H3ABioNet
Pan African Bioinformatics Network for H3Africa

WELLCOME GENOME CAMPUS
CONNECTING SCIENCE
ADVANCED COURSES+ SCIENTIFIC CONFERENCES

Next Generation Sequencing Bioinformatics Course 2021

NGS Data Analysis Overview and Data Formats

Shaun Aron

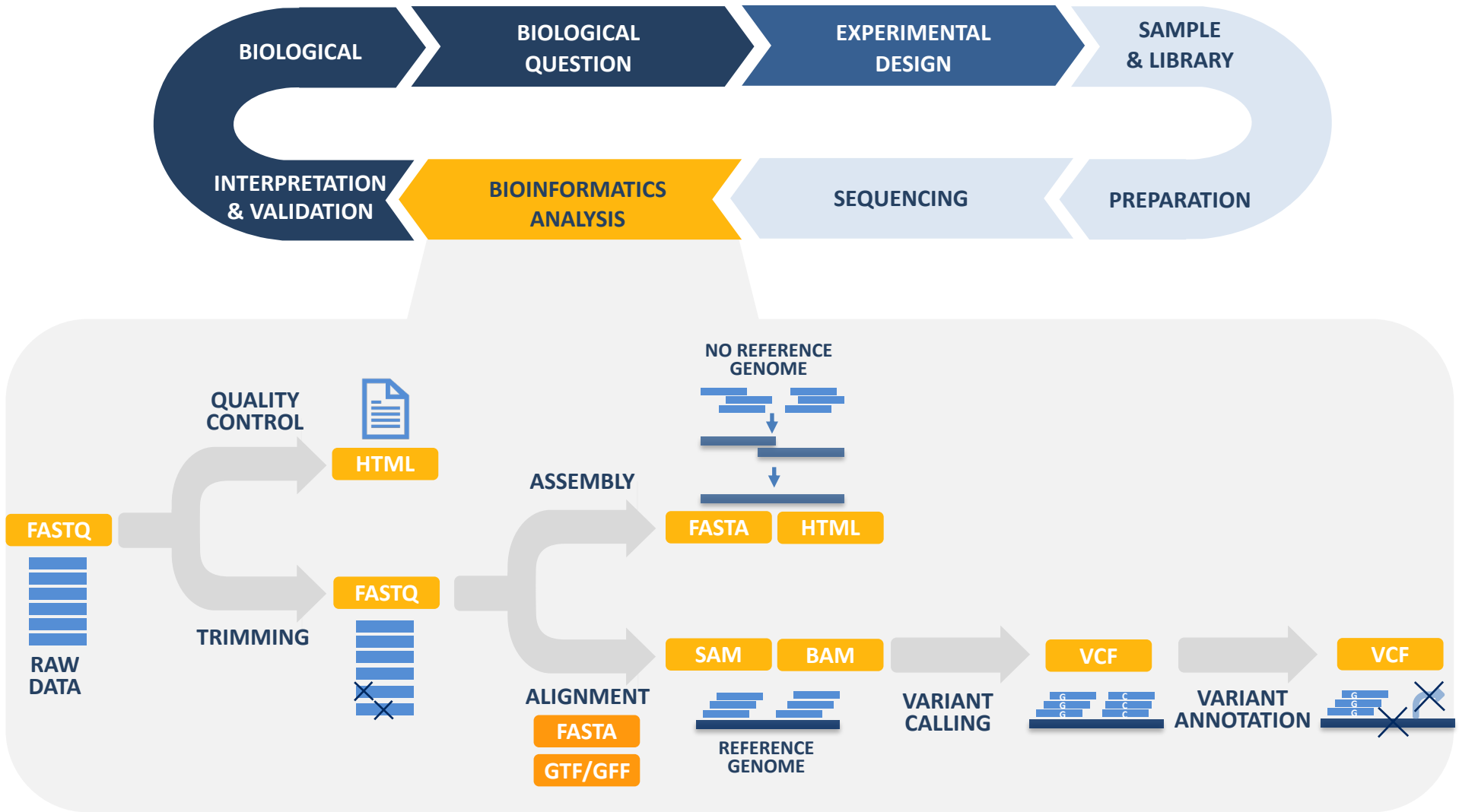
H3ABioNet
Pan African Bioinformatics Network for H3Africa

WELLCOME GENOME CAMPUS
CONNECTING SCIENCE
ADVANCED COURSES+ SCIENTIFIC CONFERENCES

Some content taken from a lecture by Petr Dancek – WTAC NGS course*

Objectives

- Brief overview of NGS file formats generated from QC to variant calling
  - Be able to describe what files are generated during the various steps of the analysis pipeline
  - Be able to describe the information contained in the different file formats
  - Be able to identify and extract specific information from the different files


BIOLOGICAL
BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
SAMPLE & LIBRARY
INTERPRETATION & VALIDATION
BIOINFORMATICS ANALYSIS
SEQUENCING
PREPARATION
QUALITY CONTROL
HTML
FASTQ
RAW DATA
TRIMMING
FASTQ
ASSEMBLY
NO REFERENCE GENOME
FASTA
HTML
ALIGNMENT
FASTA
GTF/GFF
SAM
BAM
REFERENCE GENOME
VARIANT CALLING
VCF
VARIANT ANNOTATION
VCF

QC and Trimming

- Data file input: FASTQ
- Data file output: QC'd FASTQ

Alignment

- Data file input: QC'd FASTQ
- Data file output: SAM -> BAM (CRAM)

