Next Generation Sequencing (NGS) Lecture 11

Amit Ghosh

IIT Kharagpur

http://www.energy.iitkgp.ac.in/~amitghosh/

Technical Terms

Read

ATGCAGAGAGTCGA…….

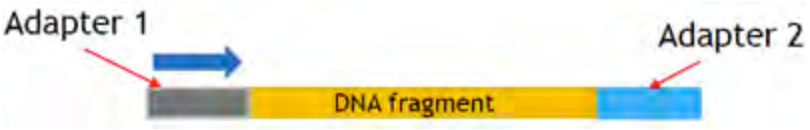
Library

ATGCAGAGAGTCGA…….

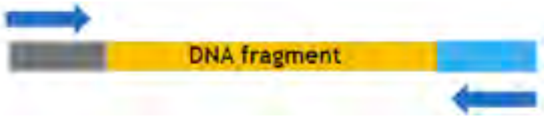
CAGAGGCTACGGATGC…….

AGTGATAGCTATGACA…….

Single-end sequencing

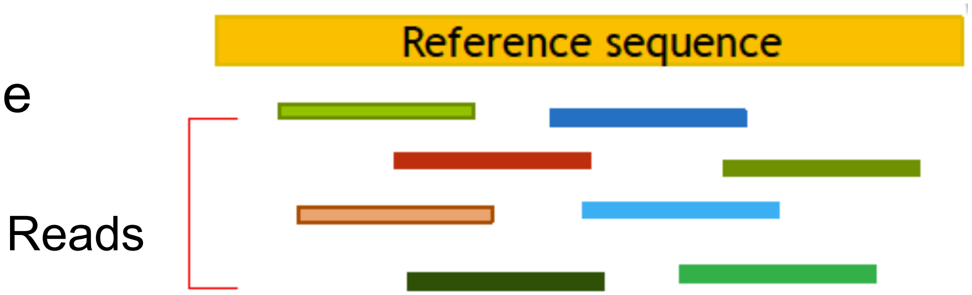


Paired-end sequencing



NGS Data Processing

Sequencing depth or coverage



Quality score or PHRED Score (Q)

$$P(\text{Error})=10^{(-Q/10)}$$

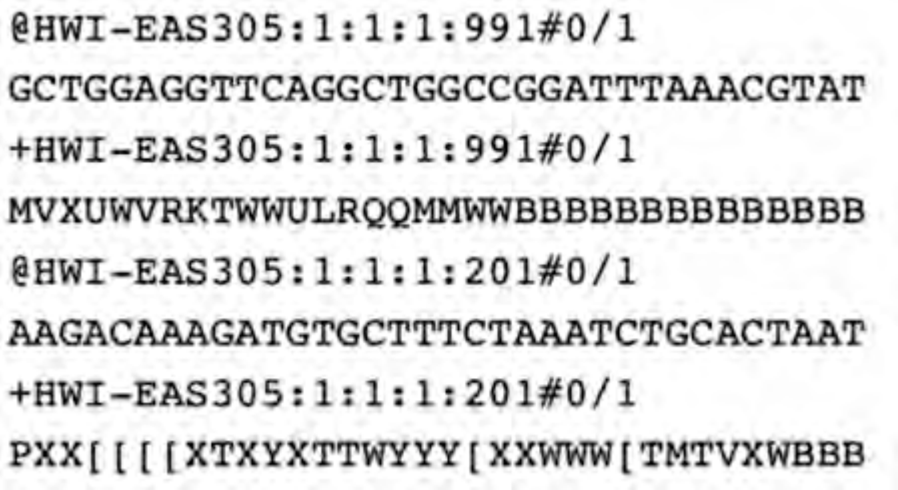
$Q = 40$, $P(\text{Error}) = 10^{-4}$
$Q = 10$, $P(\text{Error}) = 10^{-1}$

Percentage Accuracy = [1 - P(Error)] x 100

FASTQ Files

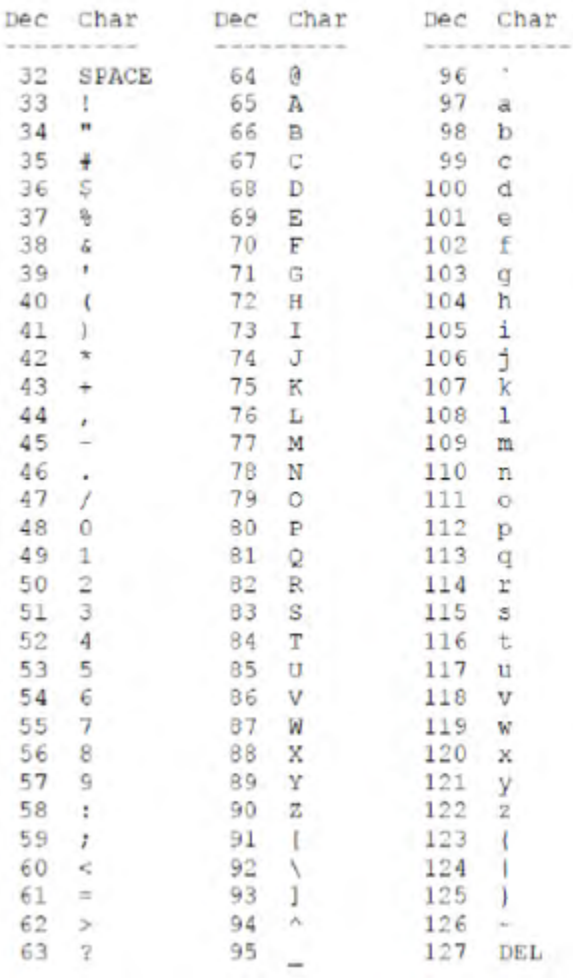
- **Format**

1. Sequence ID
2. Sequence
3. Quality ID
4. Quality Score



```
@HWI-EAS305:1:1:1:991#0/1
GCTGGAGGTTCAGGCTGGCCGGATTTAAACGTAT
+HWI-EAS305:1:1:1:991#0/1
MVXUWVRKTWWULRQQMMWWBBBBBBBBBBBBBB
@HWI-EAS305:1:1:1:201#0/1
AAGACAAAGATGTGCTTTCTAAATCTGCACTAAT
+HWI-EAS305:1:1:1:201#0/1
PXX[[[[XTXYXTTWYYY[XXWWW[TMTVXWBBB
```

