



# PO: Precision Oncology Course

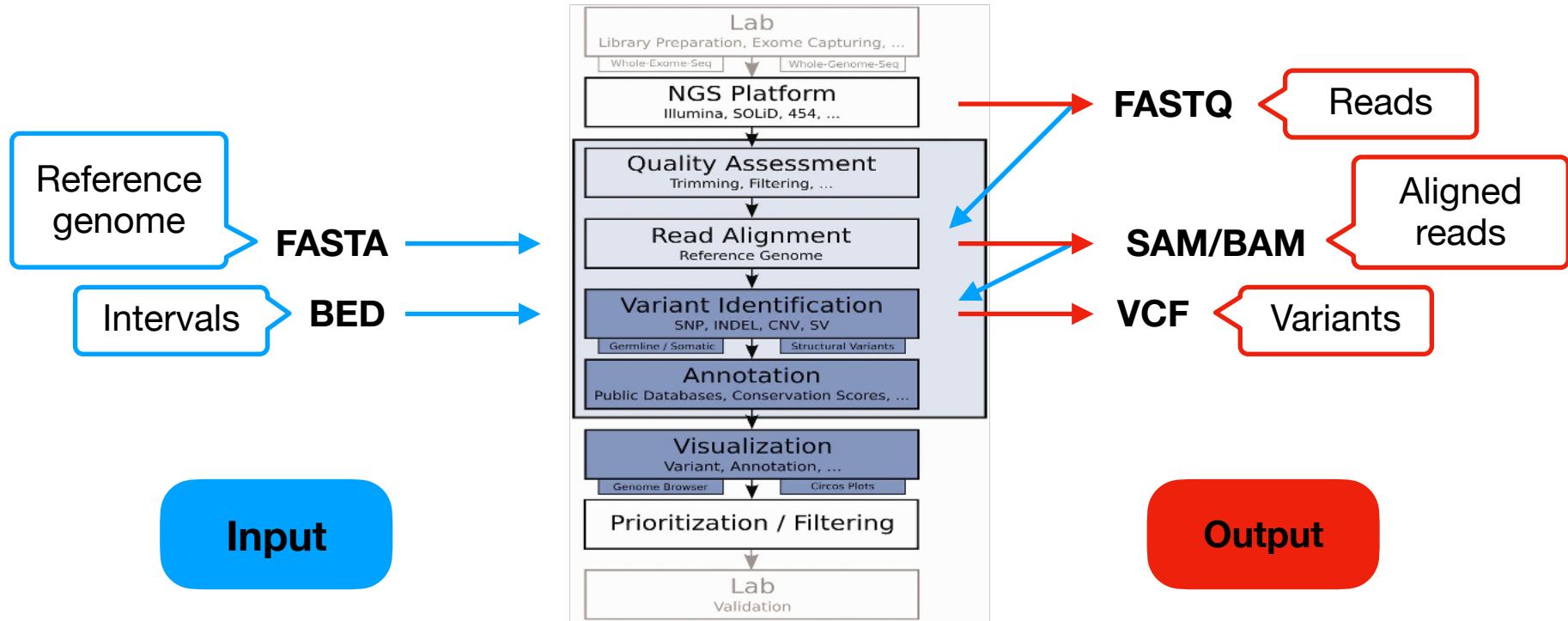
## Data Formats



CNIO BIOINFORMATICS UNIT

*cnio* stop cancer

# Formats Outline



# Sequencing

Output: Reads (FASTQ)

# FASTQ format

Typical extensions: **.fq, .fastq**

Each read is composed by **4 lines**:

- "@" Read ID and optional description (space separated)
- Sequence
- "+" (optionally: repeat the read ID)
- Encoded Base Quality Score

Identifier	• @SRR566546.970 HWUSI-EAS1673_11067_FC7070M:4:1:2299:1109 length=50
Sequence	• TTGCCTGCCTATCATTAGTCAGGTGGAGATGTGAGGATCAGT
'+' sign	• +
Quality scores	• hhhhhhhhhghhhhhfhhhhfffffe'ee['X]b[d[ed' [Y[^Y
Identifier	• @SRR566546.971 HWUSI-EAS1673_11067_FC7070M:4:1:2374:1108 length=50
Sequence	• GATTGTATGAAAGTATAACAACTAAAAGTGCAGGTGGATCAGAGTAAGTC
'+' sign	• +
Quality scores	• hhggfhhcghghggfcffdhfehhhhcehdchhdhaehffffde'bVd