Variant Calling

- Data file input: BAM (QC'd) (CRAM)
- Data file output: VCF (BCF)

Annotation

- Data file input: VCF (BCF)
- Data file output: Annotated VCF (BCF)

Raw Sequence Data

- FASTQ format
  - Derived from FASTA format
  - Original Sanger standard from capillary sequence data
  - Sequence description, sequence and associated per base quality score
  - PHRED quality scores encoded as ASCII printable characters (ASCII 33-126)

FASTQ Format

@title	@SRR010930.8436795/1
sequence	ACCCCAGGATCAACACTTCACATGCATTAGCAGAGAGAGATAAATCAA
+optional_text	+
quality	=>=??A?<@B@A:?B?D;AC@@CAAAD<AAA:99?:@=?@B@77C><4

FASTQ Format

```
@61CC3AAXX100125:7:118:2538:5577/1
GACACCTTTAATGTCTGAAAAGAGACATTCACCATCTATTCTCTTGGAGGGCTACCACCTAAGAGCCTTCATCCCC
+
?>CADFEEEBEDIEHHIDGGGEEEEHFFGIGIIFFIIEFHIIIIHIIFFIIIDEIIGIIIEHFFFIIEHIFA@?==
@61CC3AAXX100125:7:1:17320:13701/1
CTCAGAAGACCCTGAGAACATGTGCCCAAGGTGGTCACAGTGCATCTTAGTTTTGTACATTTTAGGGAGATATGAG
+
?BCAAADBBGGHGIDDDGHFEIFIIIIFGEIFIIFIGIGEFIIGGIIHEFFHHHIHEIFGHHIEFIIEECE?>@89
@61CC3AAXX100125:7:93:5100:14497/1
CTCAACTGGCTGAAAGTATTATCAATAGAAAGGAATGTTCAGGTTCTTCAATTTTAGAGTGCCCTGGCCTAGAAGA
+
?BCACEEGGGFICFFDECEGDEHFGFDEEGGEIEGFIFHIGEIGHIIHIGHGHHHEFF@GIIIIIIIIIHD@==98
@61CC3AAXX100125:6:92:7549:15004/1
CTTTTGCCAGTGACTCATCTGGCAGGTATCTCAAGTCAGCCCTTGCCTGGCCTGGCACCTTGCTGTGGTCTGAATG
+
?BBBCGFFDCDHCHHHFEHHIIIFEIEDFIIIGGEFIGEIIIIHIIIIIIIIIIIIGIIIHIIIIIIIGFHE;=:>
@61CC3AAXX100125:5:7:1488:7780/1
CCTGAGCTGCAGCACAGAGTGGAGGTAGTGGGGAGCTGTCACCTGGGTATGCCCCCCTTTCCCCTGTGCCATCACT
+
9==>:<CDEEB@FCFC@?@>G=;AF<9<8@>;4:;G@DAE@9HCIH@<>?728$'=B8@:68CB8>>8<8D=;<>8
```

Phred Quality Scores

- Encodes the probability of a call being an error
  - Phred Quality score Q = $-10*\log_{10}P$
  - Error probability $P = 10^{-Q/10}$
  - Example: call with Q=30 has an error probability
    - $P = 10^{-30/10}$
    - $P = 10^{-3} = 1$ in 1000
    - ASCII encoding – more compact – single byte

encoding	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4
Q score	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Phred Quality Scores

