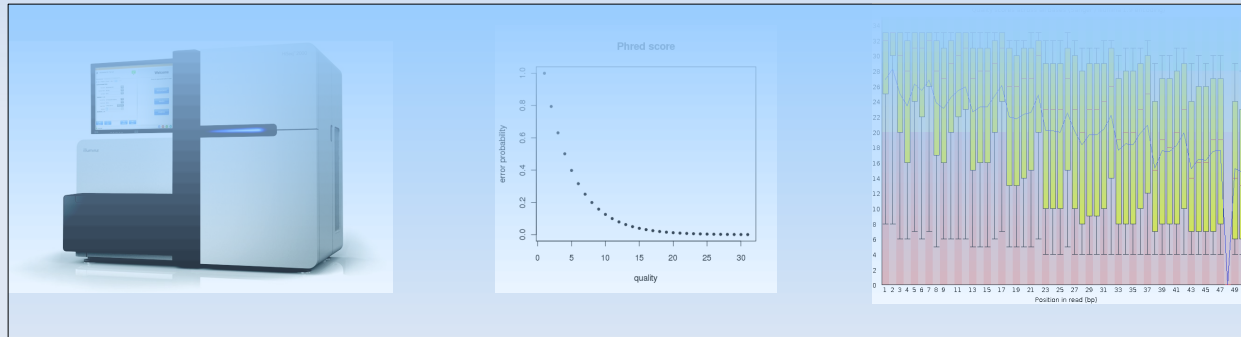


# NGS sequence preprocessing



José Carbonell Caballero  
[jcarbonell@cipf.es](mailto:jcarbonell@cipf.es)



# NGS Sequence preprocessing

- Contents

Data  
Format

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

Quality  
Control

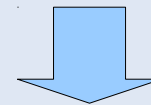
- Evaluation of sequence quality
  - Quality control tools
  - Identification of typical artifacts
  - Sequence filtering
- 
- Practical session

# NGS Sequence preprocessing

- Sequence capture



RAW data  
*Proprietary format*



FastQ

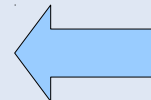
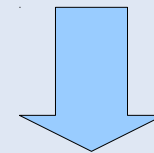
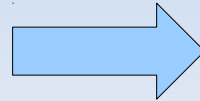
# NGS Sequence preprocessing

