Computational Genomics

SV Detection with Paired-End Reads

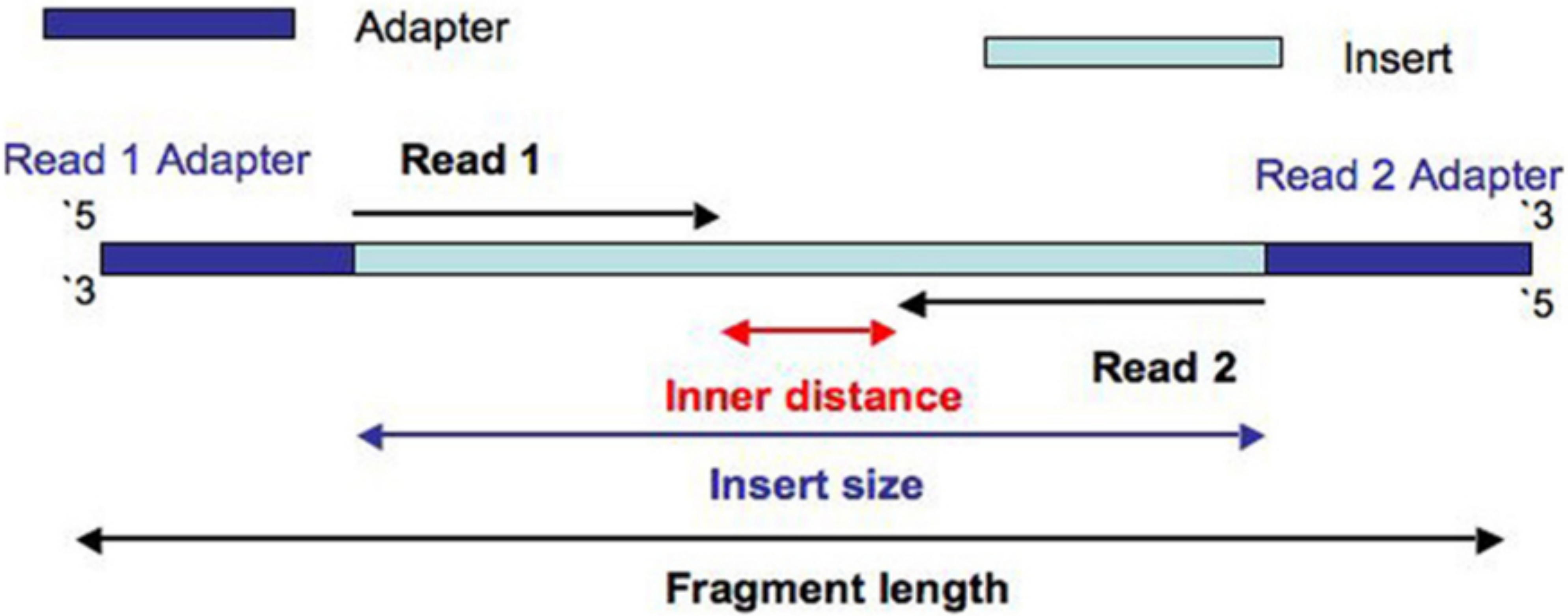
Structural Variation Detection

- Single Read: insertions, deletions, duplications, inter-chromosomal…
- Problems:
  - relying on long reads
  - long insertions
  - repeats

