NGS Data Analysis Pipeline

Álex Alemán Ramos
Mar 28-30th 2016, Valencia.

Computational • Genomics

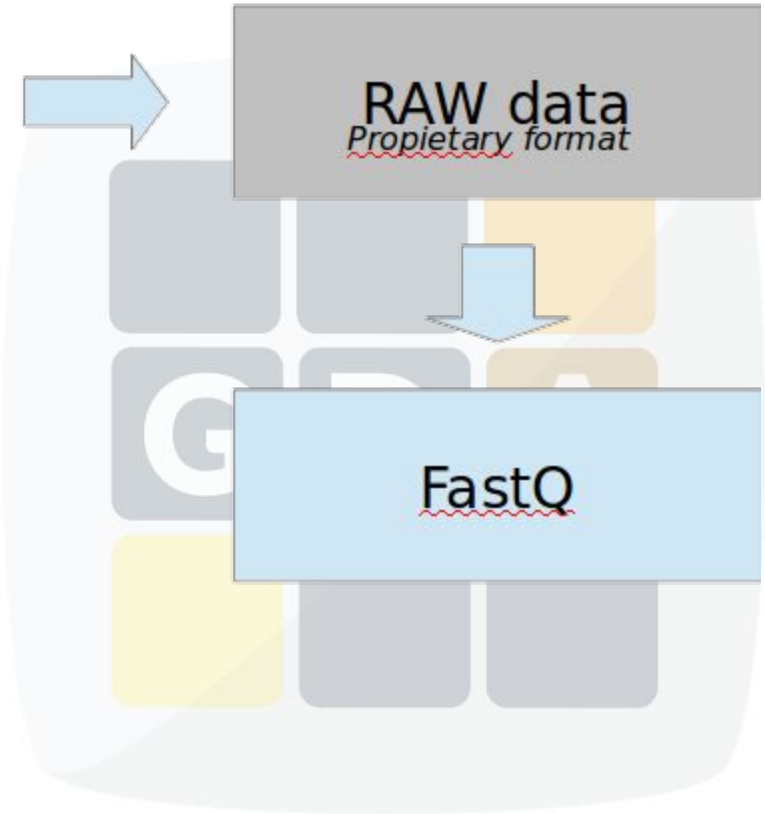
PRINCIPE FELIPE
CENTRO DE INVESTIGACION

Sequence Capture

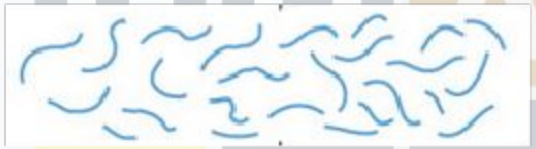
illumina

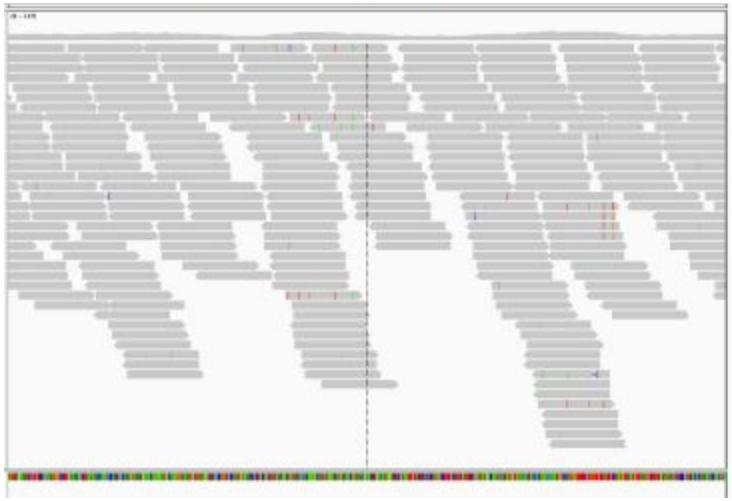
Roche

applied biosystems

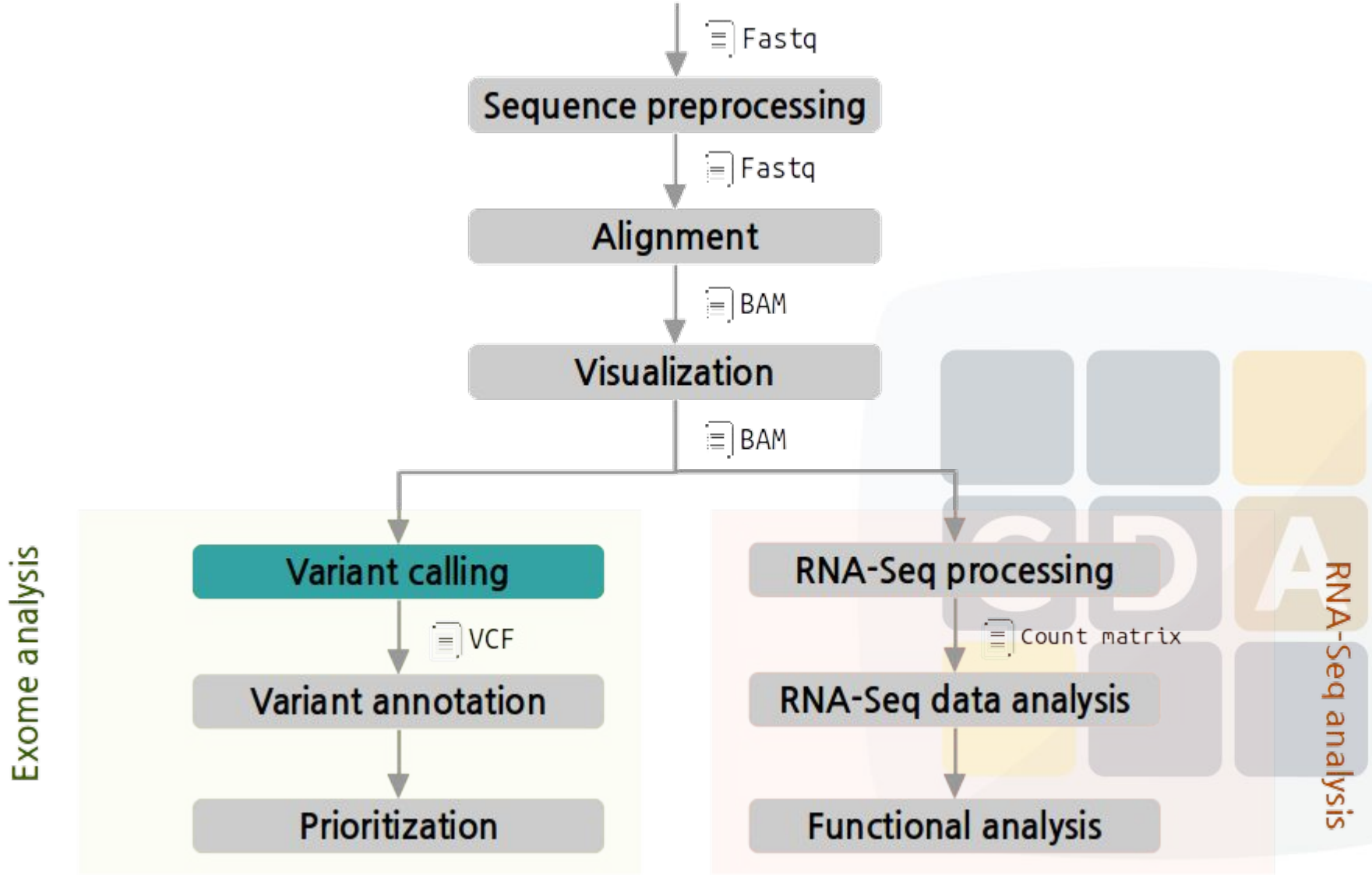


Genome Sequencing

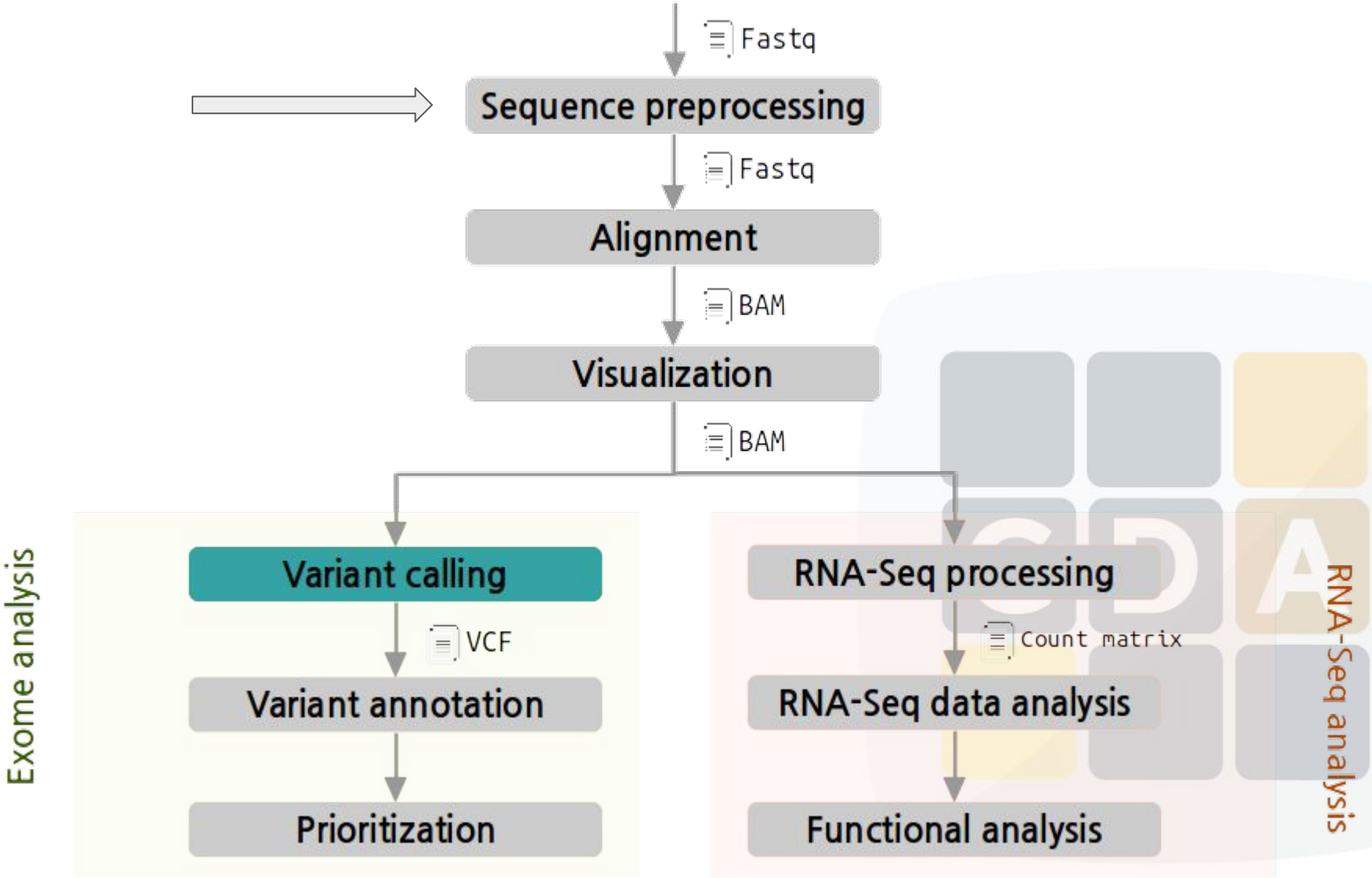




NGS basic pipeline



Where are we?


Fastq
Sequence preprocessing
Fastq
Alignment
BAM
Visualization
BAM
Exome analysis
Variant calling
VCF
Variant annotation
Prioritization
RNA-Seq analysis
RNA-Seq processing
Count matrix
RNA-Seq data analysis
Functional analysis

Contents

- Data formats
  - Sequence capture
  - Fasta and fastq formats
  - Sequence quality encoding
- Quality Control
  - Evaluation of sequence quality
  - Quality control tools
  - Identification of artifacts & filtering

From sequencers to digital data

- **What structure does the data have?**
  - Text-based formats (easy to use!)
  - If not compressed, it can be huge

- **Different data formats:**
  - Different sequencers output different files (sff, fasta, csfasta, qual file, fastq...)
  - There are some data formats widely accepted (e.g. FastQ format)

Fasta format

- Two lines per sequence:
  - 1. Header lines starts with “>” followed by a sequence ID
  - 2. Sequence (string of nt or peptides)

