

Lecture 3

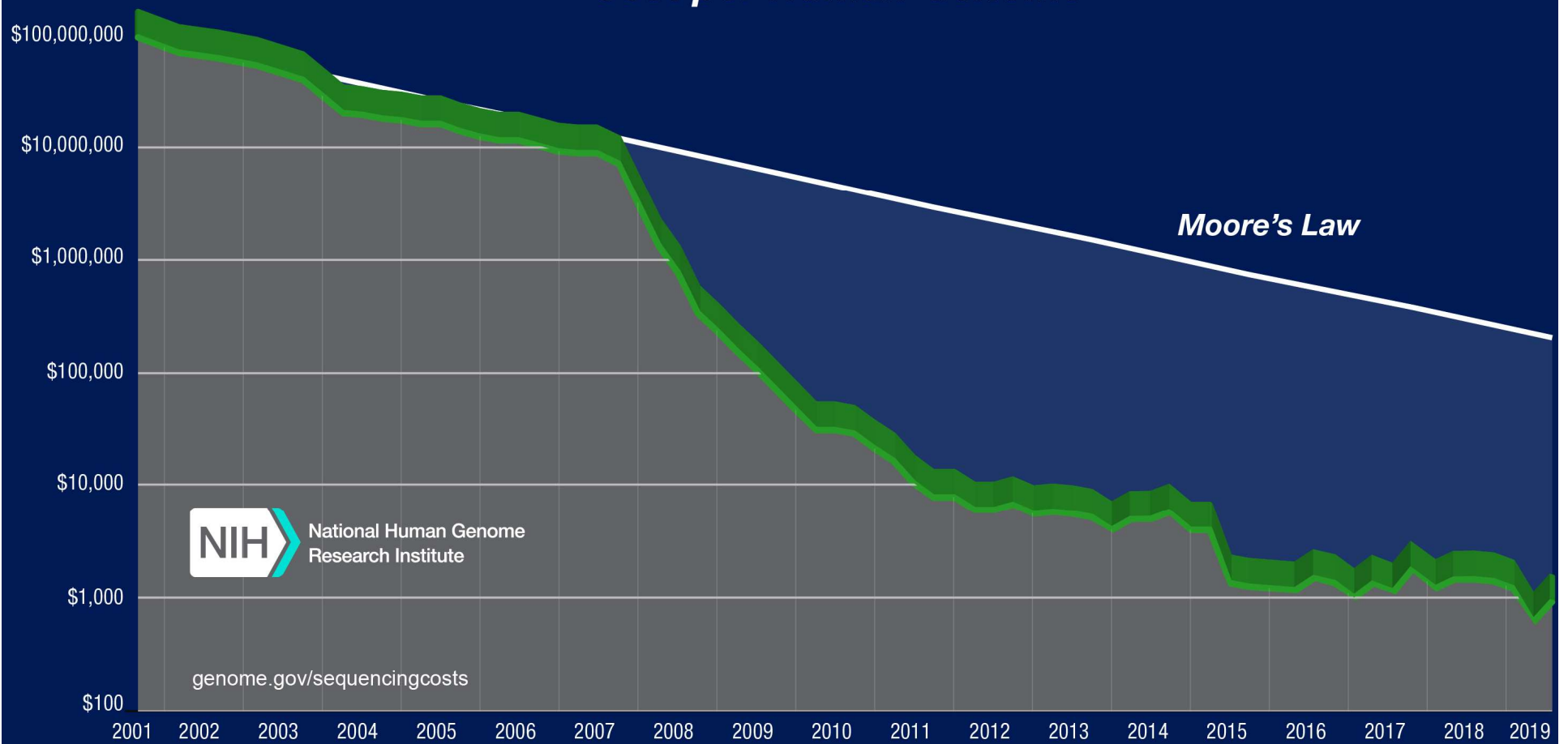
Introduction to bioinformatics (MVE510)

Autumn 2020

Repetition

- Next generation sequencing (NGS) has revolutionized the life sciences and has decreased the cost more than 100,000-fold.
- Common platforms include massively parallel pyrosequencing (454), Illumina, PacBio and Nanopore
- Important applications include genome sequencing, genome resequencing, transcriptomics (RNA-seq) and metagenomics

Cost per Human Genome



Today's agenda

- A closer look at the data from next generation sequencing (NGS)
- Challenges in the analysis of NGS data
- Errors patterns and data formats (FASTA and FASTQ)
- Preprocessing of NGS data: quality and filtering
- Genome sequencing

NGS data is highly fragmented!

- Next generation sequencing data is often highly fragmented
- Short read fragments are between 50-300 nucleotides. A single read will thus only partly cover a gene/exon/intron
- The reads are randomly selected from the input DNA
- Multiple reads are often necessary to full describe a genomic region of interest

NGS data is massive!

- Next generation sequencing generates large volumes of data!

Example: Sequencing of a human genome

- Size of the genome: 3.2 gigabases (3.2×10^9)
- Genome needs to be covered several times to avoid errors (often 50 times).

Data size: At least 160 gigabases (160×10^9 data points)

NGS data is dirty!

- NGS is error prone and contains many forms of errors:
 - Substitutions
 - Insertions/deletions
 - Duplicated reads
- The type of error depend on the sequencing chemistry. Different sequencing platforms therefore produce different forms of errors.
- The errors needs to be identified and removed before the data can be trusted.

Common errors: substitutions

Correct sequence

GGCGCTGGACTCTACAGCAGATGTGGAACTGGAGA

CGCTGGGCTCTACATCAG

GGACTCTACAGCAGATGTGG

GACTCTACAGCAGATGTGGA

TCTACATCAGATGTGGAA

CAGCAGATGTGGAACTGGAG

Sequence reads

Common errors: Insertions and deletions ('indels')

Correct sequence

CTTCATAAGCTAGATGCCAGTTAA-CTGTCGAGAGG

CTAGATG- CAGTTAA-CTGTC

AGATGCCAGTTAA A CTGTCGA

ATGCCAGTTAA-CTGTCGAGA

TGCCAGTTAA-CTGTCGAGAG

TGCCAGT- AA-CTGTCGAGAG

Sequence reads

Common errors: duplicates

- Duplicates are caused by sequencing the same DNA fragment multiple times. These reads all come from the same DNA molecule and does not describe the true diversity in the sample.
- Duplicates typically caused by biases in the amplification steps where certain DNA fragments are amplified with higher efficiency. More amplification often means more duplicates.
- In many applications, duplicates are important to remove to avoid incorrect and misleading results.

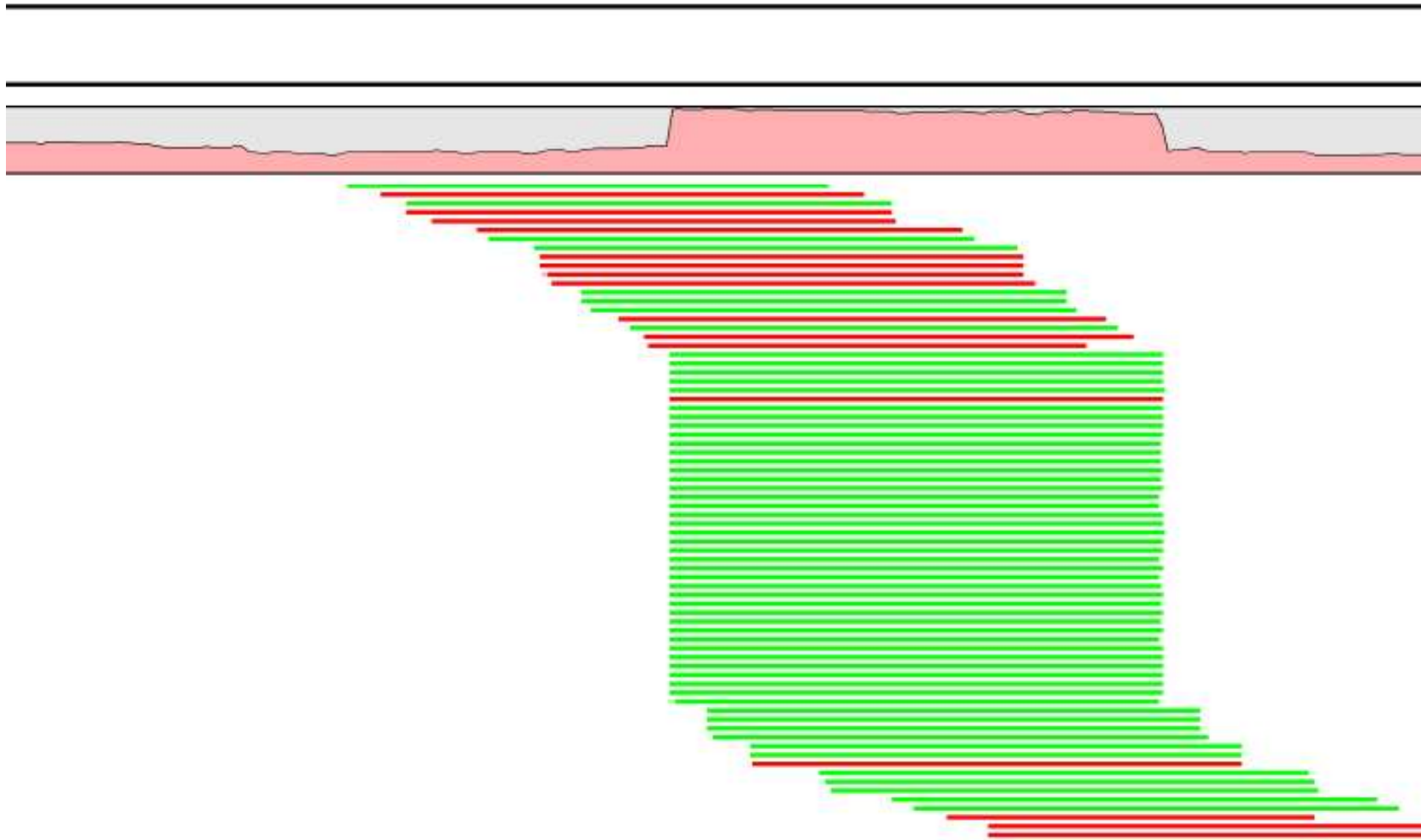
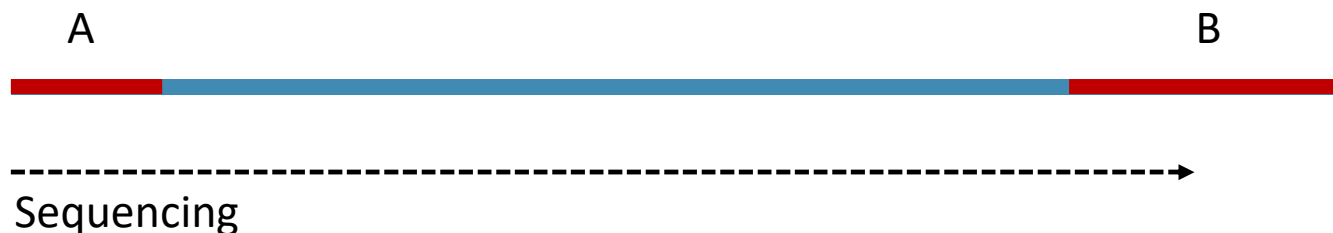