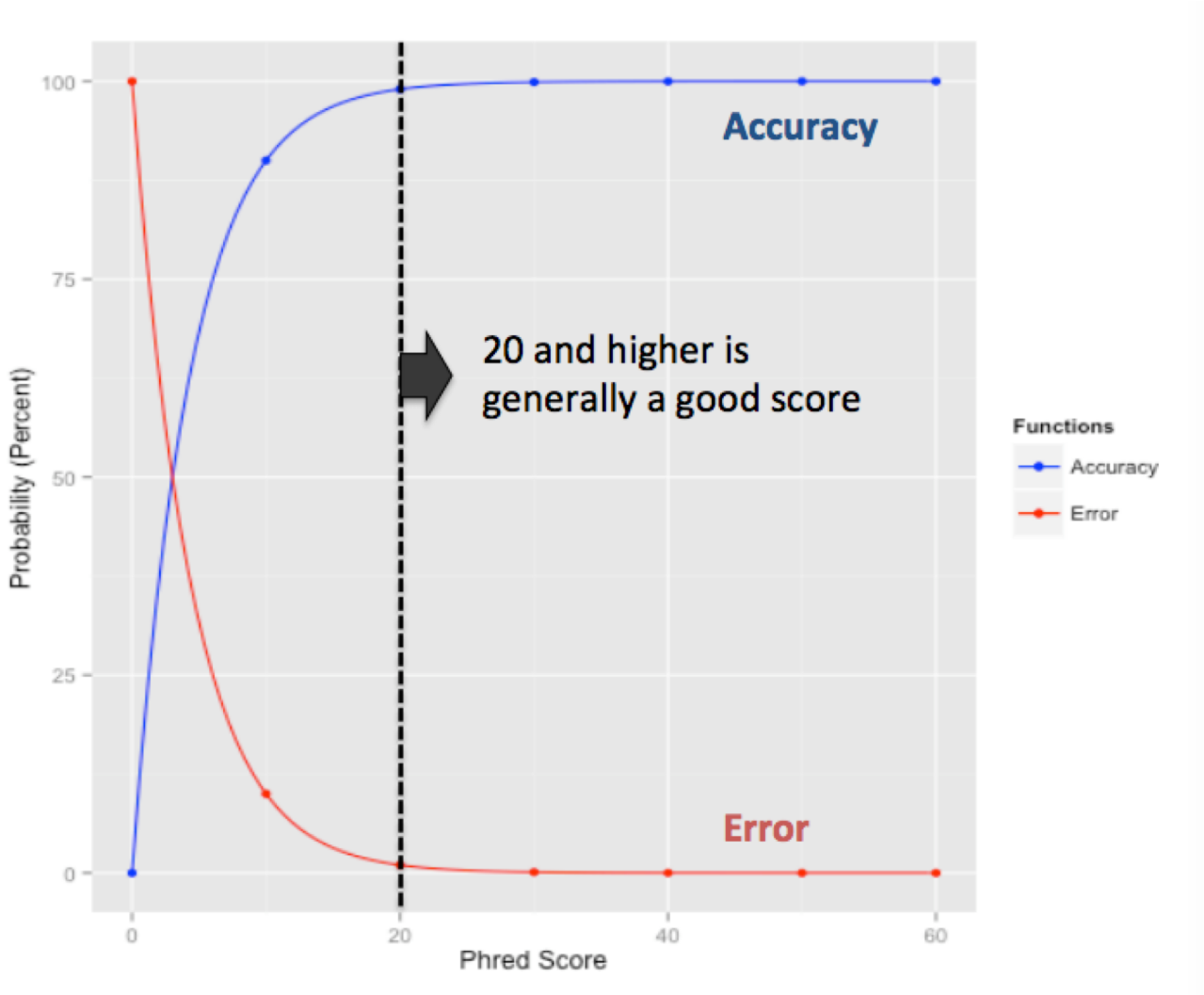
- The first 33 ASCII characters are reserved for control characters so the quality score for a base call needs to have 33 subtracted from it.
  - the ASCII character D = 68
    - Q = 68 – 33 = 35
  - the ASCII character H = 72
    - Q = 72 -33 = 39

- Beware that there are other quality score systems out there - Solexa, Sanger, etc.

Phred Quality Scores

- Encodes the probability of a call being an error
  - Phred Quality score Q = $-10*\log_{10}P$
  - Examples
    - 90% confidence (10% error rate) = Q10
    - 99% confidence (1% error rate) = Q20
    - 99.9% confidence (.1% error rate) = Q30

Phred Quality Scores



QC and Trimming

- FASTQ files -> FASTQC
- Review the various quality parameters
- Trimming software allows for bad quality reads to be removed or trimmed based on selection thresholds

QC and Trimming

- Various methods for attempting to correct bad quality data
  - Read trimming – trim off adaptors or low quality regions of reads
  - Can be fixed length or quality based
    - Fixed length – fixed size read only
    - Quality based – only retain reads that have an average quality score
- Quality score threshold + minimum length
- Adaptor trimming – platform specific based on identifying known adaptor sequences

Alignment vs. Assembly

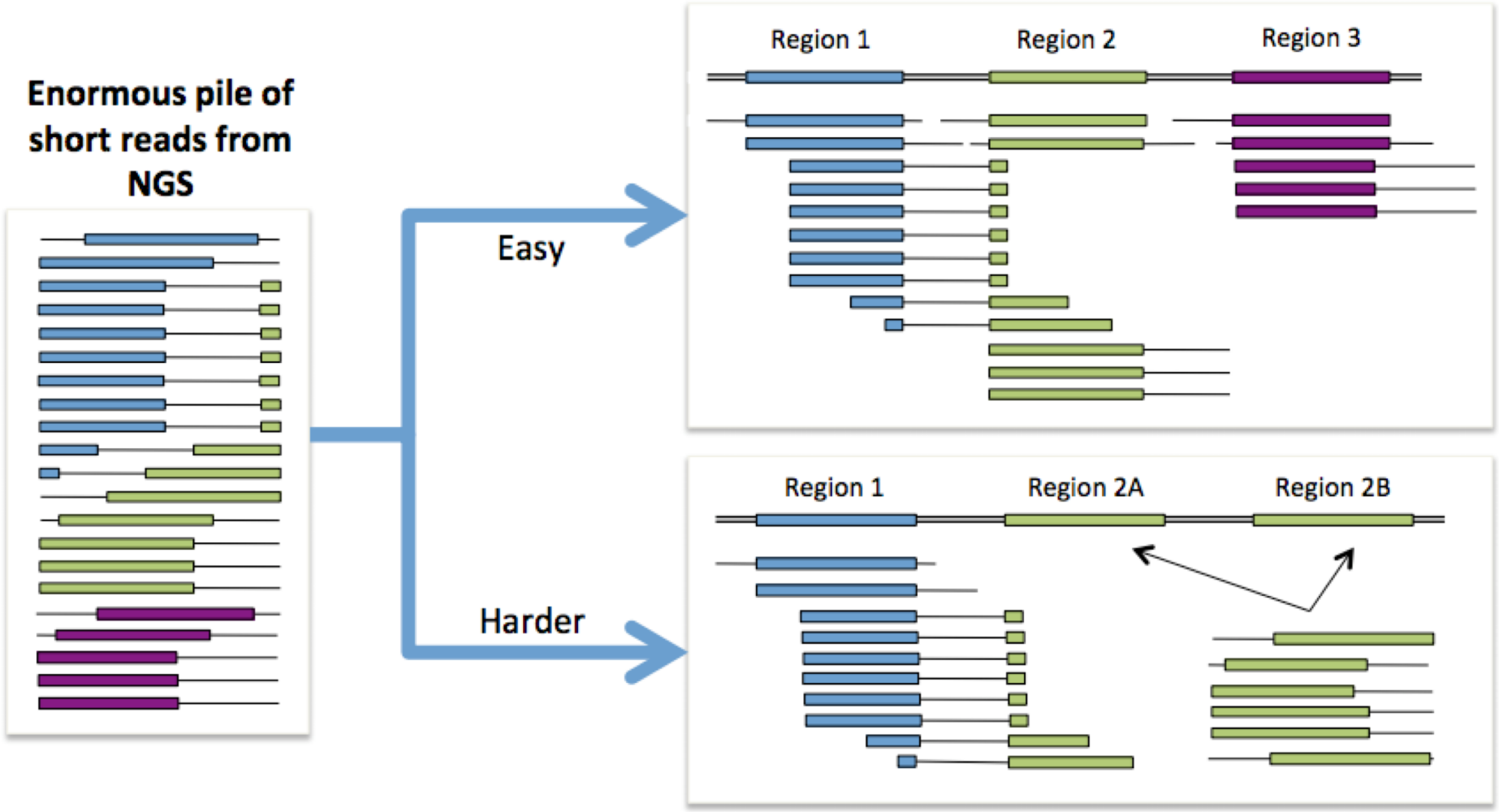
- *De Novo* Assembly
  - Merge overlapping fragments(reads) of DNA sequence into a larger full length contigs
  - Usually applicable when you do not have a reference genome