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
IENY
```

```
>BBTBSCRYR
tgcaccaaacatgtctaaagctggaaccaaaattactttctttgaagacaaaaactttca
aggccgccactatgacagcgattgcgactgtgcagatttccacatgtacctgagccgctg
caactccatcagagtggaaggaggcacctgggctgtgtatgaaaggcccaattttgctgg
gtacatgtacatcctaccccggggcgagtatcctgagtaccagcactggatgggcctcaa
```

- Typical file extensions (.fasta, .fa, .fna, .fnn, .faa, ...)

Fastq format

- We could say “it is a fasta with **qualities**”:
  - 1. Header (like the fasta but starting with “@”)
  - 2. Sequence (string of nt)
  - 3. “+” and sequence ID (optional)
  - 4. Encoded quality of the sequence

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Quality codification

- Phred quality score
  - Error probability
  - ASCII encoded

- Phred +33
  - Sanger [0,40]
  - Illumina 1.8 [0,41]
  - Illumina 1.9 [0,41]

- Phred +64
  - Illumina 1.3 [0,40]
  - Illumina 1.5 [3,40]

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		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

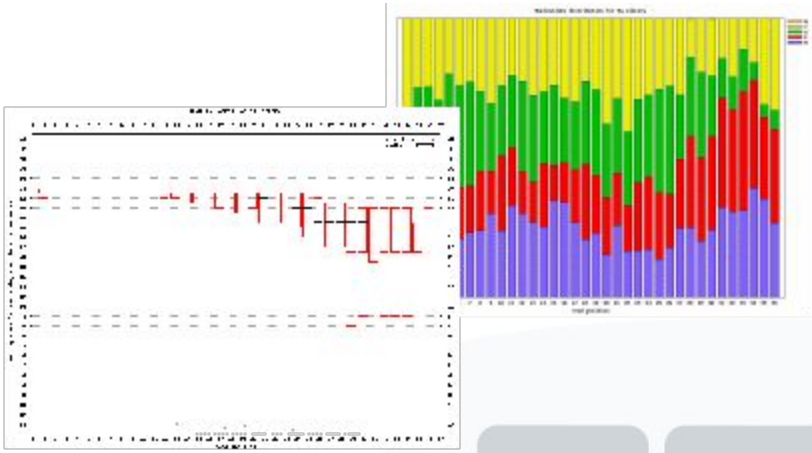
Contents

- Data formats
  - Sequence capture
  - Fasta and fastq formats
  - Sequence quality encoding

- Quality Control
  - Evaluation of sequence quality
  - Quality control tools
  - Identification of artifacts & filtering

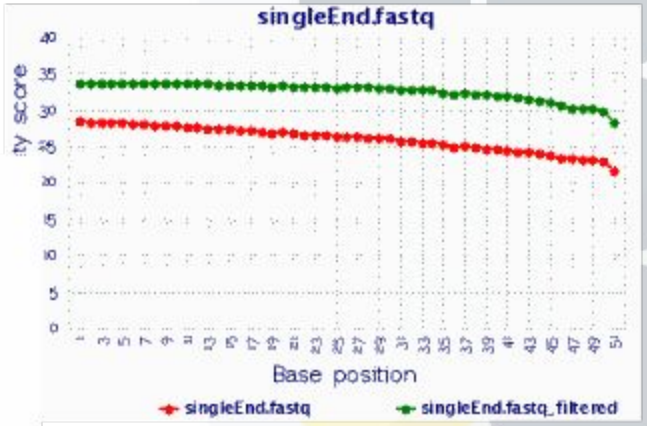
Sequence quality evaluation

- Quality control tools:
  - **Fastx-toolkit**



http://hannonlab.cshl.edu/fastx_toolkit/download.html

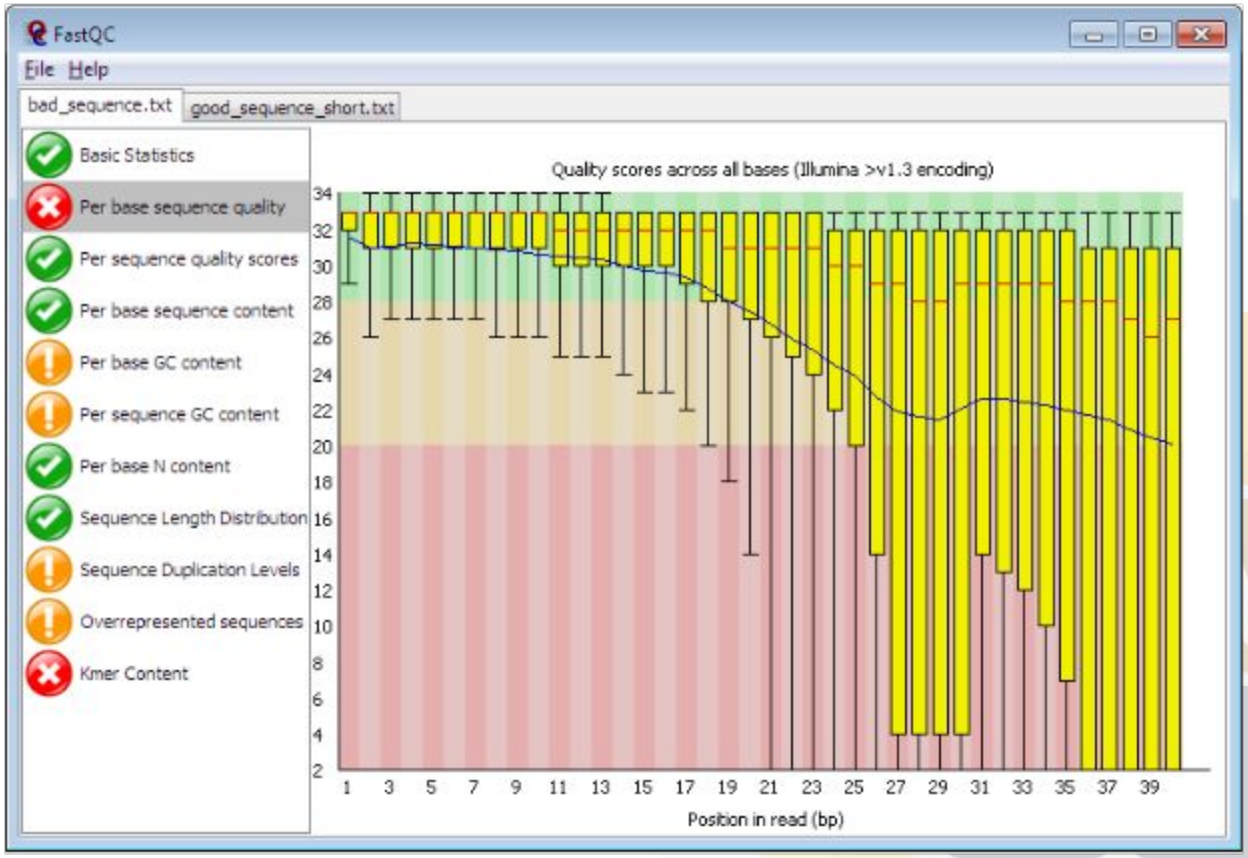
  - **NGS QC Toolkit**



http://www.nipgr.res.in/ngsqctoolkit.html

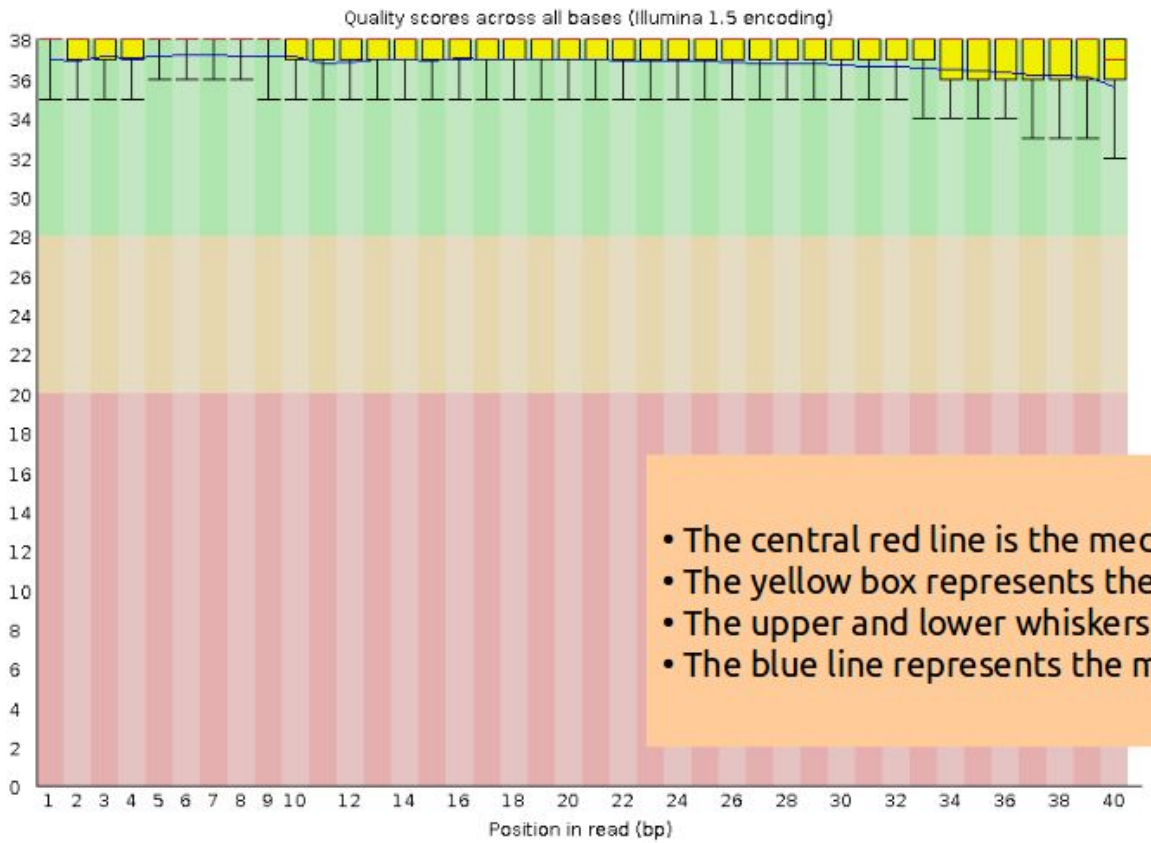
Sequence quality evaluation

- Other quality control tool: **FastQC**



http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

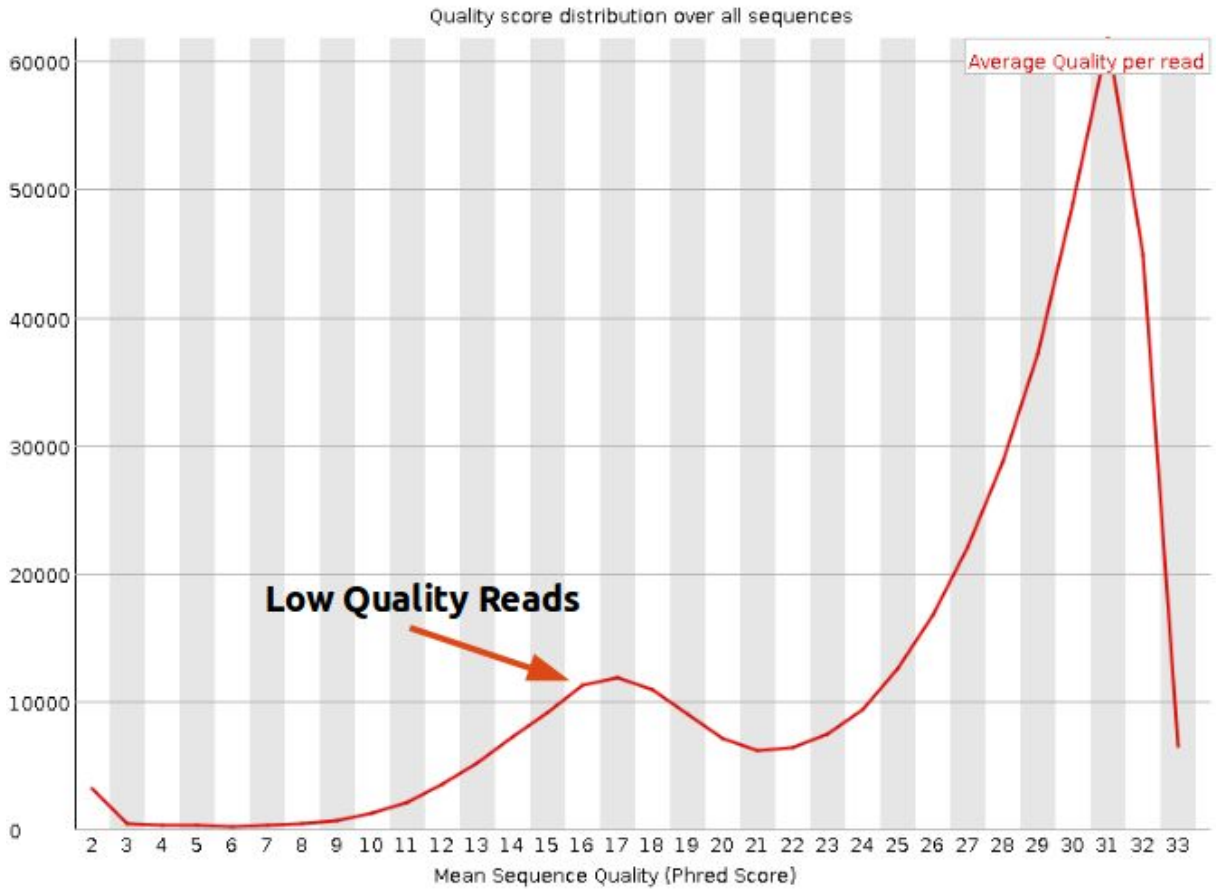
Sequence quality per base position



- **Good data**
  - Consistent
  - High quality along the read

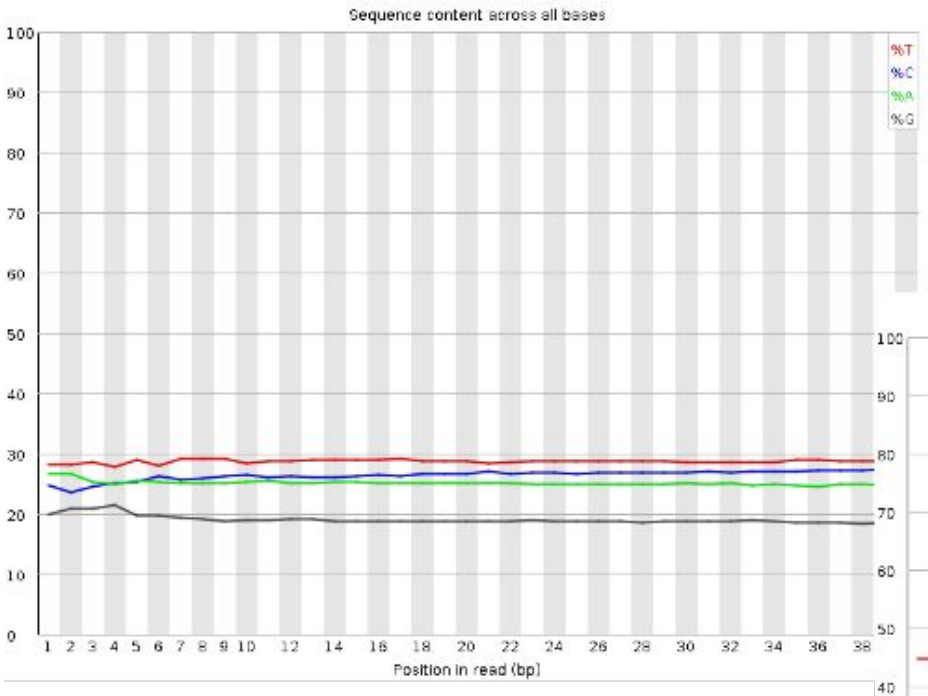
- The central red line is the median value
- The yellow box represents the inter-quartile range (25-75%)
- The upper and lower whiskers represent the 10% and 90% points
- The blue line represents the mean quality

Per sequence quality distribution

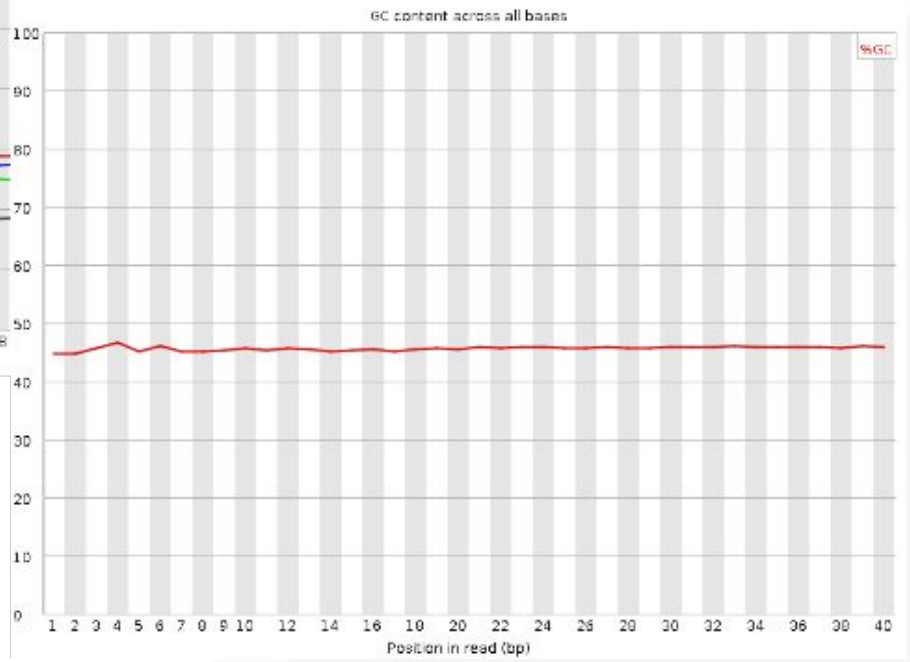


- **Bad data**
  - Non-uniform distribution

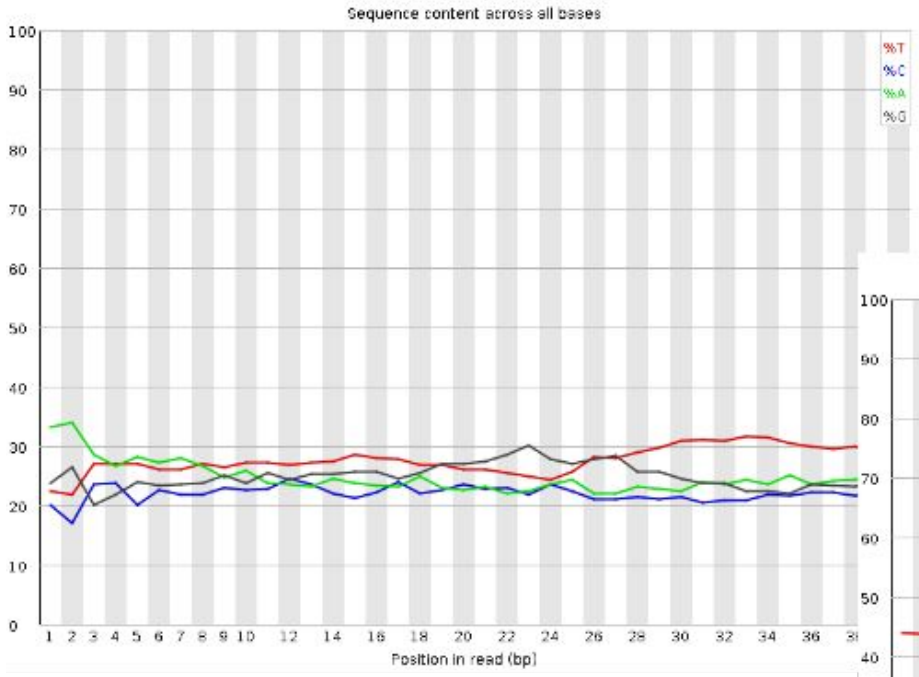
Per base sequence content



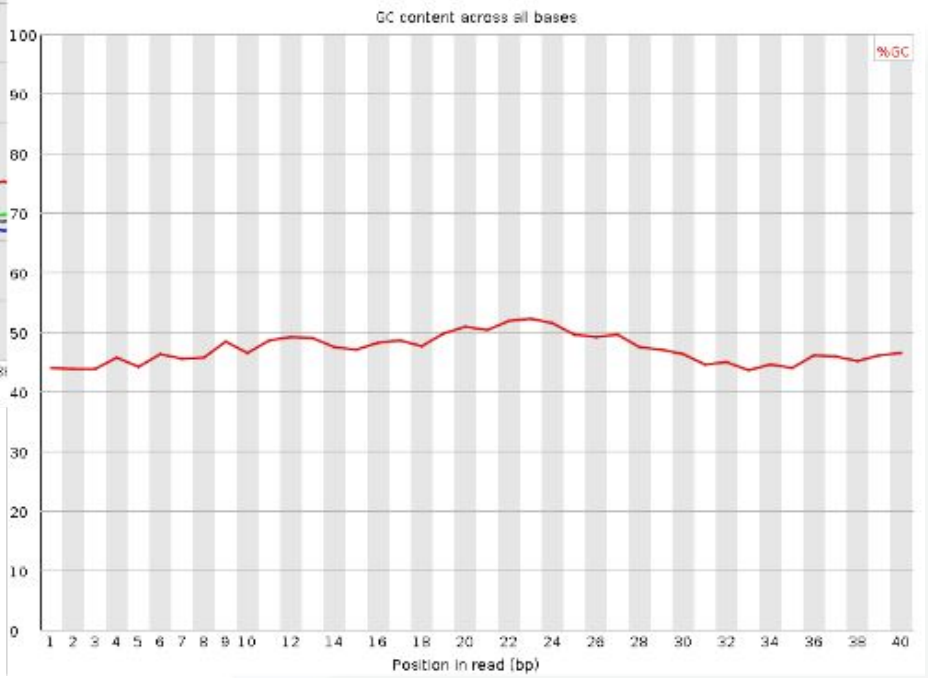
- **Good data**
  - Smooth over length
  - Organism dependent (GC)



Per base sequence content

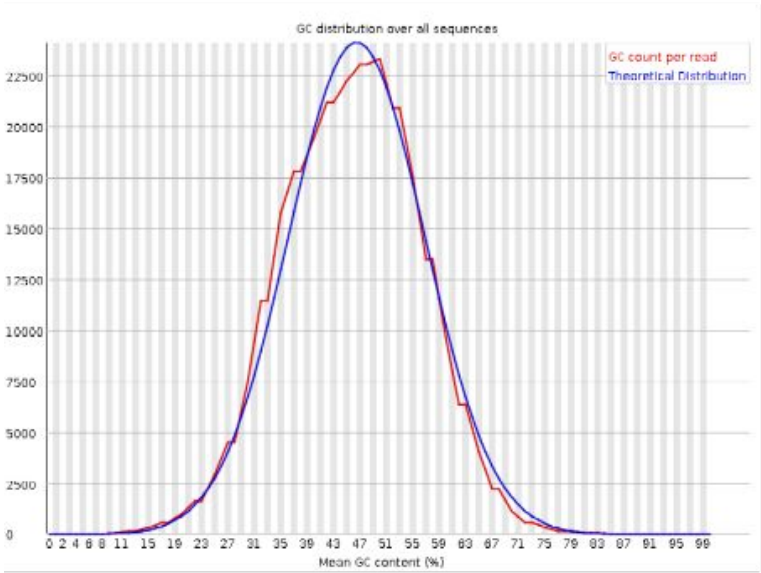


- **Bad data**
  - Sequence position bias

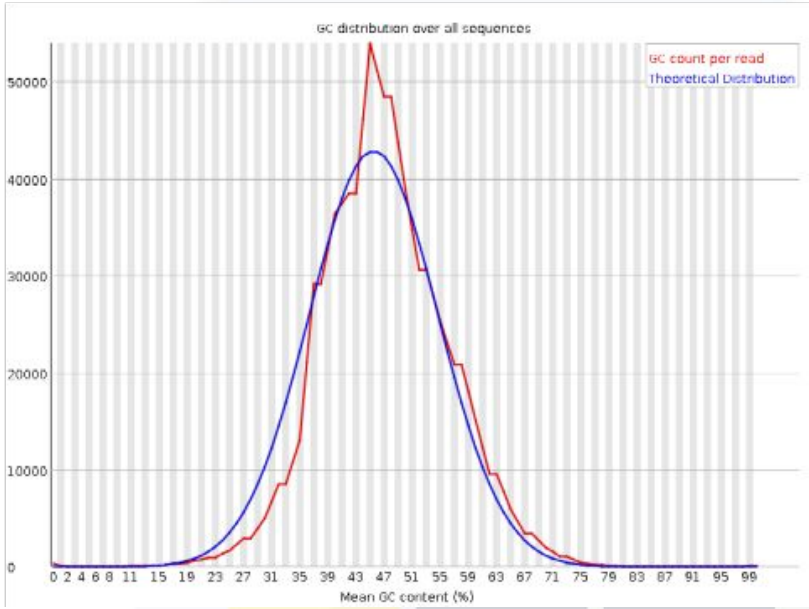


Per sequence GC content

- **Good data**
  - Fits with expected
  - Organism dependent

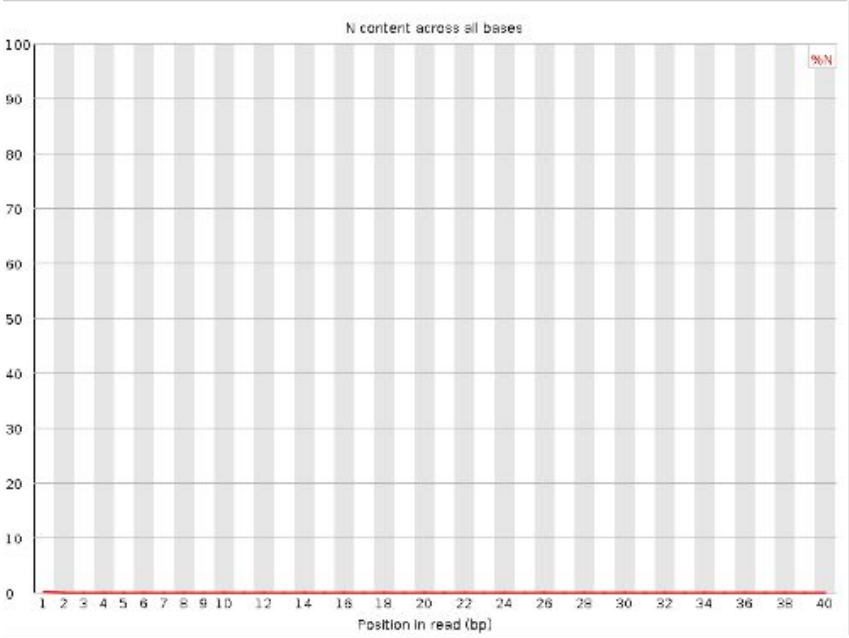


- **Bad data**
  - Does not fit with expected
  - Library contamination?

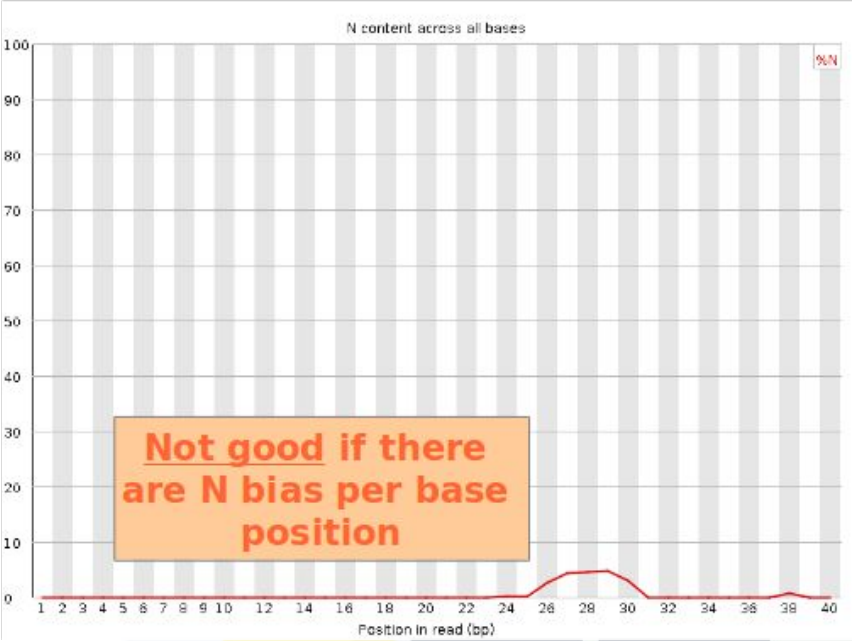


Per base N content

- **Good data**

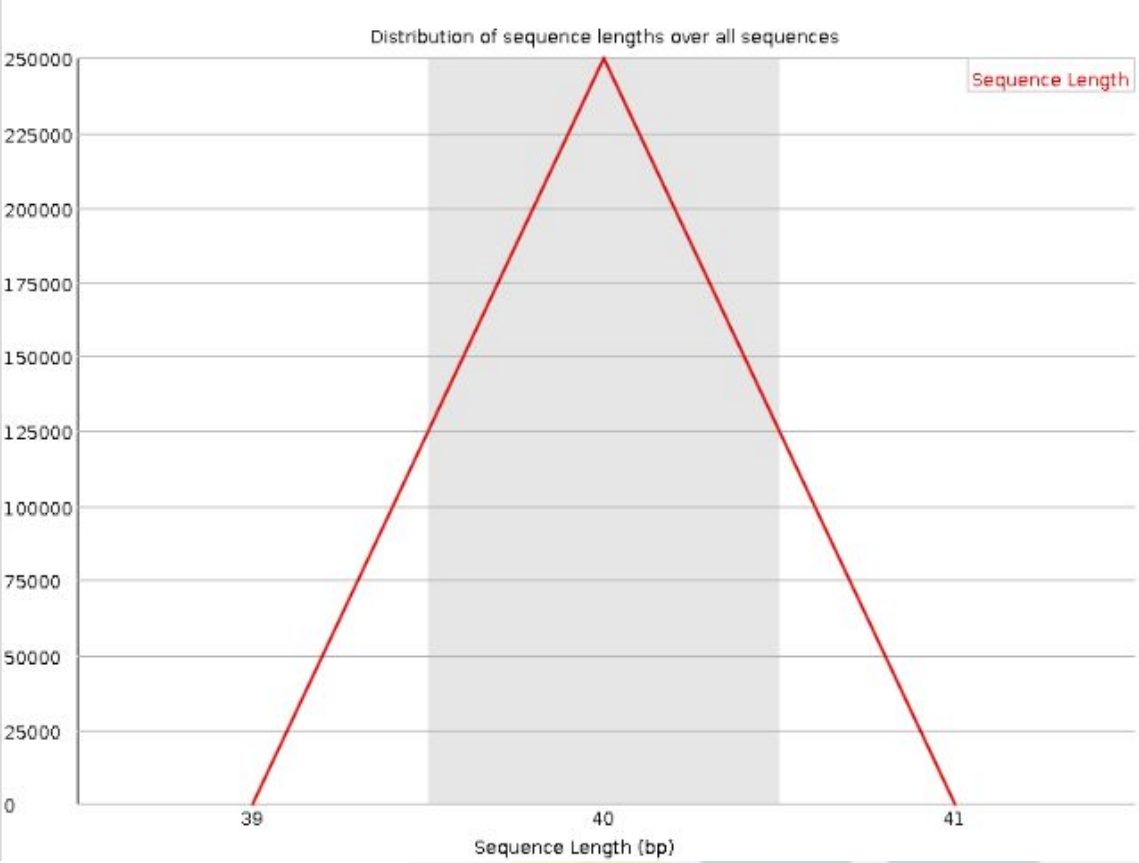


- **Bad data**



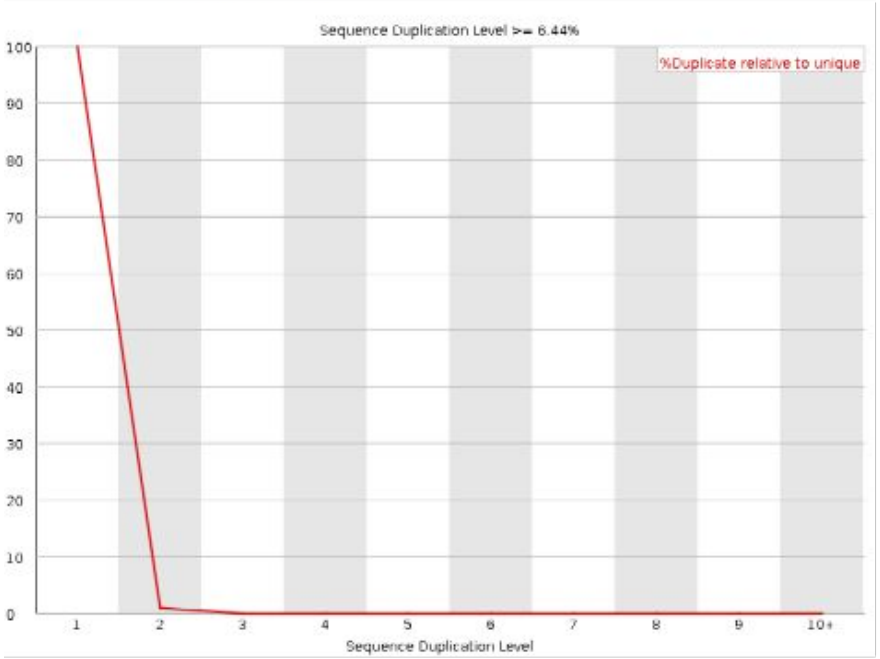
Sequence length distribution

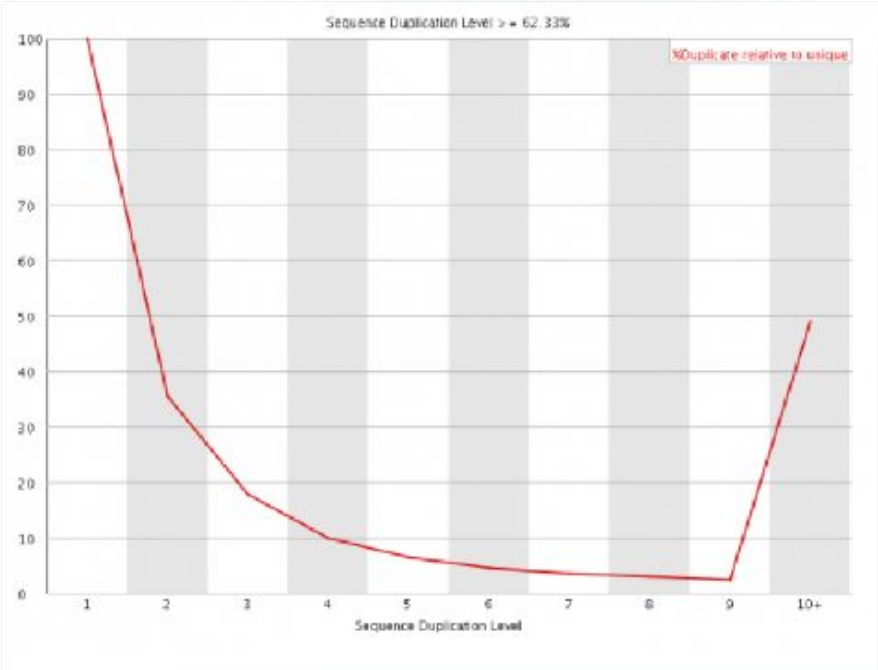
- Just descriptive:
  - Some sequencers output sequences of different length (e.g. 454)



Sequence duplication levels

- In **Transcriptomics**, you expect higher number of duplicated sequences.
- In **Genomics** you should be worried if this happens → PCR artifact?





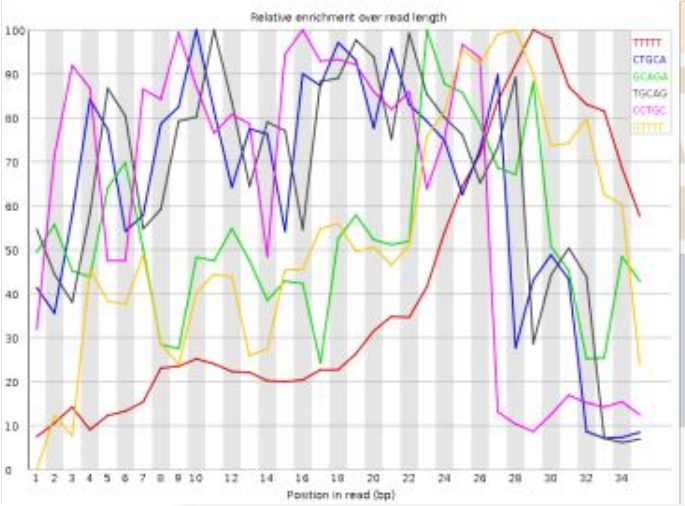
Overrepresented sequences & Kmer content

- Question:
  - If we obtain the exact same sequences too many times
    → **Do we have a problem?**

- Answer:
  - **Sometimes !**

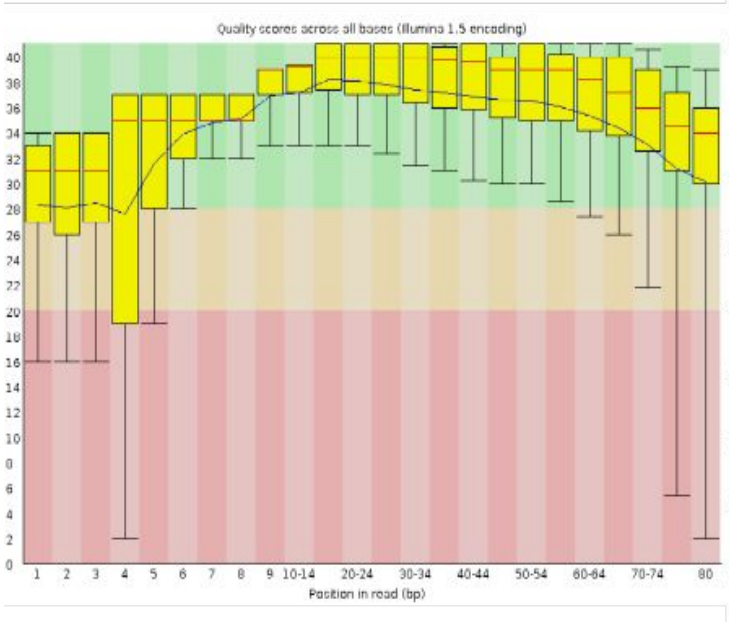
- Examples:
  - PCR primers, adapters ...

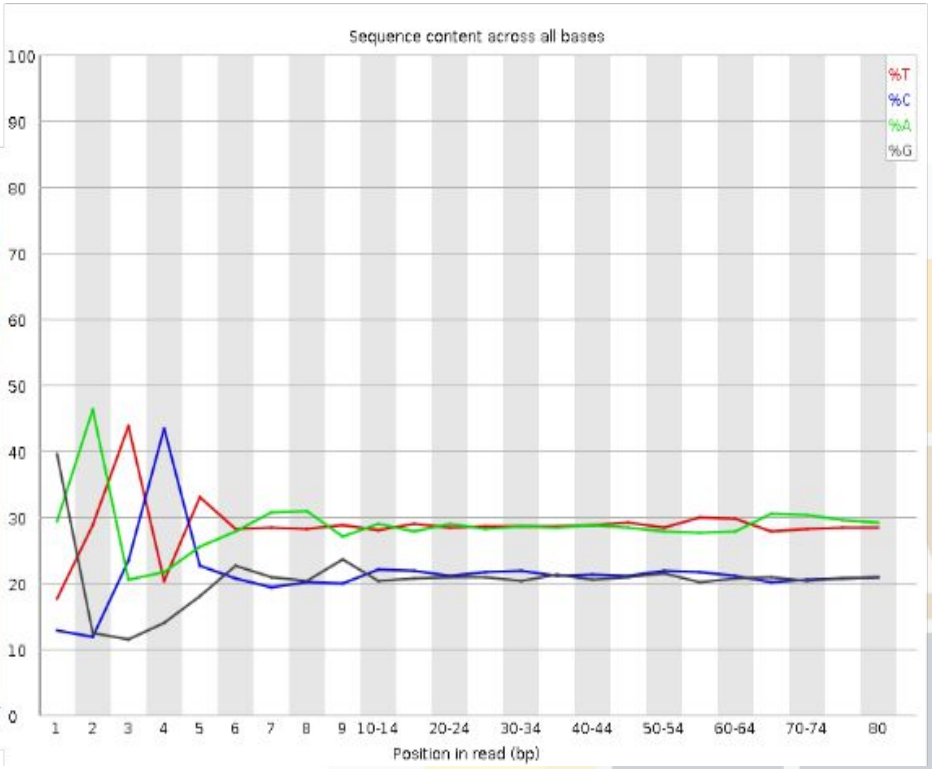
Sequence	Count	Percentage
AGAGTTTTATCGCTTCCATGACGC AGAAGTTAACACTTTC	2065	0.5224039181558763
GATTGGCGTATCCAACCTGCAGA GTTTTATCGCTTCCATG	2047	0.5178502762542754



Typical artifacts

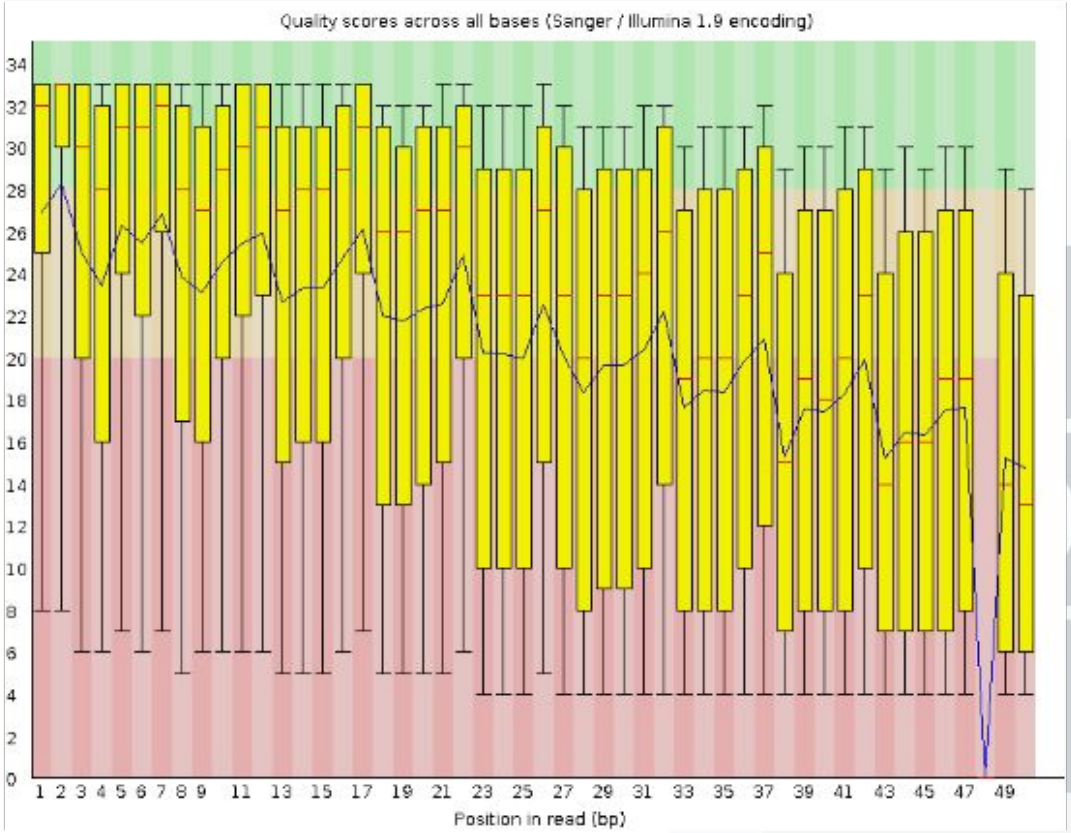
- Sequence adapters



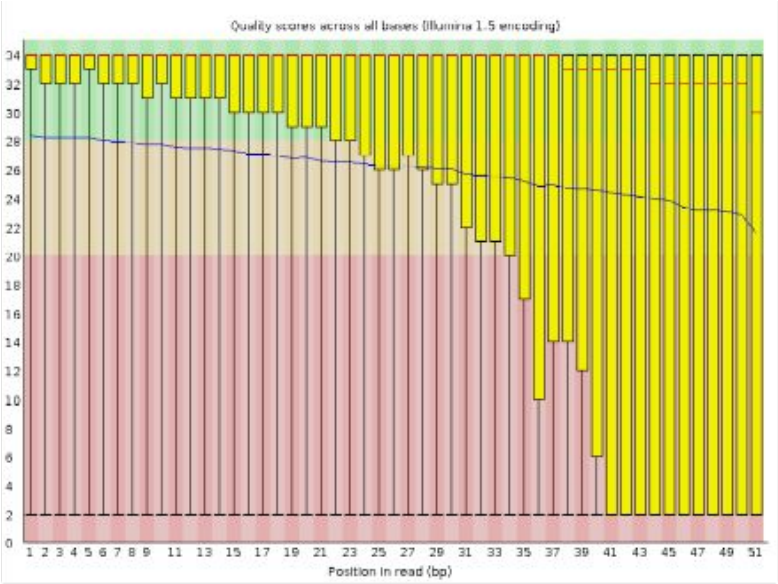


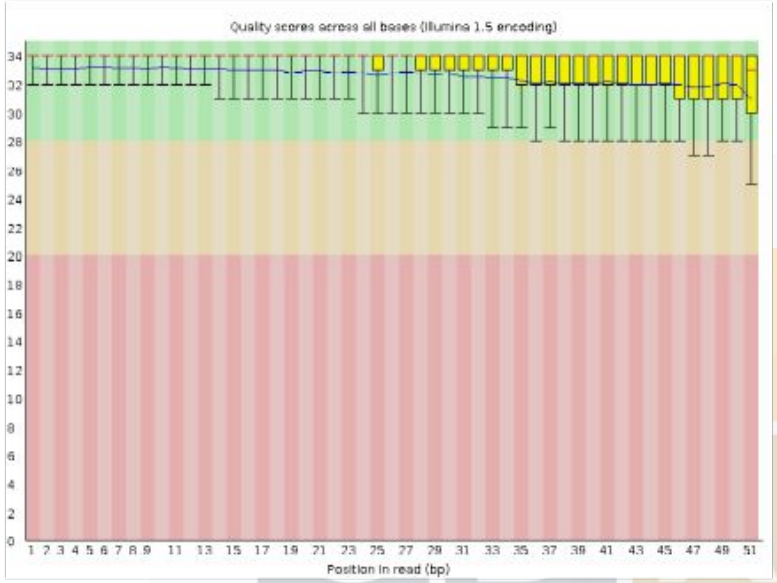
Typical artifacts

- Platform dependent



Improving sequence quality

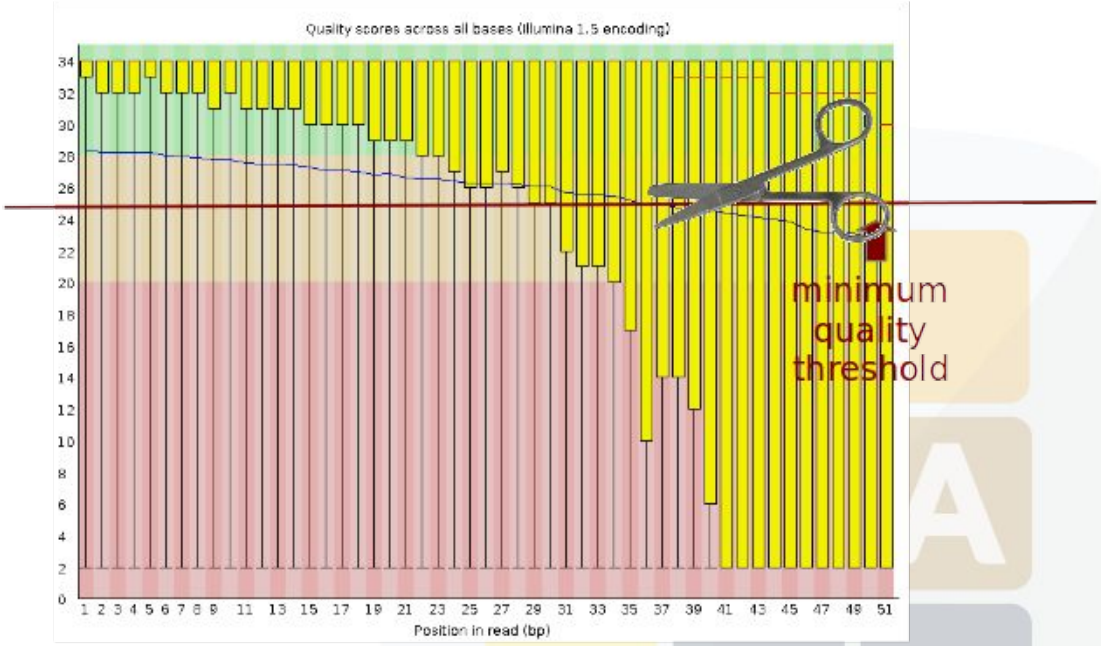




- Removing bad quality data will improve our confidence on downstream analysis

Improving sequence quality

- Sequence filtering
  - Mean quality
  - Read length
  - Read length after trimming
  - Percentage of bases above Q
  - Adapter trimming
  - Adapter reads



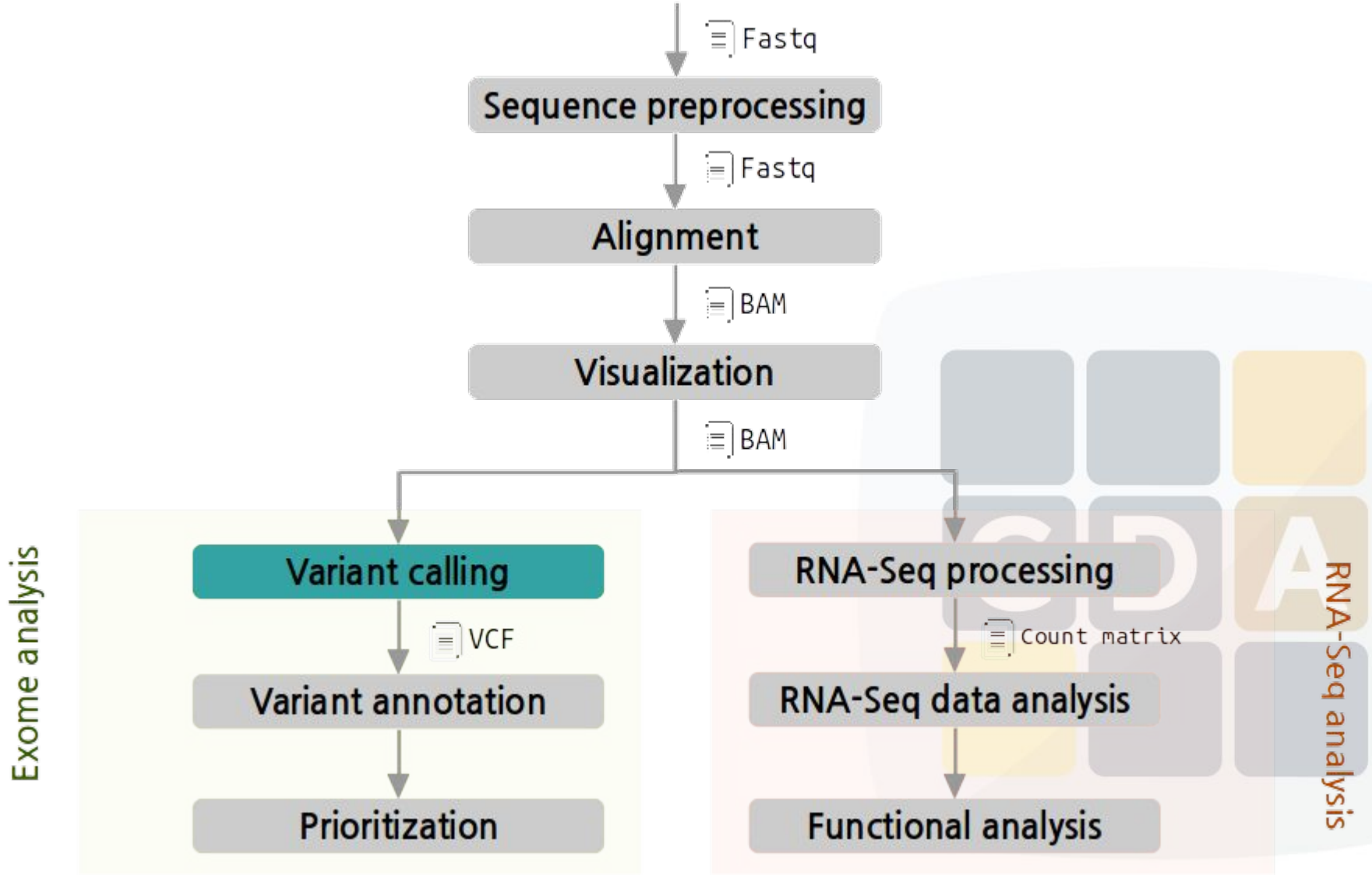
Improving sequence quality

- Sequence filtering tools
  - Fastx-toolkit
  - Galaxy (https://main.g2.bx.psu.edu/)
  - SeqTK (https://github.com/lh3/seqtk)
  - Cutadapt (http://code.google.com/p/cutadapt/)
  - Trimmomatic(http://www.usadellab.org/cms/?page=trimmomatic)
  - And more….

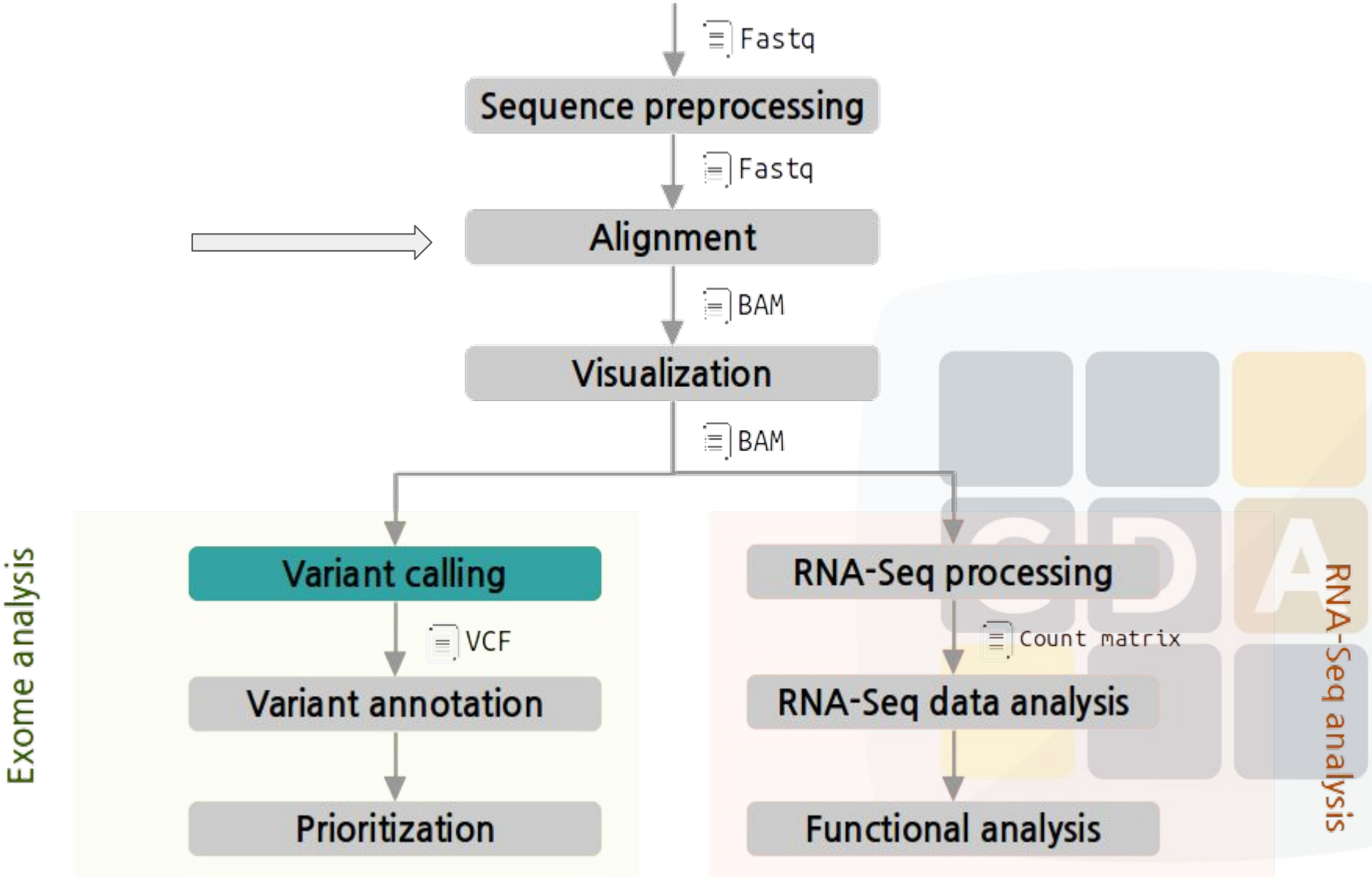
Final remarks

- After preprocessing sequences, it is important to **evaluate the quality for raw data**
- **Fastq is the standard format** for NGS raw data. This format includes a quality score for each position
- NGS Genomic Data Analysis Pipeline produces a **control quality report** to control quality for sequences

NGS basic pipeline



Where are we?


Fastq
Sequence preprocessing