# FASTQ format

## Nucleotide codes (IUPAC)

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	any base
. or -	gap

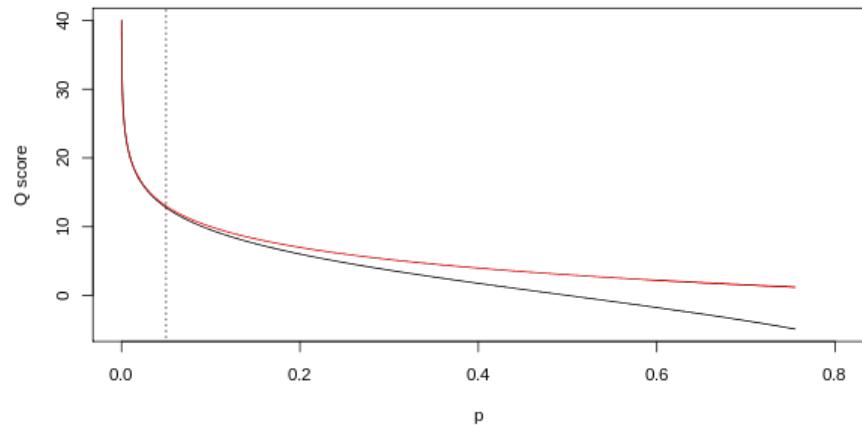
# FASTQ format

## Base Quality Score

**Phred quality score (Q):** Measure of the quality of the DNA sequencing for each base.

**P:** Base calling error probability.

$$Q = -10 \cdot \log_{10}(P)$$



**The higher the Q, the lower the P.**

[FASTQ format. Wikipedia.](#)

# FASTQ format

## ASCII code

Q is encoded in a  
single ASCII  
character for brevity.

$$ASCII = Q + 33$$

ASCII for Q=0? !  
ASCII for Q=40? |

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	NUL	(null)	32	20 040	&#32;	Space		64	40 100	&#64;	Ø		96	60 140	&#96;	`	
1	1 001	SOH	(start of heading)	33	21 041	&#33;	!		65	41 101	&#65;	A		97	61 141	&#97;	a	
2	2 002	STX	(start of text)	34	22 042	&#34;	"		66	42 102	&#66;	B		98	62 142	&#98;	b	
3	3 003	ETX	(end of text)	35	23 043	&#35;	#		67	43 103	&#67;	C		99	63 143	&#99;	c	
4	4 004	EOT	(end of transmission)	36	24 044	&#36;	\$		68	44 104	&#68;	D		100	64 144	&#100;	d	
5	5 005	ENQ	(enquiry)	37	25 045	&#37;	%		69	45 105	&#69;	E		101	65 145	&#101;	e	
6	6 006	ACK	(acknowledge)	38	26 046	&#38;	&		70	46 106	&#70;	F		102	66 146	&#102;	f	
7	7 007	BEL	(bell)	39	27 047	&#39;	,		71	47 107	&#71;	G		103	67 147	&#103;	g	
8	8 010	BS	(backspace)	40	28 050	&#40;	(		72	48 110	&#72;	H		104	68 150	&#104;	h	
9	9 011	TAB	(horizontal tab)	41	29 051	&#41;	)		73	49 111	&#73;	I		105	69 151	&#105;	i	
10	A 012	LF	(NL line feed, new line)	42	2A 052	&#42;	*		74	4A 112	&#74;	J		106	6A 152	&#106;	j	
11	B 013	VT	(vertical tab)	43	2B 053	&#43;	+		75	4B 113	&#75;	K		107	6B 153	&#107;	k	
12	C 014	FF	(NP form feed, new page)	44	2C 054	&#44;	/		76	4C 114	&#76;	L		108	6C 154	&#108;	l	
13	D 015	CR	(carriage return)	45	2D 055	&#45;	-		77	4D 115	&#77;	M		109	6D 155	&#109;	m	
14	E 016	SO	(shift out)	46	2E 056	&#46;	.		78	4E 116	&#78;	N		110	6E 156	&#110;	n	
15	F 017	SI	(shift in)	47	2F 057	&#47;	/		79	4F 117	&#79;	O		111	6F 157	&#111;	o	
16	10 020	DLE	(data link escape)	48	30 060	&#48;	0		80	50 120	&#80;	P		112	70 160	&#112;	p	
17	11 021	DC1	(device control 1)	49	31 061	&#49;	1		81	51 121	&#81;	Q		113	71 161	&#113;	q	
18	12 022	DC2	(device control 2)	50	32 062	&#50;	2		82	52 122	&#82;	R		114	72 162	&#114;	r	
19	13 023	DC3	(device control 3)	51	33 063	&#51;	3		83	53 123	&#83;	S		115	73 163	&#115;	s	
20	14 024	DC4	(device control 4)	52	34 064	&#52;	4		84	54 124	&#84;	T		116	74 164	&#116;	t	
21	15 025	NAK	(negative acknowledge)	53	35 065	&#53;	5		85	55 125	&#85;	U		117	75 165	&#117;	u	
22	16 026	SYN	(synchronous idle)	54	36 066	&#54;	6		86	56 126	&#86;	V		118	76 166	&#118;	v	
23	17 027	ETB	(end of trans. block)	55	37 067	&#55;	7		87	57 127	&#87;	W		119	77 167	&#119;	w	
24	18 030	CAN	(cancel)	56	38 070	&#56;	8		88	58 130	&#88;	X		120	78 170	&#120;	x	
25	19 031	EM	(end of medium)	57	39 071	&#57;	9		89	59 131	&#89;	Y		121	79 171	&#121;	y	
26	1A 032	SUB	(substitute)	58	3A 072	&#58;	:		90	5A 132	&#90;	Z		122	7A 172	&#122;	z	
27	1B 033	ESC	(escape)	59	3B 073	&#59;	:		91	5B 133	&#91;	[		123	7B 173	&#123;	{	
28	1C 034	FS	(file separator)	60	3C 074	&#60;	<		92	5C 134	&#92;	\		124	7C 174	&#124;		
29	1D 035	GS	(group separator)	61	3D 075	&#61;	=		93	5D 135	&#93;	]		125	7D 175	&#125;	)	
30	1E 036	RS	(record separator)	62	3E 076	&#62;	>		94	5E 136	&#94;	^		126	7E 176	&#126;	~	
31	1F 037	US	(unit separator)	63	3F 077	&#63;	?		95	5F 137	&#95;	_		127	7F 177	&#127;	DEL	