- Alignment/read mapping
  - Alignment or mapping back to a reference genome
  - Either a complete or partial (draft) reference genome is required

Alignment Mapping

- Aim is to map reads to correct positions on the reference genome
- This has to be done for each read in the dataset
- Keep in mind that reads may not match exactly to the reference genome
- There are millions of reads so computationally intensive

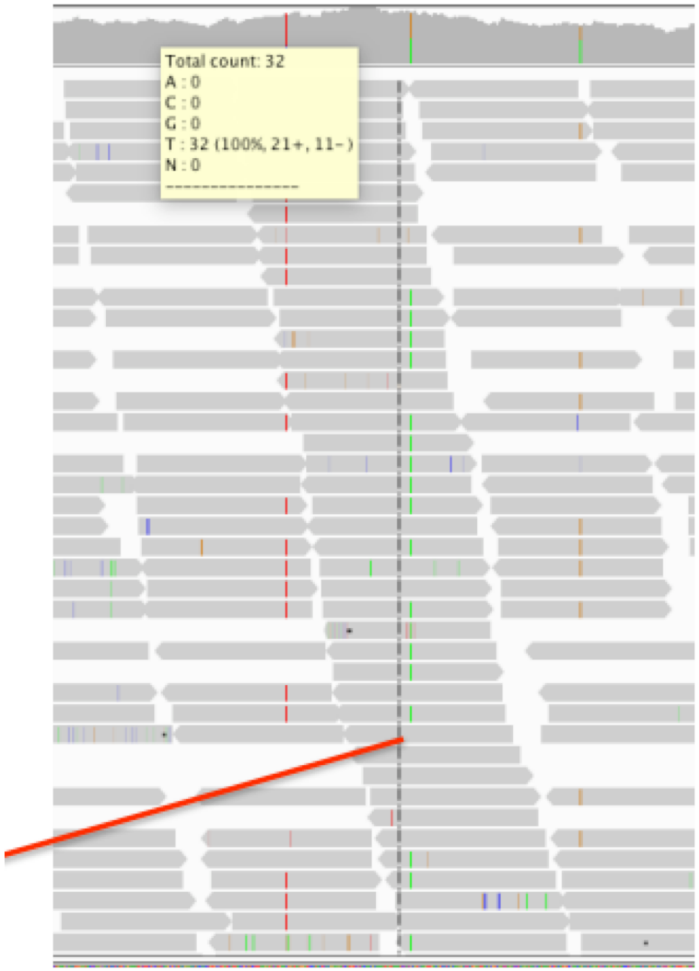
Alignment Mapping



Alignment Mapping

- Most alignment tools use an indexed genome when running the alignment
  - An “indexed” genome can be compared to the index of a book. You look through an index to easily find the topic of interest you are looking for in a book. In the same way when you index a genome you generate an index file which allows the tools to more easily and efficiently search through the genome when trying to map reads.