Fastq
Alignment
BAM
Visualization
BAM
Exome analysis
Variant calling
VCF
Variant annotation
Prioritization
RNA-Seq analysis
RNA-Seq processing
Count matrix
RNA-Seq data analysis
Functional analysis

Contents

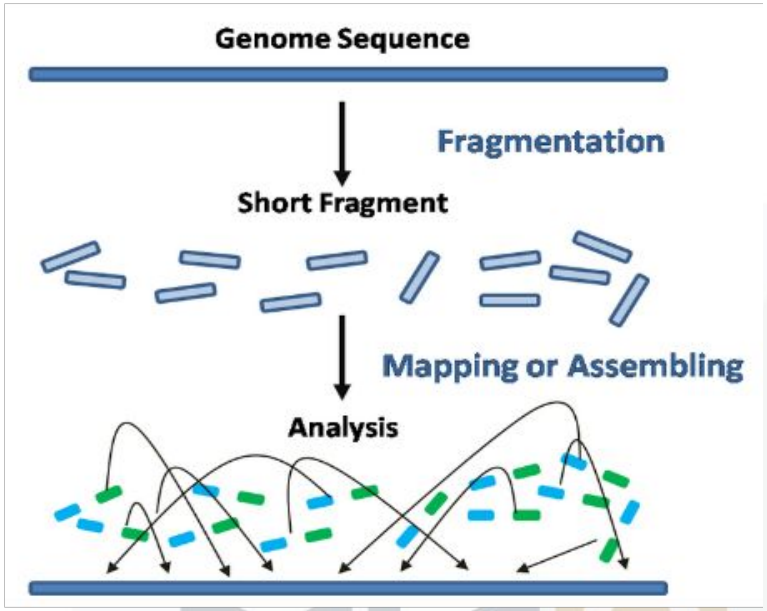
- Introduction
- Algorithms and Tools
- SAM/BAM specification
- Visualization
- Best practices
- Data repositories

Introduction

- Current read sizes ranging from 75-800bp, up to 15kb coming soon
- Single-end and paired-end reads
- Sequencing errors, low quality reads, duplicated reads
- Analysis pipelines: Exome vs Genome sequencing, RNA-seq (transcriptomics), BS-seq, ChIP-seq, ...
- Illumina **HiSeq 2500** provides high-quality 2x125bp: 176Gb in 40h, 90.2% bases above Q30
  - Human genome 3Gb ~ 60x coverage
  - Each sample produces a *fastq* file ~500GB size containing ~550M reads
- New **Illumina X Ten**: Consists if ten ultra-high-throughput sequencers. First $1000 human genome sequencer. Produces 18.000 genomes per year
- Mapping goes from FASTQ to SAM/BAM files

Aligning reads, the challenges

- Mapping reads onto a **reference genome**, a simple concept but there are some ***challenges***:
  - *Natural variability*: SNPs, *de novo* mutations, INDELS, copy number, translocations, …
  - *Repetitive regions*
  - *Sequencing errors*
  - *RNA-seq*: gapped alignment
  - *BS-seq*: C → T conversion strategy
  - *High computing resources needed: multicore CPUs and a lot of RAM*
- We must deal with genomic variation in an efficient way



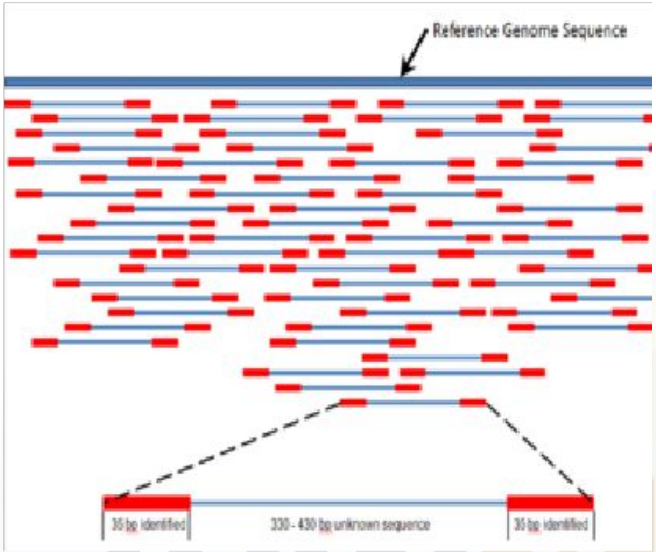
Getting a reference genome

- A ***reference genome*** is a consensus sequence built up from high quality sequencing samples from different populations. It is the control reference sequence to compare our samples
- ***Genome Reference Consortium (GRC)*** created to deliver assemblies:
  - http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/
- Current human assembly is ***GRCh38***
- Reference genomes can be downloaded from:
  - ***GRC***: Human genome available at: ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/GRCh37/Primary_Assembly/assembled_chromosomes/FASTA/
  - ***Ensembl***: many available vertebrates genomes http://www.ensembl.org/info/data/ftp/index.html
  - ***Ensembl Genomes***: http://ensemblgenomes.org/

The mapping process considerations

- Considerations:
  - Which tool to use? What am I looking for? SNVs? INDELS? Long reads?
  - Is it DNA or RNA?
  - Single-end or paired-end? Paired-end when:
    - For very short reads, reduce the number of false positives alignments
    - Re-sequencing projects, Rna-seq?
    - Am I interested in Structural variation or gene fusions?
    - Reduce number of false positive variants
  - Should I allow multiple hits?
  - Should I remove low quality reads always?

- In general for *genomic variant analysis* we need high quality reads, paired-end datasets work better, and **no** multiple hits must be allowed



Desirable features of an aligner

- Goals
  - **Sensitivity**, we are looking for genomic variants, reads with mismatches and INDELS must be properly aligned
  - **Specificity**, no wrong alignments should be provided
  - Being able to perform gapped alignments (RNA), exones must be correctly located
  - Good performance, efficiency matters
  - Easy to use
  - Open-source and maintained
  - Capable of align different data types: DNA, RNA-seq, BS-seq, …
- Unfortunately... most tools or algorithms only work well in a specific scenario

Algorithms/tools: Smith-Waterman

SW finds the optimal local alignment between:

Sequence 1 = ACACACTA
Sequence 2 = AGCACACA

Given gap-scoring penalties:

w(match) = +2
w(a,-) = w(-,b) = w(mismatch) = -1

$$H = \begin{pmatrix} & - & A & C & A & C & A & C & T & A \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 2 \\ G & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ C & 0 & 0 & 3 & 2 & 3 & 2 & 3 & 2 & 1 \\ A & 0 & 2 & 2 & 5 & 4 & 5 & 4 & 3 & 4 \\ C & 0 & 1 & 4 & 4 & 7 & 6 & 7 & 6 & 5 \\ A & 0 & 2 & 3 & 6 & 6 & 9 & 8 & 7 & 8 \\ C & 0 & 1 & 4 & 5 & 8 & 8 & 11 & 10 & 9 \\ A & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & 12 \end{pmatrix}$$

Alignment result:

Sequence 1 = A - C A C A C T A
Sequence 2 = A G C A C A C - A

- Very popular algorithm developed in 1981
- Provides a very **high sensitivity,** allowing alignments with any number of mismatches, insertions and deletions
- Gives an *optimal alignment* between two sequences given a penalties, **it is not a mapper but an sequence aligner**
- No suitable for whole genome alignment: for a 100bp read and the human genome 3Gb, the matrix dimension: 100×3.10^9, using 4 Bytes for integers: **1.2TB of RAM !!**
- Although *dynamic programming* techniques are applied to make SW more efficient, the CPU requirements are still too high, **SW is too slow for NGS**

Algorithms/tools: BLAST

Basic Local Alignment Search Tool

- BLAST is one of the most widely used programs in Bioinformatics developed in 1990 at NIH. Allows comparing and searching amino-acid and DNA sequences in a database of sequences