Quality score –ASCII table



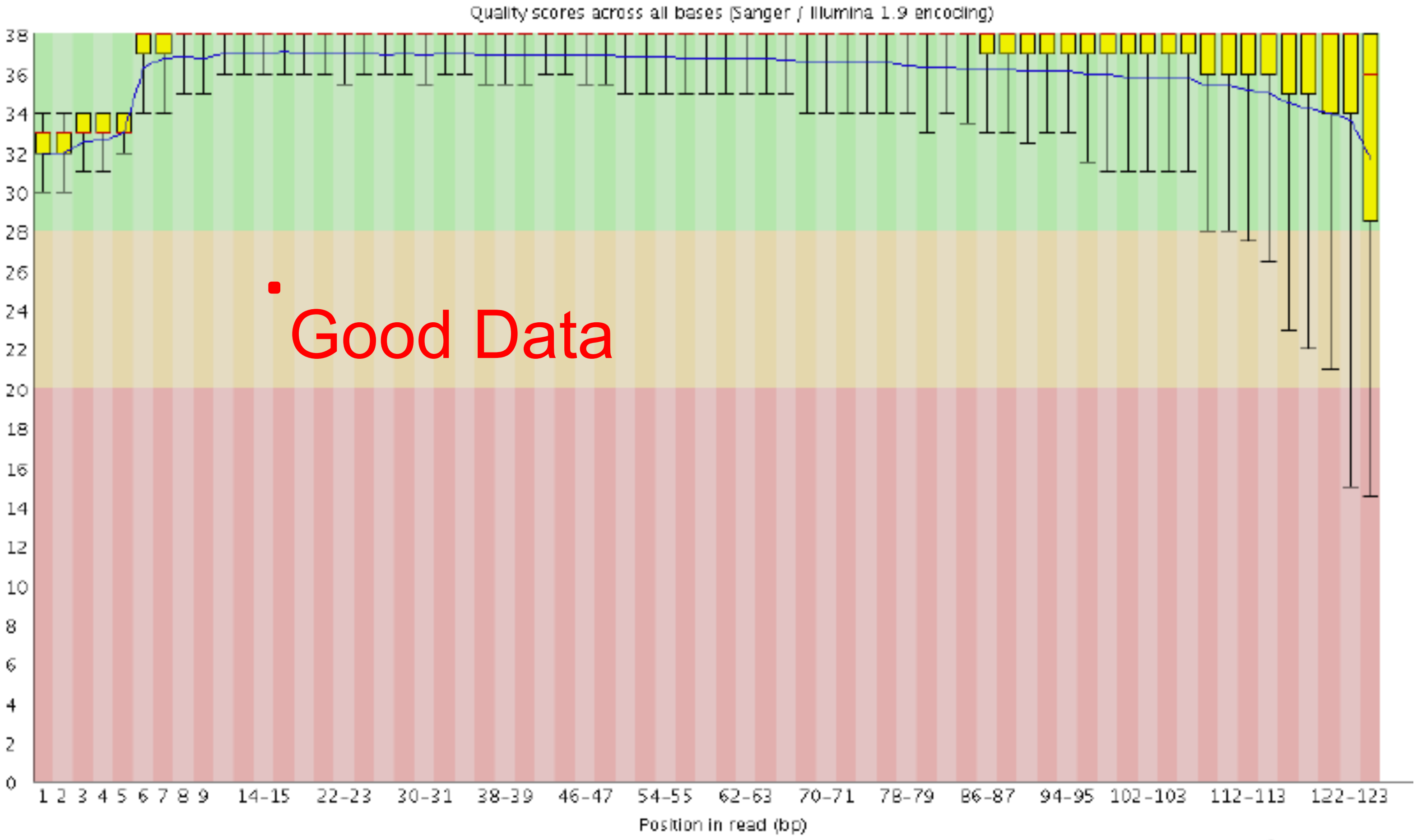
Dec	Char	Dec	Char	Dec	Char
32	SPACE	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	DEL

Checking NGS Data Quality

FastQC

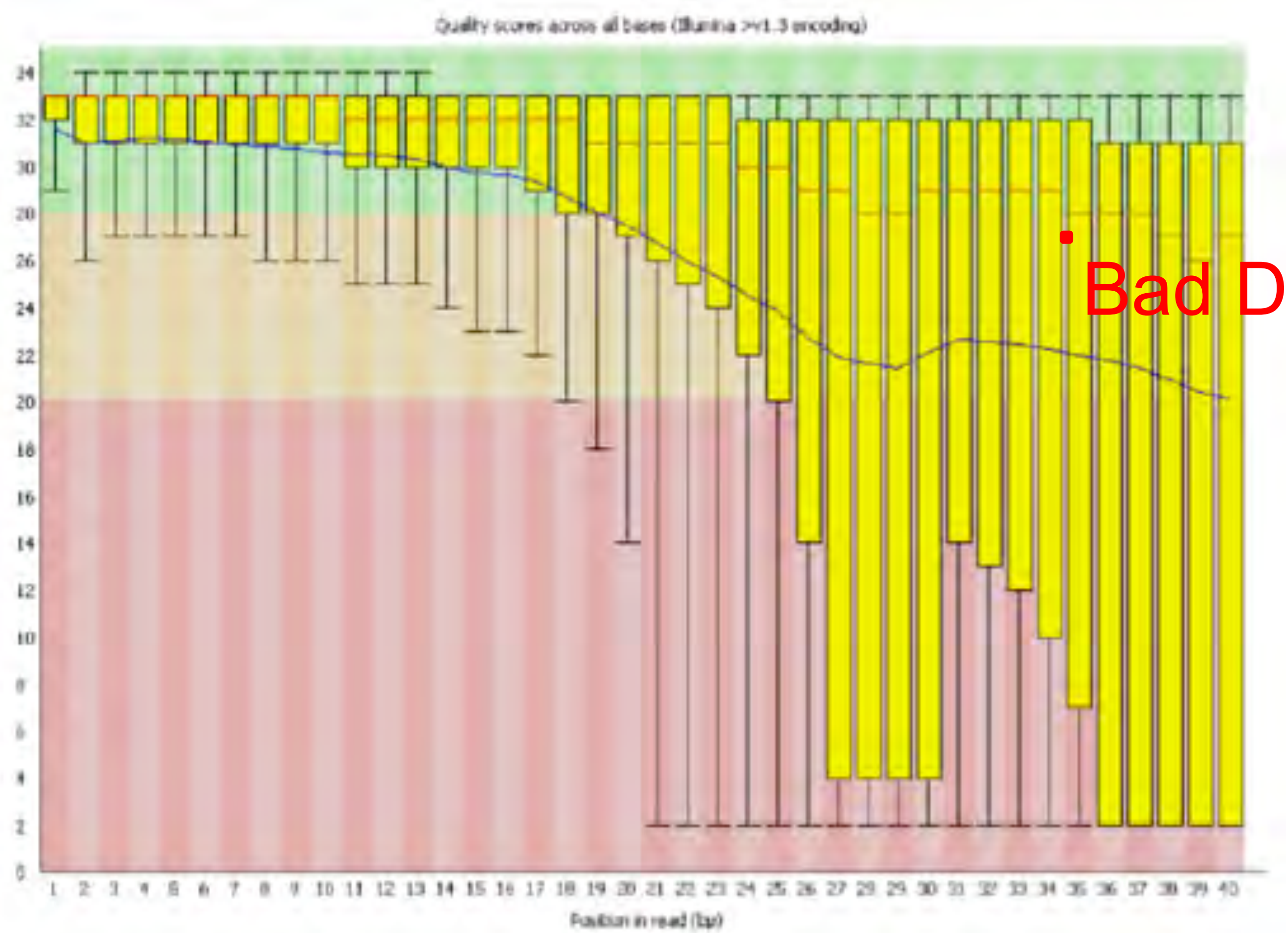
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Per base quality score