- Different technologies

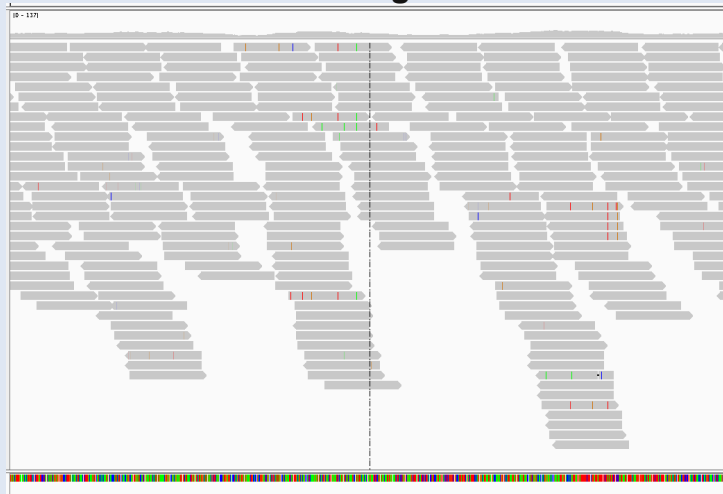


# NGS Sequence preprocessing

- Genome sequencing

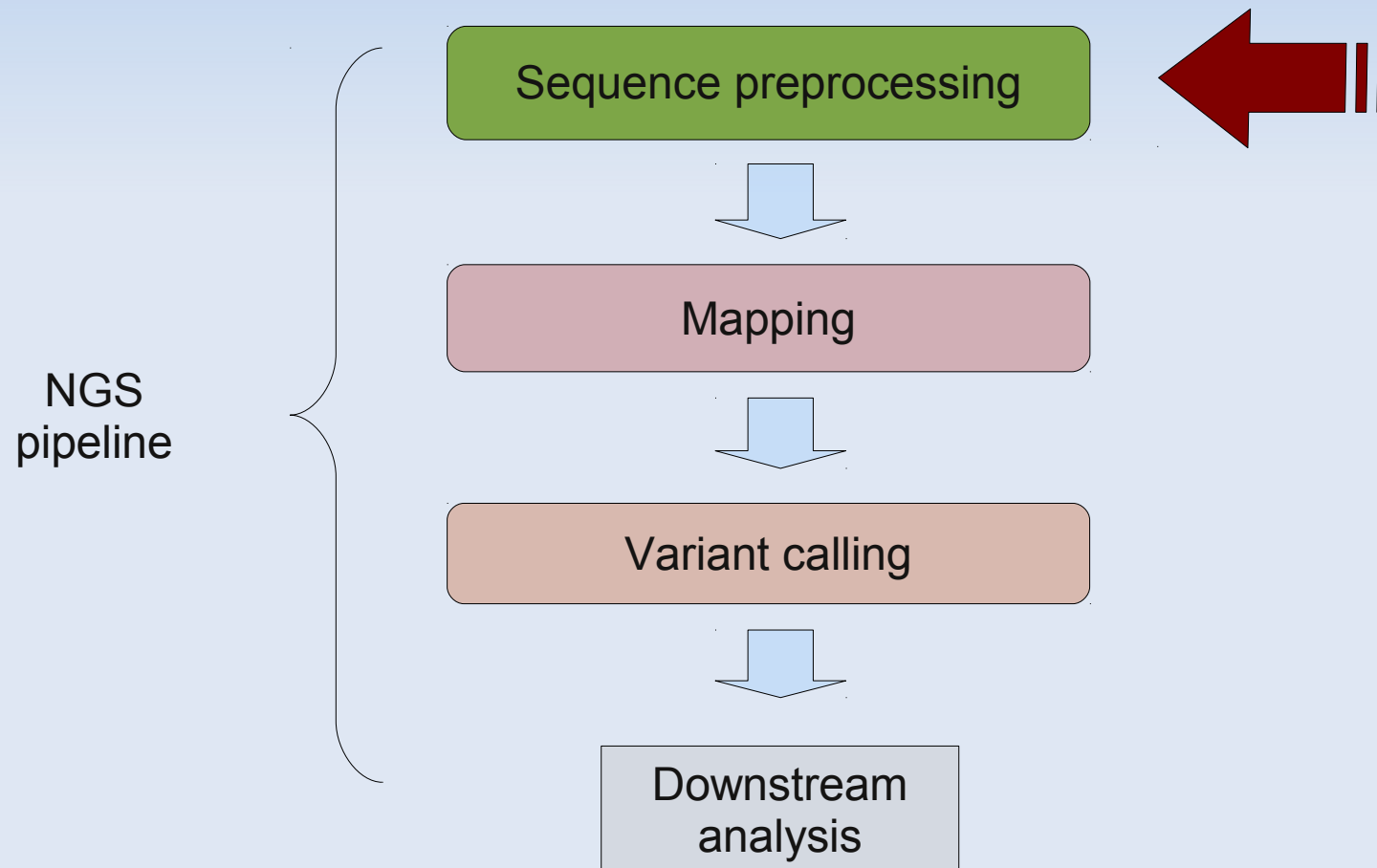


*Reference genome*



# NGS Sequence preprocessing

- Where we are?



# NGS Sequence preprocessing

- Fasta and Fastq formats
  - Standard formats for sequence storage
  - Text-based formats (easy to use!)
  - (Almost) every programming language has a parser

# NGS Sequence preprocessing

- Fasta format
  - Nucleotide or peptide sequence

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

```
>BBTBSCRYR  
tgcaccaaacatgtctaaagctggaaccaaattactttctttgaagacaaaactttca  
aggccgccactatgacagcgattgcgactgtgagatttccacatgtacctgagccgctg  
caactccatcagagtggaaggaggcacctgggctgtgtatgaaaggcccaattttgctgg  
gtacatgtacatcctaccccggggagatcctgagtagcactggatgggcctcaa
```

- Some typical file extensions (.fasta, .fa, .fna, .ffn, .faa,...)



# NGS Sequence preprocessing

- **Fasta format**
  - Allow multi-sequence (typically different chromosomes)

```
>scaffold_1
CAAGGCTATAGCCACCCGTTTTTGTGGCCTTTTTCCGCTGGACGAACTTGGCGCCCCGGCCTTCGGGTGGTTATTTTTG
GGTGCAGCCTAGTGC GCGGCCTATTTTTGGCACACGGAGGCCCTCGCAAAGTCTCGCGGCATTCCGAGTCCGGCGGACG
AAGTTGGCCAGCCCCACCCCAAGGCTATAGCCACCCGTTTTTGTGGCCTTTTTTCCGCTGGACGAACTTGGCGCCCCG
GCCTTCGGGCGGTTATTTTTGGCCGCGGCTTCGTCCGTGGCCTATTTTTGGCACACGGAGGCCCCCCGCAAACCTCTCCCG
GCATTCCGAGTCCGGCGGACGAAGTCGGCCAGCTTCACCACCCAGGCTGTTGCCACCCCTTTTTGTGGCCTTTTTCCGC
TGGACGAACTTGGCGCCCCAGCCTTTGGGCTGTTATTTATGGGTGCGGCTTTGTCTACGGCCTATTTTTGGCAGACGCAG
GCCCCTCGCAAAGTCTCGCGGCATTCCGAGTCCGGCGGACGAAGTCGGCCAGCCCCACCCCAAGGCTATAGCCACCC
GTTTTTGTGGCCTTTTTCCGCTGGACGAACTTGGCGCCCCGGCCTTCGGGCGGTTTTTTTTCTGGTGCGGTTTTGCCCCG
GCCTATTTTTGGCACTTGAGGCCCTCGTAAAGTCTCGCGGCATTCCGAGTCCGGCGGAAGAAGTTGGCCAGCCCCACC
CTCAAAGGCTATAGCCACCCGTTTTTGTGGCCTTTTTCCGCTGGACGAACTTGGGGCCCCGGCCTTCGGGCGGTTATTT
--
>scaffold_2
ACGGTCCGGGGGCATCGGGGTGGGGGGTAGCCGCGCGGCAGTTTGAACGGCGAAAAATGGGCAAGATCGGGGGCCGCTT
AGTAGGCTTTGCCGTTTCGGCCGTAACCATGATCCGGGCCTCCGATTCCGTTTCCGTTTGGTCCCACGGGACCAGCGG
TCCGGGGGCATCGGCAAGGGGGGGTAGCCGTGCGGCGGTTTGGATCGCCGAAAAATGGGTAAGATCGGGGGCCGTTTGGCCC
GTTTTGCCGTTTCGGCCCCCGGGGGGGCGGTTTCATGCCCCGGGGGGGACAGCGGGGCGGGATCGGCACACGGCGGGTGA
GGTGATCGGGTTCAGGCGGGTTCGGTTCGGCGGGGGGGCGCCGGGGGGCTATAGCGCACCCGCTTCGGGGGCCGAAATT
GGGGGCCGTAACCATGATCGGGGCCCCGATTCCGTTTCCGTTTTCGTACCACGGGACCAGCGGTCCGGGGGCATCGGGA
TGGGGGGGTAGCCGCGCGGCAGTTTGAACGGCGAAAAATGGGAAAGATCGGGGGCCGCTTAGTAGGCTTTGCCGTTTCG
GCCGTAACCATGATCCGGGCCTCCGATTCCGTTTCCGTTTGGTCCCACGGGACCAGCGGTCCGGGGGCATCGGGATGGG
GGGGTAGCCGTGCGGCGGTTTGGATCGCCGAAAAATGGGTAAGATCGGGGCCGTTTGGCCCCGTTTTGCCGTTTCGGCCCC
CCGGGGGGCGGTTTCATGCCCCGGGGGGCGGTTTCATGCCCCGGGGGGGACAGCGGGGCGGGATCGGTACACGGCGGGGAG
--
>scaffold_21
ACATATATAAAGTATTGTA CTAGAAAAACATTGTAAATGTATGCCTATTTAAACTTCAAGTATATGTAACACTTTCAAAGT
CATAAGTGTA AAACTATTATATTTAAGATGTTTTGAGTTTATAAAAAATAAAACA ACTTAACACTTCTACTCAGACGTTAAA
AAATAAAAAACA AACATATTATTCTAATATATACATT CATAACCCACTTAGACTCATTCAAGTTTAAAGATATAAAAAAAA
AAAAGTGCTACCTAGAAAATCCTTAATAGCAAGTCTTGTCTTTTATTTTCATTAAGGGTAAAAATCCTAAATGTGGCAAATGG
CGTGTATAAGAATTTTTGAGGGGTGCCAAATGCCCAATTCATT CATATTAGGCTTCTGAAAGAAGGCCATT CATTAC
GACTTCGGGTGCGCCACCTGCTTGAATCCGTTTGTGTTTTTAACGTTGCAAGTATACCCATGTAAGTAAAAACAAACAAA
```

# NGS Sequence preprocessing

- Fastq format
  - Let's say "fastq is a fasta with qualities"

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



- Fasta storages genomes...and fastq fragments

# NGS Sequence preprocessing

- Sequence quality encoding
  - Base quality must be encoded in just 1 byte!