- BLAST uses a heuristic algorithm to speed-up searches, it is **much faster** than calculating an optimal alignment with Smith-Waterman, **but it cannot guarantee the optimal alignment** of the query sequence in the database. It searches the most relevant *seeds* from query sequence in exact way and then SW is applied
- It presents a **high sensitivity,** allowing alignments with any number of mismatches, insertions and deletions, it can be used to align sequence between species
- However, it is **still too slow** for NGS mapping, blast can align few thousands sequences per hour

Algorithms/tools:

Burrows-Wheeler Transform (BWT) algorithm

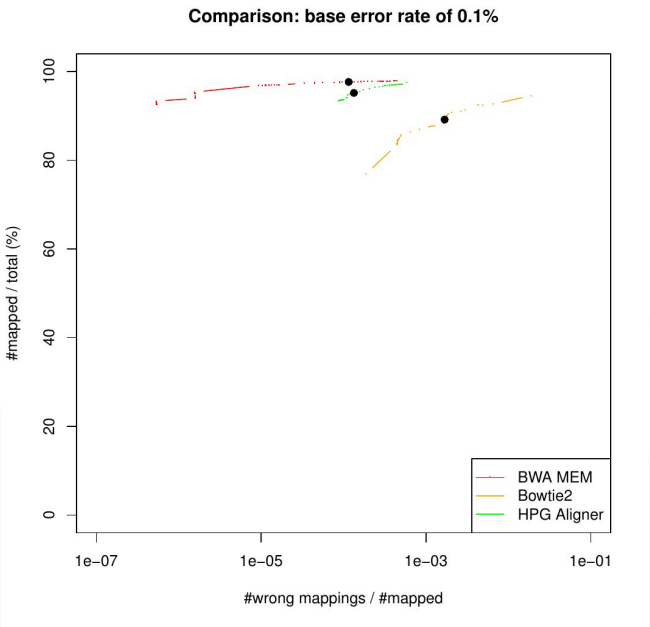
- BWT is an algorithm used in data compression techniques such as *bzip2*
- It **efficiently** align short sequencing reads against a large reference sequence such as the human genome, a **prefix tree index** is created using reference genome
- In the transformation all permutations are sorted and all suffixes are grouped
- It is **much faster** than BLAST, it can align hundred of thousands sequences per second!
- However, it presents a **lower sensitivity**, it can allow a few mismatches, and in some implementation one INDEL

$$
\begin{array}{c}
\\
\\
\begin{array}{c} 0\ 1\ 2\ 3\ 4\ 5\ 6 \\ R = \text{“}A\ G\ G\ A\ G\ C\ \$\text{”} \end{array}
\end{array}
\begin{array}{c}
0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6
\end{array}
\begin{pmatrix}
\$ & A & G & G & A & G & C \\
A & G & C & \$ & A & G & G \\
A & G & G & A & G & C & \$ \\
C & \$ & A & G & G & A & G \\
G & A & G & C & \$ & A & G \\
G & C & \$ & A & G & G & A \\
G & G & A & G & C & \$ & A
\end{pmatrix}
\begin{array}{c}
6 \\ 3 \\ 0 \\ 5 \\ 2 \\ 4 \\ 1
\end{array}
$$

Algorithms/tools:

Many aligners available, which to use?

- Many aligners available, more than 70!!
  - http://wwwdev.ebi.ac.uk/fg/hts_mappers/
- Can be difficult to select one, some criteria
  - Type of analysis: dna, rna, meth
  - Number of cites
  - ...
- **Selecting an aligner**: simulate datasets to choose the best:
  - Which one is more sensitive to INDELS?
  - Which produce less false positives alignments
  - Which RNA aligner works better with low coverage?
  - ...
- All of them work similarly
  - **Reference genome index**: this index can be a Burrows-Wheeler Transform (BWT), Suffix array (SA), ...
  - The reads are **aligned to that index or are split in seeds an then aligned**, seeds aligned are clustered together
  - In general poor performance when high number of mismatches or INDELS are present



Algorithms/tools:

DNA: BWA, BWA-SW and BWA-MEM

- BWA stands from Burrows-Wheeler Aligner, developed by R. Durbin at Sanger Institute
  - http://bio-bwa.sourceforge.net/
- It was one of the first NGS mappers and is widely used, provides very good results in common scenarios
- It implements BWT and Suffix Arrays (SA) with support for few errors:
  - *BWA-SW and BWA-MEM both tolerate more errors given longer alignment. Simulation suggests that they may work well given 2% error for an 100bp alignment, 3% error for a 200bp, 5% for 500bp and 10% for 1000bp or longer alignment*
- Implementation is in C and it is multi-thread, but lacks some features such as support for RNA-seq or big INDELS
- Not designed to take advantage of new technologies and clusters, not specially fast

Algorithms/tools:

DNA: Bowtie and Bowtie2

- Bowtie allowed a few mismatches (<3) and no gaps, claimed to be the fastest, but it missed many reads
- Bowtie2 improved sensitivity when compared to Bowtie:
  - http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
- Widely used, however it is a little bit less sensitivity than BWA, fail to correctly map many mismatches and INDELS
- Implementation is in C and it is multi-thread, but lacks some biological features such as support for RNA or big INDELS
- Not designed to take advantage of new technologies and clusters

Algorithms/tools:

RNA-seq: TopHat, the standard RNA-seq aligner

- TopHat is the standard for RNA-seq mapping
  - http://tophat.cbcb.umd.edu/
- It uses Bowtie2 to align reads, so it is not very sensitive, usually maps 75% of reads
- Not ready for long reads (>150bp), mapping decrease to below 50%
- Poor performance, can take several hours to map
- Big memory footprint and a lot of disk used
- Mapping fall down with mismatches, INDELS and longer reads
- Written in Python and C. Not designed to take advantage of new technologies and clusters

Algorithms/tools:

RNA-seq: STAR and MapSlice

- STAR developed for ENCODE project
  - https://code.google.com/p/rna-star/
  - High-performance, not very high sensitivity
- MapSplice
  - http://www.netlab.uky.edu/p/bioinfo/MapSplice2
  - Not bad sensitivity but very slow

Algorithms/tools:

Meth: Bismark, a BS-seq mapper

- Bismark can map BS-seq data:
  - http://www.bioinformatics.babraham.ac.uk/projects/bismark/
- It uses Bowtie2 for mapping
- Sensitivity and performance very poor
- Written in Perl and Python. Not designed to take advantage of new technologies and clusters

SAM/BAM specification

Mapping output: SAM/BAM format

SAM Specification: http://samtools.sourceforge.net/SAM1.pdf

Header

