Tertiary Analysis

- Working with BAM/SAM files

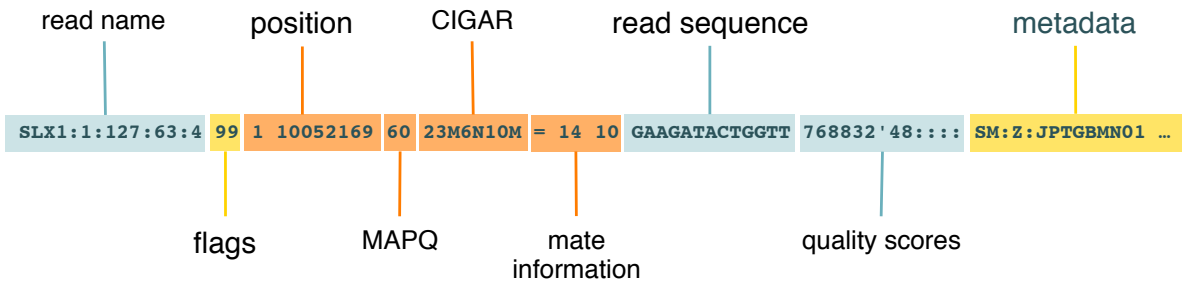
SAM Specification

- Latest vesion SAMv1 (http://samtools.github.io/hts-specs/)

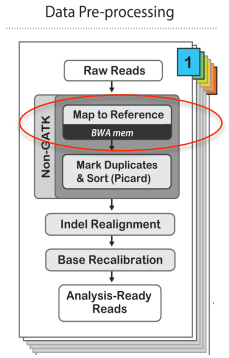
SAM File Format

Output format: Sequence/Binary Alignment Map (SAM/BAM)

HEADER containing metadata (sequence dictionary, read group definitions etc)
RECORDS containing structured read information (1 line per read record)



- Added mapping info summarizes **position**, **quality**, and **structure** for each **read**
- A BAM file can contain data from a single or from several samples



http://samtools.github.io/hts-specs/SAMv1.pdf

Header Tags

- `@HD` Version Info
- `@SQ` Genome Information (chrom, size, location, species)
- `@PG` Program tags. Information on programs that create this BAM
- `@RG` Read Groups. Information on origin of sequence data
  - Allows multiple samples to be merged into one BAM
- `@CO` Commenents

Read lines

Read Entries in SAM

No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)

Heng Li , Bob Handsaker , Alec Wysoker , Tim Fennell , Jue Ruan , Nils Homer , Gabor Marth , Goncalo Abecasis , Richard Durbin , and 1000 Genome Project Data Processing Subgroup (2009) The Sequence Alignment/Map

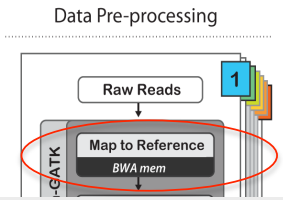
CIGAR format

CIGAR summarizes **alignment structure**

CIGAR = Concise Idiosyncratic Gapped Alignment Report

```
read1  99  ref  2  30  3M1D2M1I1M  =  14  20  CATCTAG  *
```

```
RefPos:      1 2 3 4 5 6 7   8 9
Reference:   C C A T A C T - G A
Read:          C A T - C T A G

POS: 2
CIGAR:           3M1D2M1I1M
```



Flags

- major headache for humans but the right thing to do.
  - But why on earth is strand bit 4 and not bit 1; the thing you want most should be in the first bit: even == positive, odd == negative
  - old samtools had -X option but really not that much better