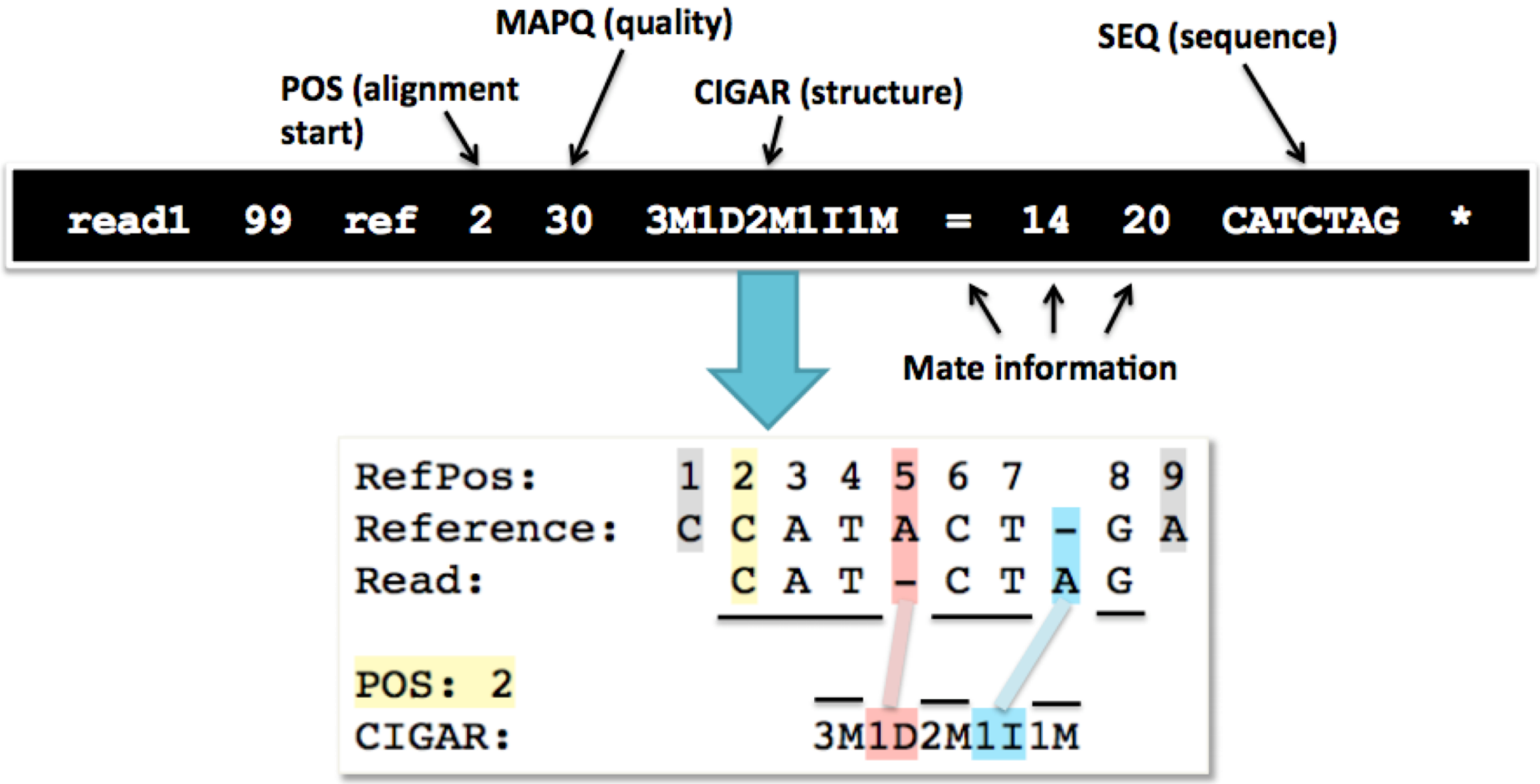
Aligned Reads



Alignment

- Input QC'd FASTQ (Tool BWA)
- Output of read alignment is generally a Sequence Alignment Map (SAM) file
  - Standardised method for storing all information relevant to how reads aligns to a reference genome
  - 11 mandatory fields + variable number of optional fields
  - One line for each read in the dataset
  - SAM files are rather big when dealing with large NGS datasets

Alignment Output - SAM



CIGAR Strings

CIGAR string

compact representation of sequence alignment:

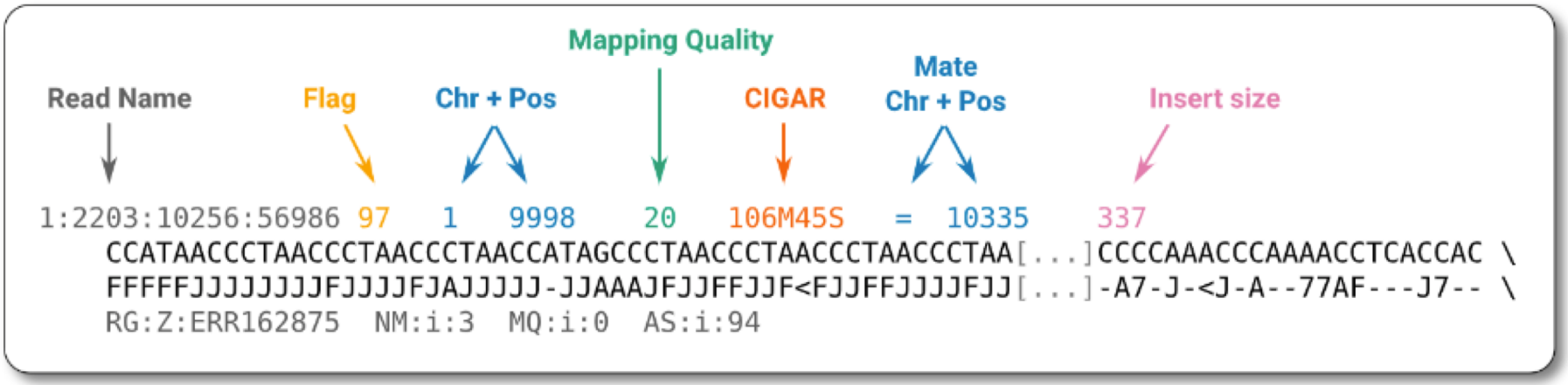
M	alignment match or mismatch
=	sequence match
X	sequence mismatch
I	insertion to the reference
D	deletion from the reference
S	soft clipping (clipped sequences present in SEQ)
H	hard clipping (clipped sequences NOT present in SEQ)
N	skipped region from the reference
P	padding (silent deletion from padded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

```
Ref:   ACGTACGTACTGT
Read:  ACGT----ACTGA
Cigar: 4M 4D 5M
```

```
Ref:   ACGT----ACGTA
Read:  ACGTACGTACGTA
Cigar: 4M 4I 5M
```

```
Ref:   CTCAGTG-GTCATCGTT
Read:  CGCA-TGAGTCTAGACG
Cigar: 4M 1D 2M 1I 3M 6S
```