```
@HD    VN:1.0  SO:coordinate
@SQ    SN:chr1 LN:249250621
@PG    ID:TopHat    VN:2.0.8    CL:/opt/soft/ngs/tophat/tophat-2.0.8.Linux_x86_64/tophat -p 4 -o
    /clinicfs/projects/3.ENCODE/mappings/Gm12878/Gm12878_Rp1_pair --no-coverage-search -r 300 --mate-std-dev 200 --
    library-type fr-unstranded /clinicfs/common/reference-genomes/homo_sapiens/bt2/hg19_ucsc/hg19_ucsc
    /clinicfs/projects/3.ENCODE/reads/Gm12878_Rp1_1.fastq /clinicfs/projects/3.ENCODE/reads/Gm12878_Rp1_2.fastq
```

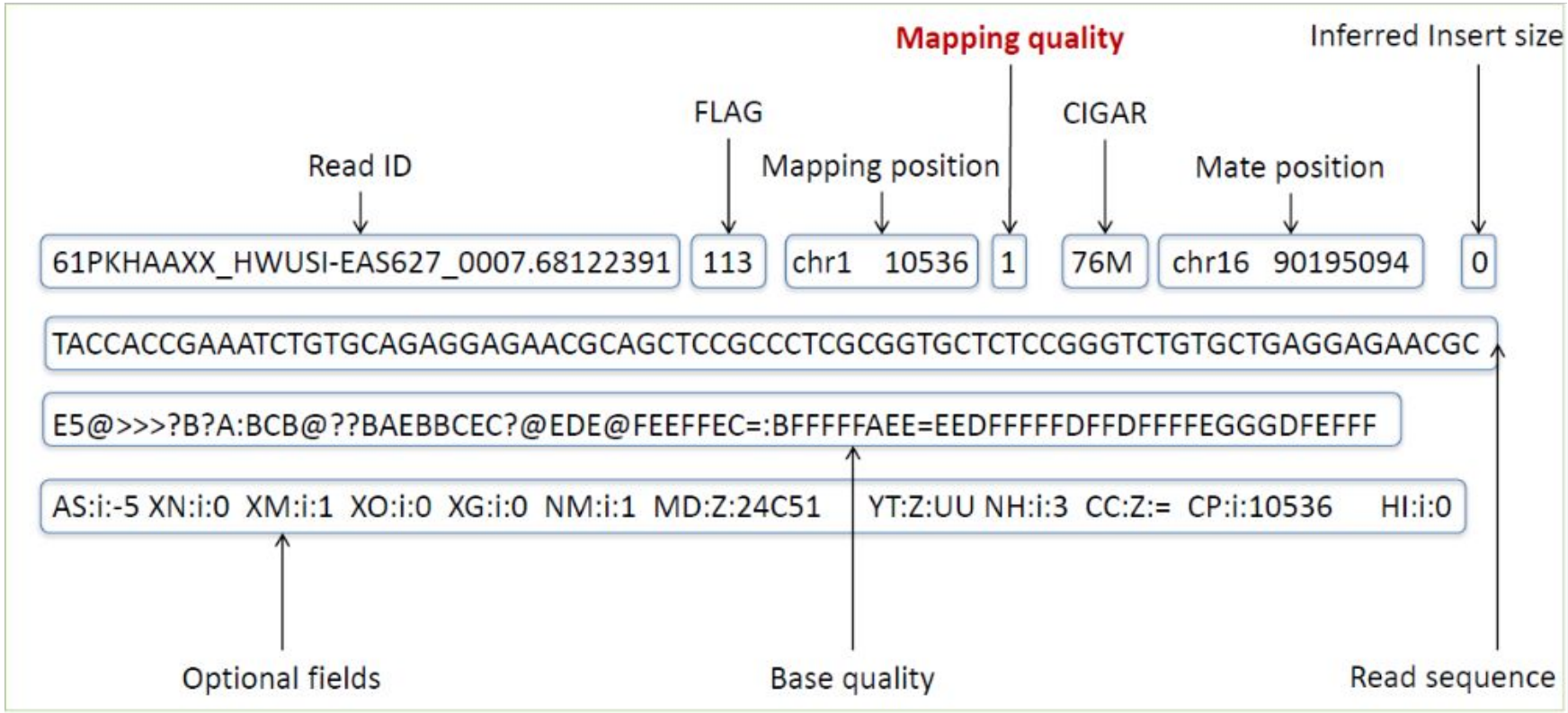
Alignments