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

- Each base has a corresponding quality value  
(quality in position  $n$ , corresponds to base of position  $n$ )
- How is the encoding?

Error probability  Phred transformation  
(inversed integer value)  ASCII encoding

# NGS Sequence preprocessing

- Sequence quality encoding

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;	@	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)

- Phred + 33

Sanger [0,40]

Illumina 1.8 [0,41]

- Phred + 64

Illumina 1.3 [0,40]

Illumina 1.5 [3,40]

[http://en.wikipedia.org/wiki/FASTQ\\_format](http://en.wikipedia.org/wiki/FASTQ_format)

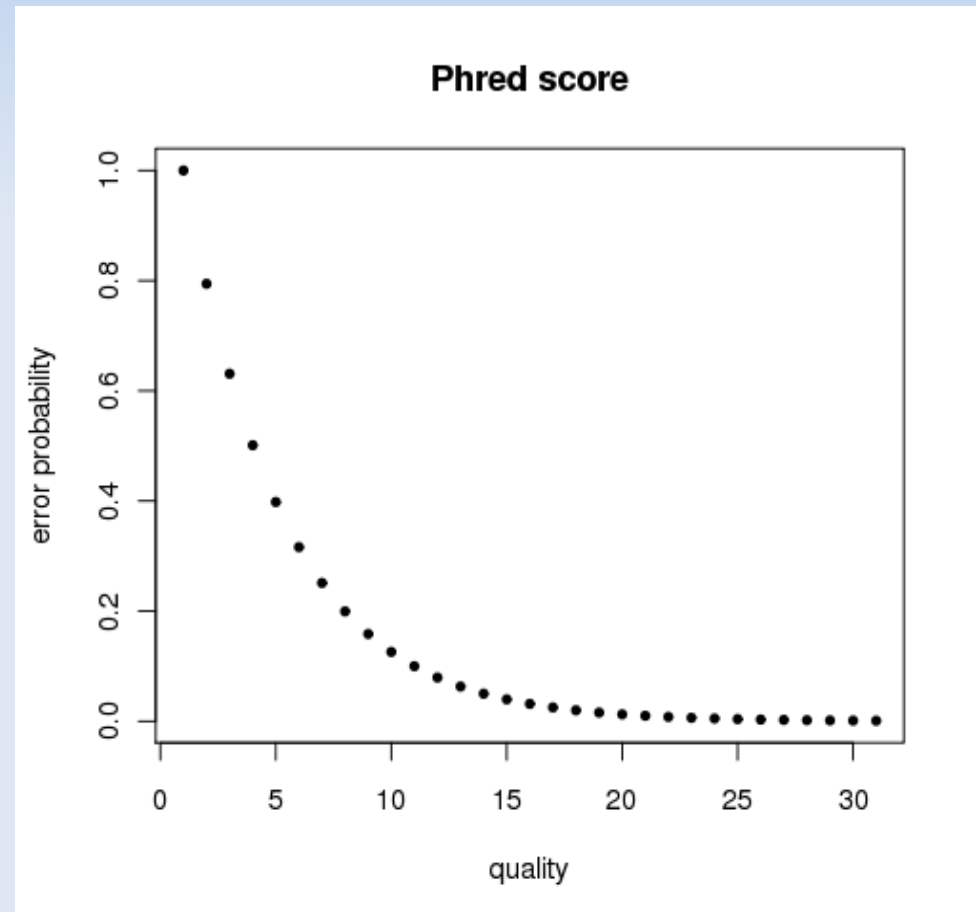
# NGS Sequence preprocessing

- Sequence quality encoding
  - Phred scores

$$Q = -10 \log_{10} P \quad \longleftrightarrow \quad P = 10^{-\frac{Q}{10}}$$

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90 %
20	1 in 100	99 %
30	1 in 1000	99.9 %
40	1 in 10000	99.99 %
50	1 in 100000	99.999 %



# NGS Sequence preprocessing

- Sequence quality encoding
  - Phred scores

Error probability  $\rightarrow$  Phred transformation (inversed integer value)  $\rightarrow$  ASCII encoding