Source: [www.LookupTables.com](http://www.LookupTables.com)

# FASTQ format

Single-end vs paired-end

One unique sample can have 1 or 2 files:

- **1 file:** Single-end sequencing  
(\*.fastq)
- **2 files:** Paired-end sequencing  
(\*\_R1.fastq and \*\_R2.fastq)

R1

```
@SN1083:379:H8VA1ADXX:2:1101:1248:2144 1:N:0:12
CCTAAATGGTGCATGCTAGGAGGCCGTGCCCTCTTGAAAAGTTGTATGTGAA
+
BBBFFFFFFFBFFFFIIIFI<FFIIIIIFIIIFBFIIIIIIIIFFIIIIIFI
```

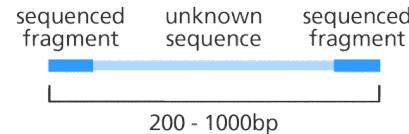
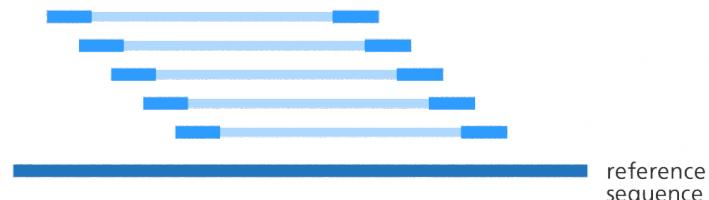
R2

```
@SN1083:379:H8VA1ADXX:2:1101:1248:2144 2:N:0:12
CATTTCGACGTTGTTAATAAGCTCTGCGTACTTGCAAGCTATCTGCGCGAACG
+
BBBFFFFFFFIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIFFF
```

Single-end reads



Paired-end reads



NGS overlapping reads. Biostars. Post 241139.

# FASTQ format

## Exercise

1. Download the data

<https://drive.google.com/file/d/15OxI4O8GO3I4e8WJFwNZet8DqOoR94y2/view?usp=sharing>

2. Go to the terminal and open the file

```
$ more WEx_Tumour_R1.fastq
```

# FASTQ format

5 min

Question

**How would you detect the number of sequences present in the file? How much reads are there?**

## Useful commands

- **grep**: Searches plain-text data sets for lines that match a regular expression.
- **wc**: Counts words
- **wc -l**: Counts lines
- **| (Pipe)**: Lets you send the output of one command to another.