```
61PKHAAXX_HWUSI-EAS627_0007.68122391   337   chr1   10536  1    76M   =    173766  163
    TACCACCGAAATCTGTGCAGAGGAGAACGCAGCTCCGCCCTCGCGGTGCTCTCCGGGTCTGTGCTGAGGAGAACGC
    E5@>>>?B?A:BCB@??BAEBBCEC?@EDE@FEEFFEC=:BFFFFFAEE=EEDFFFFFDFFDFFFFEGGGDFEFFF   AS:i:-5 XN:i:0  XM:i:1
    XO:i:0  XG:i:0  NM:i:1  MD:Z:24C51    YT:Z:UU NH:i:3  CC:Z:=  CP:i:10536    HI:i:0
61PKHAAXX_HWUSI-EAS627_0007.68122391   113   chr1   10536  1    76M   chr16  90195094    0
    TACCACCGAAATCTGTGCAGAGGAGAACGCAGCTCCGCCCTCGCGGTGCTCTCCGGGTCTGTGCTGAGGAGAACGC
    E5@>>>?B?A:BCB@??BAEBBCEC?@EDE@FEEFFEC=:BFFFFFAEE=EEDFFFFFDFFDFFFFEGGGDFEFFF   AS:i:-5 XN:i:0  XM:i:1
    XO:i:0  XG:i:0  NM:i:1  MD:Z:24C51    YT:Z:UU NH:i:3  CC:Z:=  CP:i:10536    HI:i:1
```

SAM/BAM specification

Mapping output: SAM/BAM format

SAM Specification: http://samtools.sourceforge.net/SAM1.pdf



SAM/BAM specification

Mapping output, mandatory fields

First columns are mandatory

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	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

SAM/BAM specification

Flags

https://broadinstitute.github.io/picard/explain-flags.html

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate

SAM/BAM specification

CIGAR code

CIGAR codes are strings, e.g.: 100M, 10M2D88M, 56M1I43M, 20S80M

- It contains information about indels, junctions...

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

SAM/BAM specification

Mapping output, optional fields

Some optionals fields, in the aligner section

Tag[1]	Type	Description
X?	?	Reserved fields for end users (together with Y? and Z?)
AM	i	The smallest template-independent mapping quality of segments in the rest
AS	i	Alignment score generated by aligner
BC	Z	Barcode sequence
BQ	Z	Offset to base alignment quality (BAQ), of the same length as the read sequence. At the i-th read base, $\mathrm{BAQ}_i = Q_i - (\mathrm{BQ}_i - 64)$ where Q_i is the i-th base quality.
CC	Z	Reference name of the next hit; "=" for the same chromosome
CM	i	Edit distance between the color sequence and the color reference (see also NM)
CP	i	Leftmost coordinate of the next hit
CQ	Z	Color read quality on the original strand of the read. Same encoding as QUAL; same length as CS.
CS	Z	Color read sequence on the original strand of the read. The primer base must be included.
E2	Z	The 2nd most likely base calls. Same encoding and same length as QUAL.
FI	i	The index of segment in the template.
FS	Z	Segment suffix.
FZ	B,S	Flow signal intensities on the original strand of the read, stored as (uint16_t) round(value * 100.0).
LB	Z	Library. Value to be consistent with the header RG-LB tag if @RG is present.
H0	i	Number of perfect hits
H1	i	Number of 1-difference hits (see also NM)
H2	i	Number of 2-difference hits
HI	i	Query hit index, indicating the alignment record is the i-th one stored in SAM
IH	i	Number of stored alignments in SAM that contains the query in the current record
MD	Z	String for mismatching positions. *Regex*: [0-9]+(([A-Z]|\^[A-Z]+)[0-9]+)*[2]
MQ	i	Mapping quality of the mate/next segment
NH	i	Number of reported alignments that contains the query in the current record
NM	i	Edit distance to the reference, including ambiguous bases but excluding clipping
OQ	Z	Original base quality (usually before recalibration). Same encoding as QUAL.
OP	i	Original mapping position (usually before realignment)
OC	Z	Original CIGAR (usually before realignment)
PG	Z	Program. Value matches the header PG-ID tag if @PG is present.
PQ	i	Phred likelihood of the template, conditional on both the mapping being correct
PU	Z	Platform unit. Value to be consistent with the header RG-PU tag if @RG is present.
Q2	Z	Phred quality of the mate/next segment. Same encoding as QUAL.
R2	Z	Sequence of the mate/next segment in the template.
RG	Z	Read group. Value matches the header RG-ID tag if @RG is present in the header.
SM	i	Template-independent mapping quality
TC	i	The number of segments in the template.

SAM/BAM specification

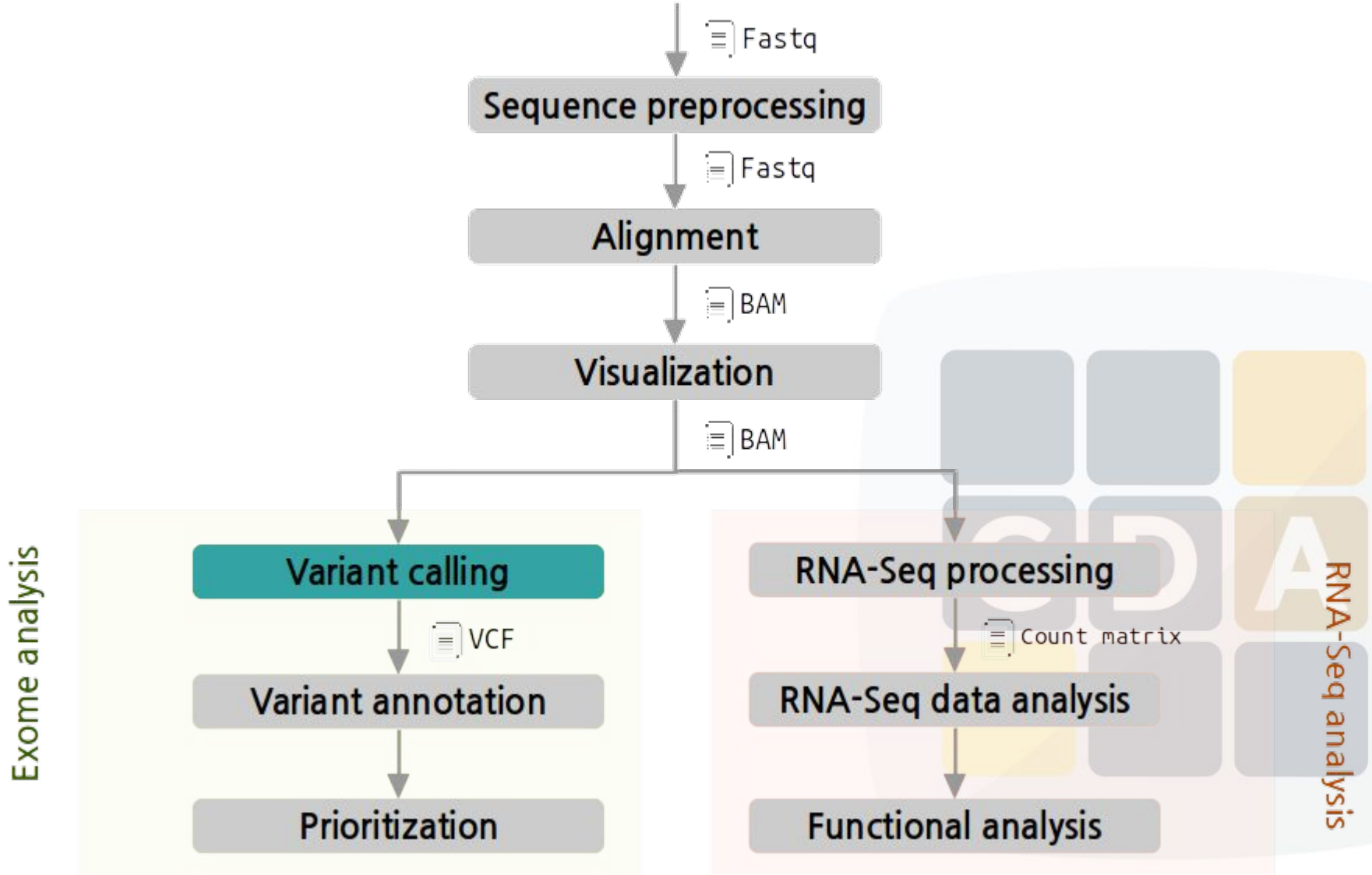
SAMtools

SAM Tools provide various utilities for manipulating alignments in the SAM/BAM format:

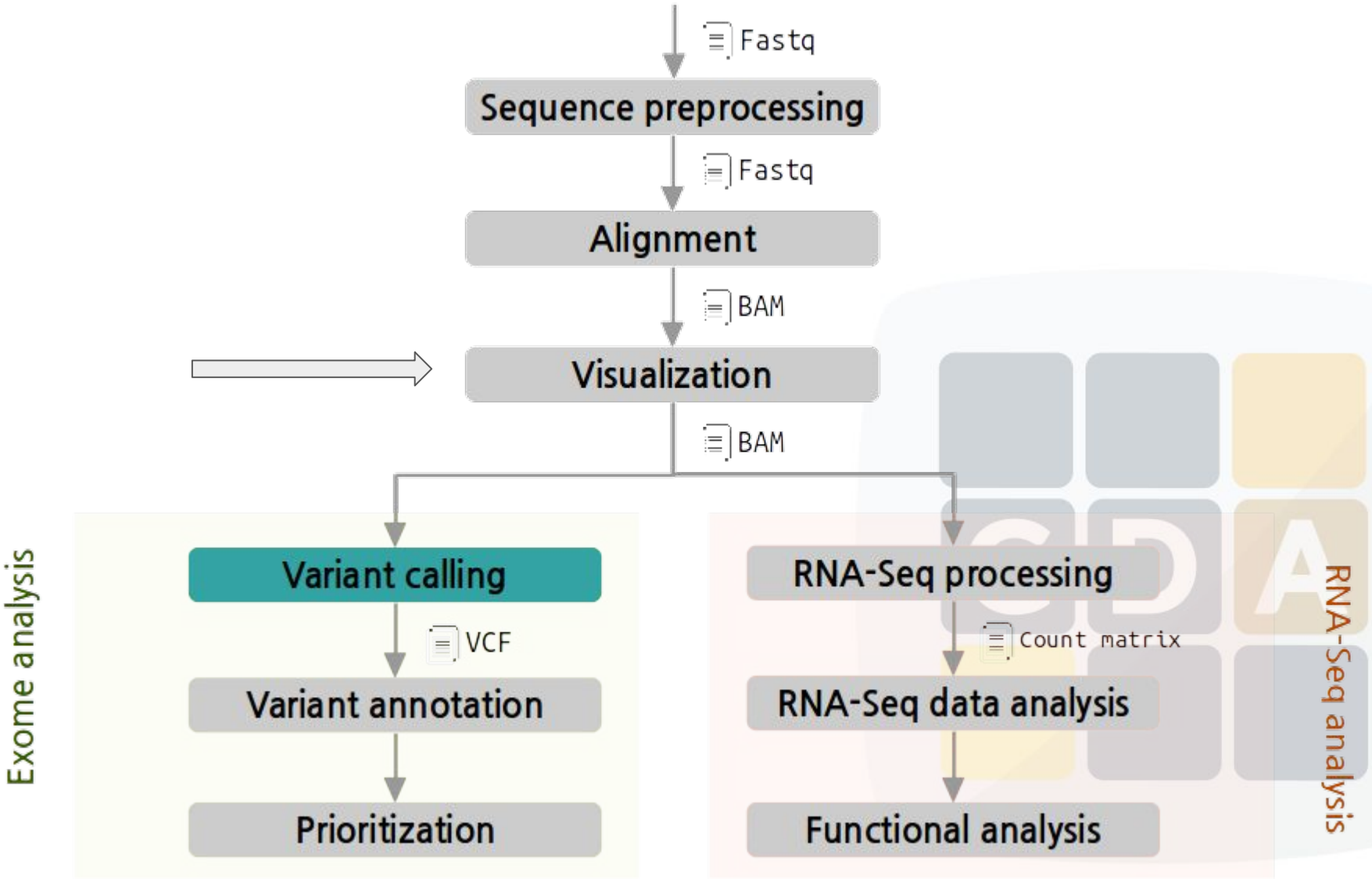
- SAM ↔ BAM conversion
- Filter by mappgin quality and flag
- Simple statistics
- Depth (coverage)
- Merge
- Sort
- ...

http://samtools.sourceforge.net/

NGS basic pipeline



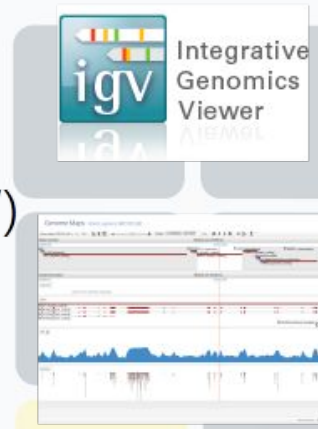
Where are we?


Fastq
Sequence preprocessing
Fastq
Alignment
BAM
Visualization
BAM
Exome analysis
Variant calling
VCF
Variant annotation
Prioritization
RNA-Seq analysis
RNA-Seq processing
Count matrix
RNA-Seq data analysis
Functional analysis

Alignment visualization

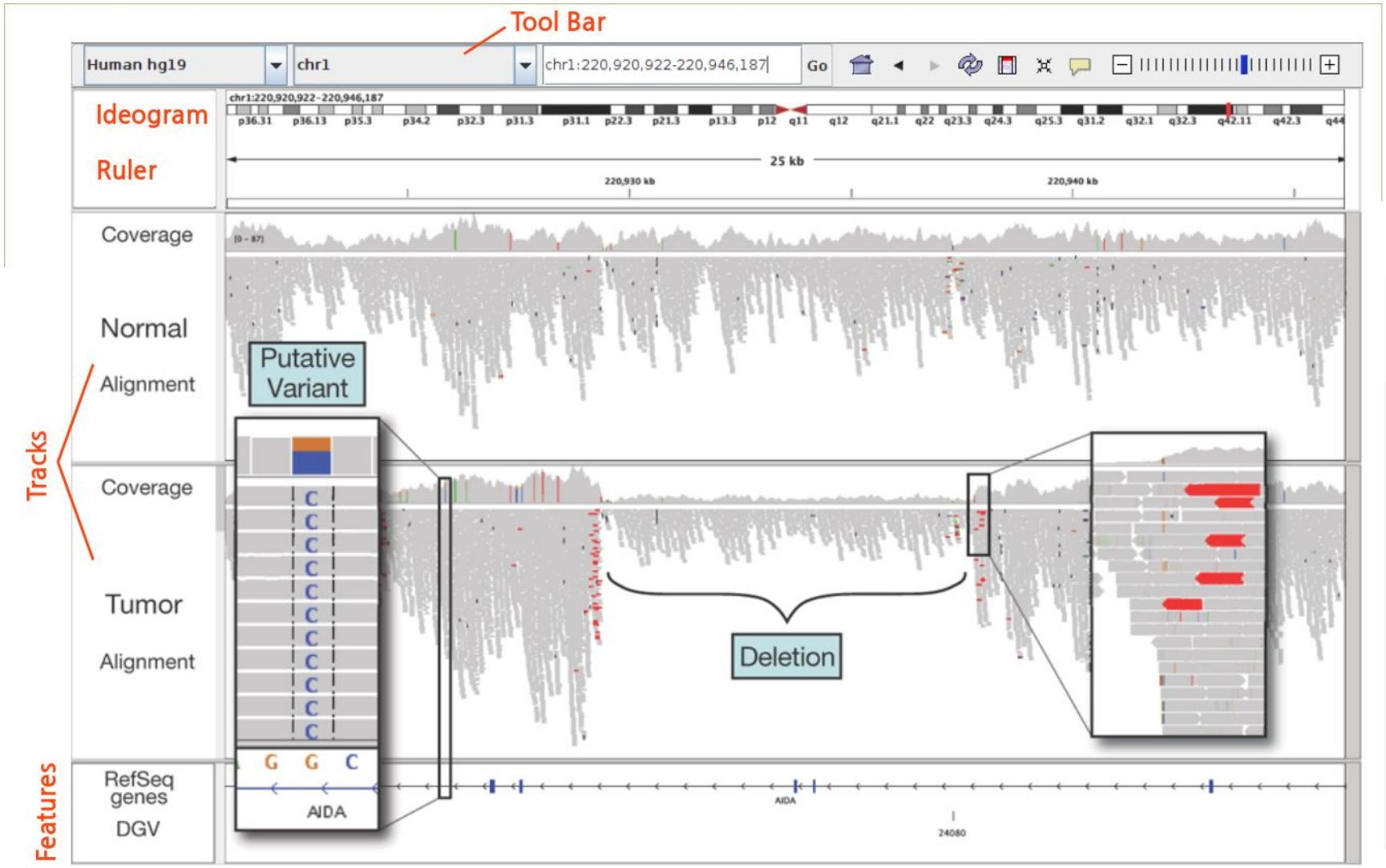
Why visualization?

- **Large** quantities of genomic **data** (NGS, array based methods...)
- **Human interpretation** and judgment using visualization can help complex biological relationships
- Two **Genomics Viewers**:
  - Integrative Genomics Viewer (IGV)
  - Genome Maps (http://genomemaps.org/)



Alignment visualization

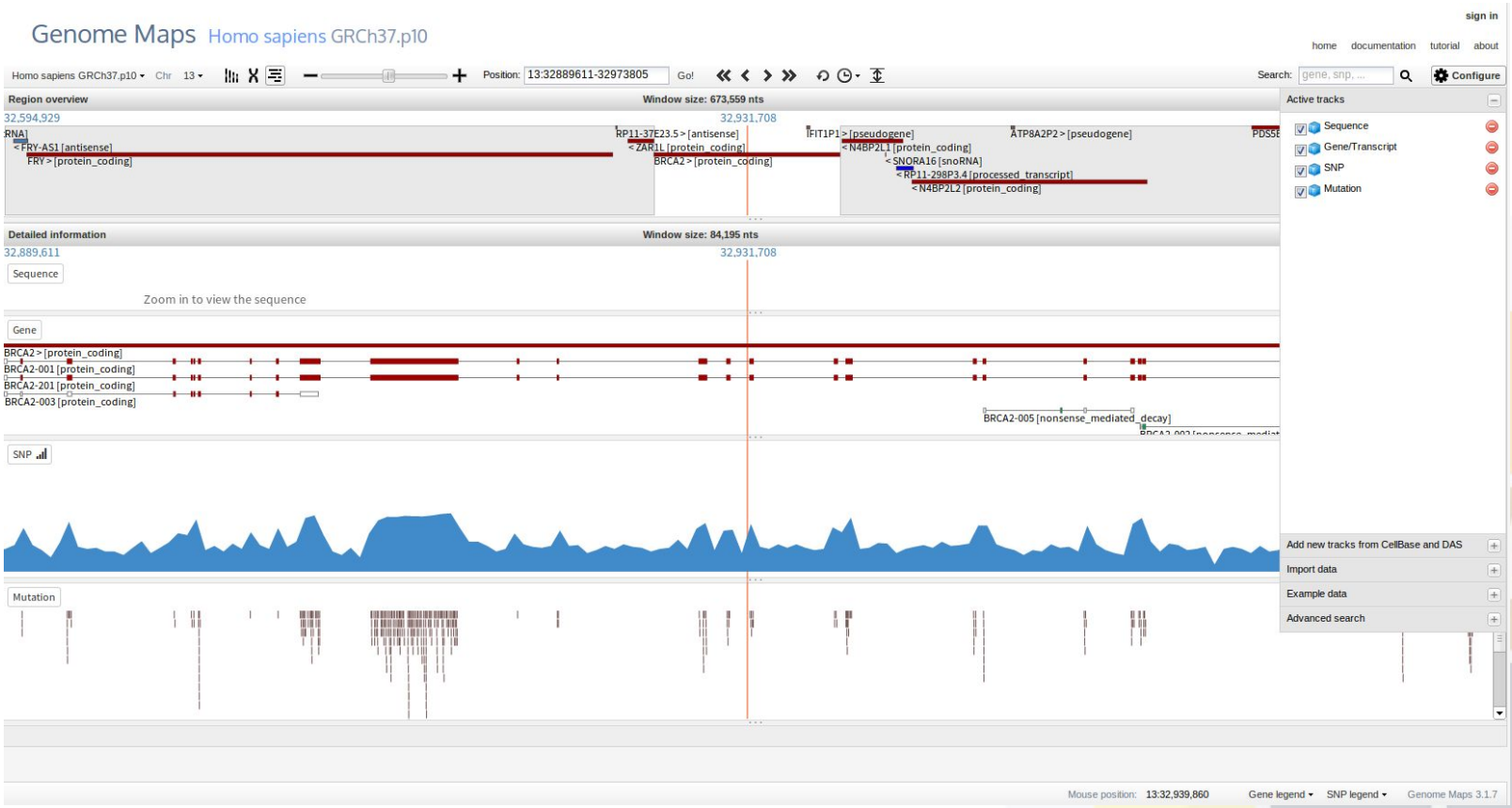
Integrative Genomics Viewer (IGV)



Alignment visualization

GenomeMaps

http://genomemaps.babelomics.org



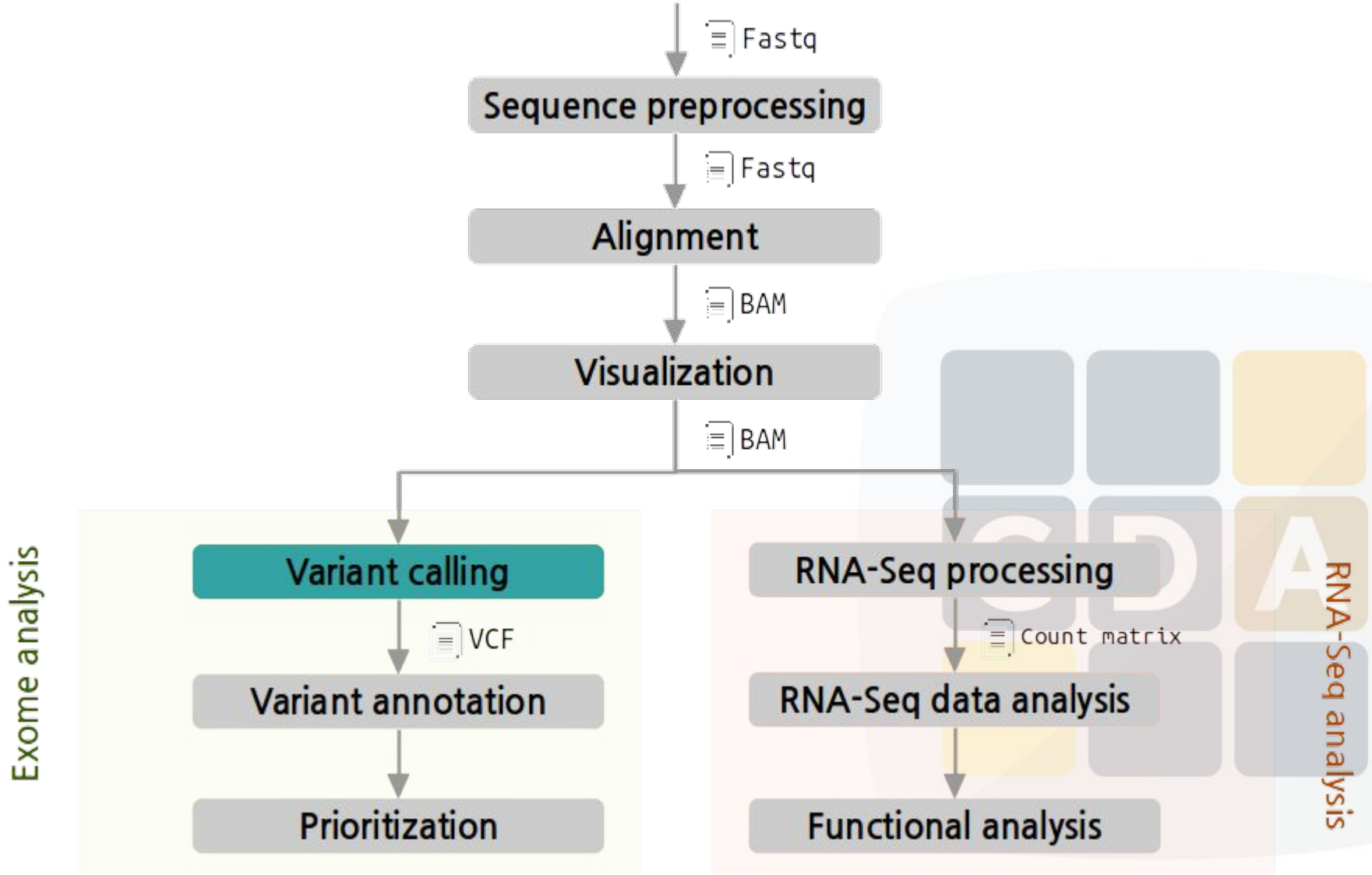
Best practices

- Choose the best aligner for your analysis and hardware
- Remove duplicated and low qualities reads from FASTQ
- Try to use paired-end datasets for variant calling and structural variation analysis. In RNA-seq paired-end can detect gene fusions
- Do **not allow multiple hits** for variant calling analysis. RNA-seq depending on read size and the analysis to perform
- Realign INDELS and recalibrate mapping quality for variant calling analysis
- **Simulation** can be very useful for choosing the right aligner

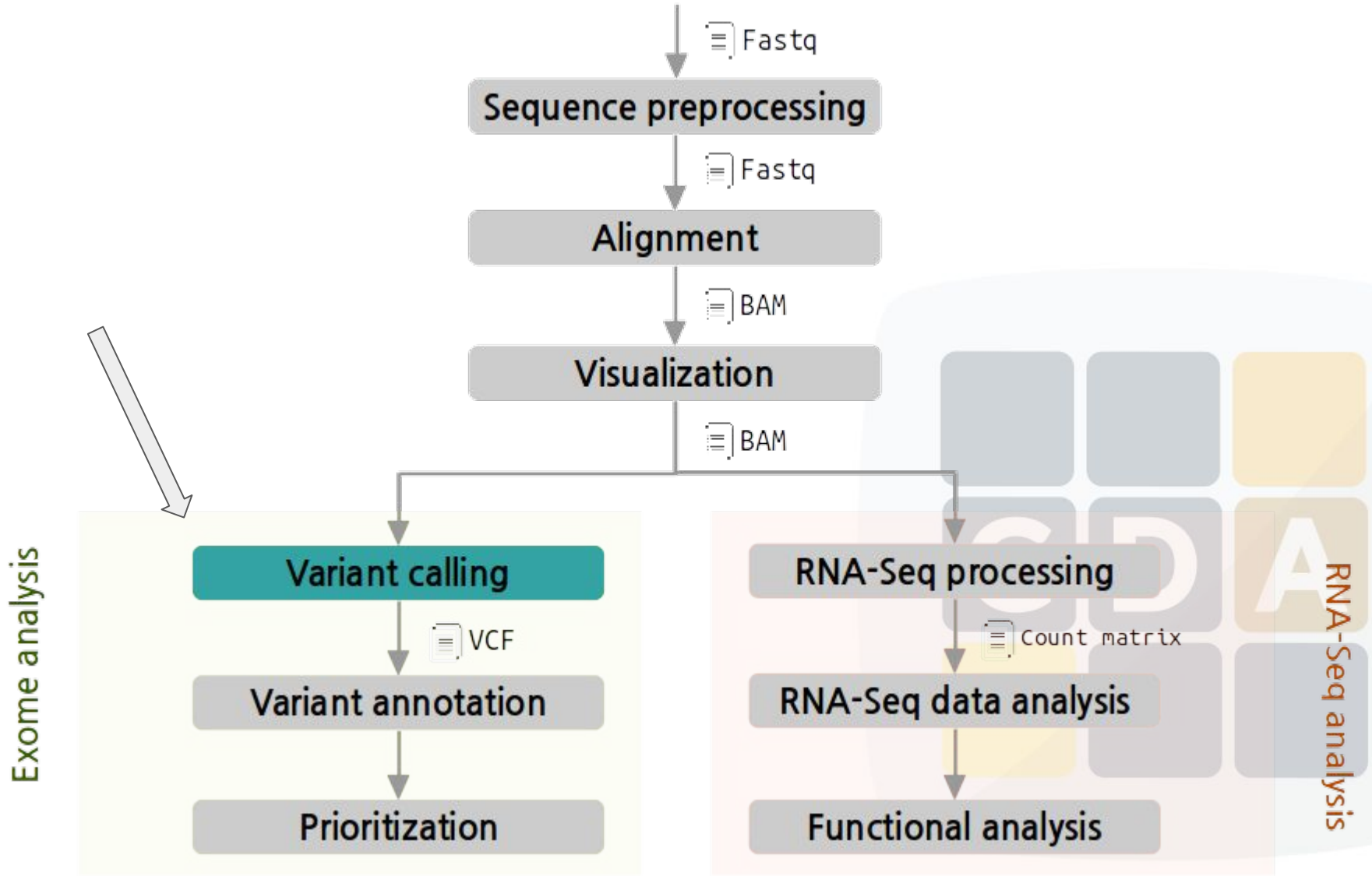
Data repositories

- **1000 Genome** project
  - http://www.1000genomes.org/
- **SRA**, *Short Read Archive*
  - http://www.ncbi.nlm.nih.gov/sra
- **EGA**, European Genome-Phenome Archive
  - https://www.ebi.ac.uk/ega
- … and many others

NGS basic pipeline



Where are we?


Fastq
Sequence preprocessing
Fastq
Alignment
BAM
Visualization
BAM
Exome analysis
Variant calling
VCF
Variant annotation
Prioritization
RNA-Seq processing
Count matrix
RNA-Seq data analysis
Functional analysis
RNA-Seq analysis

Contents

- Terminology
- Objective
- Variant Calling pipeline
- Variant Calling Format (VCF)
- Software

Genomic Variation

Terminology

- **Variant**: sequence data difference that exists between individuals in a population
- **Mutation**: molecular event that created a variant
- **Allele**: forms of the bases occupying the same position on matching chromosomes