Image from
http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/802/index.php?manual=Remove_duplicate_mapped_reads.html

Common errors: adapter contamination

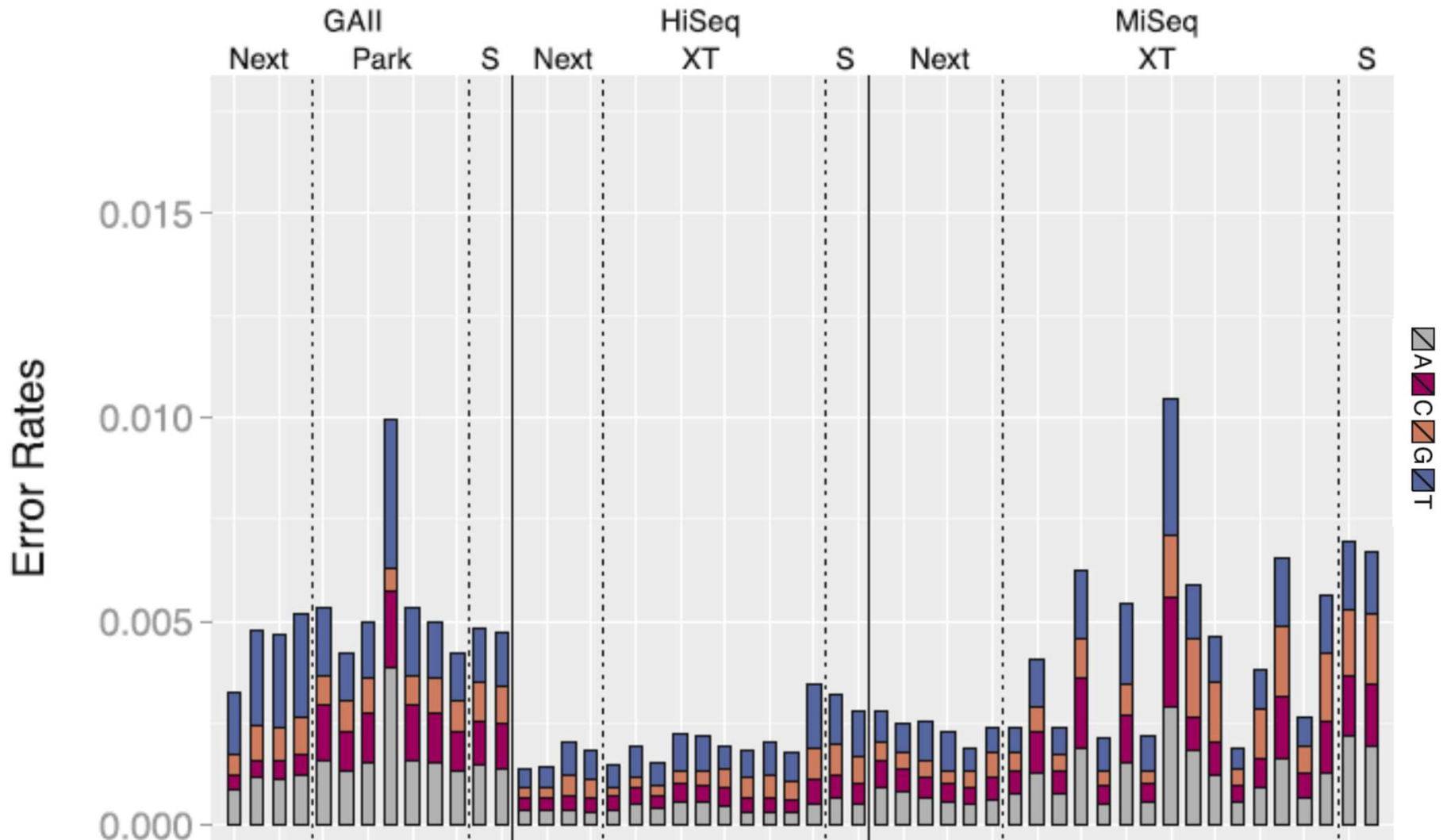
- Most sequencing platforms adds adapters to the ends of the reads
- Typically one of the adapters are sequenced and needs to be removed before analysis.
- However, if the DNA fragment is too short, the sequencing process can start to also sequence a part of the other adapter.



Errors in Illumina data

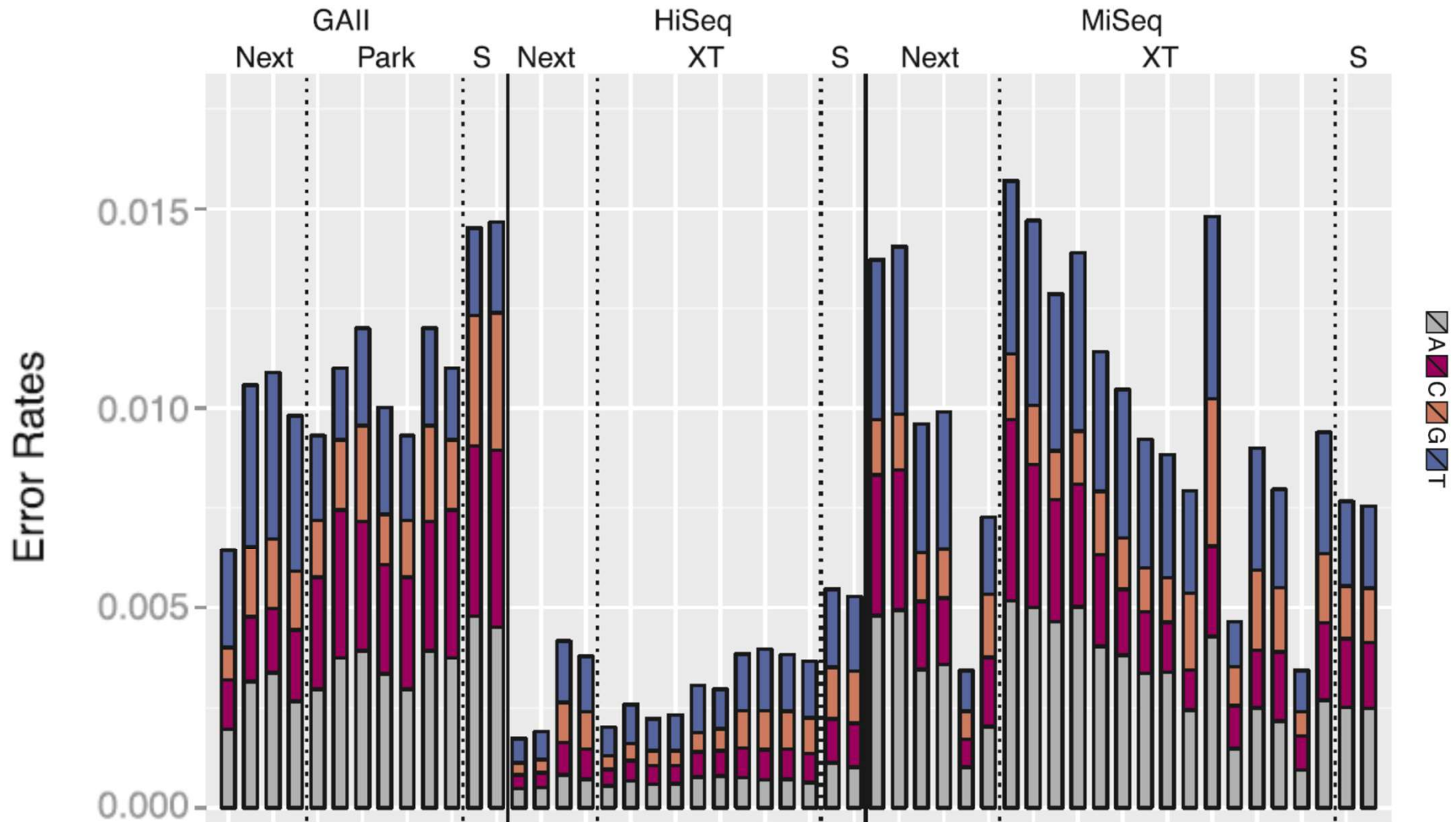
- Illumina sequencing has a error rate up to 1%.
 - Error rate for substitutions: 0.1-1%
 - Error rate for indels: 0.01%
- The error rate is dependent on
 - The position on the read. The probability for an error increase for each sequenced base pair
 - The genomic context. T has a higher error rate than A, C and G. GC-rich patterns, such as GGX has a higher error rate.
 - In paired-end sequencing, the first read has a lower error rate then the second read.
- Duplicates can also be common in Illumina data if the sample preparation is not done properly (or if the starting material is limited).

Errors in Illumina data: substitutions in the first read



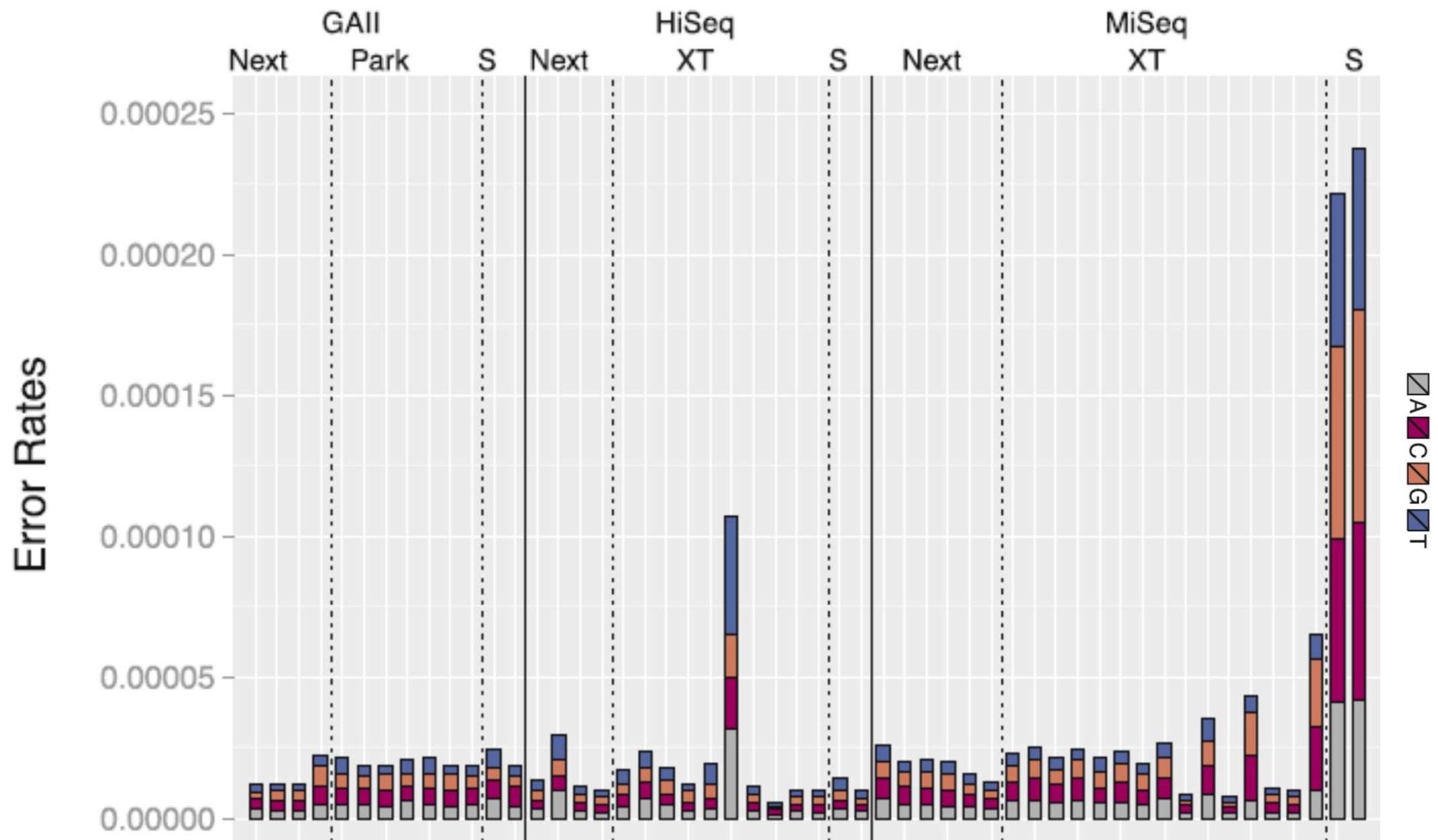
Schirmer et al, Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data, BMC Bioinformatics 17:125 2016.