# FASTQ format

10 min

Question

**How would you detect the number of sequences present in the file? How much reads are there? 1398294 reads.**

Divide by 4 the result of `cat WEx_Tumour_R1.fastq | wc -l`

# Alignment

Input: Reads (FASTQ) and Reference Genome (FASTA)

Output: Aligned reads (BAM/SAM)

# FASTA format

Typical extensions: **.fasta, .fas, .fa, .fna, .fsa**

Each sequence is composed by at least **2 lines**:

- ">" Sequence ID and optional description (space separated)
- Line(s) with the whole sequence

Header	• >VIT_201s0011g03530.1
Sequence	• AATTAAGCATAAATACTCACTCTTACCCCTTATTTCTTATCTCTCATCACTTTGGTGCGAAG
Header	• GACCATGAGAACAGCTGCAATGGGTGAGGGTCTTCGCAAGGCATGCAGCCAAGACTGCATCA
Header	• >VIT_201s0011g03540.1
Sequence	• CAGGTAGCGTGAAGTTAACCCCTAGCGCTTAGACAAACAGCTGTAGTCACCGCCCACAAACACC
Header	• AGCCTCTGAGACACCACCTCAAACCTTCCACTTAAATACACATCCCTCACACCCTTTCAATT
Header	• >VIT_201s0011g03550.1
Sequence	• CATGCAAAGCTGAACGCGATGCTGTGATTGGTGGTAAGTGGTAGTTGAGTAAATTGACAGTGAA
	• GCCGAAATGGTAAAAGACTAAGGCTAGAAGTAGAATACCACTGTTCTCATCACGTGGGCCA

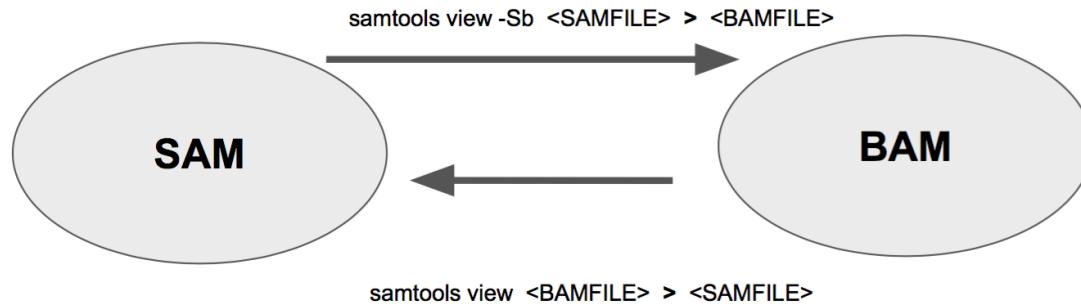
We can have  
multiple  
sequences in the  
same file:  
**multifasta**

# SAM/BAM format

**SAM** is the human readable text format ([.sam extension](#)).

**BAM** is the binary, machine efficient format ([.bam extension](#)).

Both contain exactly the **same information** and are interconvertible using **samtools**.



[File specifications](#)