Alignment Output – SAM*



Optional tags

AS	Alignment score by the aligner
NM	Edit distance to the reference
MQ	Mapping quality of the mate
RG	Read group

Read Group

ID	SRR/ERR number
PL	Sequencing platform
PU	Run name
LB	Library name
PI	Insert fragment size
SM	Individual
CN	Sequencing center

Insert size

length of the DNA fragment sequenced from both ends by paired-end sequencing:

Alignment Output - BAM

Binary Sequence Alignment (BAM) is a compressed version of SAM

- Data in BAM is binary and not human readable
- Efficient storage of alignment files
- Format can be read by downstream analysis tools

- Tools to convert SAM to BAM and process BAMs
  - Samtools, Picard, htslib

Alignment Output – CRAM*

- BAM files are still relatively large ~1.5 – 2 bytes per base pair
- Computer disk capacity is falling behind storage requirements for sequencing data
- CRAM is a reference-based compression technique
- Some quality information lost but in a controlled manner
- Used in most production pipelines now and results in up to 40% reduction in disk space usage

Alignment File QC

- The quality of the alignment can be assessed
- Various quality control measures that will be covered later in the course
- Output from this step is a QC'd BAM file

Variant Calling

- Is there a variant (SNP, indel or structural variant) at a particular position?
- Based on per-base evidence provided for all the reads that have mapped to a particular position in the sequence
- Useful to aggregate the evidence from all reads that relate to a particular base in the sequence
- This is called generating a “pile-up”
- Easier for SNPs than for indels and structural variation