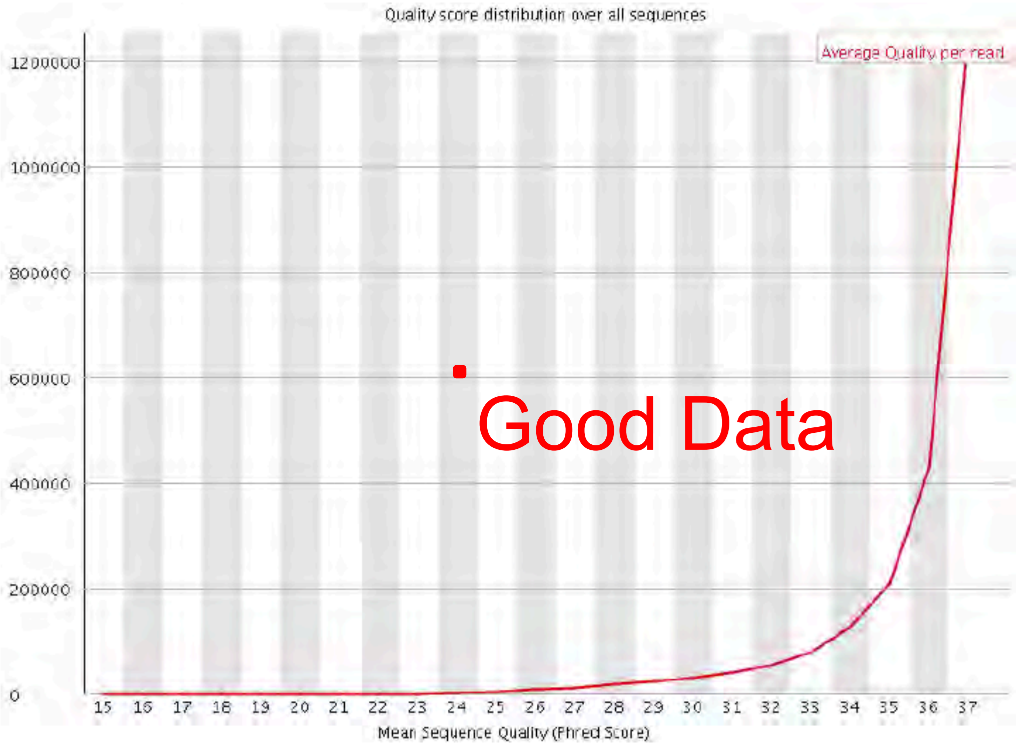
- Good Data

Per base quality score



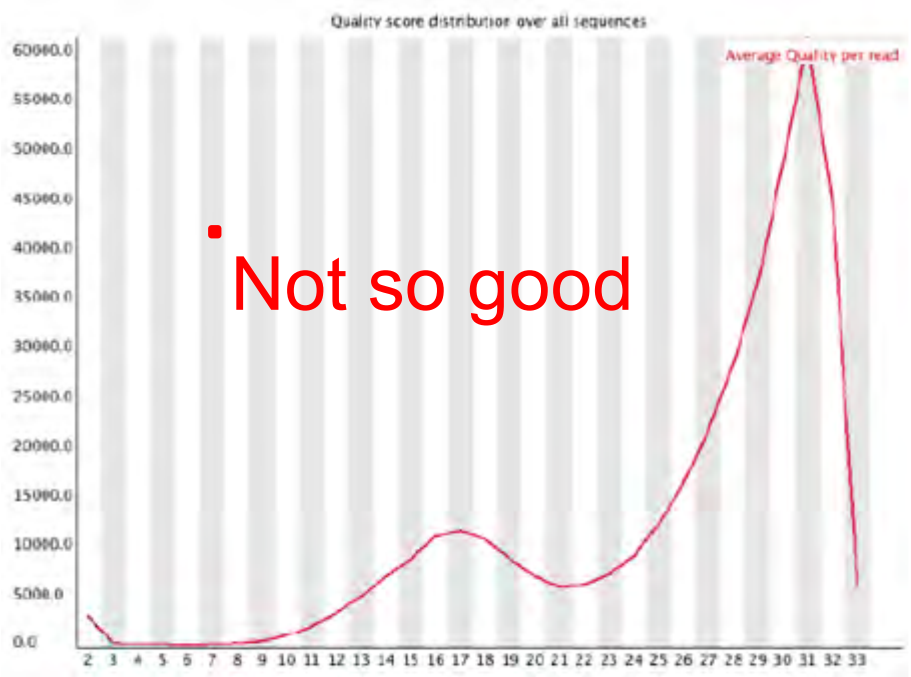
Bad Data

FASTQC: Per sequence quality scores



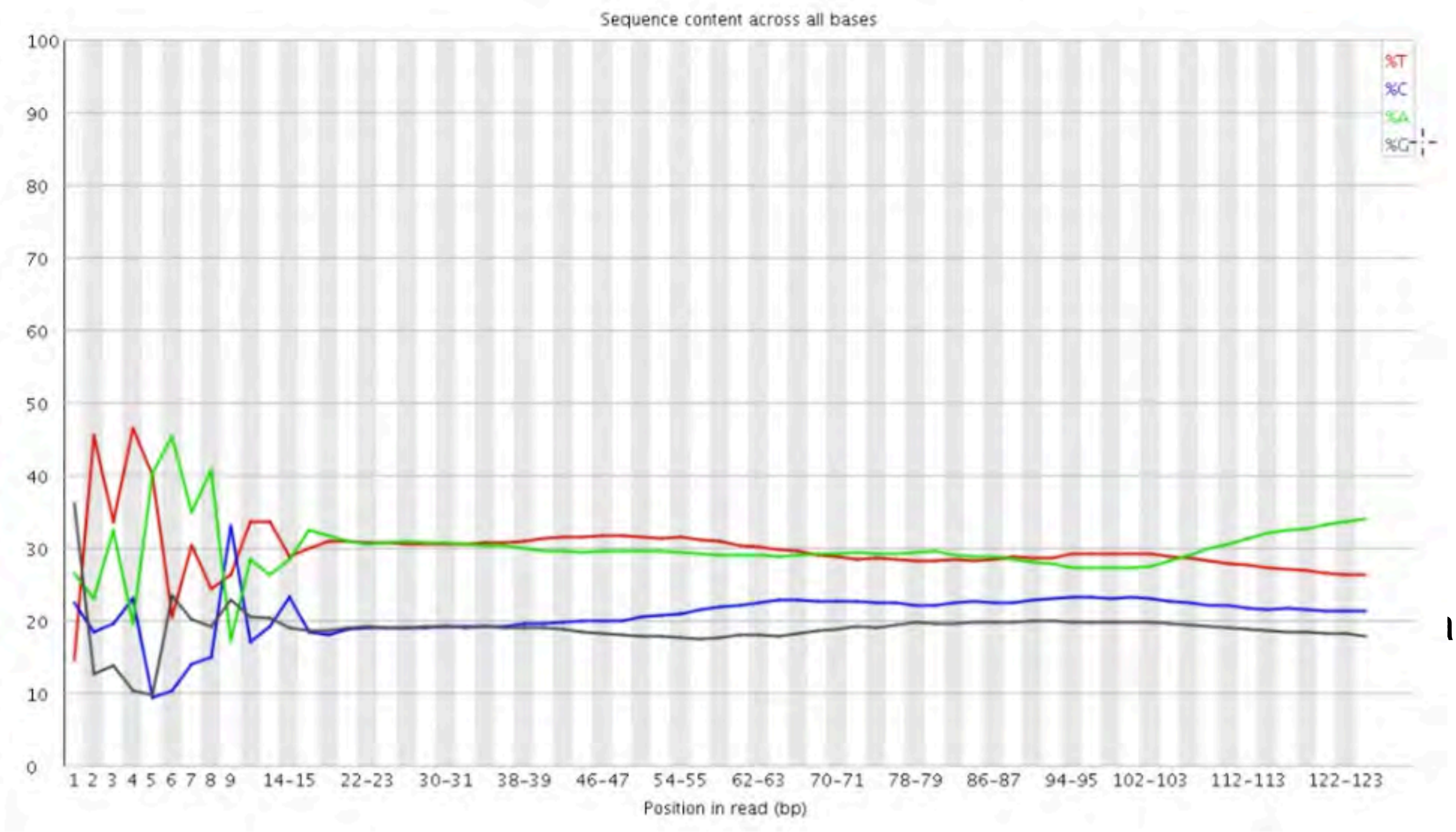
- Good Data

FASTQC: Per sequence quality scores



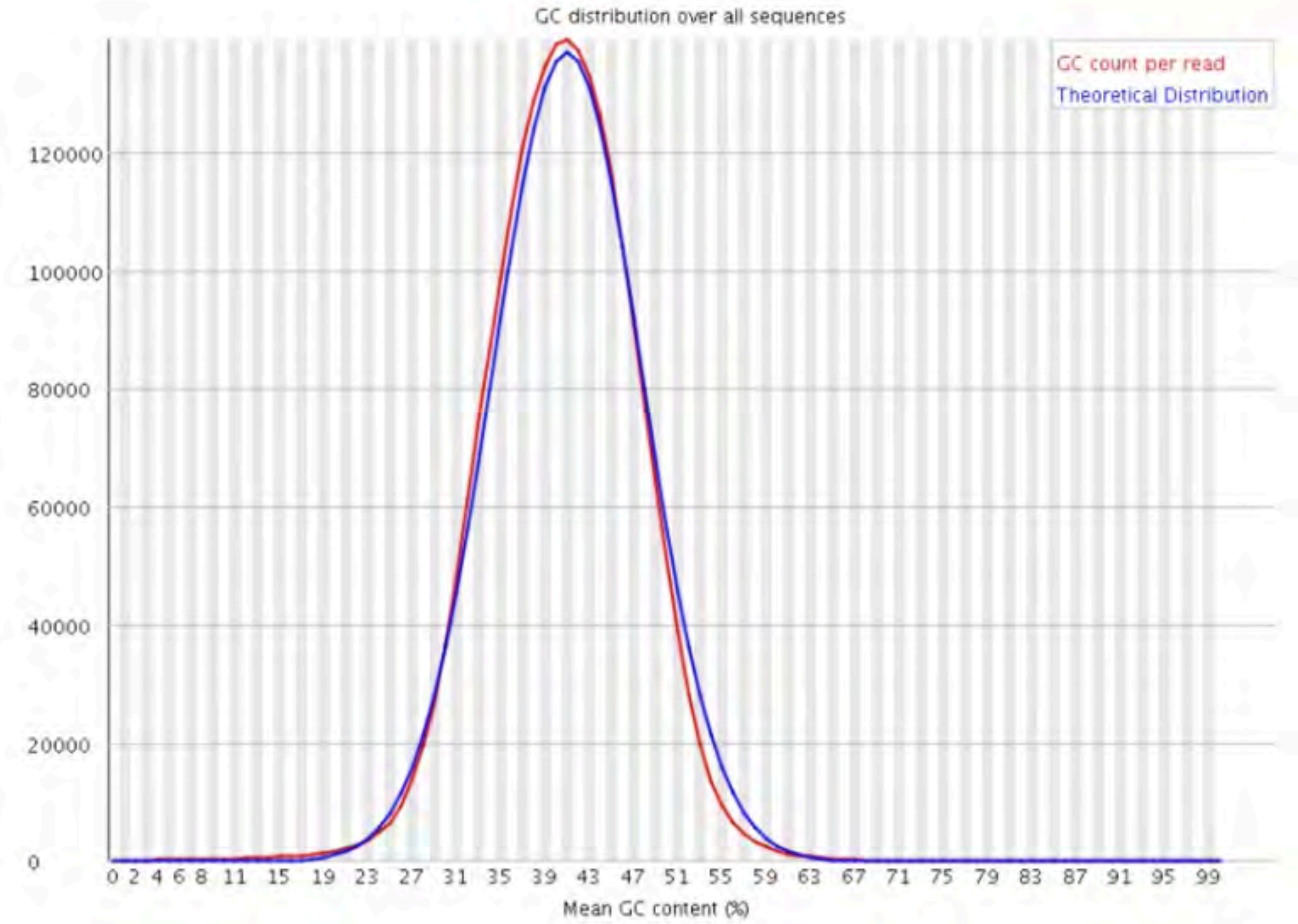
- Not so good

FASTQC: Nucleotide Content Per Position



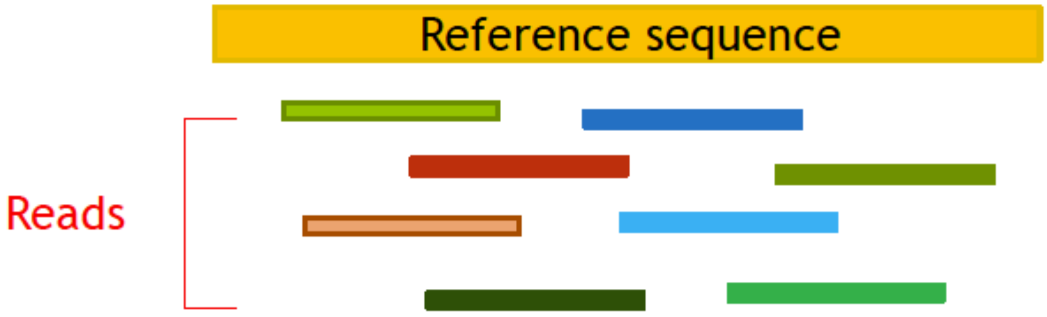
ıs