Dec	Hex	Flags	Dec	Hex	Flags	Dec	Hex	Flags
65	0x41	p1	69	0x45	pu1	73	0x49	pU1
81	0x51	pr1	97	0x61	pR1	113	0x71	prR1
117	0x75	purR1	121	0x79	pUrR1	129	0x81	p2
133	0x85	pu2	137	0x89	pU2	145	0x91	pr2
161	0xa1	pR2	177	0xb1	prR2	181	0xb5	purR2
185	0xb9	pUrR2	321	0x141	p1s	329	0x149	pU1s
337	0x151	pr1s	353	0x161	pR1s	369	0x171	prR1s
377	0x179	pUrR1s	385	0x181	p2s	401	0x191	pr2s

Flags; better solution

- PICARD page is a life saver; bookmarkit or download it https://broadinstitute.github.io/picard/explain-flags.html

SAM file example (from STAR)

```
@HD     VN:1.4
@SQ     SN:4    LN:191154276
@SQ     SN:7    LN:159138663
@SQ     SN:12   LN:133851895
@SQ     SN:17   LN:81195210
@PG     ID:STAR PN:STAR VN:STAR_2.5.1b  CL:STAR   --genomeDir /share/data/compgen2016/
day45_Intro2Seq_VarCalling/genomes/H.Sapiens/b37_hl/index/star/NoGTF   --readFilesIn /share/data/compgen2016/
day45_Intro2Seq_VarCalling/Labs/2_Mapping/data/gencodeTest2_Q30_1_R1.fastq.gz   /share/data/compgen2016/
day45_Intro2Seq_VarCalling/Labs/2_Mapping/data/gencodeTest2_Q30_1_R2.fastq.gz      --readFilesCommand gzcat
@CO     user command line: STAR --genomeDir /share/data/compgen2016/day45_Intro2Seq_VarCalling/genomes/
H.Sapiens/b37_hl/index/star/NoGTF --readFilesIn /share/data/compgen2016/day45_Intro2Seq_VarCalling/Labs/
2_Mapping/data/gencodeTest2_Q30_1_R1.fastq.gz /share/data/compgen2016/day45_Intro2Seq_VarCalling/Labs/2_Mapping/
data/gencodeTest2_Q30_1_R2.fastq.gz --readFilesCommand gzcat
ENST00000000233.5_540_1027_0:0:0_0:0:0_0        163     7       127231072       255     71M4S   =
127231609       612
ACATGCCCAACGCCATGCCCGTGAGCGAGCTGACTGACAAGCTGGGGCTACAGCACTTACGCAGCCGCACGTGGT     ??????????????????????????????
????????????????????????????????????????????     NH:i:1  HI:i:1  AS:i:144        nM:i:0
ENST00000000233.5_540_1027_0:0:0_0:0:0_0        83      7       127231609       255     75M     =
127231072       -612
AGAGGAGGAGCAGGGATCTGGGTTTCCTTTTTTTTTTCTGTTTTGGGTGTACTCTAGGGGCCAGGTTGGGAGGGG     ??????????????????????????????
????????????????????????????????????????????     NH:i:1  HI:i:1  AS:i:144        nM:i:0
ENST00000000233.5_536_1088_0:0:0_1:0:0_1        99      7       127231068       255     75M     =
127231670       677
CAGGACATGCCCAACGCCATGCCCGTGAGCGAGCTGACTGACAAGCTGGGGCTACAGCACTTACGCAGCCGCACG     ??????????????????????????????
????????????????????????????????????????????     NH:i:1  HI:i:1  AS:i:146        nM:i:1
ENST00000000233.5_536_1088_0:0:0_1:0:0_1        147     7       127231670       255     75M     =
127231068       -677    CAGGTTGGGAGGGGGAAGGTGAGGGCTTCGGGTGGTGCTTTAATGTGGCACTGGATCTTGAGTAATAAATTTGCT
```

Manipulating SAM/BAM files

Samtools vs Picard

- When there is overlap, my honest advice, use Picard
- Unless you are doing pipes/streams
  - But probably should not be doing those anyway
- However `samtools view` is prehaps the most used samfile command ever (really)
  - go over options