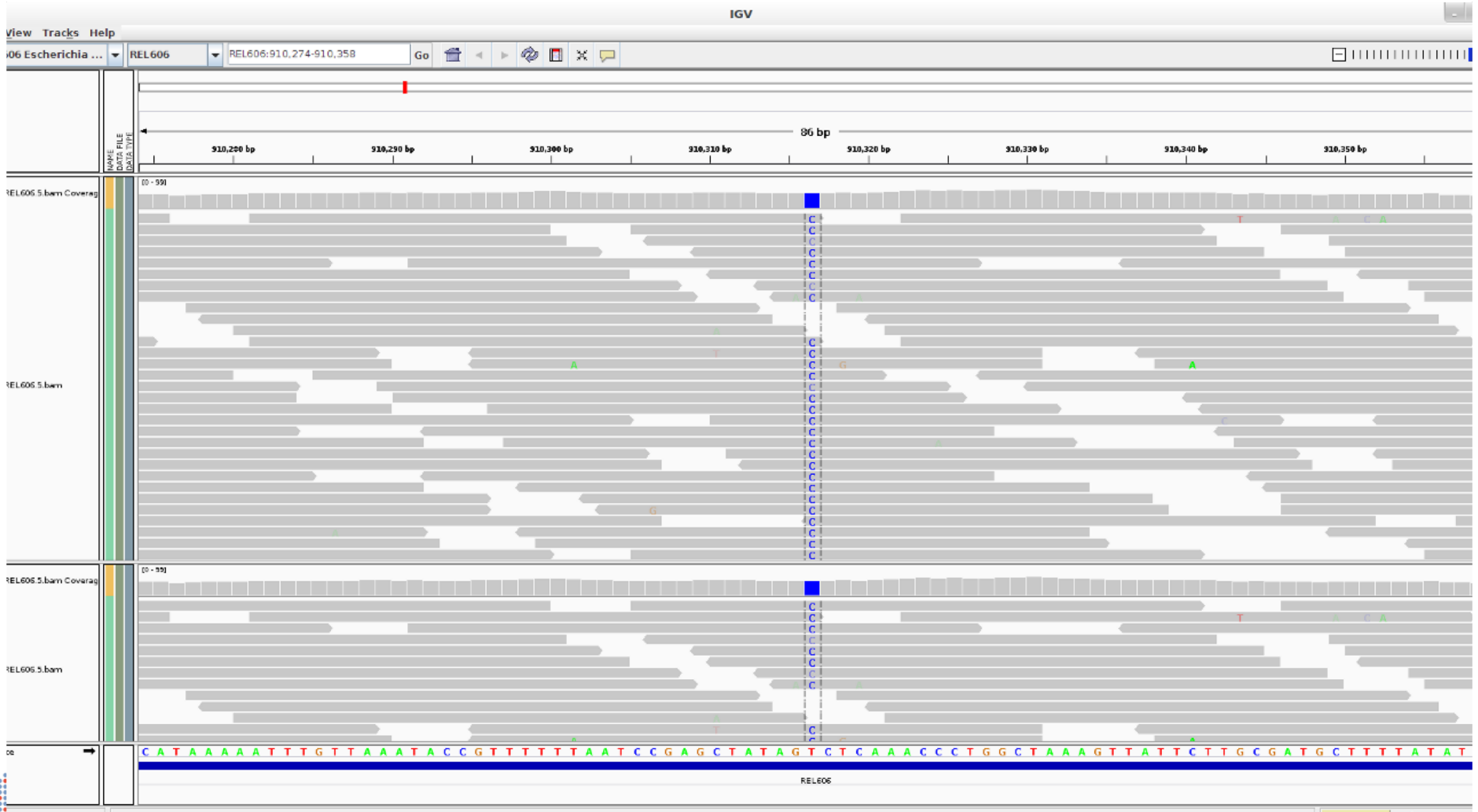
Variant Calling - PileUp

- The pileup file has as many lines as there are bases in the reference sequence that are aligned with reads in the SAM/BAM file
- Each line contains information about every base found in the sequence reads that aligns to a base in the reference sequence
- The Pileup file can therefore be filtered to retain only high quality variants

Variant Calling - PileUp

- Pileup file filtering
  - Remove poor quality base calls
  - Remove consensus base calls that have less than 10 reads supporting that position
  - Remove all bases that are the same as the reference sequence (not SNPs)

Variant Calling



Variant Calling

- There are more advanced tools for calling variants and genotypes

- Assess the likelihood of each possible genotype for each position in the reference genome, given the observed reads at that position and reports back a list of all variants.

- Most use some form of a Bayesian model variant caller to call variants and produces a quality or statistical parameter for confidence in the call of that variant

Variant Calling Output (VCF)

- Input for variant callers is a QC'd BAM file
- Variant callers produce a VCF file
- Format for SNPs, indels, structural variants and CNVs
- VCF – Variant Call Format
- Standardised format for representing variant calls
- Was developed and maintained by 1000 genomes project but now maintained by The Global Data Working Group File Formats task Force ga4gh.org

VCF format

- Stores variant data together with additional information on quality, annotations etc.
- Developed for fast searching and access (indexed)
- Includes additional metadata i.e. dbSNP accession numbers
- Additional tags can be added
- Can be zipped and indexed and read via UNIX commands

Basic Structure of a VCF

- Meta-information
  - Starts with ## and is a key=value pair
  - E.g. ##fileformat=VCFv4.2
  - INFO – information on various columns in the file
  - FILTER – information on filters that have been applied to the data
  - FORMAT – information on the format of standard columns

```
##INFO=<ID=CIGAR,Number=A,Type=String,Description="CIGAR alignment for each alternate indel allele">
##INFO=<ID=RU,Number=A,Type=String,Description="Smallest repeating sequence unit extended or contracted in
 the indel allele relative to the reference. RUs are not reported if longer than 20 bases.">
##INFO=<ID=REFREP,Number=A,Type=Integer,Description="Number of times RU is repeated in reference.">
##INFO=<ID=IDREP,Number=A,Type=Integer,Description="Number of times RU is repeated in indel allele.">
##FILTER=<ID=IndelConflict,Description="Locus is in region with conflicting indel calls">
##FILTER=<ID=SiteConflict,Description="Site genotype conflicts with proximal indel call. This is typically
 a heterozygous SNV call made inside of a heterozygous deletion">
##FILTER=<ID=LowGQX,Description="Locus GQX is less than 30 or not present">
##FILTER=<ID=HighDPFRatio,Description="The fraction of basecalls filtered out at a site is greater than 0.
3">
##FILTER=<ID=HighSNVSB,Description="SNV strand bias value (SNVSB) exceeds 10">
##FILTER=<ID=HighREFREP,Description="Locus contains an indel allele occurring in a homopolymer or dinucleo
tide track with a reference repeat greater than 8">
##FILTER=<ID=HighDepth,Description="Locus depth is greater than 3x the mean chromosome depth">
##fileDate=20140414
##source=IsaacVariantCaller
##startTime=Mon Apr 14 17:19:59 2014
```

Header

- Meta information followed by a header with 8 mandatory fields:
  - CHROM – chromosome number
  - POS – contig position of variant
  - ID – The dbSNP identifier if record exists in dbSNP
  - REF – The reference base on the forward strand
  - ALT – The alternative base observed in your sample or population
  - QUAL – The Phred scaled quality score that the ALT allele exists at the site (higher, less probabilty that call occurs due to chance)
  - FILTER – PASS of FAIL call based on filtering approach used for calling variants
  - INFO – Additional information – various number of acceptable information encoded as key:value pairs

Genotype Entries

```
[HEADER LINES]
#CHROM  POS ID      REF ALT QUAL    FILTER  INFO          FORMAT
   NA12878
chr1    873762  .       T   G   5231.78 PASS    [ANNOTATIONS] GT:AD:DP:
GQ:PL  0/1:173,141:282:99:255,0,255
chr1    877664  rs3828047   A   G   3931.66 PASS    [ANNOTATIONS] GT:AD
:DP:GQ:PL  1/1:0,105:94:99:255,255,0
chr1    899282  rs28548431  C   T   71.77   PASS    [ANNOTATIONS] GT:AD
:DP:GQ:PL  0/1:1,3:4:25.92:103,0,26
chr1    974165  rs9442391   T   C   29.84   LowQual [ANNOTATIONS] GT:AD
:DP:GQ:PL  0/1:14,4:14:60.91:61,0,255
```