FASTQC: Per sequence GC content



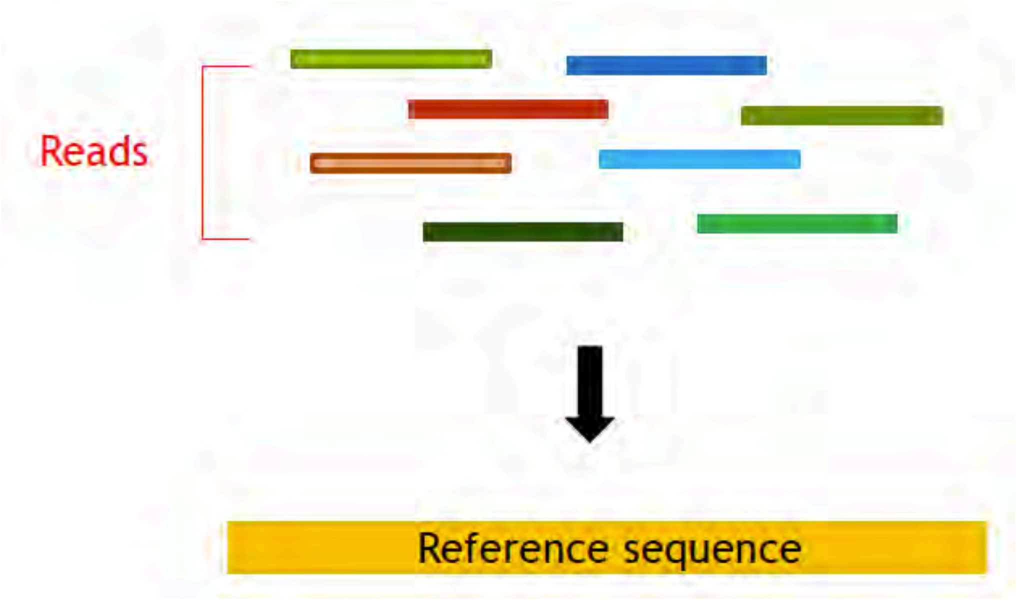
Data Processing

1. Read mapping against available reference genome sequence



Data Processing

2. De novo genome assembly



Mapping read data

Challenges

- Long reference sequences
- Large number of reads
- Reads are small
- Normal algorithms will take too much time (in years!)

Algorithms

Trivial search (slowest)

Blast etc.