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

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTTT
+
!''*(((((***+))%%#+))%%%) .1***-+*'' ) **5GCCF>>>>>CCCCCCC65
```

$P = 0.01$   $\rightarrow$   $Q = -10 \log_{10}(0.01) = 20$   $\rightarrow$  ASCII  $33+20 = 53 \Rightarrow$  **G**  
*Phred+33*

$P = 1$   $\rightarrow$   $Q = -10 \log_{10}(0.01) = 0$   $\rightarrow$  ASCII  $33+0 = 33 \Rightarrow$  **!**  
*Phred+33*

# NGS Sequence preprocessing

- Evaluation of sequence quality
  - Primary tool to assess sequencing
  - If we **evaluate** our sequence quality in deep...  
... then we will know how **reliable** are our results
  - QC determines posterior filtering
  - We must be consistent with any filtering decision...  
...if not, downstream analysis will suffer the consequences
  - QC must be test after every critical step

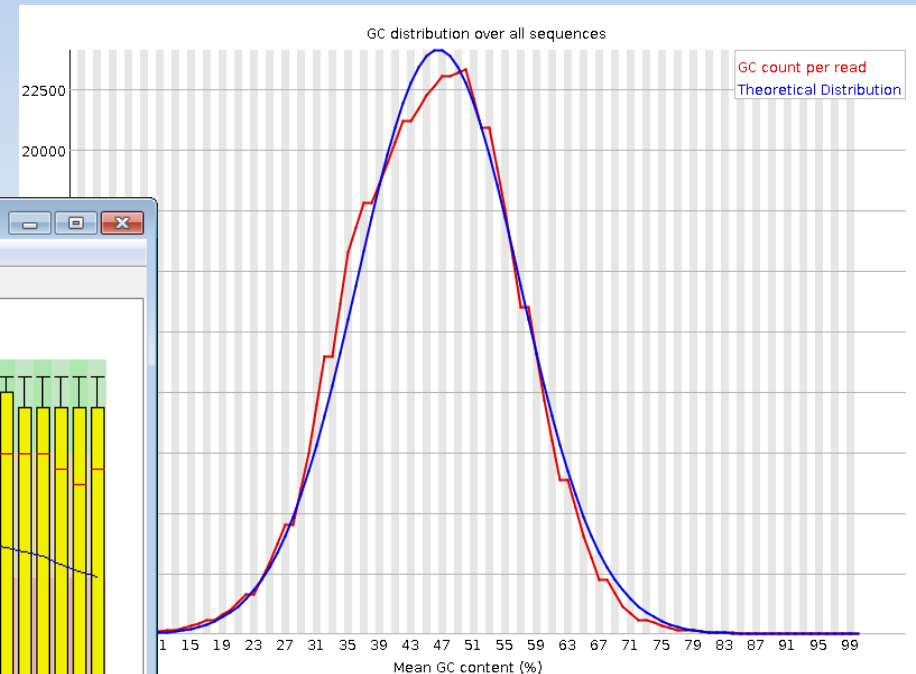
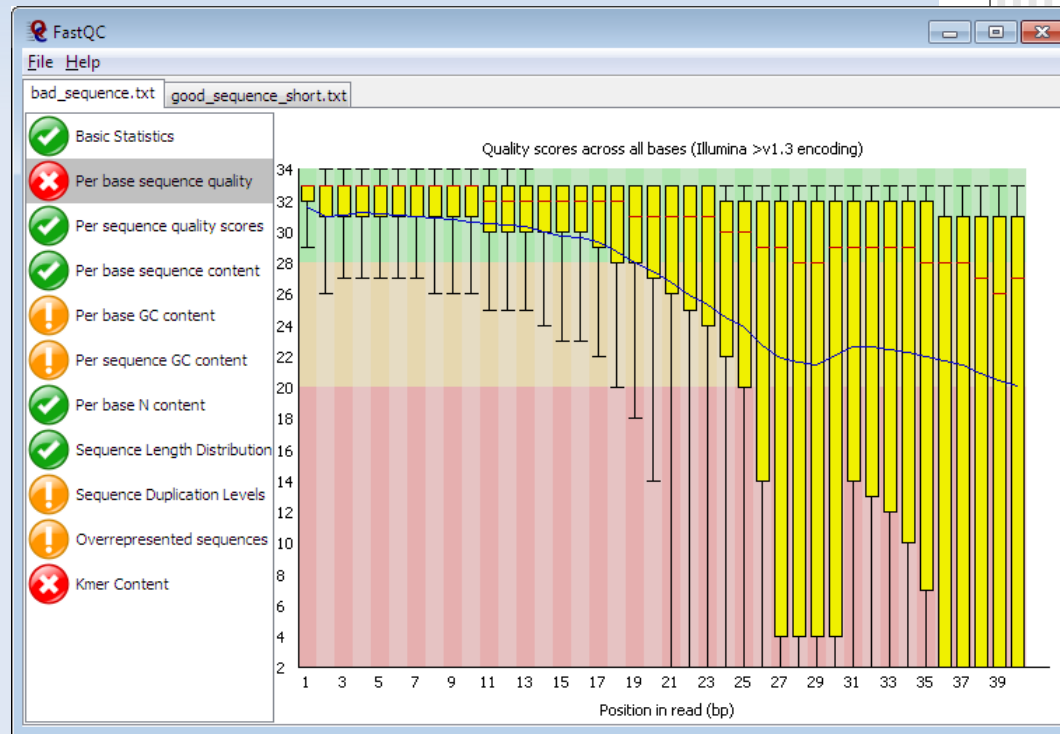
# NGS Sequence preprocessing

- Evaluation of sequence quality
  - How? quality per base
  - Quality (or error probability) will be also a topic in next pipeline steps



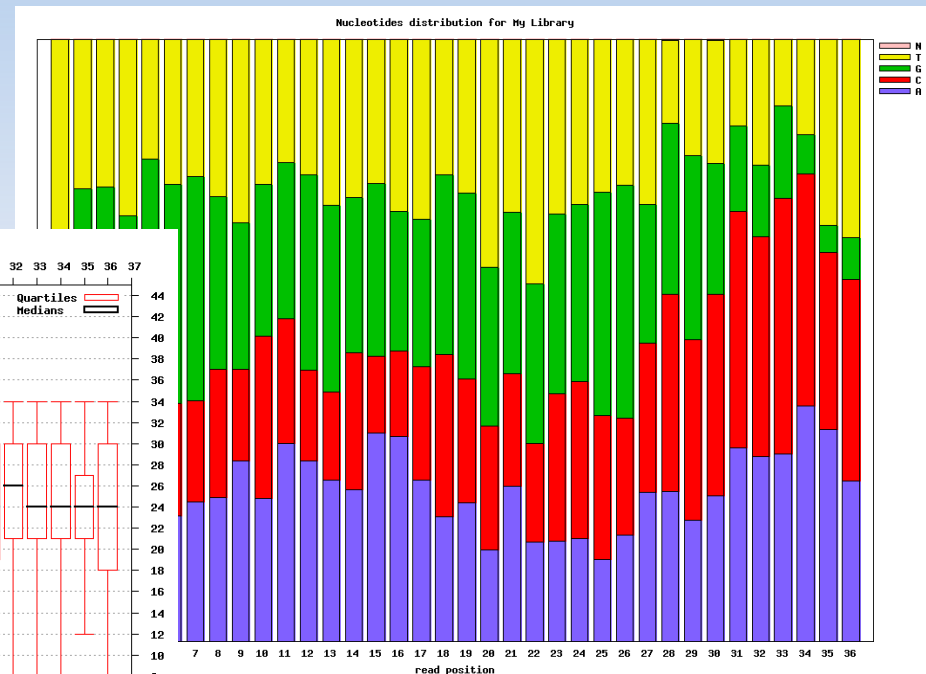
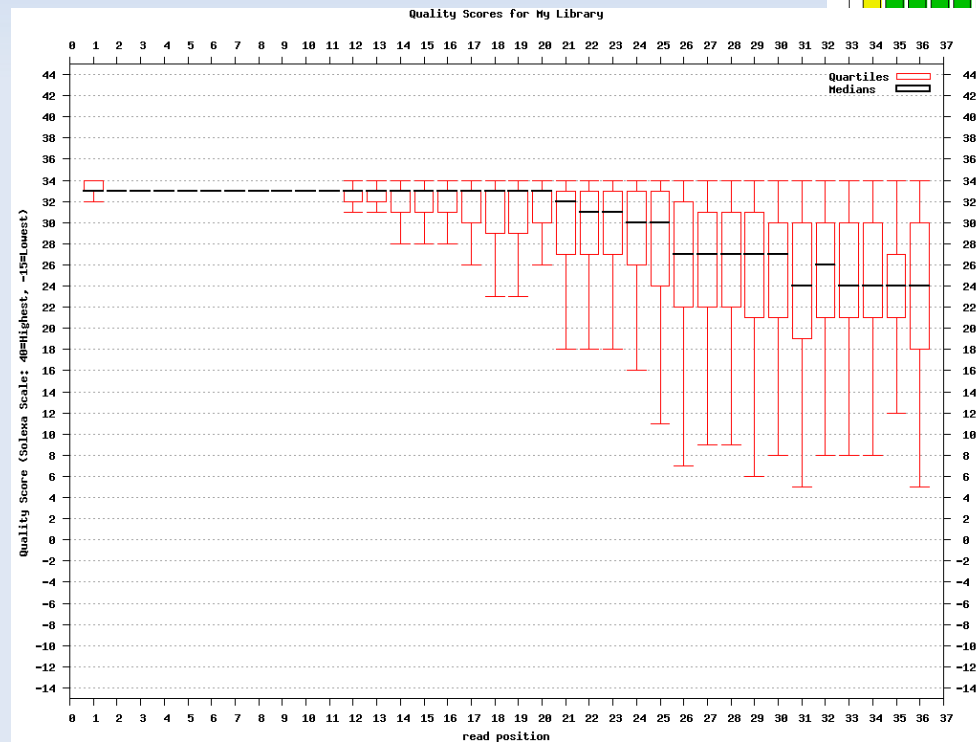
