

# SAM, BAM and samtools

Formats and tools for managing next-  
gen sequencing data

<http://samtools.sourceforge.net/SAM-1.3.pdf>

# Outline

- Project Two overview
  - Counting reads to estimate gene expression
- Intro to SAM and BAM
  - Compact alignment formats for NGS
- Samtools
  - Unix utilities for working with SAM/BAM files, NGS

<http://samtools.sourceforge.net/SAM-1.3.pdf>

# Project Two

- `countReads.py`
  - Reports the number of “reads” in a SAM file that overlap each gene model in a “bed” format file
- Run from the command line like this:
  - `cat sample.sam | countReads.py -b TAIR.bed > out.txt`
  - `countReads.py -a sample.sam -b TAIR.bed > out.txt`

# Supported methods - getStartEnd

- Accepts a SAM format line and returns a tuple with start and end position of the alignment

```
# just a stub to get you started
def getStartEnd(sam_line):
    return (0,0)
```

<http://samtools.sourceforge.net/SAM-1.3.pdf>

# Supported methods - reportReads

- Accepts a three named arguments – files opened for reading (bed\_fh, sam\_fh) and one file opened for writing (out\_fh)
- Counts the number of reads overlapping each gene model in the bed file
- Writes bed format lines with “score” field containing the number of reads overlapping the model

```
# just a stub to get you started
def reportReads (bed_fh=None, sam_fh=None, out_fh=None) :
    pass
```

# Example

- SAM file has 10 reads that overlap gene model AT1G07350.1

Original line is:

```
'chr1\t2257424\t2260101\tAT1G07350.1\t0\t-\t2257731\t2260101\t0\t9\t0,101,109,150,64,460,482,\t2587,2061,1852,1578,1114,563,0,'
```

New line is:

```
'chr1\t2257424\t2260101\tAT1G07350.1\t10\t-\t2257731\t2260101\t0\t9\t0,101,109,150,64,460,482,\t2587,2061,1852,1578,1114,563,0,'
```

# What is a SAM file?

- SAM - Sequence Alignment/Map format
- Tab-delimited text format with two sections:
  - Header – meta-data about the alignments
  - Alignments themselves
    - Each alignment is on one line, has 11 required fields, many optional fields that use an easy-to-parse syntax
- SAM files can be sorted
  - Makes searching faster

<http://samtools.sourceforge.net/SAM-1.3.pdf>

# BAM is binary SAM

- BAM uses a compression scheme to make alignments more compact
- BAM files can be sorted and indexed
  - Makes accessing data very fast
- BAI (extension .bai) is the index for a BAM file
  - Sample.bam has index file Sample.bam.bai

<http://samtools.sourceforge.net/SAM-1.3.pdf>

# Use samtools to view contents of BAM

- samtools
  - Unix-friendly command line program for viewing contents of BAM file
  - Open source project
  - Started mainly to support 1000 genomes project
- Course Web site has a link to “zip” file containing samtools compiled for Mac
- Demo and description of the format

# Header

- Each line starts with @ and is tab-delimited
  - @HD comes first, followed by version string indicating the SAM file version and SO tag indicating whether the file is sorted
  - @SQ comes next (if file is sorted in any way)
    - This is the “sequence dictionary” and reports the names and lengths of the reference sequences used to make the alignments

# View just the header

```
$ samtools view -H treatment_drought2_out.bam
@HD VN:1.0 SO:sorted
@SQ SN:chr1 LN:30427671
@SQ SN:chr2 LN:19698289
@SQ SN:chr3 LN:23459830
@SQ SN:chr4 LN:18585056
@SQ SN:chr5 LN:26975502
@SQ SN:chrC LN:154478
@SQ SN:chrM LN:366924
@PG ID:TopHat VN:1.1.4 CL:/common/lorainelab/sw/
tophat-1.1.4.Linux_x86_64/tophat --max-multihits 1 --min-
intron-length 20 --solexa1.3-quals -I 1200 -p 8 -F 0 -o
tophat-1.1.4-pilot-pollen/treatment_drought2_out -G /storage/
lorainelab/fastapub/tair/TAIR9_GFF3_genes-fixed.gff /storage/
lorainelab/bowtieindex/a_thaliana chapel_hill_1/
s_8_sequence.txt,chapel_hill_2/s_4_sequence.txt
```

# Alignments

## 1.4 The alignment section: mandatory fields

Each alignment line has 11 mandatory fields. These fields always appear in the same order and must be present, but their values can be '0' or '\*' (depending on the field) if the corresponding information is unavailable. The following table gives an overview of the mandatory fields in the SAM format:

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2 <sup>16</sup> -1]	bitwise FLAG
3	RNAME	String	\*  [!-( )+-<>-~] [!-~]*	Reference sequence NAME
4	POS	Int	[0,2 <sup>29</sup> -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 <sup>8</sup> -1]	MAPping Quality
6	CIGAR	String	\*  ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	\* =  [!-( )+-<>-~] [!-~]*	Ref. name of the mate/next fragment
8	PNEXT	Int	[0,2 <sup>29</sup> -1]	Position of the mate/next fragment
9	TLEN	Int	[-2 <sup>29</sup> +1,2 <sup>29</sup> -1]	observed Template LENgth
10	SEQ	String	\*  [A-Za-z=.]+	fragment SEQUENCE
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

1. QNAME: Query template NAME. Each template has a unique name.

# Use samtools view to retrieve reads overlapping a region of interest

```
$ samtools view treatment_drought2_out.bam
chr1:1,000,000-1,000,001 | more
61C2DAAXX:4:91:1662:10658#0      16      chr1      999935
255      75M      *      0      0
TTAAGGCTCCCATTTACACTATCGAAAAAGATGGGACAAGTGCTGAAACGTGTATGAT
CCGGTTCCATGCTGGTC
4BBBBBCBBBACBBBBBCBCBCDC@BCCCC@CBCACBCCCCCCCCCCCCBCCCCCCCCCCC
CCCCCCCCCCCCCCCC      NM:i:0  NH:i:1
```

## Mandatory Fields

- |                                                     |                                         |
|-----------------------------------------------------|-----------------------------------------|
| 1 – QNAME name of the query sequence                | 6 – CIGAR describes the alignment       |
| 2 – FLAG bitwise flag score (ignore for now)        | 7 – RNEXT (ignore for now)              |
| 3 – RNAME name of the reference sequence            | 8 – PNEXT (ignore for now)              |
| 4 – POS start position of the alignment (one-based) | 9 – TLEN (ignore for now)               |
| 5 – MAPQ alignment quality score (ignore for now)   | 10 – SEQ query sequence                 |
|                                                     | 11 – QUAL base quality (ignore for now) |

# Optional fields

- Most alignment tools include at least some in their output

- Syntax

- TAG:TYPE:VALUE

- Example

- NM:i:0

- NH:i:1

**NM** – edit distance to the reference  
(number of bases that need to change to perfectly match the reference)

**NH** – Number of reported alignments that contain the query in the current record – the number of times the read aligned

**i** – type is integer

For more, see the SAM spec

# CIGAR

- Use this to calculate the end position of the alignment
- Example
  - Read X starts at position 100 and has CIGAR code 75M (75 matches)
  - Start position in interbase is  $100-1=99$
  - End position is  $99+75=74$

More next time – How to handle “N”