Hash-table based

Suffix-tree based

-large memory requirement

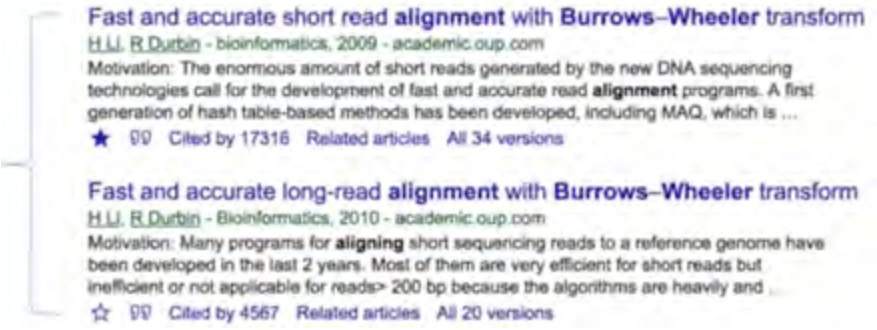
if one mapping takes 0.1 sec, mapping 100 million reads will take –
$0.1 \times 100 \times 10^6 = 10^6$ seconds = 11.5 days

Burrows-Wheeler Aligners

Most widely used tools:

bwa: http://bio-bwa.sourceforge.net/
Bowtie: http://bowtie-bio.sourceforge.net/index.shtml

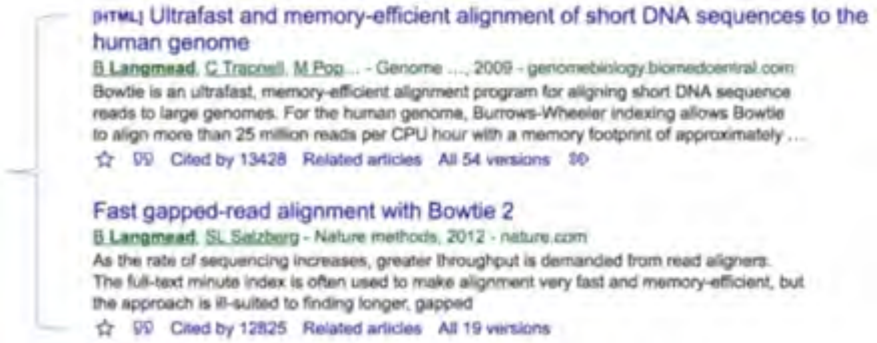
BWA



Fast and accurate short read **alignment** with **Burrows–Wheeler** transform
H Li, R Durbin - bioinformatics, 2009 - academic.oup.com
Motivation: The enormous amount of short reads generated by the new DNA sequencing technologies call for the development of fast and accurate read **alignment** programs. A first generation of hash table-based methods has been developed, including MAQ, which is ...
★ ❝ Cited by 17316 Related articles All 34 versions

Fast and accurate long-read **alignment** with **Burrows–Wheeler** transform
H Li, R Durbin - Bioinformatics, 2010 - academic.oup.com
Motivation: Many programs for **aligning** short sequencing reads to a reference genome have been developed in the last 2 years. Most of them are very efficient for short reads but inefficient or not applicable for reads> 200 bp because the algorithms are heavily and ...
☆ ❝ Cited by 4567 Related articles All 20 versions

Bowtie



[HTML] Ultrafast and memory-efficient alignment of short DNA sequences to the human genome
B Langmead, C Trapnell, M Pop ... - Genome ..., 2009 - genomebiology.biomedcentral.com
Bowtie is an ultrafast, memory-efficient alignment program for aligning short DNA sequence reads to large genomes. For the human genome, Burrows-Wheeler indexing allows Bowtie to align more than 25 million reads per CPU hour with a memory footprint of approximately ...
☆ ❝ Cited by 13428 Related articles All 54 versions

Fast gapped-read alignment with Bowtie 2
B Langmead, SL Salzberg - Nature methods, 2012 - nature.com
As the rate of sequencing increases, greater throughput is demanded from read aligners. The full-text minute index is often used to make alignment very fast and memory-efficient, but the approach is ill-suited to finding longer, gapped
☆ ❝ Cited by 12825 Related articles All 19 versions

Burrows-Wheeler transformation

Step 1: Add $at the end and $<a lexicographically

Reference string T: acaacg$

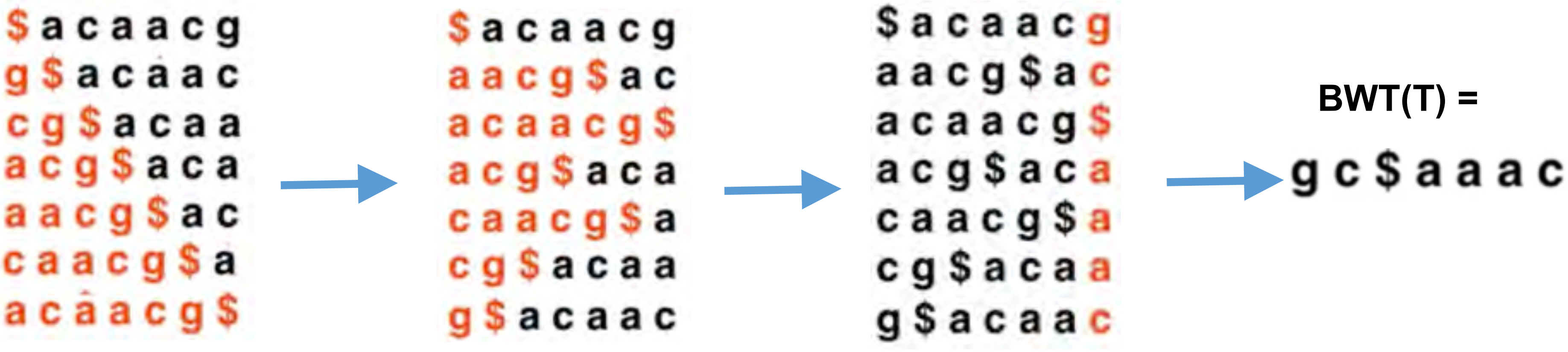
Step 2: Rotate the string counter-clockwise and get all possible rotation

Step 3: Sort alphabetically and store the last column

Burrows-Wheeler transformation

Index the reference sequence so that searching is efficient

Reversible permutation used originally in compression



$ a c a a c g
g $ a c a a c
c g $ a c a a
a c g $ a c a
a a c g $ a c
c a a c g $ a
a c a a c g $