# NGS Sequence preprocessing

- Quality control tools
  - FastQC



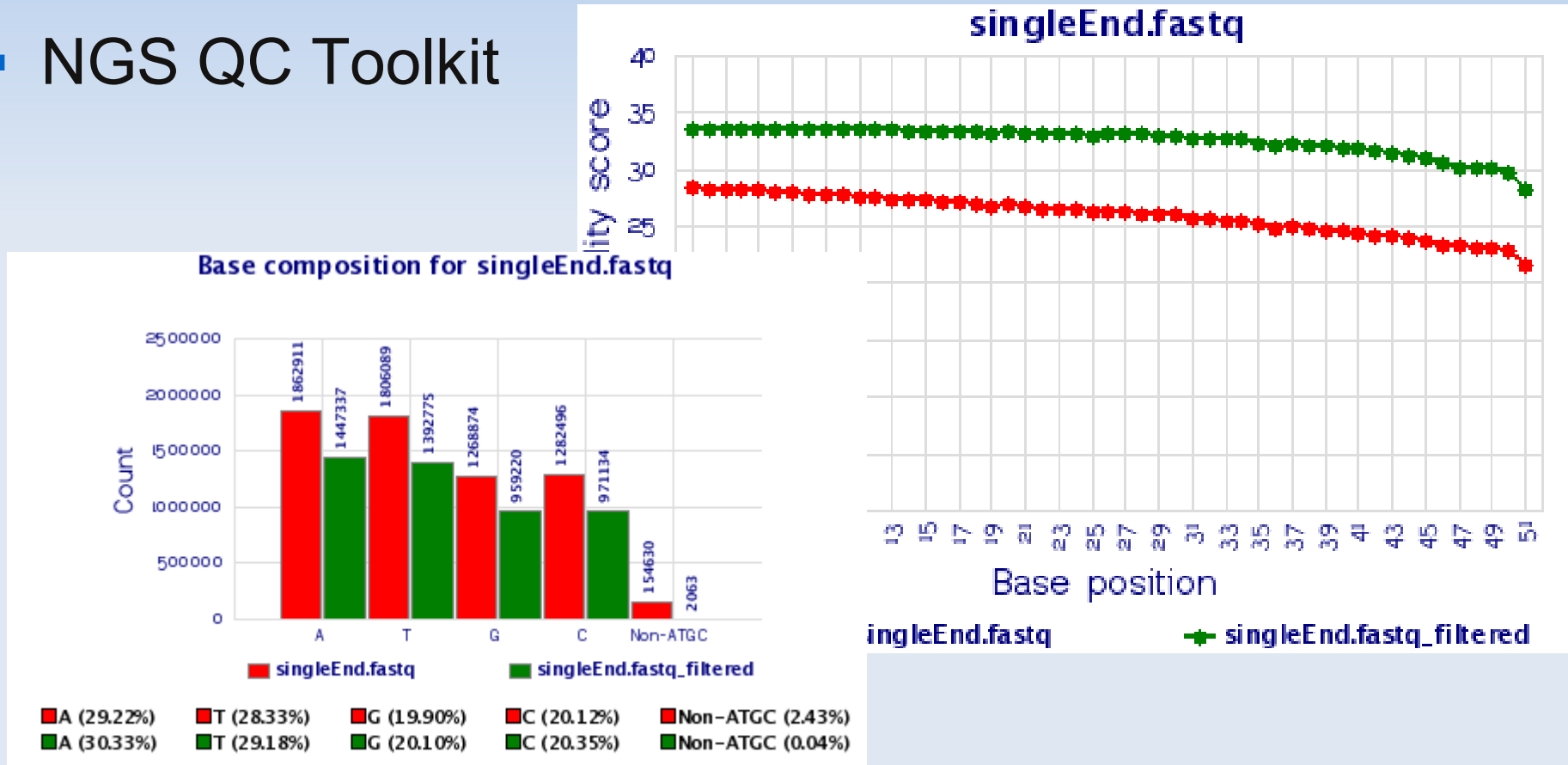
# NGS Sequence preprocessing

- Quality control tools
  - fastx-toolkit



# NGS Sequence preprocessing

- Quality control tools
  - NGS QC Toolkit



# NGS Sequence preprocessing

- Quality control tools
  - Example

## **GOOD quality**

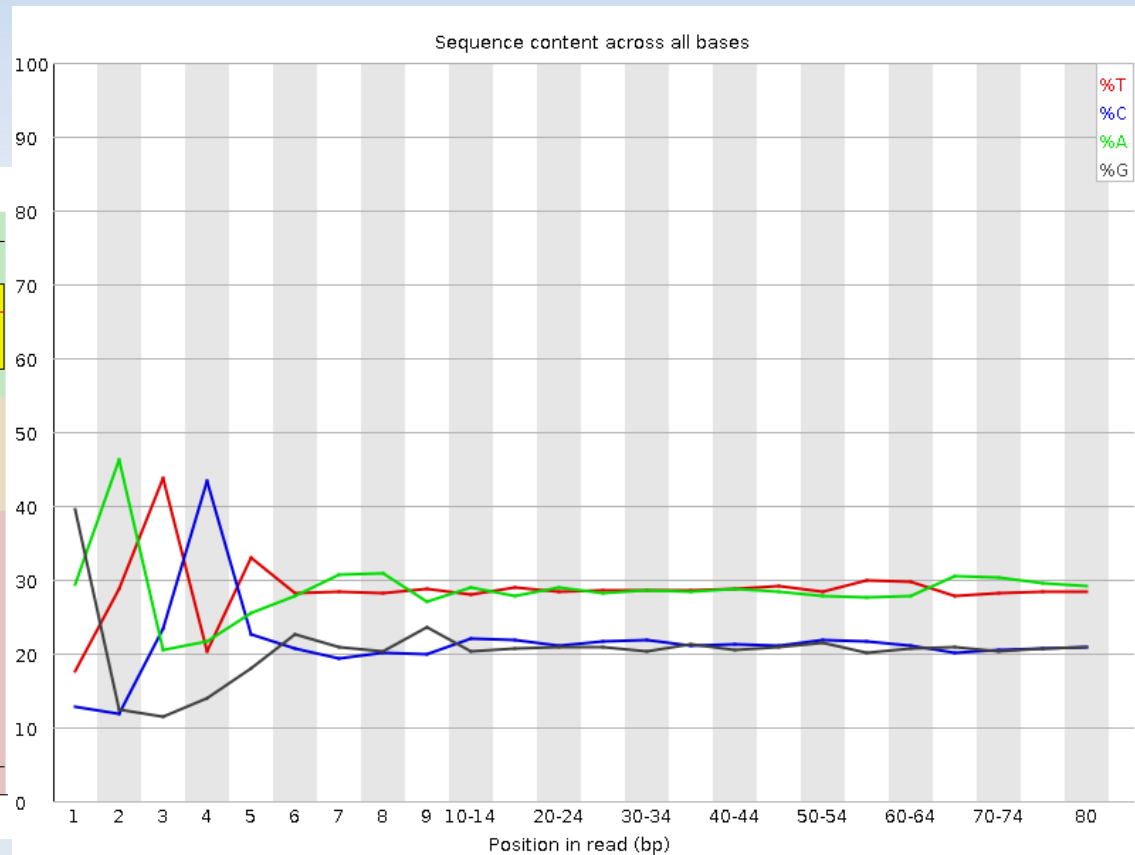
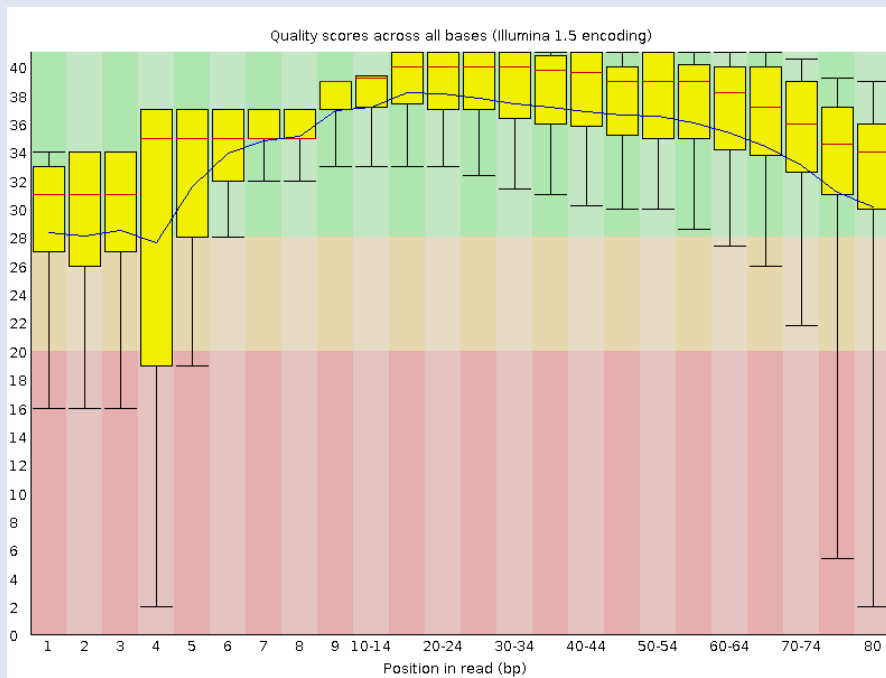
[http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good\\_sequence\\_short\\_fastqc/fastqc\\_report.html](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc/fastqc_report.html)