- **Genotype**: allelic state in a specific individual
  - AA homozygous or AT heterozygous at specific base
- **Polymorphism**: sequence variation that is common within a population
  - "SNP on chromosome 16 associated with obesity"

Types of Genome Sequence Variants

1. **Single Nucleotide Variants (SNVs)**

   Single base changes, e.g., A→T.

2. **Insertions-Deletions (Indels)**

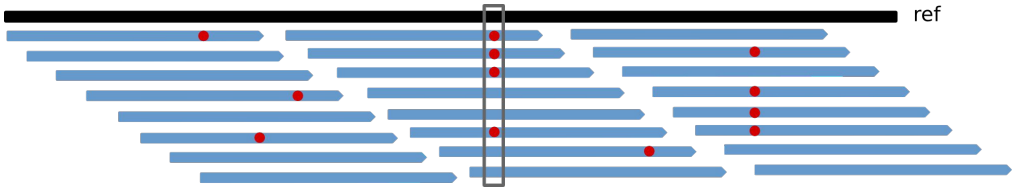
   Consisting of one or a few bases, e.g., +ATGA, ΔT.

3. **Structural Variants (SVs)**

   Everything else: large deletions, insertions, duplications, inversions, translocations, mobile element insertions, horizontal gene transfer

Objective

Assign a genotype to each position



Problems

Some variation observed in BAM files is caused by mapping and sequencing artifacts:

- **PCR artifacts**:
  - Mismatches due to errors in early PCR rounds
  - PCR duplicates
- **Sequencing errors**: erroneous call, either for physical reasons or to properties of the sequenced DNA
- **Mapping errors**: often happens around repeats or other low-complexity regions

Separate **true variation** from machine artifacts

Variant calling process pipeline

1. **Mark duplicates**

   Duplicates should not be counted as additional evidence

2. **Local realignment around INDELS**

   Reads mapping on the edges of INDELS often get mapped with mismatching bases introducing false positives

3. **Base quality score recalibration (BQSR)**

   Quality scores provided by sequencing machines are generally inaccurate and biased

4. **Variant calling**

   Discover variants and their genotypes

Mark duplicates

- All NGS **sequencing platforms are NOT single molecule sequencing** → the same DNA molecule can be sequenced several times
- **PCR** → duplicate DNA fragments in the final library
- If there is a base variation it will have **high depth support**
- Can result in **false variant calls**

Tools

- **Samtools**: samtools rmdup or samtools rmdupse
- **Picard**: MarkDuplicates

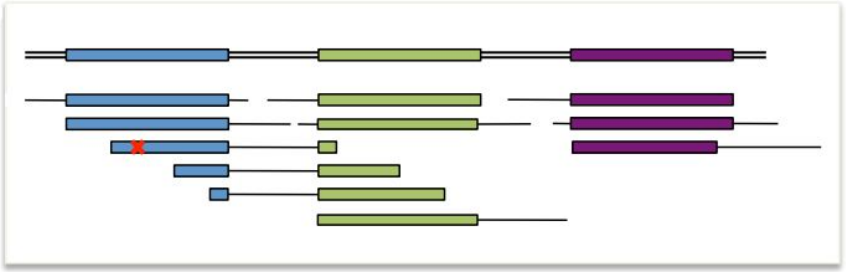
Mark duplicates

The reason why duplicates are bad

✖ = sequencing error propagated in duplicates

FP variant call (bad)

After marking duplicates, the GATK will only see :

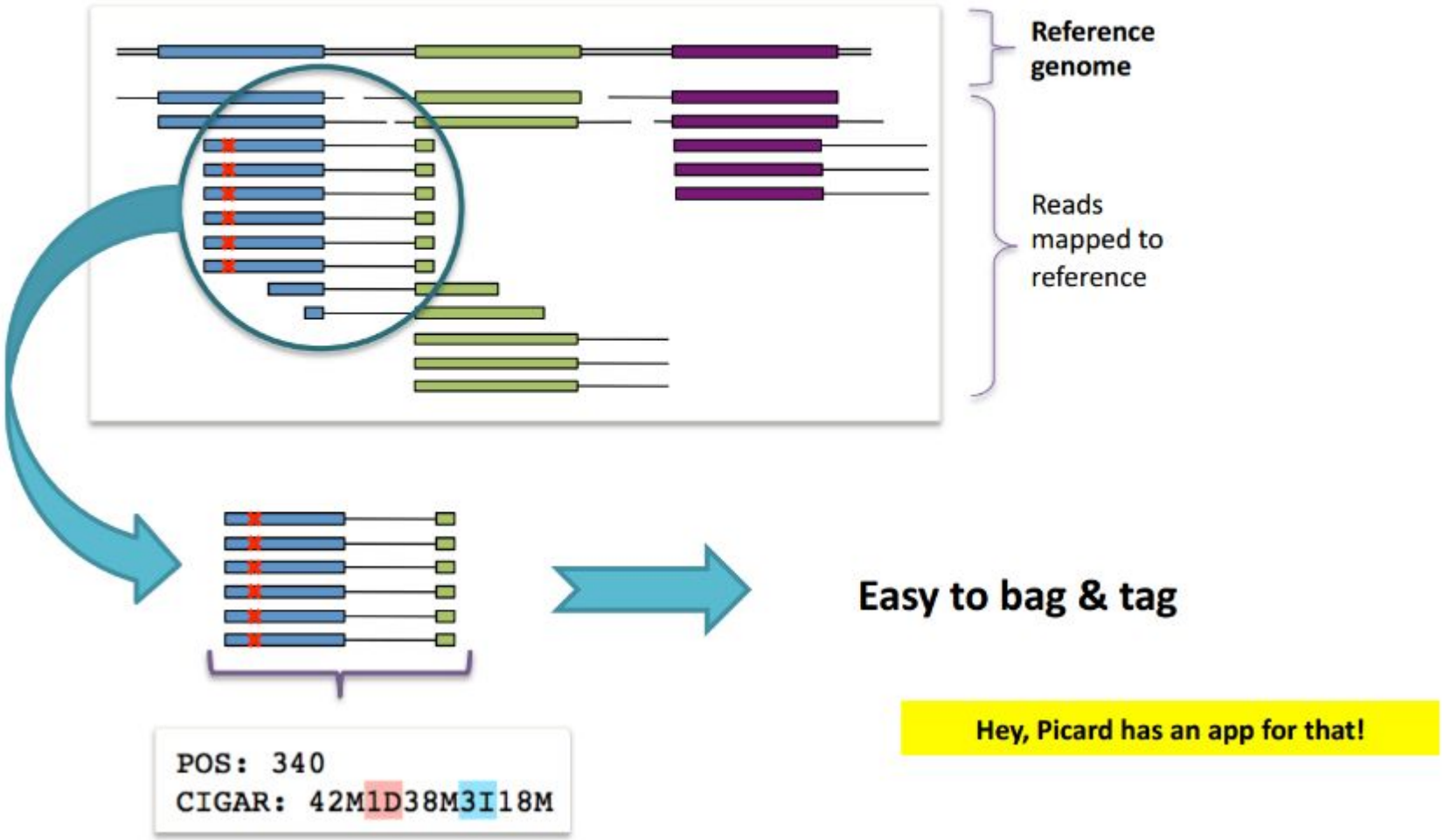


... and thus be more likely to make the right call

Mark duplicates

Duplicate identification

Duplicates have the **same starting position** and the **same CIGAR** string

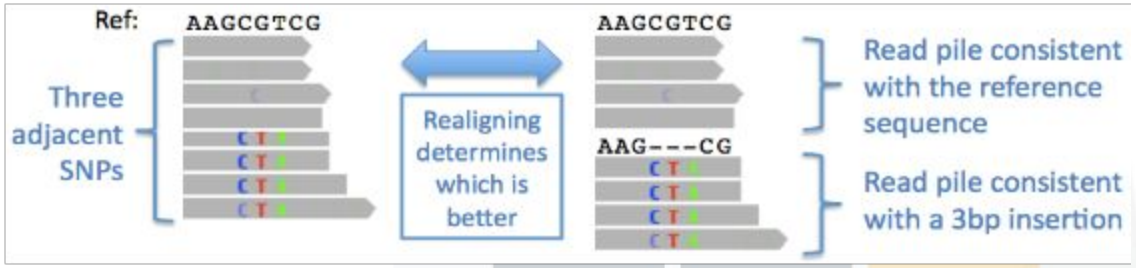


Local realignment around INDELS

- Reads **near INDELS** are mapped with mismatches
- **Realignment** can identify the most consistent placement for these reads
  1. **Identify** problematic regions
  2. **Determine the optimal** consensus sequence
- **Minimizes mismatches** with the reference sequence
- **Refines** location of **INDELS**

Local realignment around INDELS

- Reads **near INDELS** are mapped with mismatches
- **Realignment** can identify the most consistent placement for these reads
  1. **Identify** problematic regions
  2. **Determine the optimal** consensus sequence
- **Minimizes mismatches** with the reference sequence
- **Refines** location of **INDELS**



Base quality score recalibration

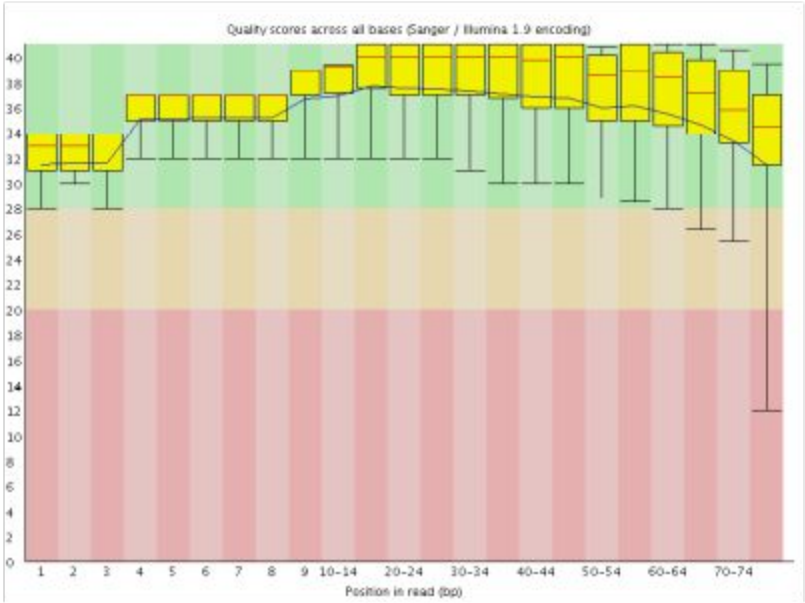
- **Calling algorithms rely** heavily on the **quality scores** assigned to the individual base calls in each sequence read
- Unfortunately, the scores produced by the machines are subject to various sources of **systematic error**, leading to over- or under-estimated base quality scores in the data

How?

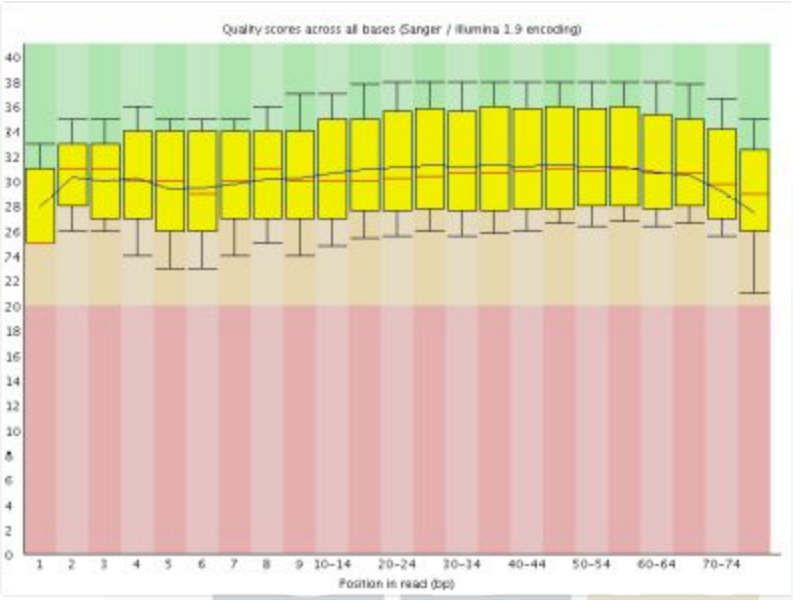
1. **Analyze covariation** among several features of a base:
   - Reported quality score
   - Position within the read
   - Preceding and current nucleotide
2. Use a set of **known variants** (i.e.: dbSNP) to model error properties of real polymorphism and determine the **probability that novel sites are real**
3. **Adjust** the quality scores of all reads in a BAM file

Base quality score recalibration

Before



After



Phred Quality score:

$$Q_{\text{Phred}} = -10 \log_{10} P(\text{error}).$$

Variant calling

Variant discovery process

Steps

1. **Variant calling:** Identify the positions that differ from the reference
2. **Genotype calling**: calculate the genotypes for each sample at these sites

Initial approach

Independent base assumption

Counting the number of times each allele is observed

Evolved approach

Bayesian inference → Compute genotype likelihood

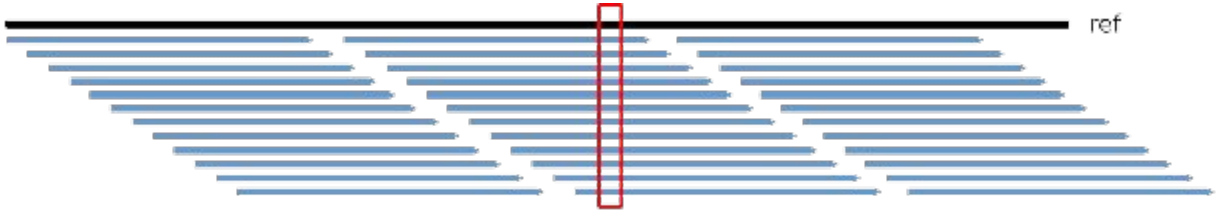
Advantages:

Provide statistical measure of **uncertainty**

Lead to **higher accuracy** of genotype calling

Variant calling

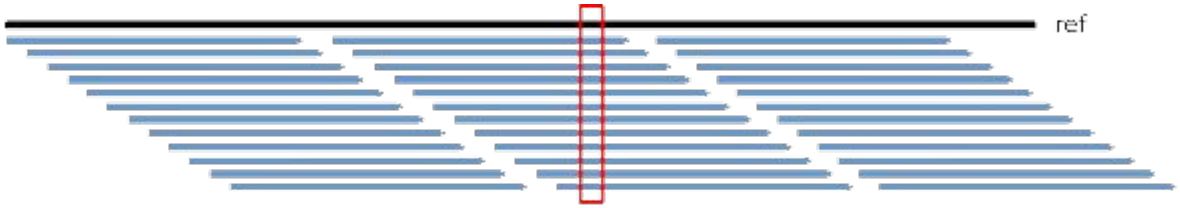
Variant discovery process



Reference = A

Variant calling

Variant discovery process



Reference = A

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG

AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG

AAAAAAAAAAAAAAAGGGGGGGGGGGGCT

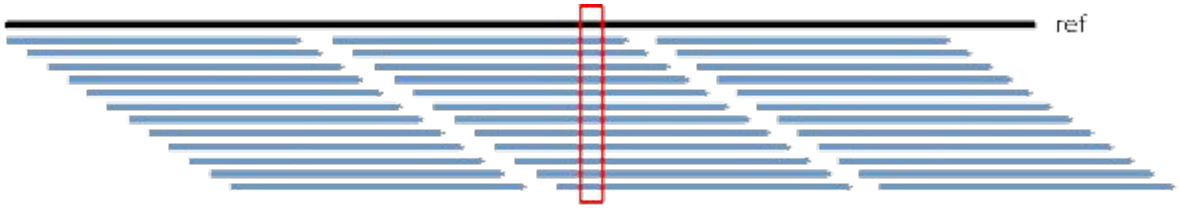
AAAGGGCCTT

N = nucleotides
G = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

Variant calling

Variant discovery process



Reference = A

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	N=30, X=0
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	N=30, X=30
AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG	N=30, X=15
AAAAAAAAAAAAAAAGGGGGGGGGGGGCT	N=30, X=12
AAAGGGCCTT	N=10, X=3

Cutoff for X → value or proportion

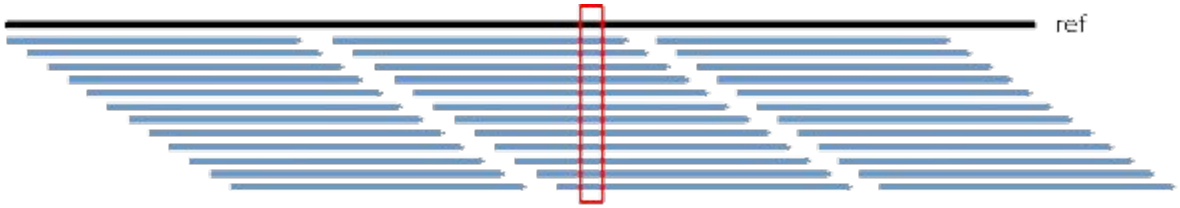
- c = 30% $X \leq c$ → **RR**, $X > c$ → **RV**
- c_1 = 10%, c_2 = 30% $X \leq c_1$ → **RR**
  $c_1 < X < c_2$ → **RV**
  $X \geq c_2$ → **RR**

N = nucleotides
G = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

Variant calling

Variant discovery process



Reference = A

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	N=30,	X=0	→ **RR**
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	N=30,	X=30	→ **VV**
AAAAAAAAAAAAAAAGGGGGGGGGGGGGGG	N=30,	X=15	→ **RV**
AAAAAAAAAAAAAAAGGGGGGGGGGGGCT	N=30,	X=12	→ **RV**
AAAGGGCCTT	N=10,	X=3	→ **RV?**

Cutoff for X → value or proportion

- $c = 30\%$ $\quad X \leq c$ → **RR**, $\quad X > c$ → **RV**
- $c_1 = 10\%$, $c_2 = 30\%$ $\quad X \leq c_1$ → **RR**
  $c_1 < X < c_2$ → **RV**
  $X \geq c_2$ → **RR**

N = nucleotides
G = true genotype
R = reference base
V = variant base
X = variant nucleotides

Outcomes:
RR RV VV

Variant Calling Format

VCF file format

- Specification defined by the 1000 genomes (current version **4.2**): http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41
- Commonly **compressed and indexed** with bgzip/tabix
- Single-sample or multi-sample VCF