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

BWT(T) =

g c $ a a a c

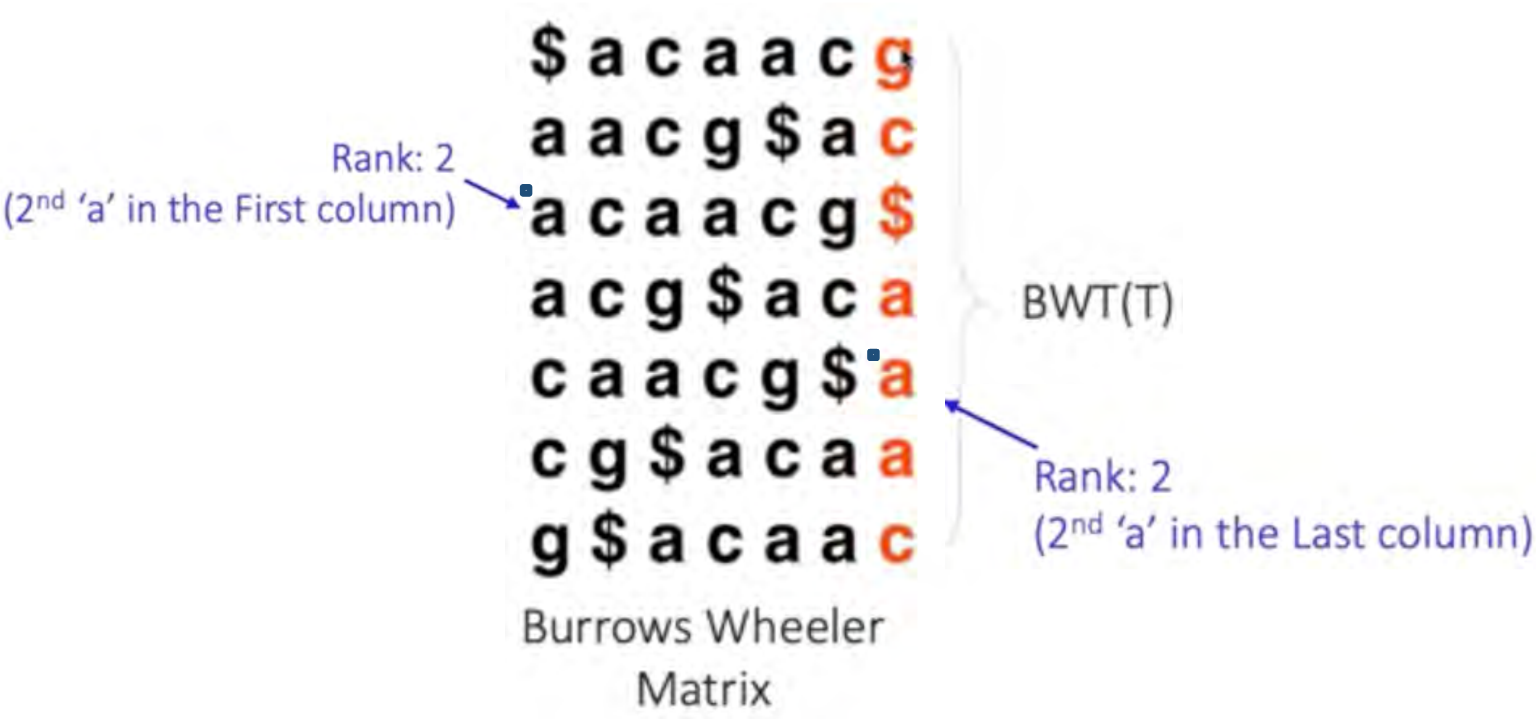
Burrows
Wheeler
Matrix

Last column

Burrows, M; Wheeler, DJ. A block sorting lossless data compression algorithm, Digital Equipment Corporation, Palo Alto, CA 1994, Technical Report 124, 1994

Last-First (LF) mapping

i-th instance of a letter in the last column corresponds to the i-th instance of the same letter in the first column

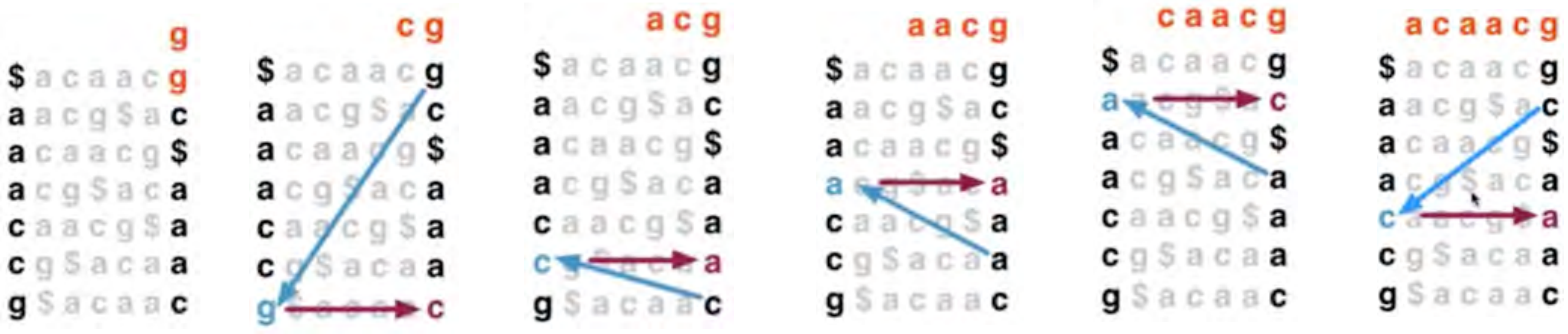


```
$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c
```

Burrows Wheeler Matrix

Last-First (LF) mapping

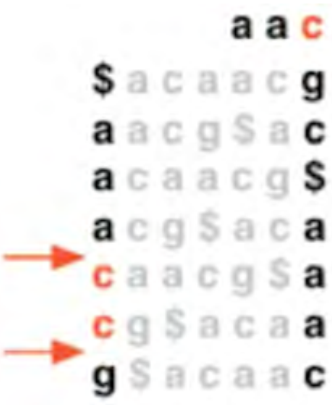
i-th instance of a letter in the last column corresponds to the i-th instance of the same letter in the first column



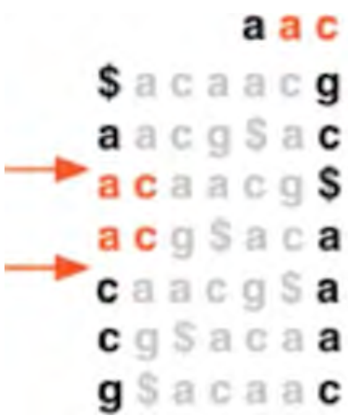
```
g
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

cg
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

acg
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

aacg
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

caacg
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

acaacg
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac
```

BWT(T) to Retrieve Alignments

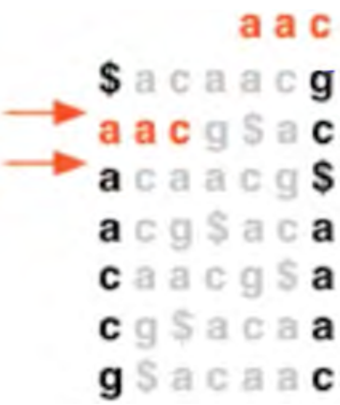
Query Q = aac



```
      a a c
$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c
```



```
      a a c
$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c
```



```
      a a c
$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c
```

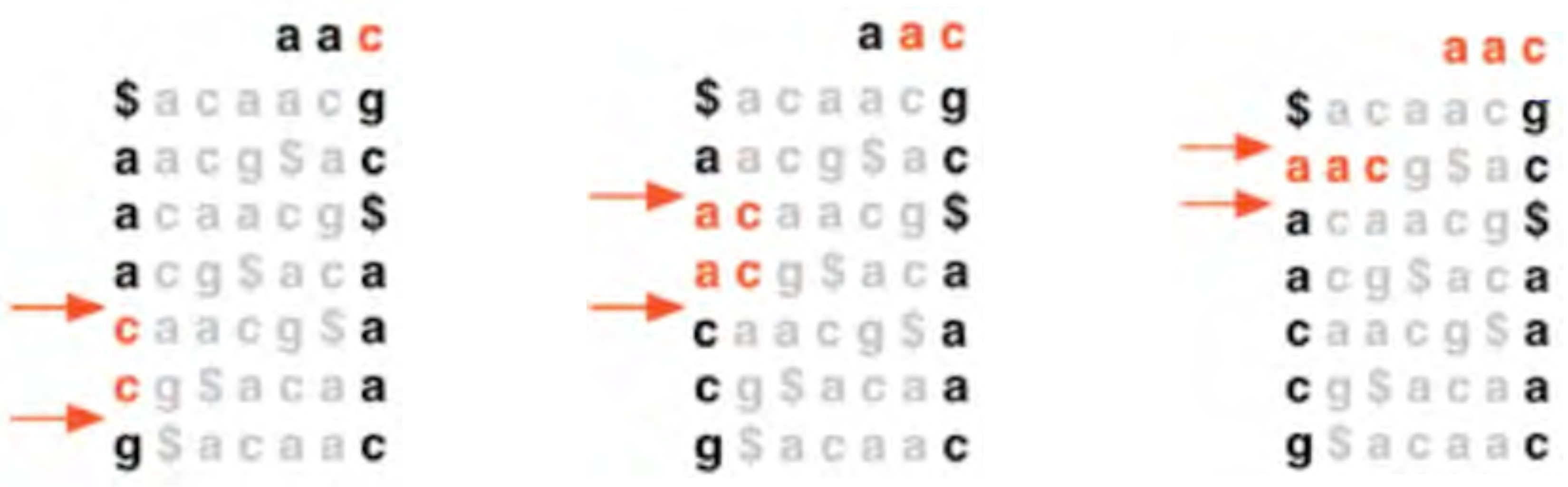
BWT(T) to Retrieve Alignments

- In progressive rounds, top & bottom delimit the range of rows beginning

- If range becomes empty the query does not occur in the text

- If no match, instead of giving up, try to backtrack to a previous position and try a different base (mismatch, much slower)