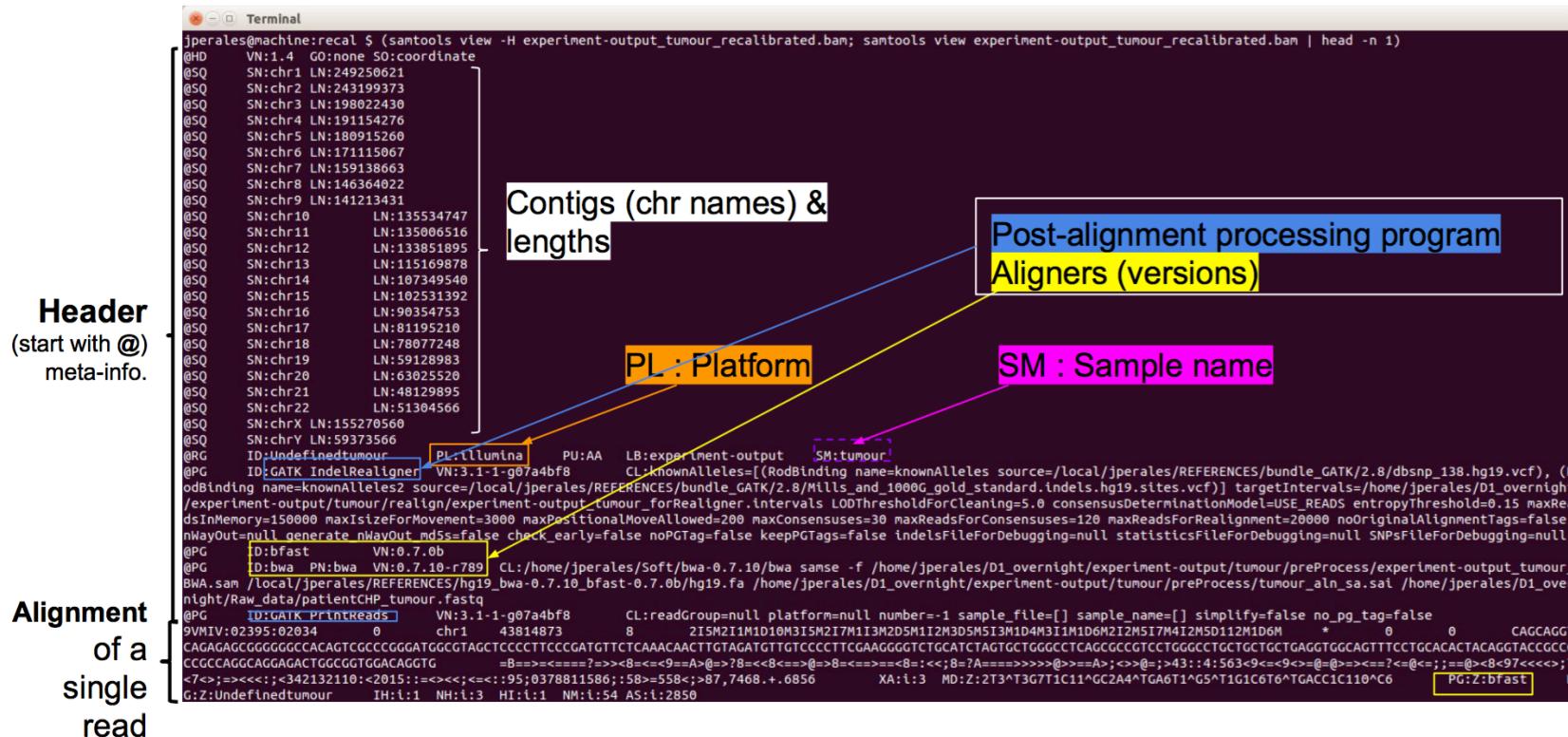
# SAM/BAM format

It has two different parts

- **Header:** All lines that start with “@“. Contains information about the alignment.
- **Alignments:** The rest of the lines, with 12 tab separated columns. Information relative to each aligned read.

# SAM/BAM format

## Parts



# SAM/BAM format

## Alignment

#1 ReadName	#2 99	#3 chr10	#4 2	#5 30	#6 <b>3MD2M1I1M</b>	#7 =	#8 14	#9 20	#10 CATCTG	#11 jjjjjjj	#12 z:Aligner
----------------	----------	-------------	---------	----------	------------------------	---------	----------	----------	---------------	----------------	------------------

#Col	Field	FIELDS
1.	QNAME	read name
2.	FLAG	bitwise FLAG* (unmapped, pair unmapped, properly mapped, ...)
3.	RNAME	Reference sequence name (e.g. chr1).
4.	POS	1-based leftmost position.
5.	MAPQ	Mapping Quality (Phred-scaled). Scale 0 to 255.
6.	CIGAR	<a href="#">extended CIGAR string</a>
7.	MRNM	Paired-end: Mate Reference sequence Name (= if same as RNAME).
8.	MPOS	Paired-end: 1-based Mate position.
9.	TLEN	Paired-end: Insert size
10.	SEQ	Read sequence
11.	QUAL	Base Quality Score from the Read sequence.
12.	OPT	Optional Tags

### If single-end:

7. Reference sequence name of the alignment of the next read in sequence.
8. Position in the alignment of the next read in sequence.
9. Number of bases covered by reads from the same fragment. Plus/minus means the current read is the leftmost/rightmost read.

# SAM/BAM format

## Alignment - FLAG

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
ReadName	99	chr10	2	30	3MD2M1I1M	=	14	20	CATCTG	jjjjjjjj	z:Aligner

Decimal	Description of read
1	Read paired
2	Read mapped in proper pair
4	Read unmapped
8	Mate unmapped
16	Read reverse strand
32	Mate reverse strand
64	First in pair
128	Second in pair
256	Not primary alignment
512	Read fails platform/vendor quality checks
1024	Read is PCR or optical duplicate
2048	Supplementary alignment

# SAM/BAM format

## Alignment - CIGAR

### CIGAR: Concise Idiosyncratic Gapped Alignment Report

- Compressed representation of an alignment.
- A CIGAR string is made up of <integer><op> pairs.
- Where "op" is an operation specified as a single character, usually an uppercase letter (see table).

RefPos:	1 2 3 4 5 6 7 8 9
Reference:	C C A T A C T - G A
Read:	C A T - C T A G
POS:	2
CIGAR:	3M1D2M1I1M

Op	Description
M	alignment match (can be a sequence match or mismatch)
I	insertion to the reference
D	deletion from the reference
N	skipped region from the reference
S	soft clipping (clipped sequences present in SEQ)
H	hard clipping (clipped sequences NOT present in SEQ)
P	padding (silent deletion from padded reference)
=	sequence match
X	sequence mismatch