## **POOR quality**

[http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad\\_sequence\\_fastqc/fastqc\\_report.html](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc/fastqc_report.html)

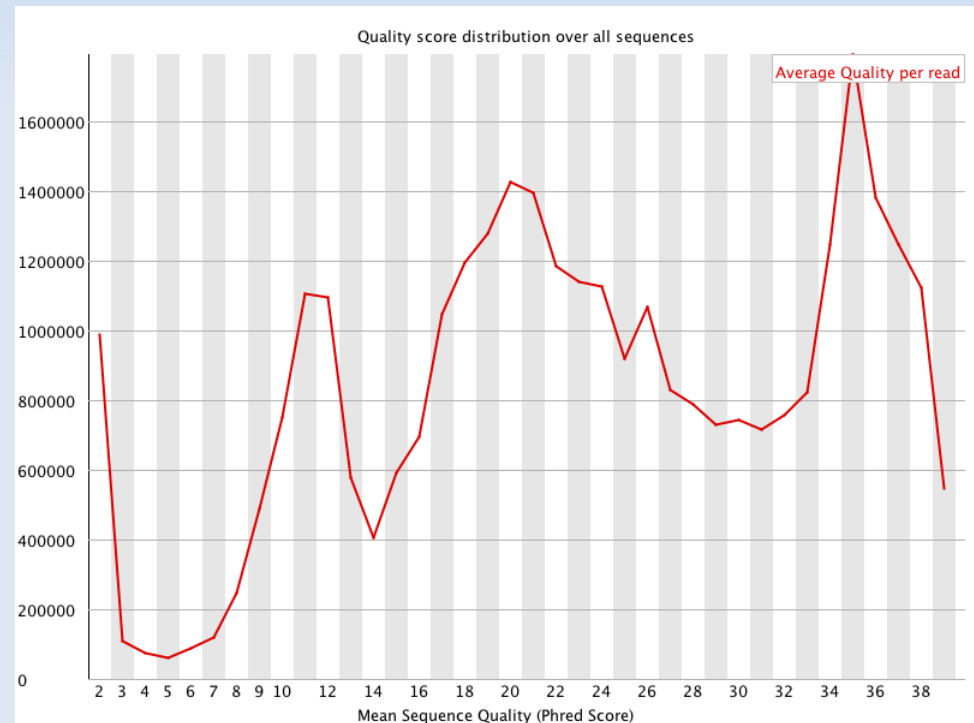
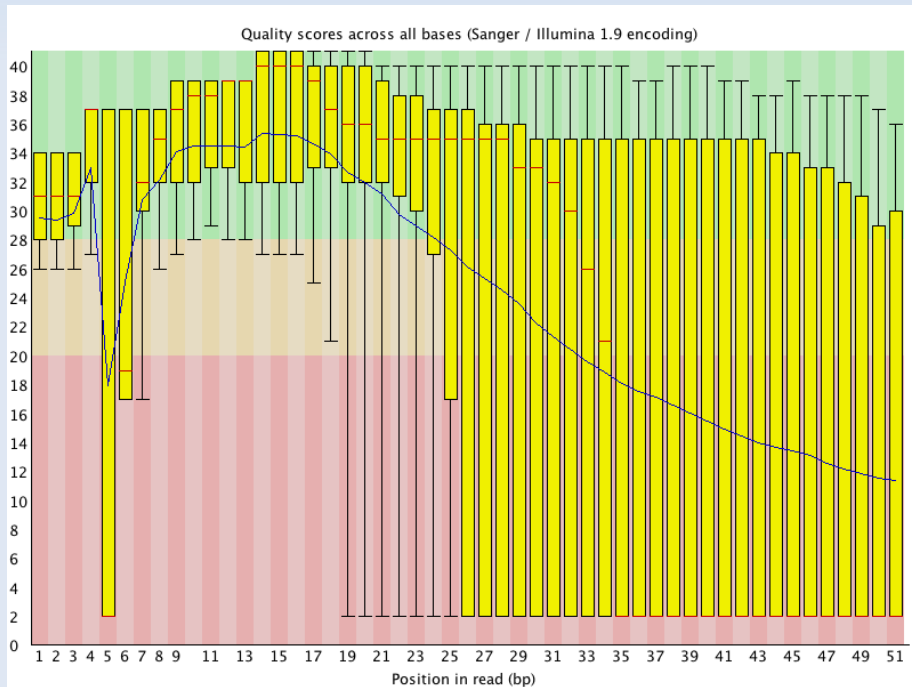
# NGS Sequence preprocessing

- Typical artifacts
  - Sequence adapters



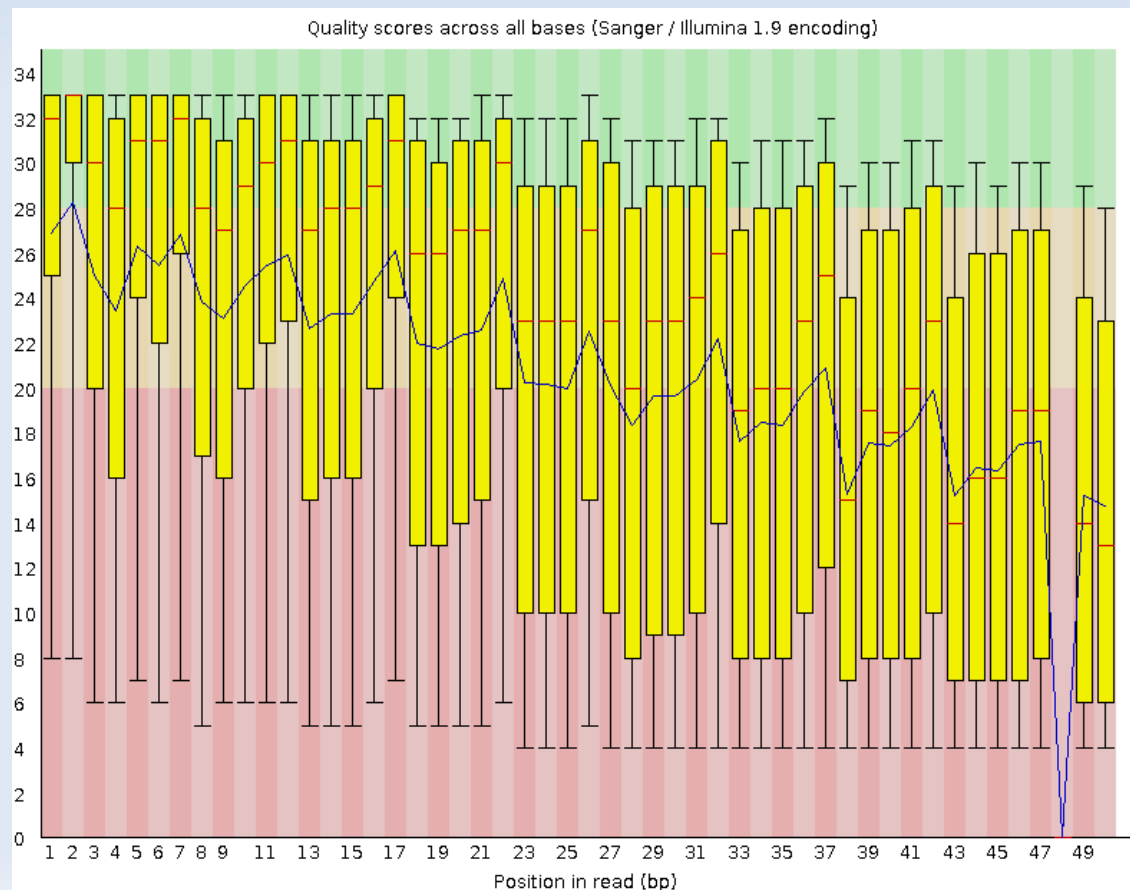
# NGS Sequence preprocessing

- Typical artifacts
  - Poor quality data



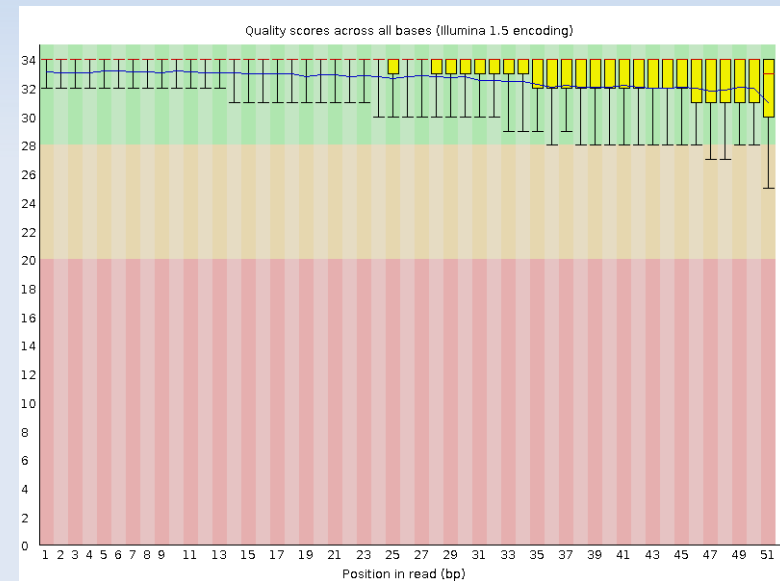
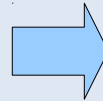
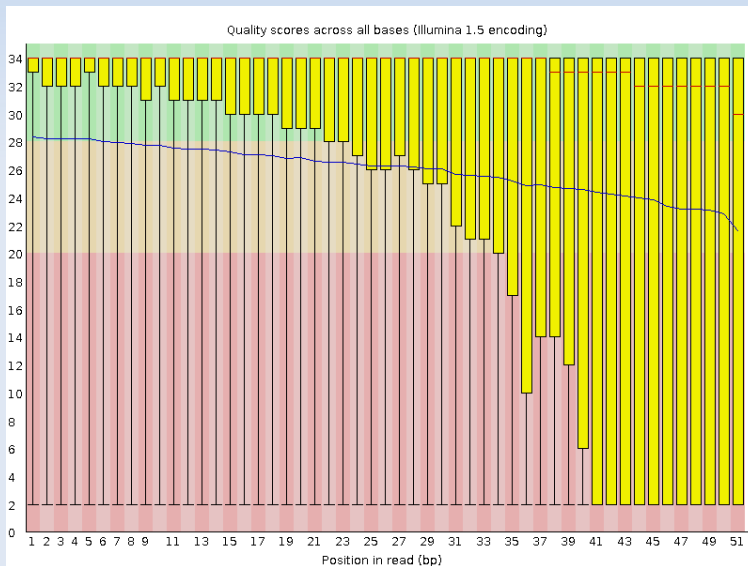