Genotype Entries

- Looks complicated but not:
  - GT : the genotype of the sample. For diploid organisms the GT field indicates two alleles in the sample 0 for REF allele and 1 for the ALT allele. So GT is either 0/0, 0/1 or 1/1
  - GQ : Genotype quality – Phred-scaled confidence that the true genotype is the one provided in GT – higher more confidence in call
  - AD and DP are complementary fields that represent the depth of the data for the sample at the site. DP (coverage) AD (DepthPerAlleleBySample)
  - PL provides the likelihood of the given genotypes 0/0, 0/1, 1/1 – Phred scaled likelihoods of all three genotypes. Likelihoods are normalised and log10-scaled.

Interpretation Example

```
chr1    899282  rs28548431  C   T   [CLIPPED] GT:AD:DP:GQ:PL    0/1:1,3
:4:25.92:103,0,26
```

- Site on chr1 at position 899282
- Has been identified before indicated by rsID
- REF allele C and ALT allele T
- Genotype = 0/1 = C/T – Het for ALT allele
- GQ = 25.92 – low as DP is only equal to 4 reads of which 1 read was REF and 3 ALT alleles AD=1,3
- PL (0/1) = (likelihood 1), PL(1/1) = 26 (likelihood $10e^{-2.6}$ or 0.0025), PL(0/0) = 103 (likelihood $10e^{-10.3}$) very small number

Interpretation Example*

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLE1	SAMPLE2	SAMPLE3
11	24535	.	G	A	243	PASS	DP=221;AF=0.5	GT:AD	0/1:73,15	0/0:48,0	0/1:71,14
12	153927	.	C	CA,T	15	LowQ	AF=0,0.1	GT	2/2	1/2	0/1

Additional Tags

- Variants in a VCF can be annotated with additional tags:
  - Formatted in a similar way to the genotype tags
  - Various set of commonly used built in annotations
  - Additional functional annotations can be added to VCF

```
##INFO=<ID=hgmd_id,Number=.,Type=String,Description="HGMD Variant ID">
##INFO=<ID=hgmd_disease,Number=.,Type=String,Description="HGMD Disease">
##INFO=<ID=hgmd_alleles,Number=.,Type=String,Description="HGMD wild-type/mutant annotation">
##INFO=<ID=hgmd_gene,Number=.,Type=String,Description="Gene with an annotated variant in HGMD">
##INFO=<ID=gwas_sig,Number=.,Type=String,Description="GWAS significance exponent">
##INFO=<ID=gwas_rr,Number=.,Type=String,Description="GWAS relative risk">
##INFO=<ID=gwas_raf,Number=.,Type=String,Description="GWAS risk allele frequency">
##INFO=<ID=gwas_id,Number=.,Type=String,Description="GWAS associated variant ID">
##INFO=<ID=gwas_allele,Number=.,Type=String,Description="GWAS associated allele">
##INFO=<ID=gwas_association,Number=.,Type=String,Description="GWAS description">
##INFO=<ID=phastCons,Number=0,Type=Flag,Description="overlaps a phastCons element">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 135">
```

Genome VCF (gVCF)*

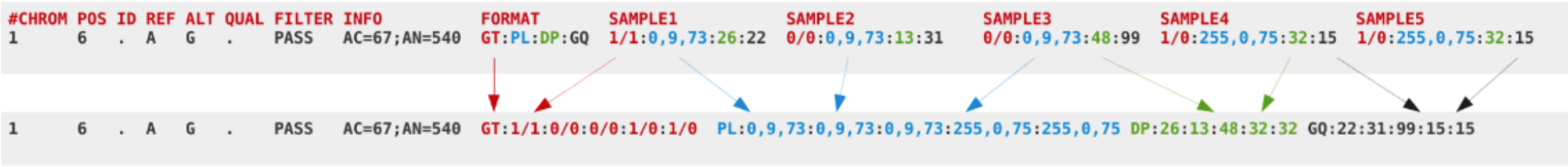
- While we are usually only interested in variant sites, we may also want to know if at a non-variant site, no alternative allele was observed or there was not data or coverage for that position

gVCF

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
19	9902	.	G	.	.	.	MinDP=0;END=9905
19	9906	.	G	.	.	.	MinDP=5;END=9909
19	9910	.	G	A	.	.	DP=15
19	9911	.	T	.	.	.	MinDP=14;END=9915
19	9916	.	G	T	.	.	DP=18
19	9917	.	A	.	.	.	MinDP=16;END=9920

VCF -> BCF*

- VCFs can become very large with multiple samples in one file ~680 GB for WGS in 3781 samples
- BCF is a binary version of the VCF with fields rearranged for faster access



Tools for working with VCFs

- Number of tools developed to work with and query VCF files
  - VCFTools (http://vcftools.sourceforge.net/)
  - Samtools (https://samtools.github.io/)
  - BCFtools (https://samtools.github.io/bcftools/)
- Will explore this further in the practical
- Interpretation, filtering and validation

Summary

- NGS analysis encompasses various steps
- Each step has a specific input and output file format
- The formats are specific to the tools used at each of the analysis steps
- Knowledge of the various formats allows for the identification and extraction of specific information from the different files