Reference sequence with aligned reads	CIGAR string	Explanation
C T G C A T G T T A G A T A A * * G A T A G C T G T G C T A A A G G A T A * C T G G A T A A * G G A T A T G T T A [REDACTED] T G C T A	1M2I4M1D3M	Insertion & Deletion
a a a C A T G T T A G A A A C A T G T T A G	5M1P1I4M	Padding & Insertion
	5M1N5M	Spliced read
	3S8M	Soft clipping
	3H8M	Hard clipping

# Variant Identification

Input: Alignments (SAM/BAM) and Intervals (BED)

Output: Called Variants (VCF)

# BED format

**Intervals:** Regions of interest (important in targeted sequencing and WES). **More info here:** <https://gatk.broadinstitute.org/hc/en-us/articles/360035531852-Intervals-and-interval-lists>

## Typical extension: .bed

Tab delimited file with **up to 12 different columns. The first 3 are mandatory:**

1. **chrom:** The name of the chromosome (e.g. chr3, chrY, chr2).
2. **chromStart:** The starting position of the feature in the chromosome 0-based (The first base in a chromosome is numbered 0).
3. **chromEnd:** The ending position of the feature in the chromosome.

```
track name="CHP2_Designed" description="Amplicon_Insert_CHP2" type=bedDetail ionVersion=4.0
chr1 43814968    43815086    CHP2_MPL_1    .    GENE_ID=MPL
chr1 115252185   115252269   CHP2_NRAS_3   .    GENE_ID=NRAS
chr1 115256504   115256584   CHP2_NRAS_2   .    GENE_ID=NRAS
chr1 115258689   115258774   CHP2_NRAS_1   .    GENE_ID=NRAS
chr2 29432572    29432680    CHP2_ALK_2    .    GENE_ID=ALK
chr2 29443607    29443729    CHP2_ALK_1    .    GENE_ID=ALK
chr2 209113103   209113206   CHP2_IDH1_1   .    GENE_ID=IDH1
chr2 212288904   212288990   CHP2_ERBB4_8  .    GENE_ID=ERBB4
chr2 212530051   212530180   CHP2_ERBB4_7  .    GENE_ID=ERBB4
```

# BED format

## Typical extension: .bed

Tab delimited file with **up to 12 different fields. The rest are optional:**

4. **Name:** Defines the name of the BED line
5. **Score:** “.” or a number between 0 and 1000
6. **Strand:** + forward; - reverse
7. **thickStart**
8. **thickEnd**
9. **itemRgb:** (255,0,0)
10. **blockCount**
11. **blockSizes**
12. **blockStarts**

```
track name="CHP2_Designed" description="Amplicon_Insert_CHP2" type=bedDetail ionVersion=4.0
chr1 43814968 43815086 CHP2_MPL_1 . GENE_ID=MPL
chr1 115252185 115252269 CHP2_NRAS_3 . GENE_ID=NRAS
chr1 115256504 115256584 CHP2_NRAS_2 . GENE_ID=NRAS
chr1 115258689 115258774 CHP2_NRAS_1 . GENE_ID=NRAS
chr2 29432572 29432680 CHP2_ALK_2 . GENE_ID=ALK
chr2 29443607 29443729 CHP2_ALK_1 . GENE_ID=ALK
chr2 209113103 209113206 CHP2_IDH1_1 . GENE_ID=IDH1
chr2 212288904 212288990 CHP2_ERBB4_8 . GENE_ID=ERBB4
chr2 212530051 212530180 CHP2_ERBB4_7 . GENE_ID=ERBB4
```

# VCF format

Typical extension: .vcf

It has two different parts

- **Header:** All lines that start with “##”. Contains information about the variant annotation process.
- **Body:** The rest of the lines, with at least 8 tab separated fields. Information relative to each identified variant.

# VCF format

- **QUAL** is the **score** assigned to a given call **in log-scale**.
- The **FILTER** column specifies which records **passed the calling**.

The diagram illustrates the structure of a VCF file, divided into **VCF header** and **Body**.

**Mandatory header lines:** These are the first few lines starting with `##`, such as `##fileformat=VCFv4.0` and `##fileDate=20100707`. They provide metadata about the file.

**Optional header lines (meta-data about the annotations in the VCF body):** These lines start with `#` and provide detailed information about the annotations in the body, such as `#INFO` and `#FORMAT` tags.

**Body:** This section contains the main variant data.

**Annotations for the Body:**

- Reference alleles (GT=0):** The `REF` column, which contains "A" for the first variant and "ACG" for the second.
- Alternate alleles (GT>0 is an index to the ALT column):** The `ALT` column, which contains ".", "C", "T, CT", "G", and "<DEL>" respectively.
- Phased data (G and C above are on the same chromosome):** The `FORMAT` column, which includes "GT:DP", "GT:GQ", and "GT:GQ:DP". The "GT:GQ:DP" entry shows "1|0:77" for the third variant, indicating phase information.
- Sample Data:** The `SAMPLE` columns show genotype calls for two samples: SAMPLE1 and SAMPLE2. For the first variant, the calls are "1/2:13" and "0/0:29". For the second variant, they are "0|1:100" and "2/2:70". For the third variant, they are "1|0:77" and "1/1:95". For the fourth variant, they are "1/1:12:3" and "0/0:20".