# NGS Sequence preprocessing

- Typical artifacts
  - Platform dependent



# NGS Sequence preprocessing

- Sequence filtering (and editing)

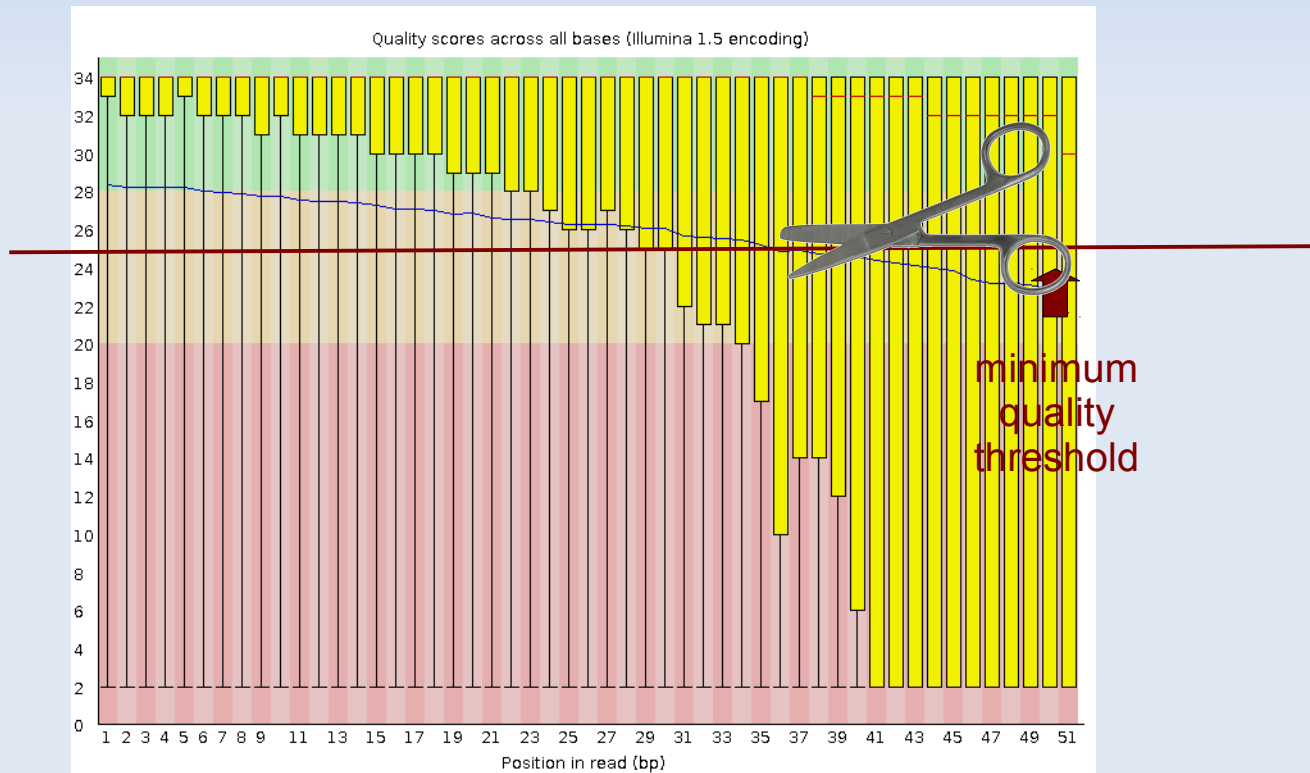


- remove bad quality data
- Improve confidence of downstream analysis



# NGS Sequence preprocessing

- Sequence filtering (and editing)
  - Tail quality trimming



# NGS Sequence preprocessing

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

# NGS Sequence preprocessing

- 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/>)
  - And more....

# NGS Sequence preprocessing

Any question?