Errors in Illumina data: substitutions in the second read



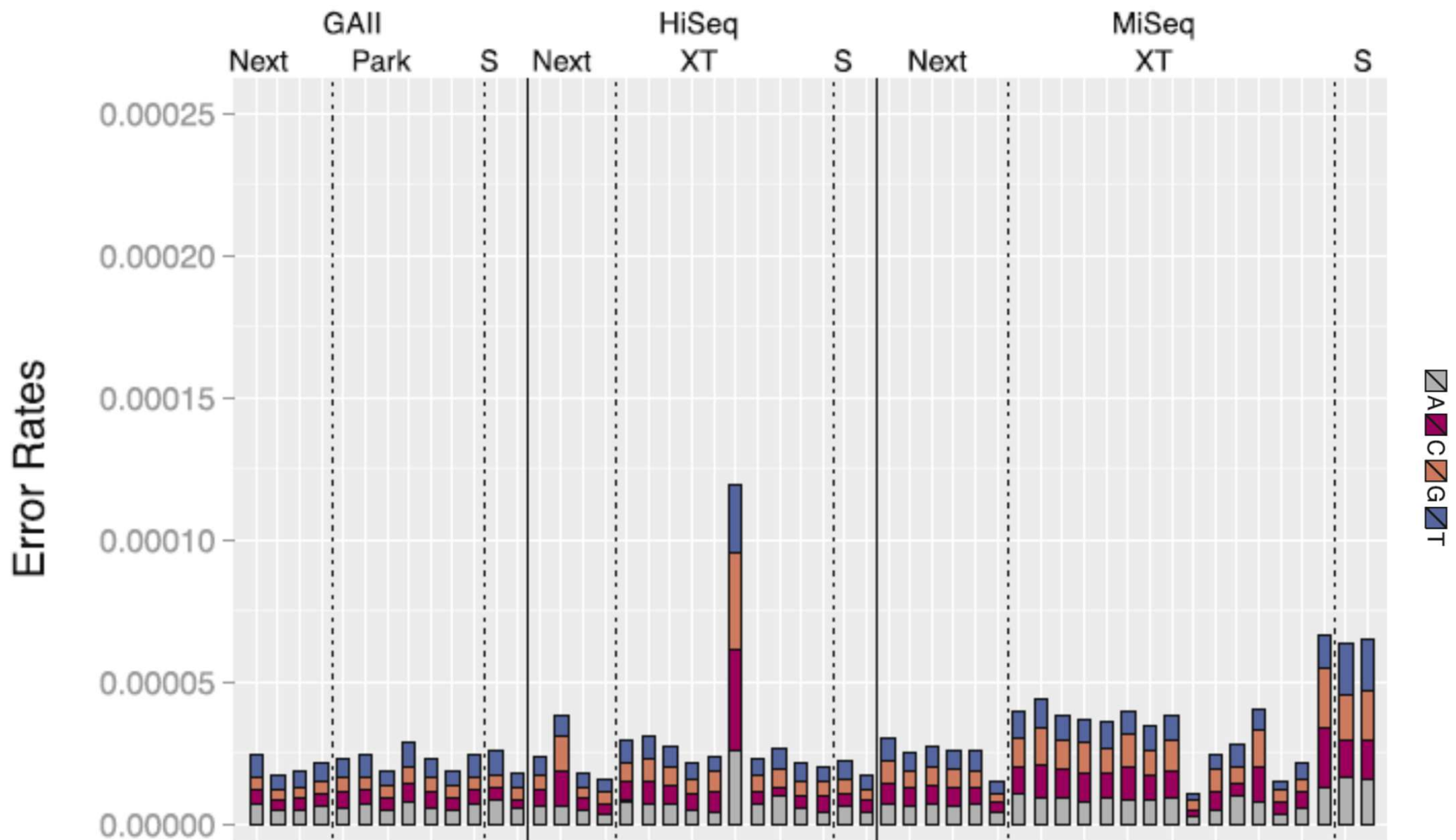
Schirmer et al, Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data, BMC Bioinformatics 17:125 2016.

Errors in Illumina data - insertions



Schirmer et al, Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data, BMC Bioinformatics 17:125 2016.