**Variant Types:** The variants are categorized as follows:

- Deletion:** The first variant, where the reference allele "A" is missing.
- SNP:** The second variant, where the reference allele "A" is replaced by "C".
- Large SV:** The third variant, where the reference allele "A" is replaced by "G" and "T, CT".
- Insertion:** The fourth variant, where the reference allele "A" is followed by "<DEL>".
- Other event:** The fifth variant, where the reference allele "A" is replaced by "T".

**Annotations for the Variant Data:**

- QUAL:** The quality score for each variant.
- FILTER:** The filter applied to each variant, with "PASS" for all variants shown.
- INFO:** Additional information about the variant, such as "H2:AA=T" for the second variant and "SVTYPE=DEL;END=300" for the third variant.

# VCF format

- **INFO:** List with **further info** about the variations. The fields are **separated by**  
“,”.

The diagram illustrates the structure of a VCF file, divided into two main sections: the **VCF header** and the **Body**.

**Mandatory header lines** (red arrows):

```
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
```

**Optional header lines** (meta-data about the annotations in the VCF body) (black arrow):

```
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
```

**Body** (grey background):

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLE1	SAMPLE2
1	1	.	ACG	A,AT	.	PASS	.	GT:DP	1/2:13	0/0:29
1	2	rsl1	C	T,CT	.	PASS	H2;AA=T	GT:GQ	0 1:100	2/2:70
1	5	.	A	G	.	PASS	.	GT:GQ	1 0:77	1/1:95
1	100		T	<DEL>	.	PASS	SVTYPE=DEL;END=300	GT:GQ:DP	1/1:12:3	0/0:20

Annotations for the Body:

- Deletion**: Points to the first row (ID=rsl1).
- SNP**: Points to the second row (ID=rsl1).
- Large SV**: Points to the fourth row (ID=100).
- Insertion**: Points to the third row (ID=5).
- Other event**: Points to the fifth row (ID=100).

**Reference alleles (GT=0)** (blue arrows):

- GT:DP (row 1)
- GT:GQ (row 2)
- GT:GQ (row 3)
- GT:GQ:DP (row 4)

**Alternate alleles (GT>0 is an index to the ALT column)** (blue arrows):

- GT:DP (row 1)
- GT:GQ (row 2)
- GT:GQ (row 3)
- GT:GQ:DP (row 4)

**Phased data** (G and C above are on the same chromosome):

- GT:DP (row 1)
- GT:GQ (row 2)
- GT:GQ (row 3)
- GT:GQ:DP (row 4)

# VCF format

- **FORMAT** (optional): List with fields for describing the samples.
- **SAMPLEs** (optional): A column per sample with the values defined in FORMAT.

The diagram illustrates the structure of a VCF file, divided into **VCF header** and **Body**.

**Mandatory header lines:** These are at the top of the header section and include meta-data about the file and its source.

