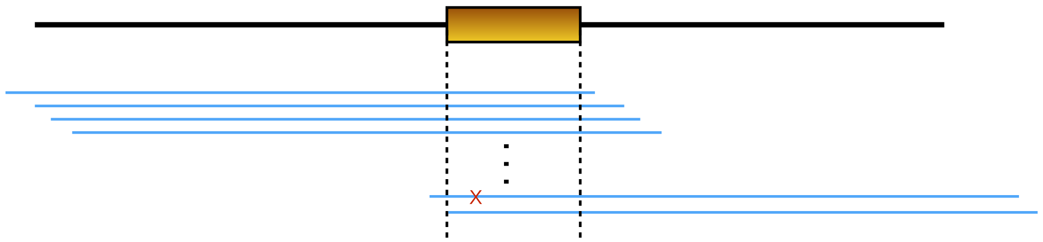
# k-mer frequency distribution

Errors

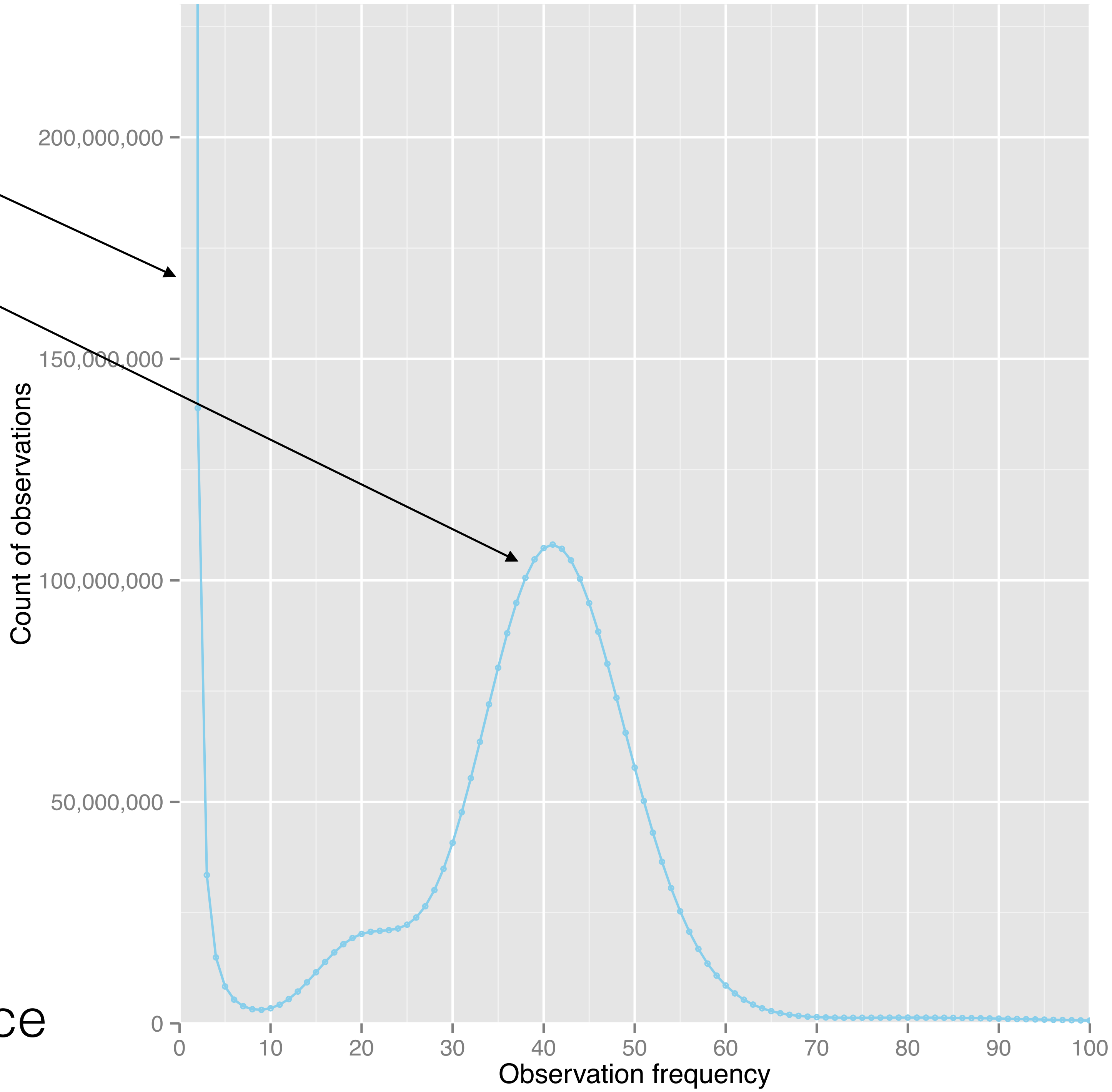
Peak at sequencing coverage



If the coverage is 40x, we observe the k-mer 40 times



k-mer with error, we only observe once



# Summary

- Many mapping strategies
  - Hash based mapping
  - Burrows-Wheeler transform
  - Split read mapping
  - Local graph alignment
- Overlap assembly
- de Bruijn graph assembly
- Choose a strategy (or combination of strategies) based on the experiment and the available data

# Mapping tools

## Mappers:

Mosaik: https://github.com/wanpinglee/MOSAIK

BWA: http://bio-bwa.sourceforge.net/

STAMPY: http://www.well.ox.ac.uk/project-stampy

## Split-read aligners:

SCISSORS: https://github.com/wanpinglee/scissors

Pindel: http://gmt.genome.wustl.edu/pindel/current/