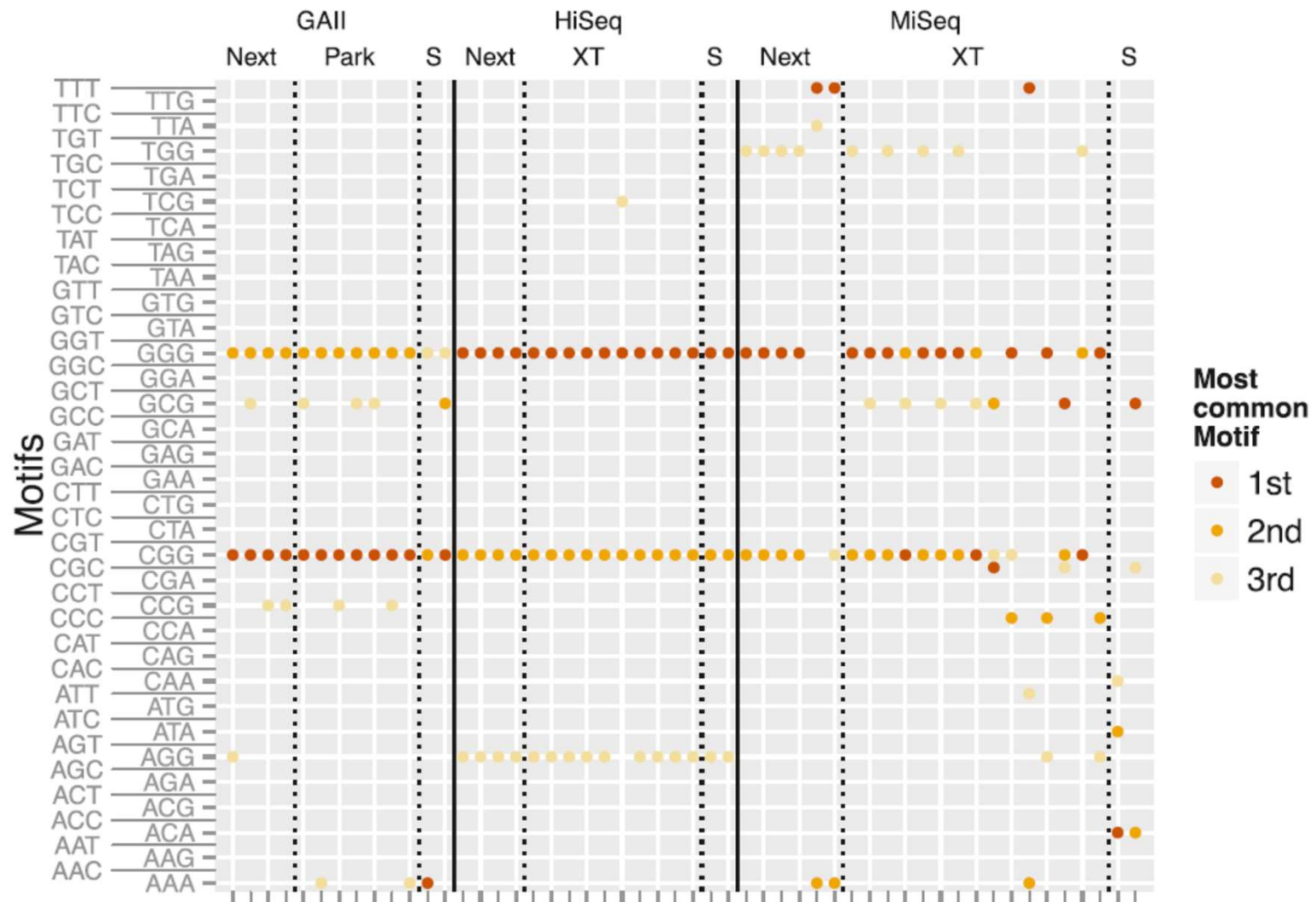
Errors in Illumina data - deletions



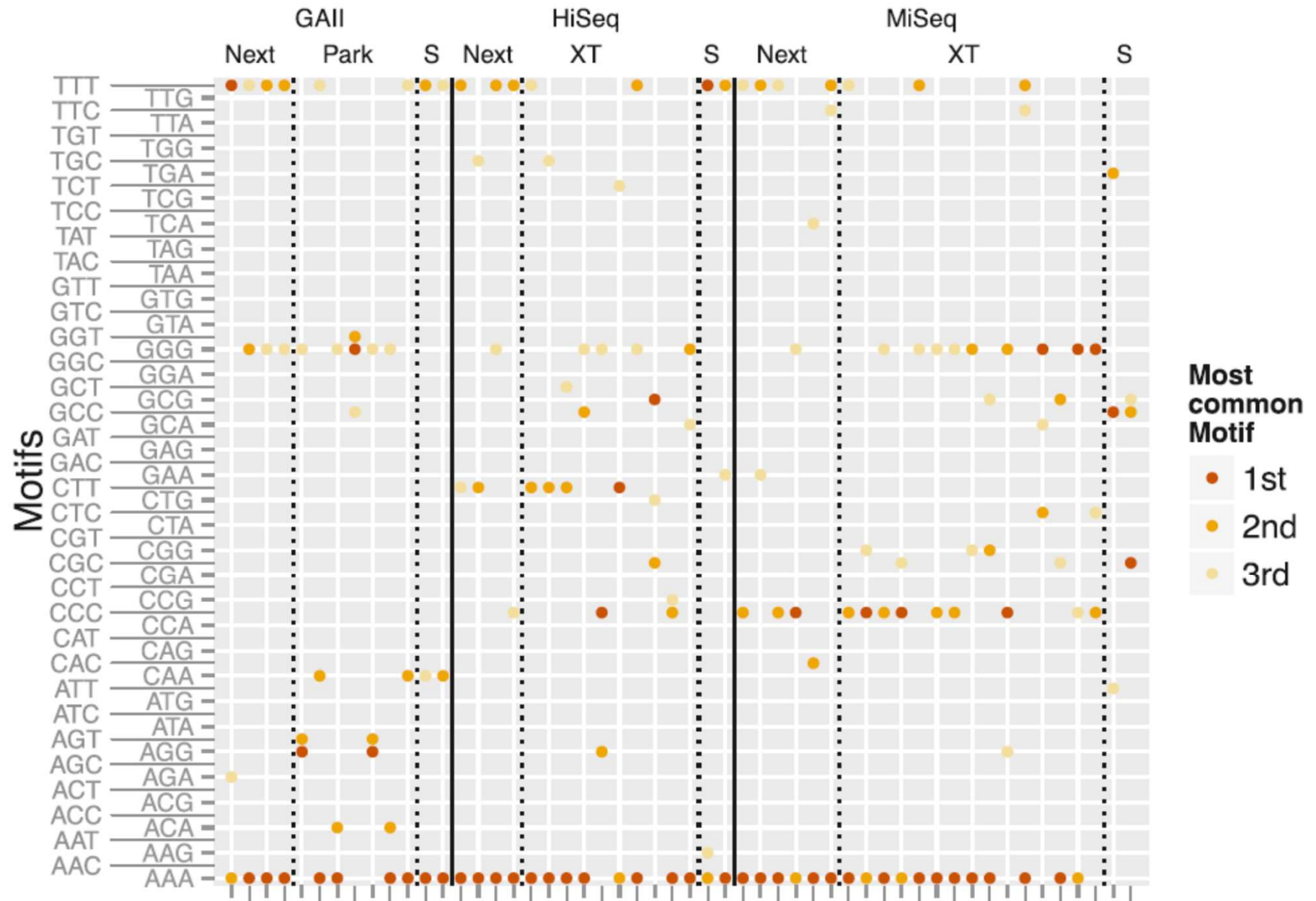
Schirmer et al, Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data, BMC Bioinformatics 17:125 2016.

Site specific errors (SSE)

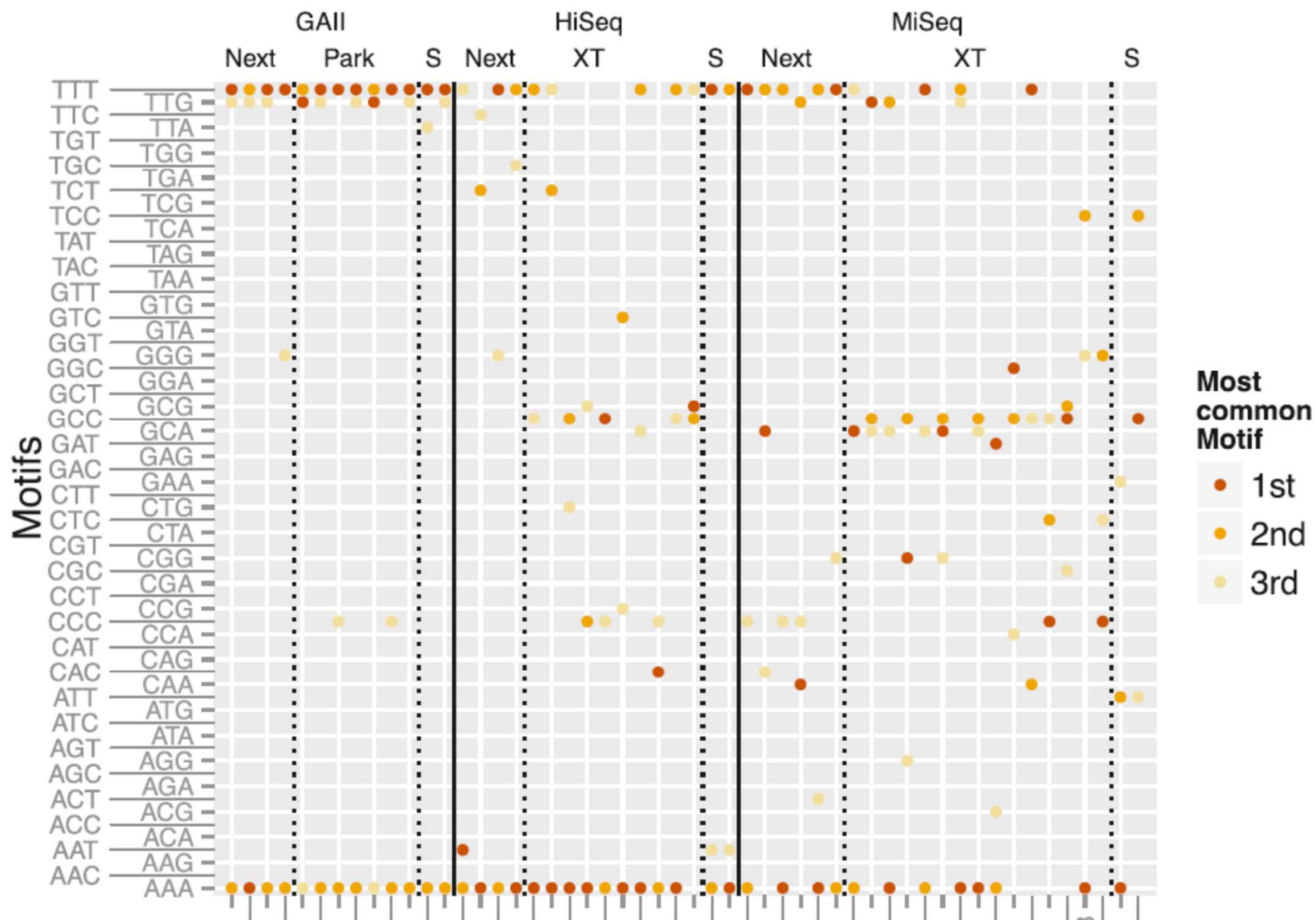
Top 3 Motifs in R1 reads for Substitutions across all DS

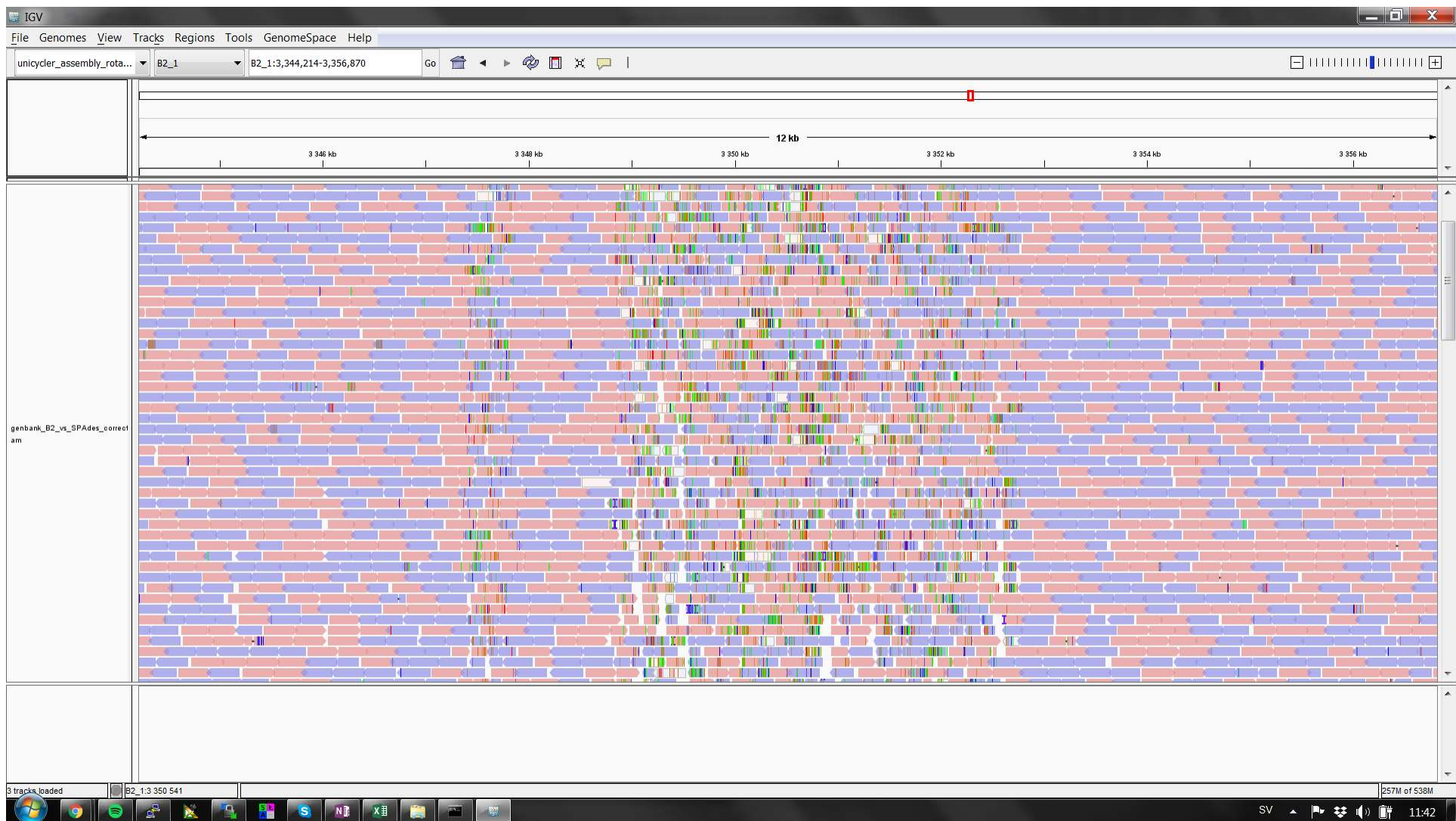


Top 3 Motifs in R1 reads for Insertions across all DS



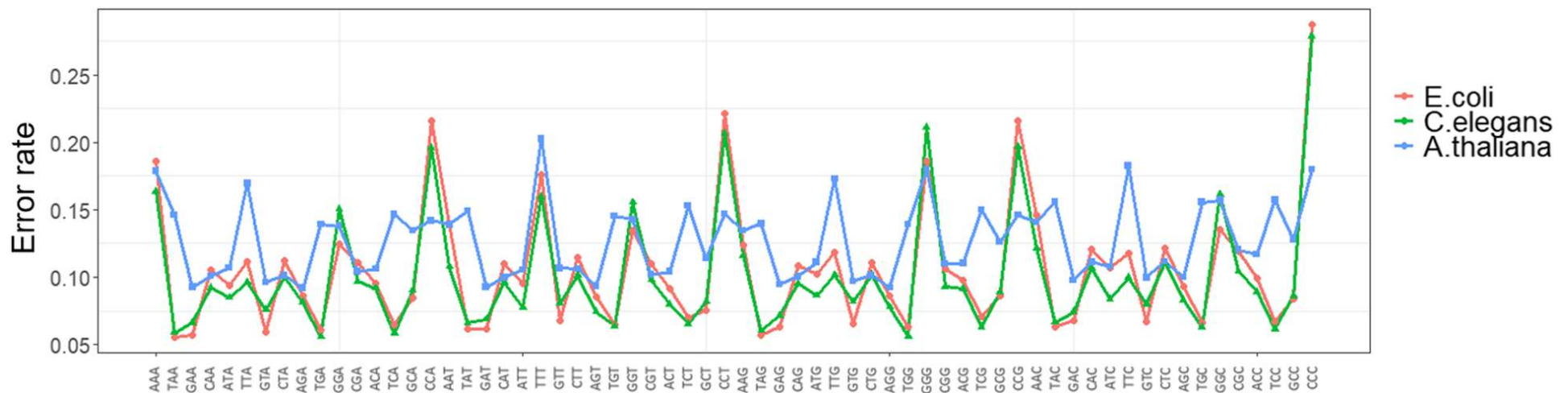
Top 3 Motifs in R1 reads for Deletions across all DS