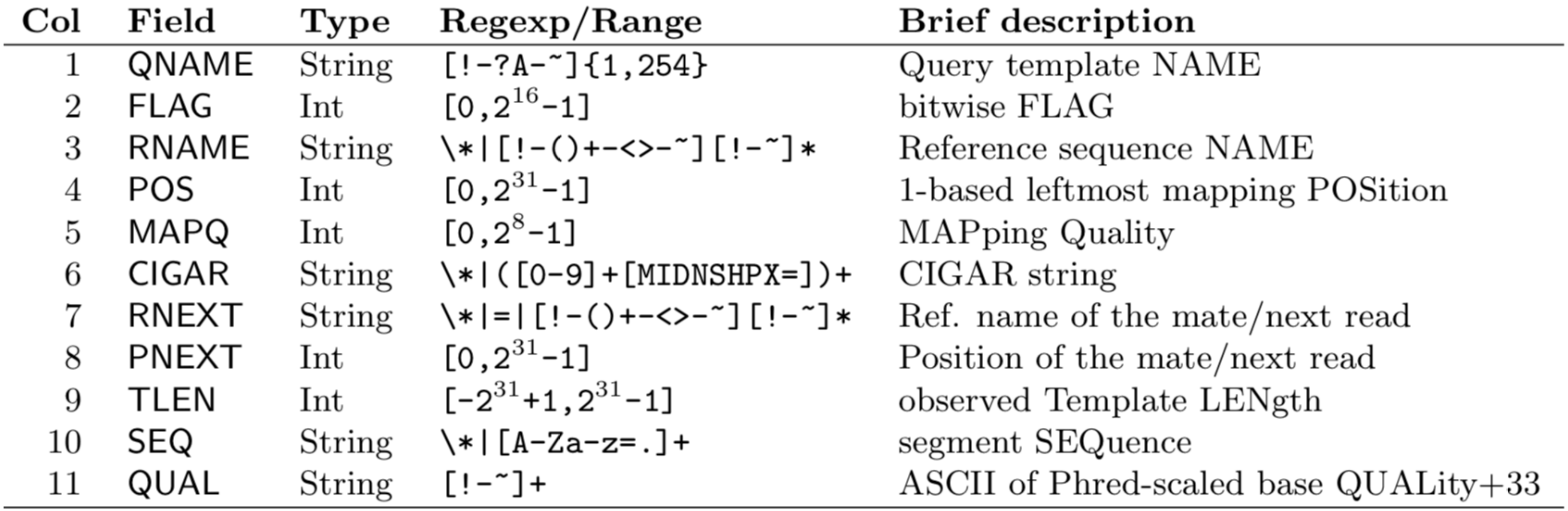
Structural Variation Detection

- Paired-end Reads: expand the search range.
- Method:
  - Set the expected insert size range
  - Find out the read pairs with abnormal insert sizes

Insert Size


Adapter
Insert
Read 1 Adapter
Read 1
Read 2 Adapter
`5
`3
`3
`5
Read 2
Inner distance
Insert size
Fragment length