**Optional header lines:** These provide meta-data about the annotations in the VCF body.

**Body:** This section contains the variant data.

**Annotations:**

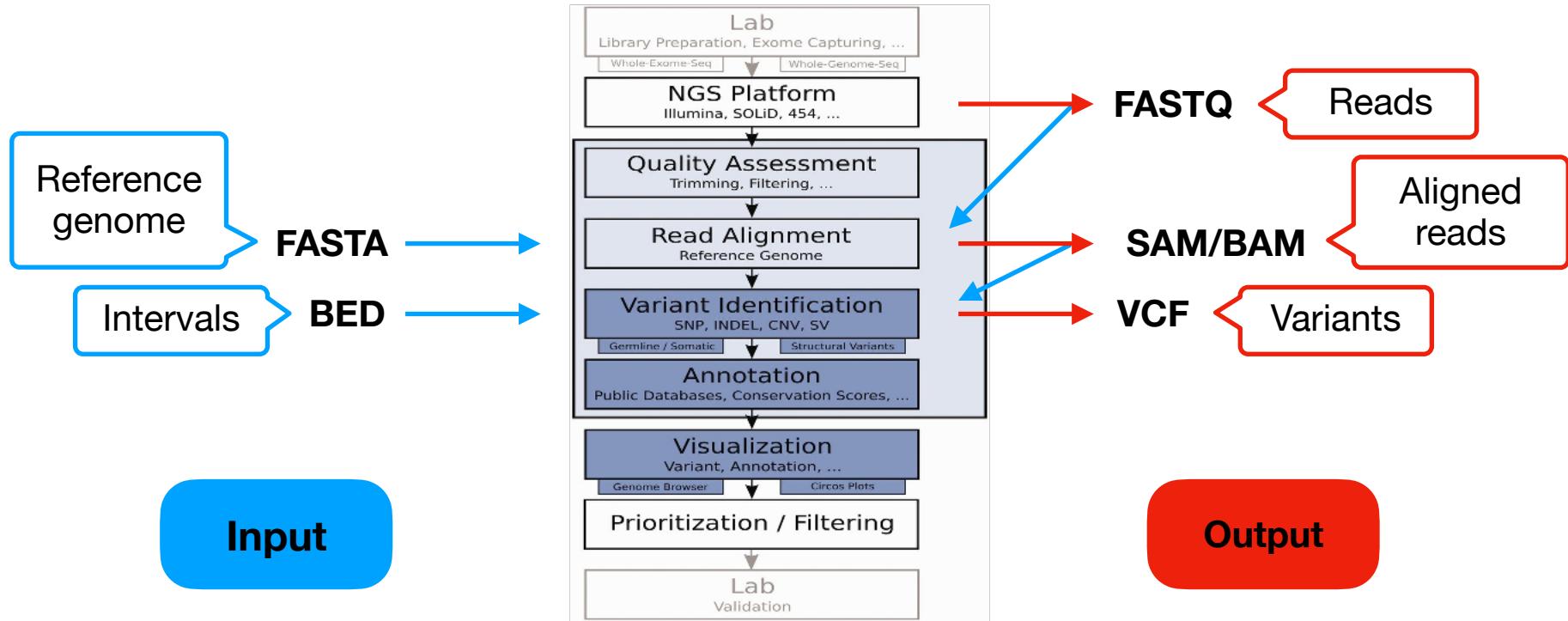
- Deletion:** A variant where the reference allele is missing.
- SNP:** A single nucleotide polymorphism.
- Large SV:** A large structural variation.
- Insertion:** A variant where additional nucleotides are present.
- Other event:** Other types of variants.
- Phased data:** G and C are on the same chromosome.
- Reference alleles (GT=0):** The reference allele for each sample.
- Alternate alleles (GT>0 is an index to the ALT column):** The alternate alleles for each sample.

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLE1	SAMPLE2
1	1	.	ACG	A, AT	.	PASS	.	GT:DP	1/2:13	0/0:29
1	2	rsl1	C	T, CT	.	PASS	H2:AA=T	GT:GQ	0 1:100	2/2:70
1	5	.	A	G	.	PASS	.	GT:GQ	1 0:77	1/1:95
1	100		T	<DEL>	.	PASS	SVTYPE=DEL;END=300	GT:GQ:DP	1/1:12:3	0/0:20

# Data formats cheat sheet

Format	Uses	Example	Example	Software Management	File Extension
Fasta	<p><b>Reference genome</b> Define biological sequences (DNA, RNA, cDNA, proteins).</p>	human_genome.fa	Plain text	samtools, picard-tools	.fasta; .fas; .fa; .fna; .fsa
FastQ	<p><b>Raw sequencing data</b> Single-end sequencing → 1 file Paired-end sequencing → 2 files (R1 and R2 for each end, respectively)</p>	DNAseq_raw_data.fastq (DNAseq_R1.fastq and DNAseq_R2.fastq)	Plain text	samtools, picard-tools <b>Aligners</b>	.fq; .fastq
SAM	<p><b>Define read alignments</b> Store alignment meta-info (reference, methods, one- or multi-sample).</p>	mapped_reads.sam	Plain text	samtools, picard-tools	.sam
BAM	<p><b>Visualize alignments (IGV)</b> The same as SAM, but compressed and indexed. Also to store UNMAPPED reads (compressed).</p>	mapped_reads.bam unmapped_reads.bam	Binary	samtools, bcftools, picard-tools, <b>IGV</b> (Integrative Genome Viewer)	.bam
VCF	<p><b>SNV &amp; Indels calls</b> Indicates genomic variations. Store Variant calling meta-info (reference, methods, one- or multi-sample).</p>	point_variants.vcf	Plain text	bcftools, Unix	.vcf
BED	<p><b>Intervals</b> Delimit genomic regions (i.e. intervals) w/ or w/o annotations.</p>	targeted_regions.bed intervals.bed	Plain text	bedtools, Unix <b>GATK, picard-tools</b>	.bed

# Formats Outline





# Thanks!



CNIO BIOINFORMATICS UNIT

*cnio* stop cancer