Errors in PacBio data

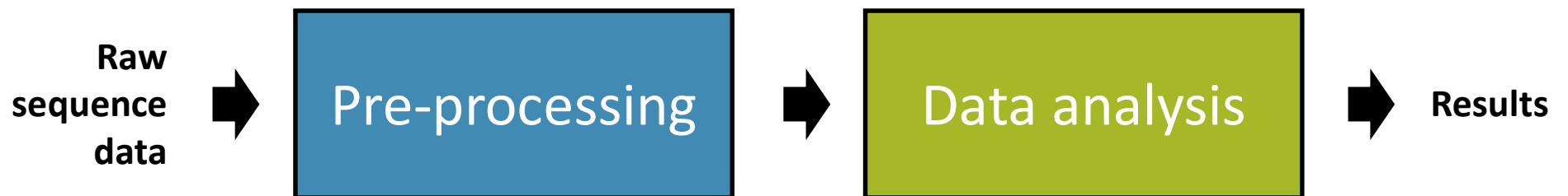
- The error rate in PacBio data is high, up to 15%
- The errors should not be dependent on context and uniformly distributed over the read
- Sequencing the same region many times is therefore an efficient way to remove errors in PacBio data



Analysis of NGS data

Pre-processing is the first step used to 'clean' NGS data

- Identifies erroneous reads and base pairs
- Cleans data by remove errors
- Important to ensure a correct down-stream analysis



Data formats for NGS data

- FASTA is the standard format for storing DNA sequences (.fasta, .fa, .fna)
- Text-based and 'human readable'
- Many programming languages has parsers for FASTA files.
- Can also be used to store peptide sequences.
- FASTQ is similar to FASTA but contains also a quality score for each nucleotide (.fastq, .fq).

Representation of sequence data – the FASTA file

Read name

Read
sequence

1

```
>M00283:44:000000000-A52U8:1:1101:14152:2164 1:N:0:10  
GTATANCCTGCTGATGAACAACGACAGCCTGTTACNGNNNNNNNACCATTCAC TNACAAT  
GGCGGTGTTTTTGGCGACGTTTGCTACGCCTTTCCGTTCACTGACGGCAGTTTCGCNNNN  
NCGTTATCGCTTTTCGNNNNNNNNCCGGTCGTTAGCGGAATGCTGNTTNNNGTGTTGATGTC  
ACNAAATATCGTAATGGCGC NNTNNCAGCGTGTGCTGTTTATNCTGCTTGGTATTCAACC  
TGGCTTCTGGT
```

2

```
>M00283:44:000000000-A52U8:1:1101:18996:2165 1:N:0:10  
ACAAANATGGCACAGGCGATAAAAAGCACCGCAAACANNNNC NNCATGCCGTA CTGATCC  
CAGATACGCCCGAAGCTGAATGACGACTTAGGTGCGCCAGAACCCGATGTAGAAACNNNN  
NACATCATACTCTCCNNNNNNNGGCAACTGCCTGGCTGACTTTNGGN NNCGCAAGGCTCA  
GTNCCTGACGCTCATCTGCCNGC NNGTGTAACAATTCACCGGNGATTTGCCTTCCCGCA  
TCACCACAATC
```

3

```
>M00283:44:000000000-A52U8:1:1101:14478:2166 1:N:0:10  
CGCATNATGTATTGATAACCGCGTTGGGT CAGCTCCGNNNNGN NCGCTCCCGGCGAGATC  
ATCAGAATACCTTCGTCTTCATAGATATCTGATGCAGGT TGGGTAGAGGAGGAACANNNN  
NGCCCAATAACGTATNNNNNNNCGTCGTTAACGATTTTGT TGGNGANN NCAACGGCTTGT  
TTNGGGT CGCAGGCGTCGTCNTANNCCACGCCAACCAGTTTANCGCCCTTAATCCCCCT  
TTGGCATTAAAT
```

4

```
>M00283:44:000000000-A52U8:1:1101:13779:2176 1:N:0:10  
TGCTGNAAACGTGGTGTACCGGCTGTCCGGTATGTATNNNN TNGTGGTGAATAATGCCCC  
CGAACAGGCAGAGGACGCCGGGCCTGCAGAGCCTGTTTCTGCGGGAAGTGTT CGACNNT  
NAGCTGAGTTTTGCCNNNNNNNTGGCGCGTGAGATGGGGCGACCCGANNGGCGTGCCATG  
CTNGCCGGGATGTCATCCACGGAGTATGCCGACTGGCACCGCNTTTACAGTACCCATTAT  
TTTCATGATGT
```

IUPAC DNA codes

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

Representation of sequence data – the FASTQ file

Read name

Read sequence

1

@M00283:44:000000000-A52U8:1:1101:12571:2232 1:N:0:1

ACGACCAGAACGACCGCGCAACTGGCTATCGATACGACGGGATTCGTGACGCACGGTACCGATGATATGCAGGCCACCTGCTTCCA
GTACCGCATCGTGACGTACCTGCCAGTCGGCTTTAATTTTTTCAATTTGCTCTGCGGTCGGATTTTCCAGCGCGGCAACTTCTGCC
TGCCAGCTACCACCGAGCACAAATATCTGTACCACGACCCGCCATATTGGTCGCGATAGTCACCGCATCCGGATACCTTG

+

A?A1ADDDFFFAA100AAAA/EGFB/BFAF///F///AA/E/?01/B/0F/>/>E/?EFE///FD22FGBGGECFC<F0FE1FG1
GFDBDA/BCCCFA0?CFGHGHH0<1?F1C-CEGB<1<<BD0CH00=DGB00<=0C?C:@C--9ABFFFFEF-A@@@-9EFFFFF
B/FF//:B//-/BEF-9---BFB///:999/;9BB-@-@9@@?-/99BFF/-----:9;/99BF@---/-:-9-//;-/-

2

@M00283:44:000000000-A52U8:1:1101:13922:2234 1:N:0:1

AACGGCACGGTAACCTGACGCCGAGGAAGCTATTTTCCAGGCCTGTCTGCTGCGTTTTCGCCGATTATGATGACTACCCTGGCG
GCGCTGTTTGGCGCGCTGCCGCTGGTGTGTGTCGGGCGGTGACGGCTCGGAGCTGCGGCAACCCCTGGGGATCACCATTGTCGGCG
ACTGGTAATGAGCCAGCTCCTTACGCTGTATACCACGCCGGTGGTGTATCTCTTTTTTCGACCGTCTGCGGCTGCGTTTT

+

BCCCBCCCCCBGGGGGGGGGGGGGGGGGGHHHHHHHHHHHHHHHGGGGHHHHHHHGEggGHGGGGGGGGHHHHHHHHHHHHFFHGE
GGGGGGGGHHHGHGGGGGCGHGFGGGFFHGHGGGGGGGGGGGDDGBFDFFFFFFFFFAFFFFEFFFFEFFFFFFFFFFFFFFFFFFFFF
@FA.EFFFFFFFFFFFFEBFFFFFFFFFDFFFFFFFFBFFFEFD@-DAEFFFFFFFFFFFFCEFFFAA.AF-@DDBEFD.@E

3

@M00283:44:000000000-A52U8:1:1101:16240:2234 1:N:0:1

TCACTTTCGCCTCAATATGAACCTCACCACGATCGTTAAAGATCCTGACTTTATCGCCGTTGTTGATACCGCGTTTTTGCGCATCA
AGCGGGTTGATCCACATTTCTGACGGCAAGCTGCTTTCAGCACATCAACGTTGCCGTAAGTTGAGTGAACGCGAGATTTATAGTG
GAAACCCGTAAGCTGCAGCGGATACTGTTTGTTCAGCGGATCCTGATAACTTTCAAAGCCCGCGGTGTAGATCGGCAGG

+

BBBBBB4DABBBGFGGGGGGGGGHHHHHGH2AFHGGGHGFGHHFHGHGGGEFHHHHGDGGGGGEHBFHGHGGGEEGGFFG?EGGGH
FHGDC@<FEFFHHGFGHHGGFHF3FE?CF/FCCCGHFHHHG1GGFGFHBG<FGGGHC<?EAHHFFHHFDGH.CCDGCC0CHB00CG
.FFGF?DDGCFGBBFFF9C9---AFFFFFEHBBFFFBDF@-@FFFFBFFFFFFFFFE9;;FFFA;DFFF;BFFBF.9BD--

Sequence quality score

Quality scores (PHRED scores)

Describes the probability of errors. If a base i has an error probability p the quality score q is given by

$$q = -10 \log_{10}(p)$$

If a base i has a quality score q the error probability is given by

$$p = 10^{-q/10}$$

Quality scores (PHRED scores)

$$q = -10 \log_{10}(p)$$

$$p = 10^{-q/10}$$

Error probability (p)	Accuracy	Quality score (q)
0.25	75%	6
10^{-1}	90%	10
10^{-2}	99%	20
10^{-3}	99.9%	30
10^{-4}	99.99%	40

Quality scores (PHRED scores)

The quality score are encoded to save disk space.