```
##fileformat=VCFv4.1
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Variant Calling Format

VCF file format

```
#CHROM  POS     ID        REF    ALT    QUAL   FILTER   INFO
20      14370   rs6054257 G      A      29     PASS     NS=3;DP=14;AF=0.5;DB;H2

FORMAT        NA00001          NA00002          NA00003
GT:GQ:DP:HQ   0|0:48:1:51,51   1|0:48:8:51,51   1/1:43:5:.,.
```

genotype genotype quality read depth haplotype qualities

- **CHROM**: chromosome
- **POS**: position
- **ID**: identifier
- **REF**: reference base(s)
- **ALT**: non-reference allele(s)
- **QUAL**: quality score of the calls (phed scale)
- **FILTER**: "PASS" or a filtering tag
- **INFO**: additional information
- **FORMAT**: describes the information given by sample

Software

Software	Available from	Calling method	Prerequisites	Comments	Refs
SOAP2	http://soap.genomics.org.cn/index.html	Single-sample	High-quality variant database (for example, dbSNP)	Package for NGS data analysis, which includes a single individual genotype caller (SOAPsnp)	15
realSFS	http://128.32.118.212/thorfinn/realSFS/	Single-sample	Aligned reads	Software for SNP and genotype calling using single individuals and allele frequencies. Site frequency spectrum (SFS) estimation	-
Samtools	http://samtools.sourceforge.net/	Multi-sample	Aligned reads	Package for manipulation of NGS alignments, which includes a computation of genotype likelihoods (samtools) and SNP and genotype calling (bcftools)	53
GATK	http://www.broadinstitute.org/gsa/wiki/index.php/The_Genome_Analysis_Toolkit	Multi-sample	Aligned reads	Package for aligned NGS data analysis, which includes a SNP and genotype caller (Unifed Genotyper), SNP filtering (Variant Filtration) and SNP quality recalibration (Variant Recalibrator)	32,33
Beagle	http://faculty.washington.edu/browning/beagle/beagle.html	Multi-sample LD	Candidate SNPs, genotype likelihoods	Software for imputation, phasing and association that includes a mode for genotype calling	42
IMPUTE2	http://mathgen.stats.ox.ac.uk/impute/impute_v2.html	Multi-sample LD	Candidate SNPs, genotype likelihoods	Software for imputation and phasing, including a mode for genotype calling. Requires fine-scale linkage map	44
QCall	ftp://ftp.sanger.ac.uk/pub/rd/QCALL	Multi-sample LD	'Feasible' genealogies at a dense set of loci, genotype likelihoods	Software for SNP and genotype calling, including a method for generating candidate SNPs without LD information (NLDA) and a method for incorporating LD information (LDA). The 'feasible' genealogies can be generated using Margarita (http://www.sanger.ac.uk/resources/software/margarita)	54
MaCH	http://genome.sph.umich.edu/wiki/Thunder	Multi-sample LD	Genotype likelihoods	Software for SNP and genotype calling, including a method (GPT_Freq) for generating candidate SNPs without LD information and a method (thunder_glf_freq) for incorporating LD information	-

A more complete list is available from http://seqanswers.com/wiki/Software/list. LD, linkage disequilibrium; NGS, next-generation sequencing.

GATK (Genome Analysis ToolKit)

http://www.broadinstitute.org/gatk/

- Probabilistic method: **Bayesian estimation** of the most likely genotype
- Calculates many **parameters** for each position of the genome
- INDEL realignment
- Base quality recalibration
- SNP and INDEL calling
- **Multi-sample** calling
- Uses standard input and output files
- Used in **many NGS projects**, including the 1000 Genomes Project, The Cancer Genome Atlas, etc.