Understanding the Reads

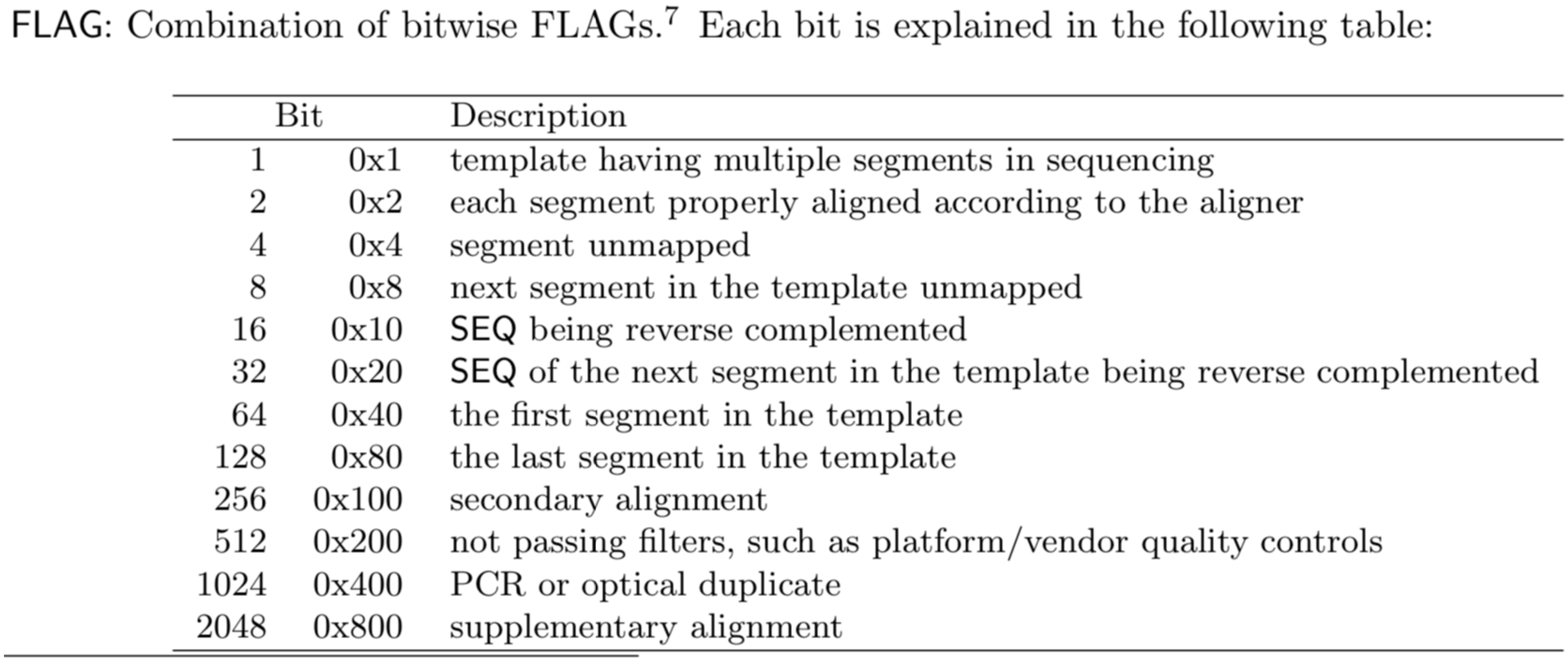


Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,254}`	Query template NAME
2	FLAG	Int	$[0,2^{16}-1]$	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	$[0,2^{31}-1]$	1-based leftmost mapping POSition
5	MAPQ	Int	$[0,2^{8}-1]$	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	$[0,2^{31}-1]$	Position of the mate/next read
9	TLEN	Int	$[-2^{31}+1,2^{31}-1]$	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

Understanding the Reads

- ecoli_3_492_3:0:0_1:0:0_126b4 163 0 2 44 100M 0 393 100 CTTTTCATTCT…AGTAACTTA array('B', [20, 20, …, 20, 20])
- QNAME, FLAG, RNAME, POS, MAPQ, CIGAR, RNEXT, PNEXT, TLEN, SEQ, QUAL
- https://samtools.github.io/hts-specs/SAMv1.pdf
  - Read *1.4 The alignment section: mandatory fields*
- Try identify the field which indicate the pairing of reads.

Understanding the Reads



FLAG: Combination of bitwise FLAGs.[7] Each bit is explained in the following table:

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Understanding the Reads

- ecoli_3_492_3:0:0_1:0:0_126b4 163 0 2 44 100M 0 393 100 CTTTTCATTCT…AGTAACTTA array('B', [20, 20, …, 20, 20])

- QNAME, FLAG, RNAME, POS, MAPQ, CIGAR, RNEXT, PNEXT, TLEN, SEQ, QUAL

- What is the insert size of the paired read?

Understanding the Reads

- ecoli_3_492_3:0:0_1:0:0_126b4 163 0 2 44 100M 0 393 100 CTTTTCATTCT…AGTAACTTA array('B', [20, 20, …, 20, 20])

- QNAME, FLAG, RNAME, POS, MAPQ, CIGAR, RNEXT, PNEXT, TLEN, SEQ, QUAL

- What is the insert size of the paired read?
- Ans: 392-2+100 = 491

Implementation

- Task1: Read in file *paired_reads.bam* with pysam. Print the first 10 reads, and check whether they are properly paired. Verify with pysam functions.

Implementation

- Task2: Extend the script to parse 10,000 reads and record the insert sizes. Report the mean and standard deviation of the results. Use proper graphical tools, and discuss whether the insert sizes look *normally distributed*.

Tip: *numpy* has built in functions for mean and standard deviation calculation.

Implementation

- Task3: Parse the entire file this time, and report any value that falls outside of the range: $mean \pm 2\times standard\ deviation$. For a normal distribution, we would expect 95% of the data to fall within the range. Does the data conform to the expectation? What does it say about the existence and abundance of SV in this data?

More readings for fun!

- Standard deviation:

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

- Normal distribution:
  https://en.wikipedia.org/wiki/Normal_distribution

- For the range:

  https://en.wikipedia.org/wiki/68–95–99.7_rule