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

Source: www.LookupTables.com

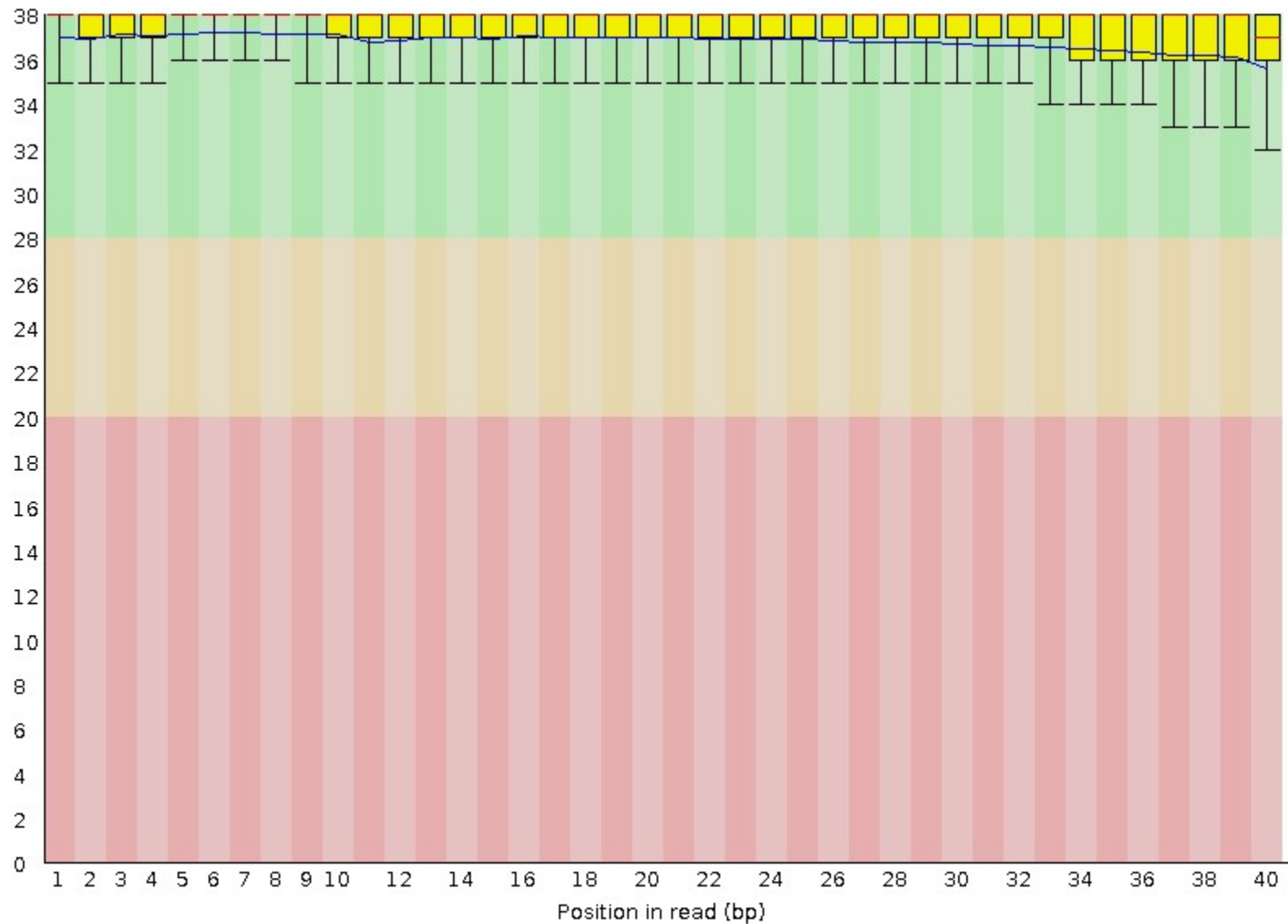
Encoding rule
33+quality score

Examples

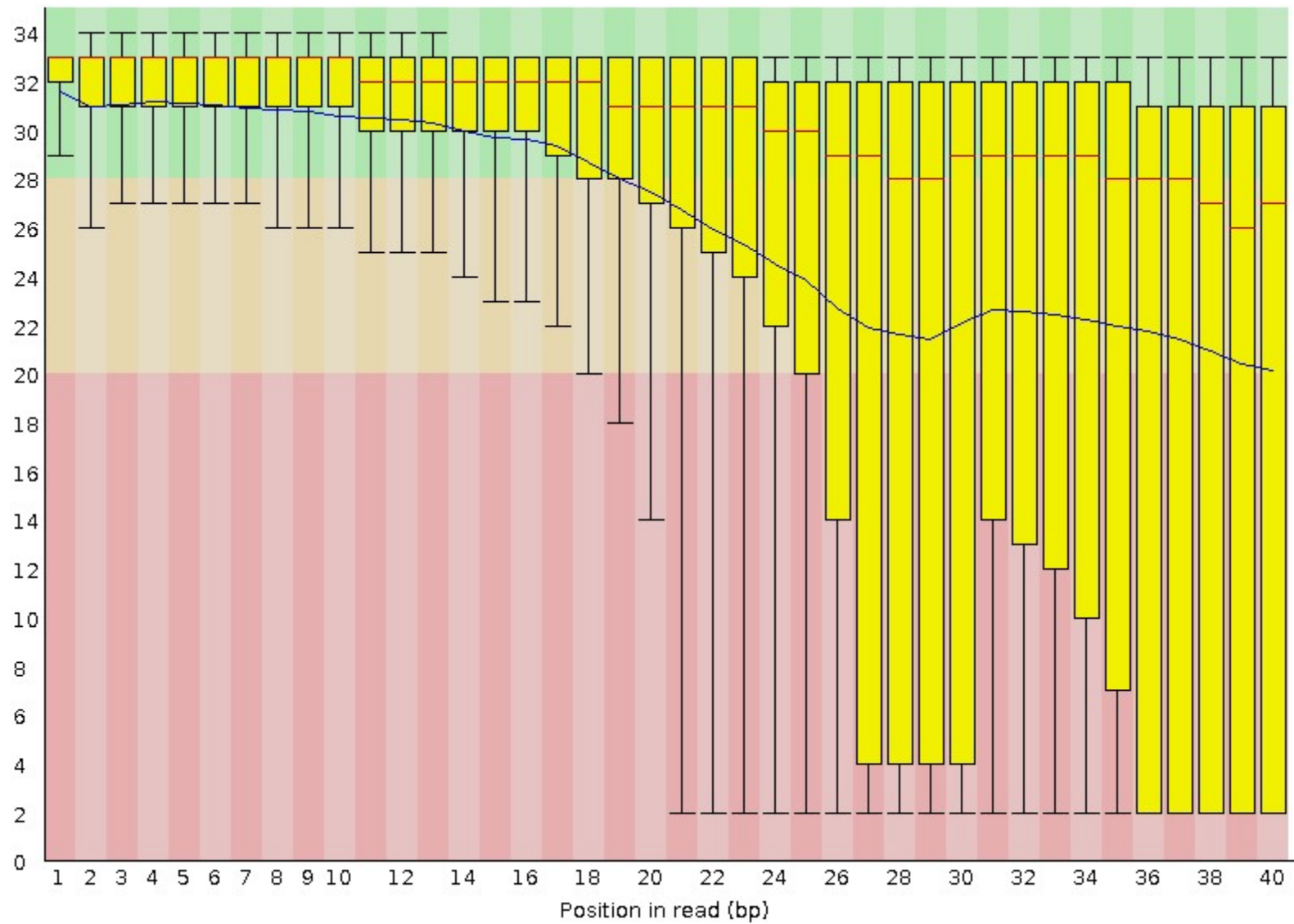
Quality score 2 gives '#'. Quality score 20 gives '5'

Quality score 37 gives 'F'.

Pre-processing - FASTQC

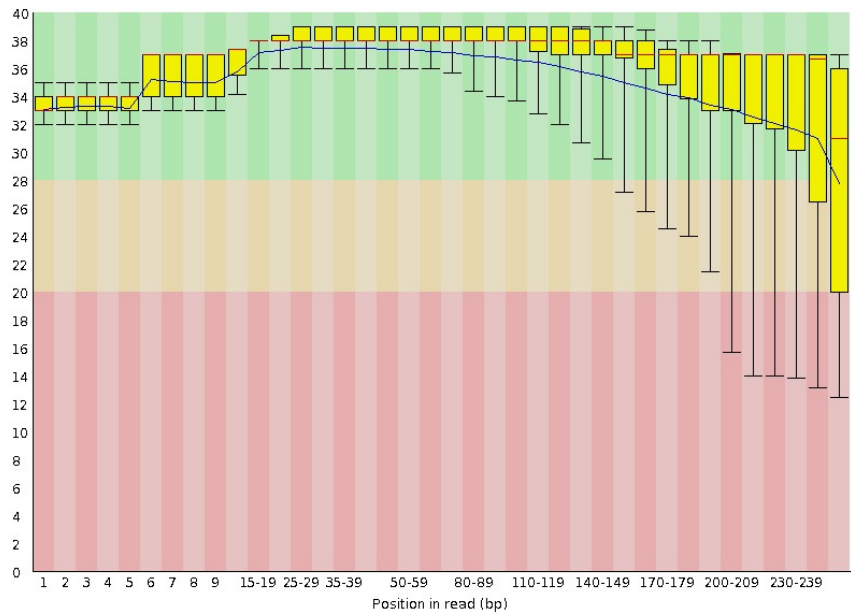


Pre-processing - FASTQC

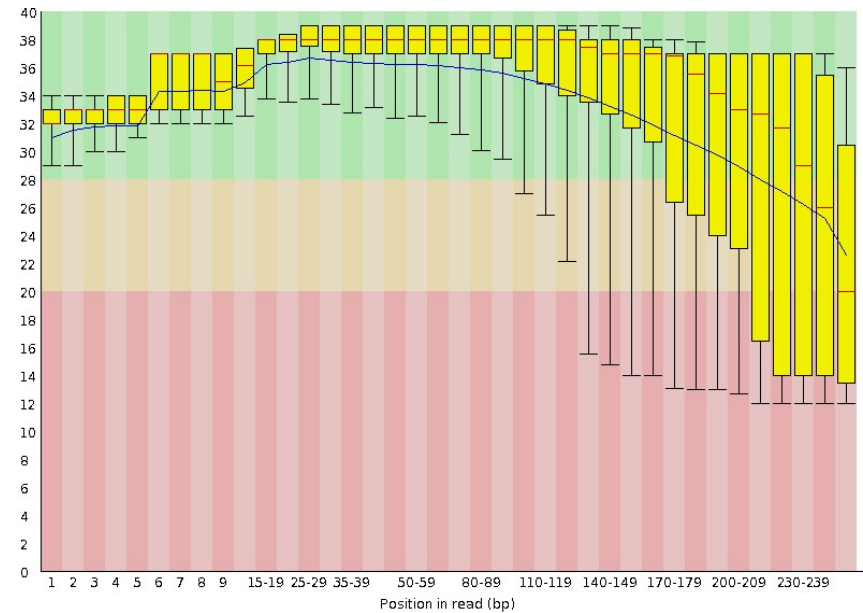


Pre-processing - FASTQC

- The second read in pair-end sequencing has often a lower quality.