Mapping a substring in the reference string



a a c

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

a a c

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

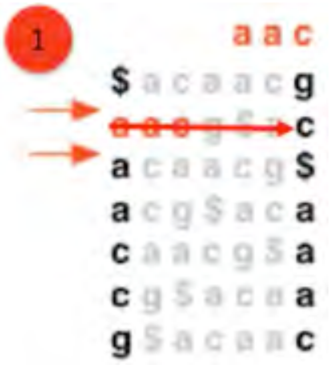
a a c

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

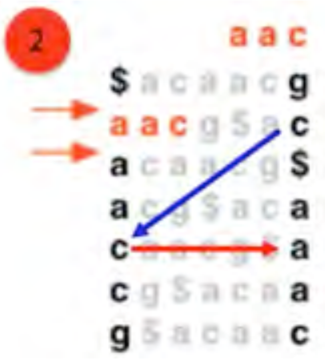
BWT(T) to Retrieve Alignments

- How to recover the query sequence (Q) alignment position in the reference sequence T: LF mapping

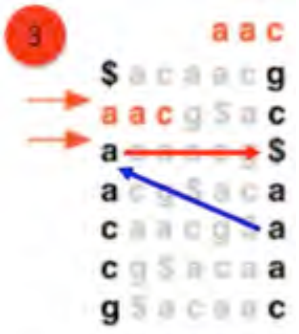
```
          3
T  =  acaacg$
        | | |
Q  =    aac
```



aac
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac



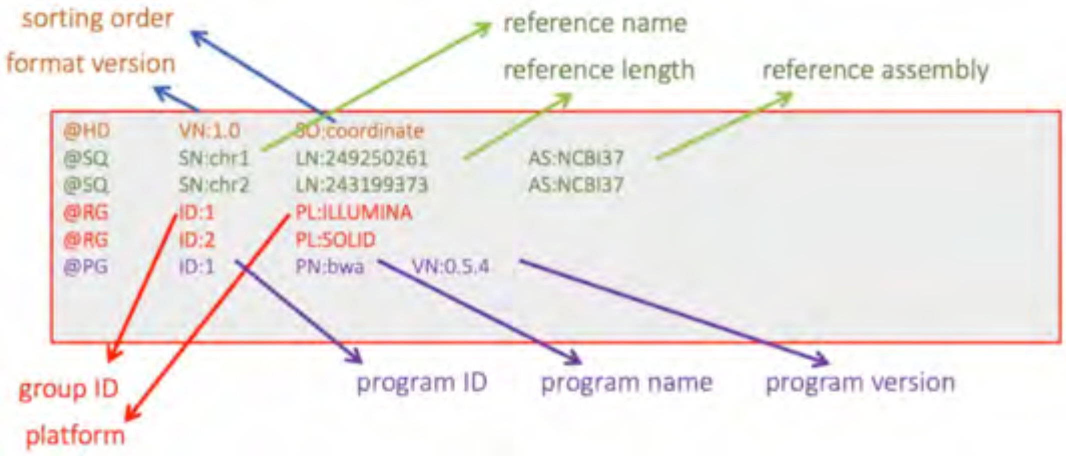
aac
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac



aac
$acaacg
aacg$ac
acaacg$
acg$aca
caacg$a
cg$acaa
g$acaac

Alignment output: SAM File - Header

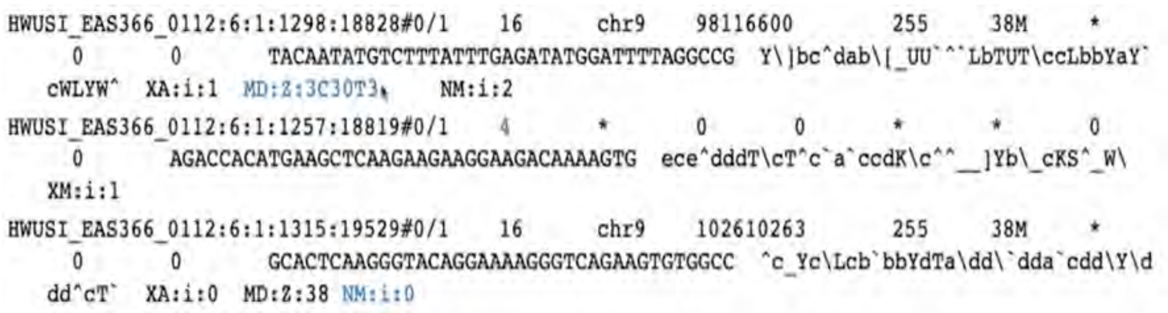
- @HD – Header line.
- @SQ – Reference genome information.
- @RG – Read group information.
- @PG – Program (software) information.



```
@HD    VN:1.0    SO:coordinate
@SQ    SN:chr1   LN:249250261    AS:NCBI37
@SQ    SN:chr2   LN:243199373    AS:NCBI37
@RG    ID:1      PL:ILLUMINA
@RG    ID:2      PL:SOLID
@PG    ID:1      PN:bwa   VN:0.5.4
```

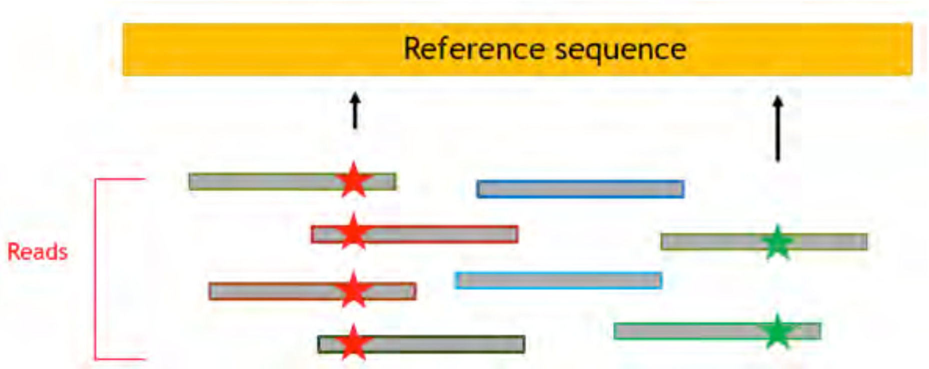
Mapped Seq Files in SAM format

- Read Name
- Map: 0 OK, 4 unmapped, 16 mapped
- Sequence, quality score
- MD: mismatch info: 3 match, then C ref, 30 match, then T ref, 3 match
- NM: number of mismatch
- BAM: binary compressed SAM format