Samtools

```
$ samtools

Version: 1.3.1 (using htslib 1.3.1)

**    faidx          index/extract FASTA
      index          index alignment

      reheader       replace BAM header
!!    rmdup          remove PCR duplicates # Careful Do not Use

**    mpileup        multi-way pileup
      sort           sort alignment file # Sort to pipes
      quickcheck     quickly check if SAM/BAM/CRAM file appears intact

      flagstat       simple stats
      idxstats       BAM index stats

      flags          explain BAM flags
*     tview          text alignment viewer
**** view            SAM<->BAM<->CRAM conversion # cat for BAMs
```

PICARD

- manipulating SAM/BAMs
  - AddRG, Sort, Index & MarkDup in almost every pipeline
  - Mark Duplicates a key step in many cases
- BAM stats
  - Alignment Stats
  - Insert Size
  - Duplicates Stats
  - and a bunch of misc other stuff
- Wins award for friendliest bioinformatics tool
  - Again honest advice if Picards does what you need use it over other tools.

Core modules for Variant Pipeline

- AddOrReplaceReadGroups (AddCommentsToBam)
  - This one module can do three key step to convert raw SAM output from mappers to BAM
    - Add ReadGroups
    - Sort (in same step)
    - Index
- MergeSamFiles
  - Often the Mapping phase is chunked into blocks need to merge before next step
- MarkDuplicates (MarkDuplicatesWithMateCigar)
  - Gets it own slides
- Metrics

Metrics

```
CalculateHsMetrics
CollectAlignmentSummaryMetrics
CollectBaseDistributionByCycle
CollectGcBiasMetrics
CollectHiSeqXPfFailMetrics
CollectHsMetrics
CollectInsertSizeMetrics
CollectJumpingLibraryMetrics
CollectMultipleMetrics
CollectOxoGMetrics
CollectQualityYieldMetrics
CollectRawWgsMetrics
CollectRnaSeqMetrics
CollectRrbsMetrics
CollectSequencingArtifactMetrics
CollectTargetedPcrMetrics
CollectVariantCallingMetrics
CollectWgsMetrics
CollectWgsMetricsFromQuerySorted
CollectWgsMetricsFromSampledSites
CompareMetrics
ConvertSequencingArtifactToOxoG
EstimateLibraryComplexity
MeanQualityByCycle
QualityScoreDistribution
```

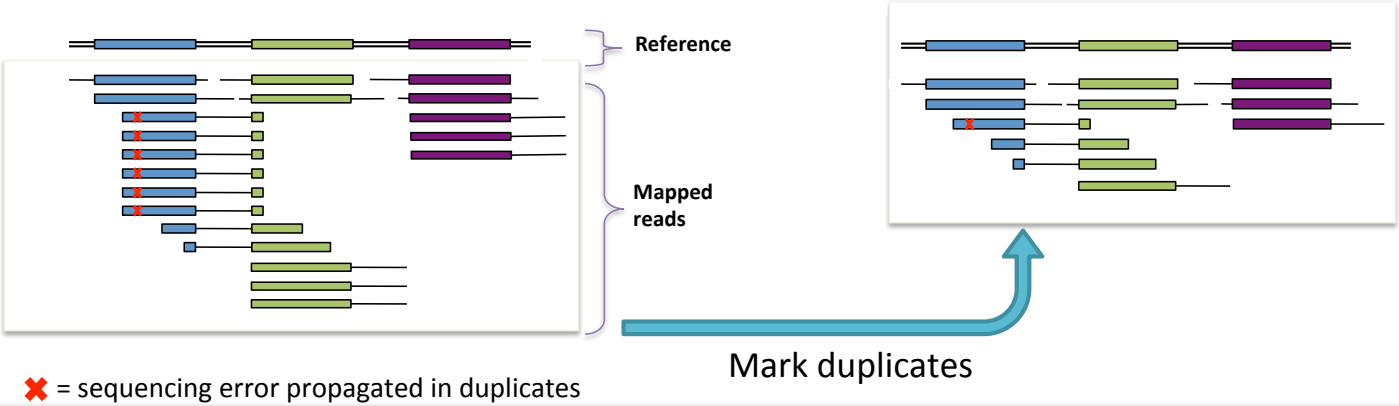
Metrics

```
CalculateHsMetrics
CollectAlignmentSummaryMetrics

CollectHsMetrics
CollectInsertSizeMetrics

                                        ConvertSequencingArtifactToOxoG
CollectOxoGMetrics

CollectRnaSeqMetrics
```

Mark Duplicates

Why mark duplicates?