Pair 1



Pair 2

Pre-processing: Removal of sequencing errors

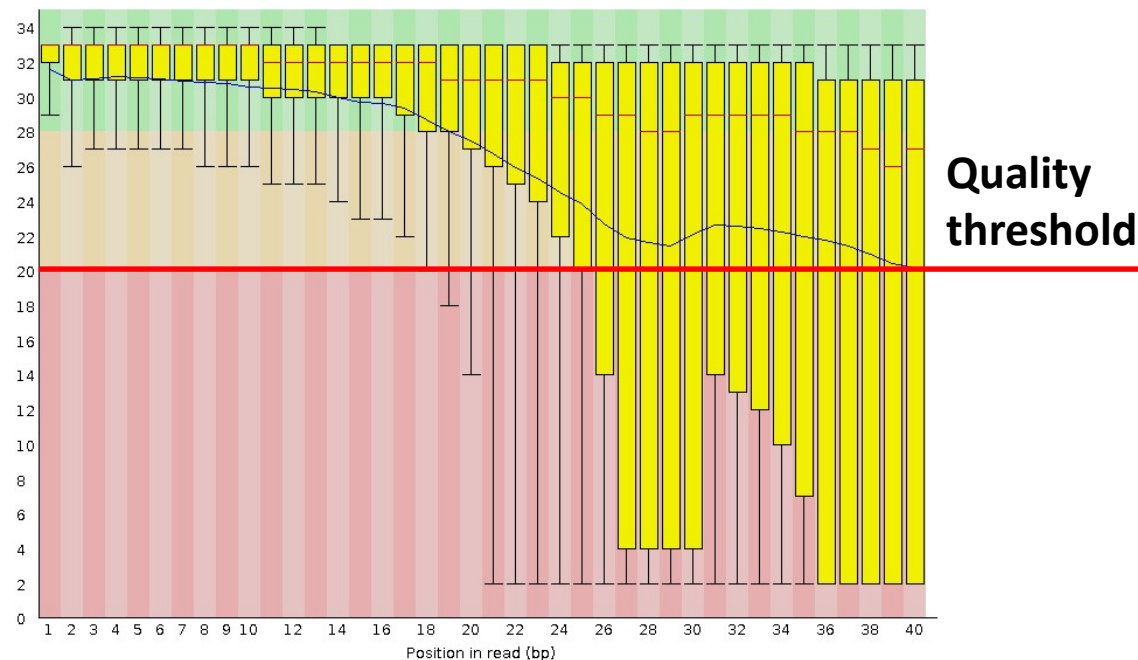
- Three main approaches
 - Filtering: completely remove bad reads
 - Trimming: remove problematic parts of the reads
 - Correcting ('denoising'): correct errors encountered in the reads
- Other forms of errors can also be corrected
 - Adapter contamination
 - Removal of read duplicates

Pre-processing: Filtering

- The overall score of a read is calculated. Reads with an overall score below a threshold are removed.
- Common threshold:
 - Minimum score over the read (or a proportion of the read)
 - Average score over the read
 - Minimum score over a 'sliding window' (e.g. 50 bp)
- Many reads have only a low quality in certain regions. Filtering of reads may therefore throw away good data.

Pre-processing: Trimming

- Removal of regions that are bad
- Trimming is almost always done from the end of the read
- The read is trimmed until a quality level is achieved



Pre-processing of NGS data - FASTX

- FASTX is a tool kit for preprocessing of NGS data

Filtering with FASTX

Remove reads with a score less 20 for 90% or more nucleotides.

```
> fastq_quality_filter -Q33 -q 15 -p 90  
-i input.fastq -o output.fastq
```

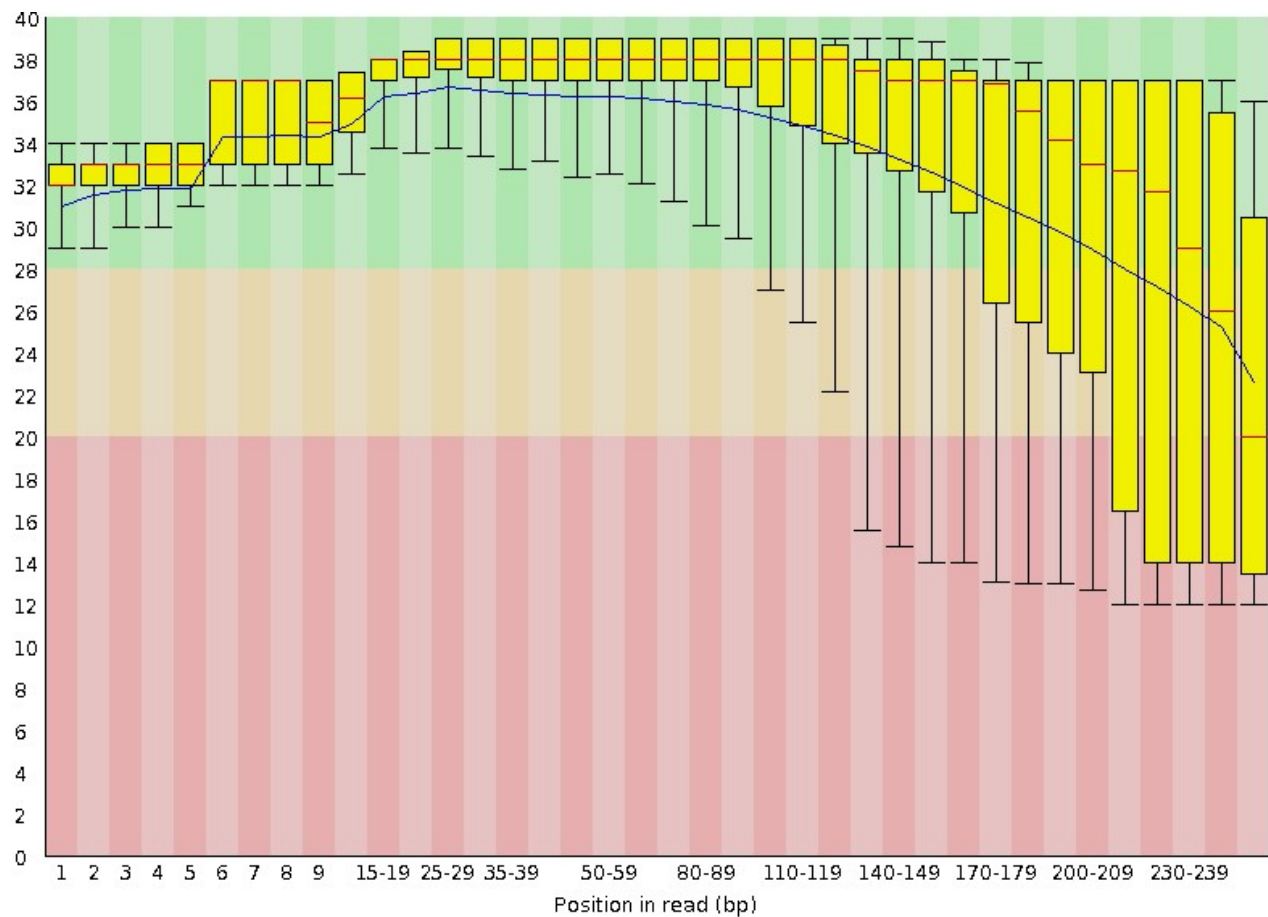
Trimming with FASTX

Trim reads until the score is at least 15. Remove trimmed reads if they are shorter than 50 nucleotides.

```
> fastq_quality_trimmer -Q33 -t 25 -l 50  
-i input.fastq -o output.fastq
```

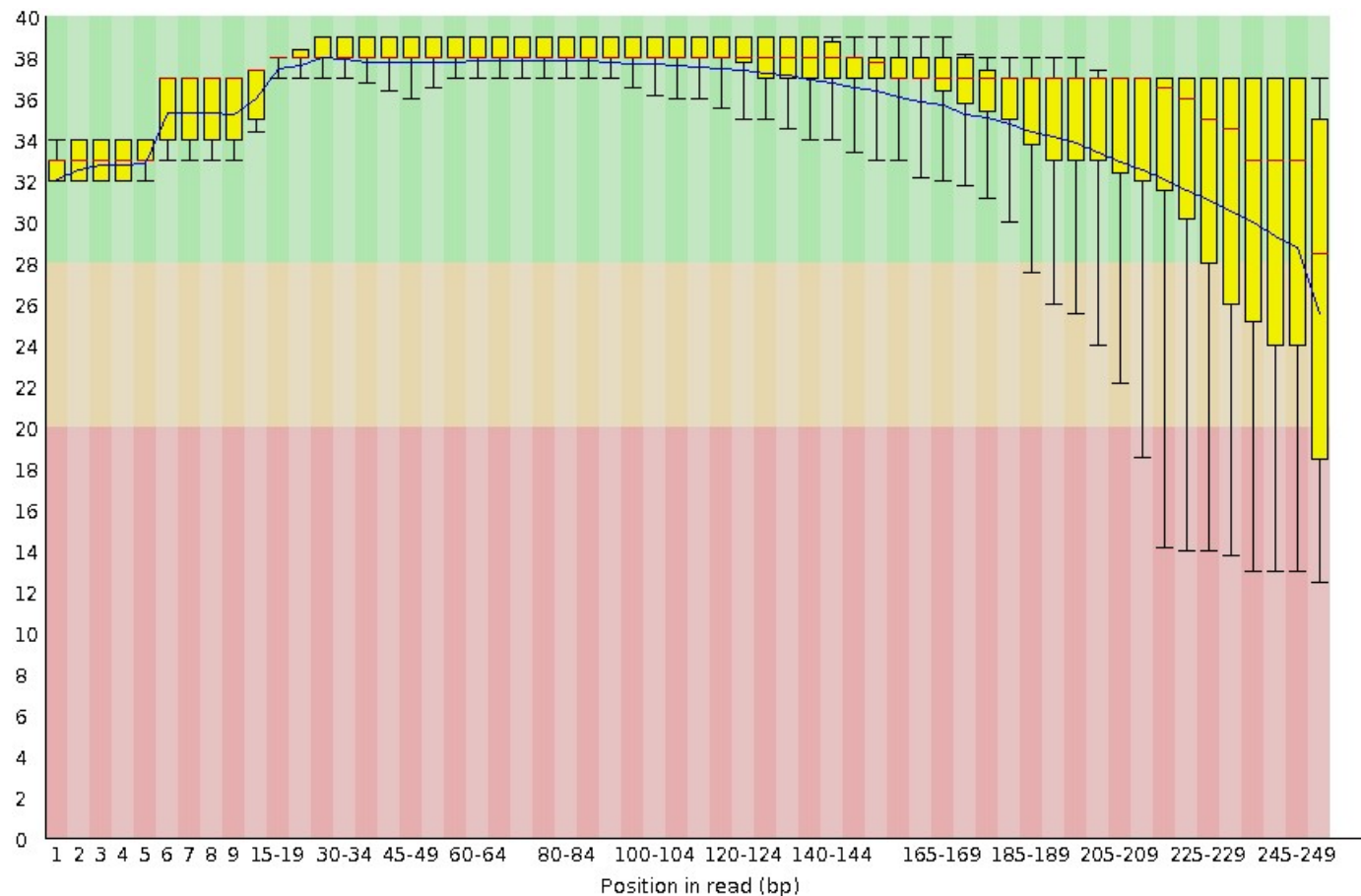
Pre-processing - FASTQC

Before pre-processing. 1548467 reads.