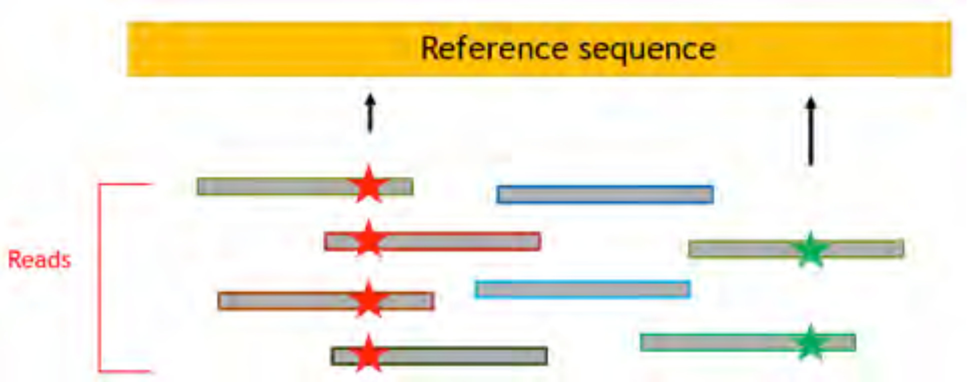


```
HWUSI_EAS366_0112:6:1:1298:18828#0/1    16      chr9    98116600        255     38M     *
    0       0       TACAATATGTCTTTATTTGAGATATGGATTTTAGGCCG  Y\]bc^dab\[_UU`^`LbTUT\ccLbbYaY`
   cWLYW^  XA:i:1  MD:Z:3C30T3     NM:i:2
HWUSI_EAS366_0112:6:1:1257:18819#0/1    4       *       0       0       *       *       0
    0       AGACCACATGAAGCTCAAGAAGAAGGAAGACAAAAGTG  ece^dddT\cT^c`a`ccdK\c^^__]Yb\_cKS^_W\
   XM:i:1
HWUSI_EAS366_0112:6:1:1315:19529#0/1    16      chr9    102610263       255     38M     *
    0       0       GCACTCAAGGGTACAGGAAAAGGGTCAGAAGTGTGGCC  ^c_Yc\Lcb`bbYdTa\dd\`dda`cdd\Y\d
   dd^cT`  XA:i:0  MD:Z:38 NM:i:0
```

Single nucleotide polymorphisms (SNP)



SNP call with number of reads and quality cut-off



Quality cut-off for SNP base: Q

Cut-off for number of reads showing the reads: C

SNP call with number of reads and quality cut-off

Cut-off for number of reads showing the reads: C

Quality cut-off for SNP base: Q

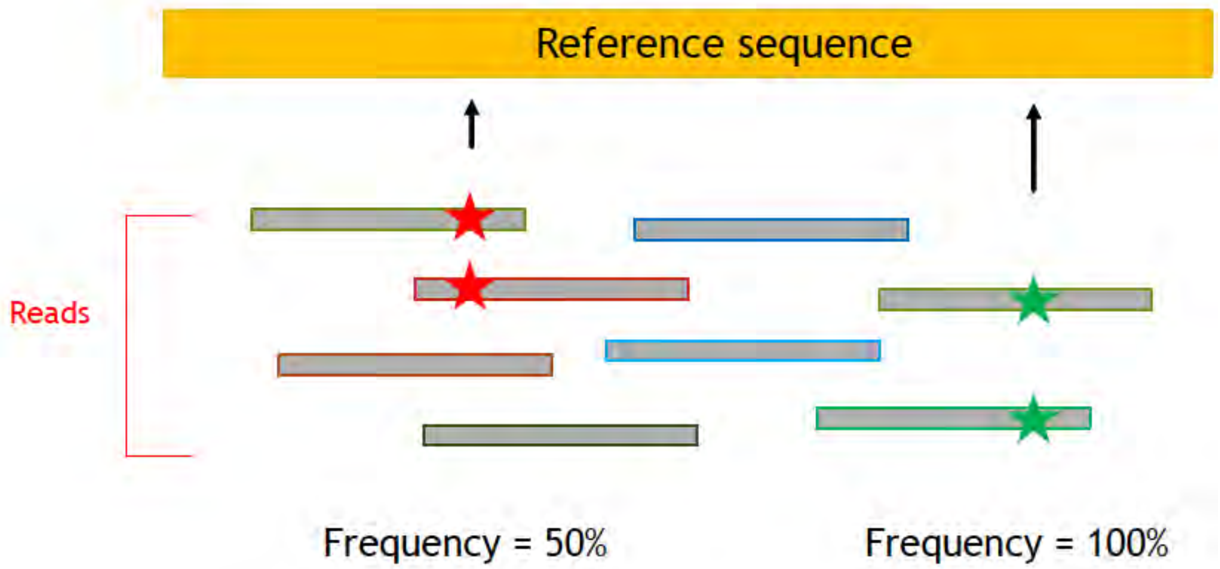
How does it help?

Higher confidence in the SNP call and reduced number of false positives

Prob. of error/false call = $10^{-Q/10}$x $10^{-Q/10}$…. x $10^{-Q/10}$(C times)

= $10^{-cQ/10}$

Frequency of SNP



Copy number variations (CNVs)

Comparison between treatment vs control group

Diseased vs healthy

Cancer vs Normal

$$\text{Coverage ratio (CR)} = \frac{\text{Coverage in diseased sample}}{\text{Coverage in healthy sample}}$$

Coverage ratio (CR)

$$\text{Coverage ratio (CR)} = \frac{\text{Coverage in diseased sample}}{\text{Coverage in healthy sample}}$$

Region	Coverage in diseased	Coverage in healthy
Region 1	100	50
Region 2	100	100
Region 3	50	150

Coverage ratio (CR)

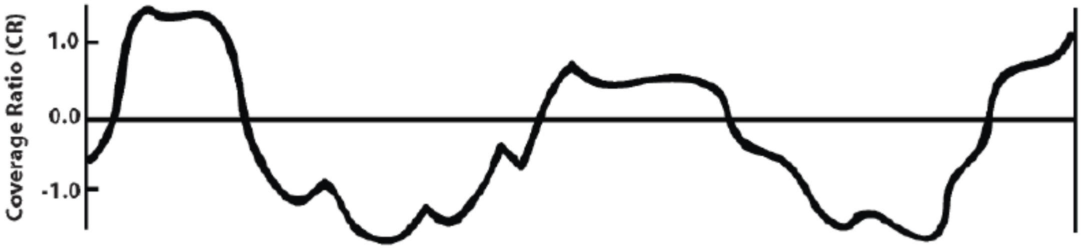
$$\text{Coverage ratio (CR)} = \frac{\text{Coverage in diseased sample}}{\text{Coverage in healthy sample}}$$

The total coverage might vary across samples

$$\text{Coverage ratio (CR)} = \frac{\text{(Coverage in diseased sample)/(Total no of reads in diseased sample)}}{\text{(Coverage in healthy sample)/(Total no of reads in healthy sample)}}$$

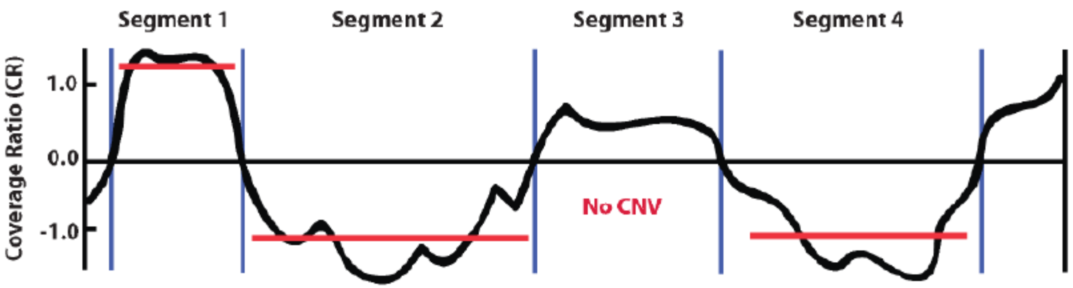
Segmentation algorithm

Coverage ratio (CR) values in a genomic region



Segmentation algorithm

Coverage ratio (CR) values in a genomic region



Multiple windows each with CR ≥ CR_{thr}

Summary

- Sequencing technologies: 1st, 2nd, 3rd generation
- Illumina (2nd gen) has taken most of the market
- Sequences are sorted in FASTQ file
- After sequencing, perform quality assessment (FASTQC)
- Sequenced “reads” need to be aligned back to reference genome
    - BLAST
    - Suffix Array
    - BWA/Bowtie: Burrows-Wheeler transformation, LF mapping
- Aligned reads are stored in SAM/BAM files