- Duplicates are sets of reads pairs that have the same unclipped alignment start and unclipped alignment end
- They're suspected to be **non-independent measurements** of a sequence
  - Sampled from the exact same template of DNA
  - Violates assumptions of variant calling
- What's more, errors in sample/library prep will get propagated to *all* the duplicates
  - Just pick the "best" copy – mitigates the effects of errors



How to indentify duplicates

- Duplicates might come from the same input DNA template, so we will assume that reads will have same start position on reference
  - "Where was the first base that was sequenced?"
  - For paired-end (PE) reads, same start for both ends
- Identify duplicate sets, then choose representaive read based on base quality scores and other criteria
- Lots of complications:
  - clipping (MarkDuplicatesWithMateCigar)
  - ...

Picard tool MarkDuplicates

- Duplicate status is indicated in SAM flag
- Duplicates are not removed, just tagged (unless you request removal)
- Downstream tools can read the tag and choose to ignore those reads
- Most GATK tools ignore duplicates by default

Sometimes do not want to do this.

- Amplicon sequencing (PCR based assay)
  - all reads start at same position by design

In somecase if the depth is too large MarkDup's will crash

Different kinds of Noise

- Random/uncorrelated (White) vs correlated/structured/biases (colored)
- Both present challenges for algorithms but non-white noise in many contexts can be especially difficult (if not impossible).
  - PCR Duplicates (MarkDups)
  - Adapter sequences (Clip)

Multi-mapper issue

- Many pipeline simple filter these reads out.
- BWA MEM problem
  - No longer sets simple flag
  - if using filter on MAPQ
- If using multi-mappers in uniq-mode need to really make sure:
  - how the algorithm deals with high multiplicity
  - random choice?
- Bowtie/SHRiMP for exhaustive multi-mappers
- CSEM (http://deweylab.biostat.wisc.edu/csem/)
  - impute likely position of multi-mappers by looking at surronding unique mappers.

Other bioinformatics file formats: BED files

- BED (0-offset)
  - standard 3 column format:
    - chromsome
    - start (first base is 0)
    - end
  - various extended version

Picard interval lists

- Used by Picard:
- Genome Header so you know what the reference is
- Standard 5 column format
  - Chromsome
  - Start (first base is 1)
  - End
  - Strand (REQUIRED)
  - Feature Name (REQUIRED)

Other range formats: GTF,GFF

- GFF/GTF: General Feature Format (1-offset)
  - 9 Columns (see http://www.ensembl.org/info/website/upload/gff.html)
    - but 9th column is a COMMENT field that can pretty much hold anything arbtrary key/value pairs
- GTF: General Transfer Format == GFF v2
  - GFF with "rules" (kind of) about what goes in column 9

Other range formats; UCSC

General Formats:

UCSC Genome Bioinformatics

Genomes | Genome Browser | Tools | Mirrors | Downloads | My Data

Frequently Asked Questions: Data File Formats

General formats:

- Axt format
- BAM format
- BED format
- BED detail format
- bedGraph format
- bigBed format
- bigGenePred table format
- bigWig format
- Chain format
- GenePred table format
- GFF format
- GTF format
- HAL format
- MAF format

Swiss Army knife of range formats

BEDTOOLS

- Genome Arithmetic
- Handles:
  - BED
  - BAM
  - GFF/GTF
  - VCF
- Another package that is also very useful: GenomicRanges in `R`

Lab 3