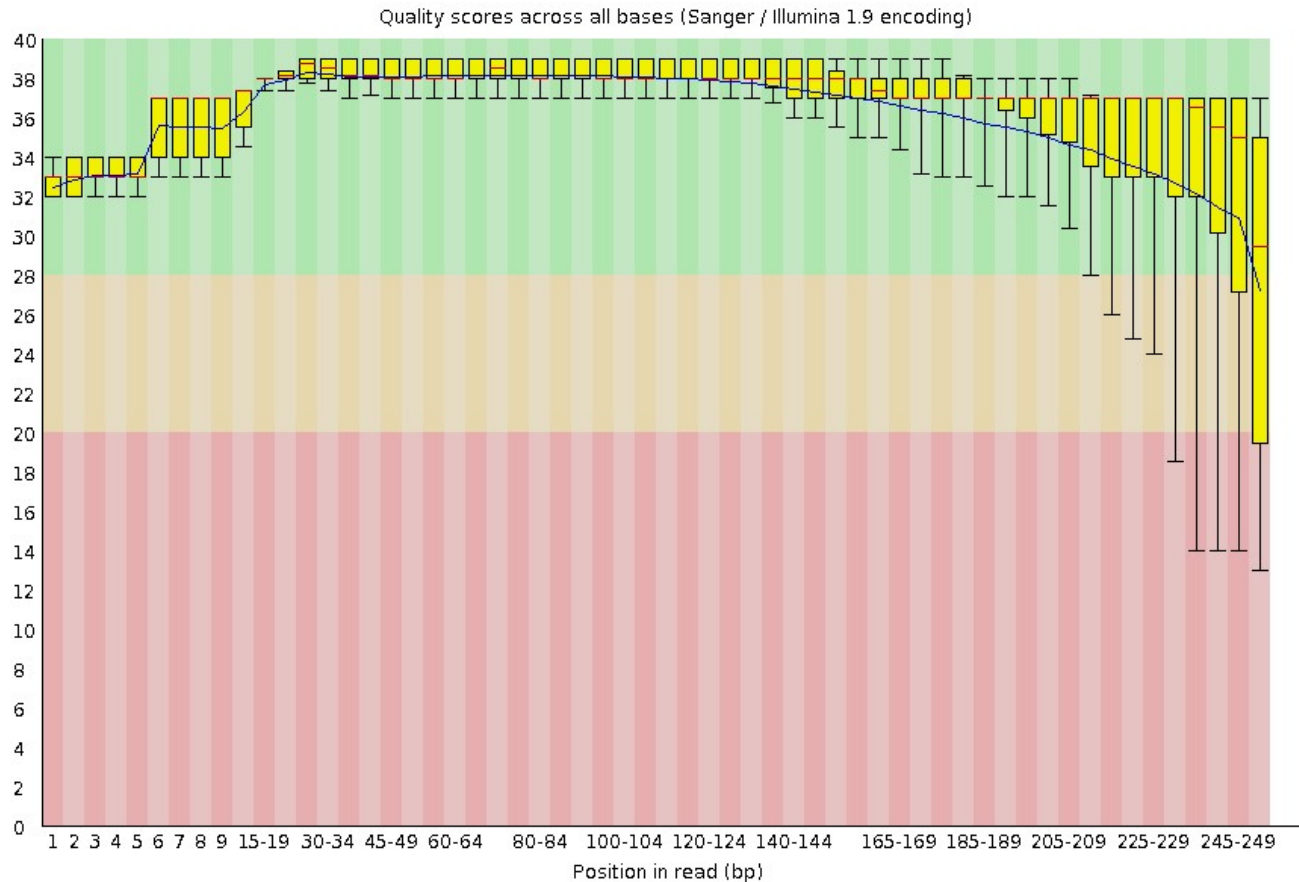
Pre-processing of NGS data - FASTX

After filtering. Quality score >15 over 90% of the reads. 958404 reads left (61.9% left).



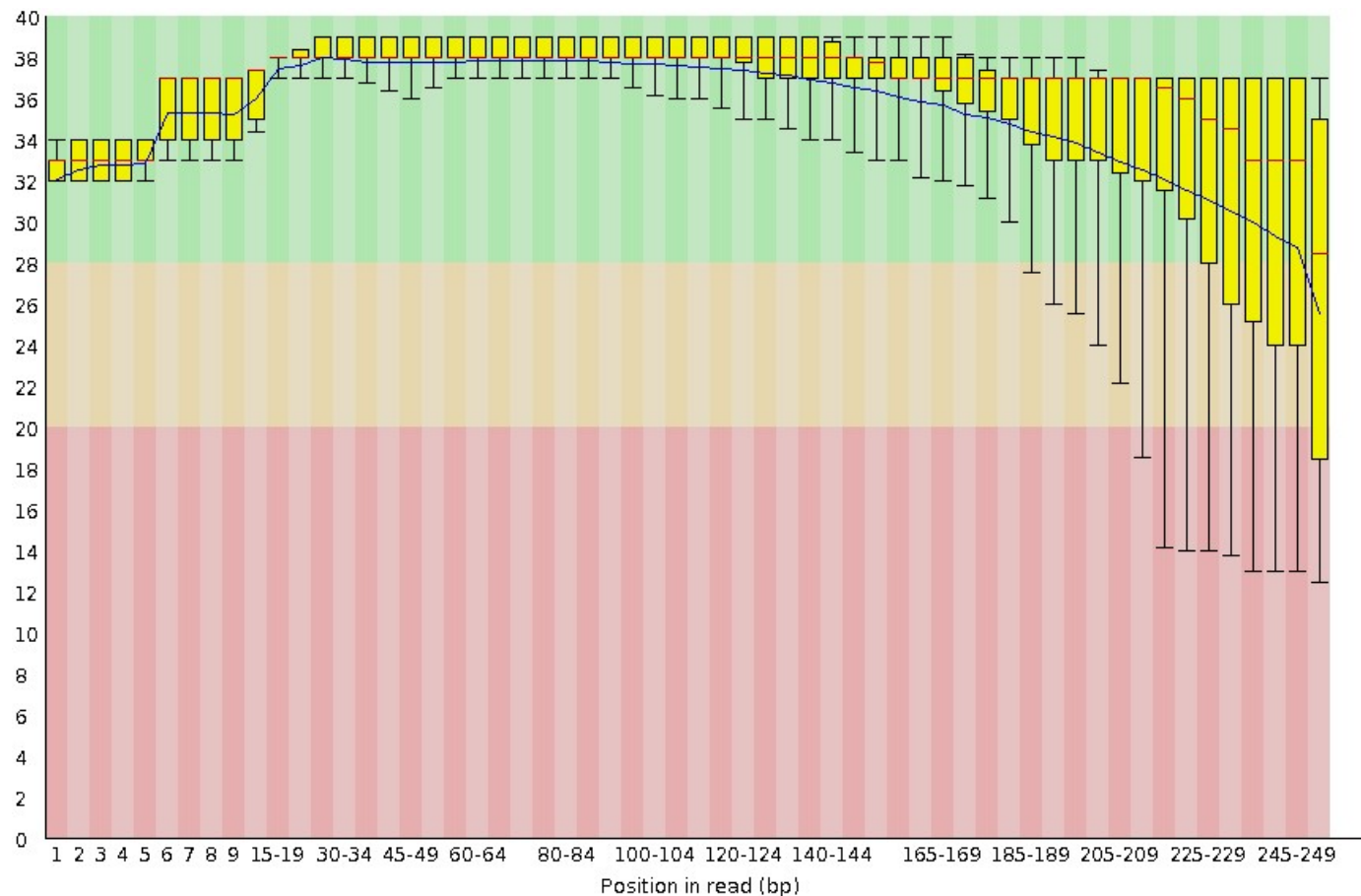
Pre-processing of NGS data - FASTX

After filtering. Quality score >30 over 90% of the reads. 643922 reads left (41.6% left).



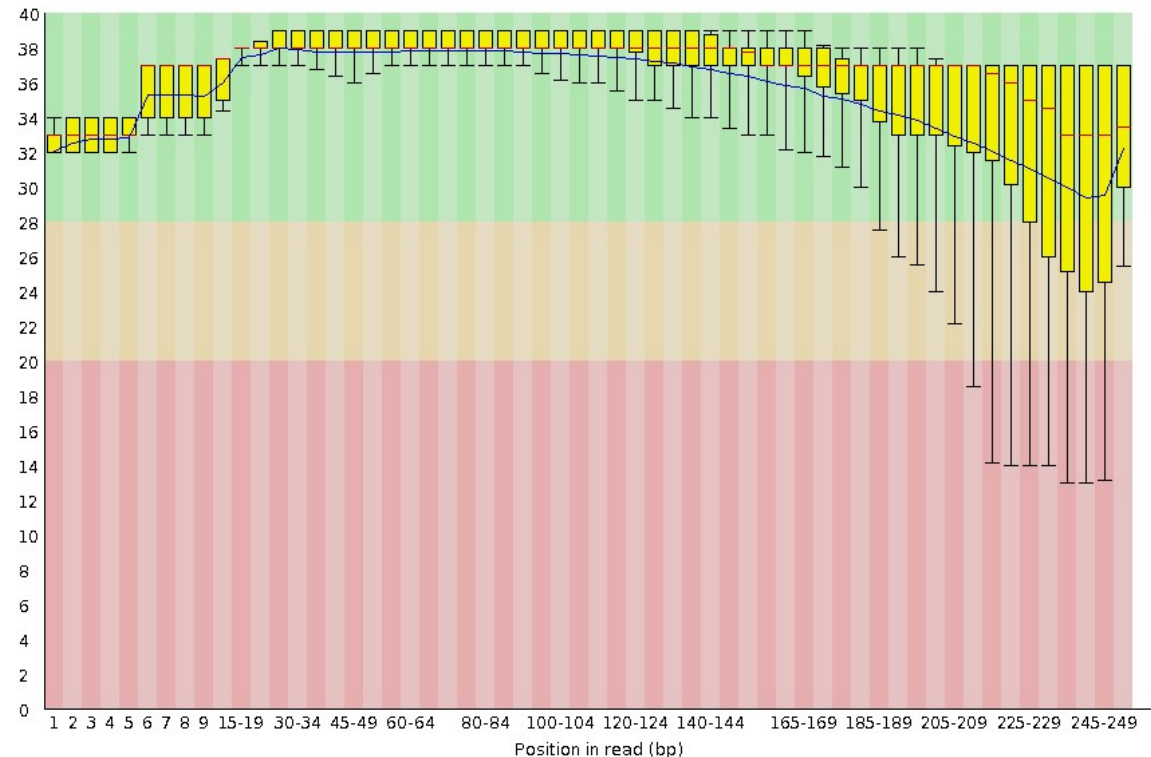
Pre-processing of NGS data - FASTX

After filtering. Quality score >15 over 90% of the reads. 958404 reads left (61.9% left).



Pre-processing of NGS data - FASTX

After trimming from the end until quality score >25.
Trimmed reads shorter than 50 nucleotides were removed. 958320 (<0.01% removed).



Error scores are far from perfect

- Many sequencing errors in Illumina data have a high score!
- For substitutions, >50% of the errors have a high error score (>30)
- For indels, >10% of the errors have a high error score
- Pre-processing of data is thus not a guarantee that the data is error free!

Pre-processing – further remarks

- There are a lot of different software and tools for pre-processing of NGS data.
- Read Österlund et al 2017 and Wang pages 76-86 for more information.

Summary: pre-processing

- NGS data contains a lot of errors and needs to be pre-processed to remove incorrect reads and bases
- The pre-processing is based on the quality score, which estimates the probability that a specific nucleotide is incorrect
- Common pre-processing approaches are filtering and trimming of reads
- There is a trade-off between data volume and quality. The stricter the we filter/trim, the more reads will be discarded but the pre-processed data will have a higher quality.