Genomics Part 3: DNA-seq Common Files and Pipelines

Bridging the Bench-Machine Learning Gap

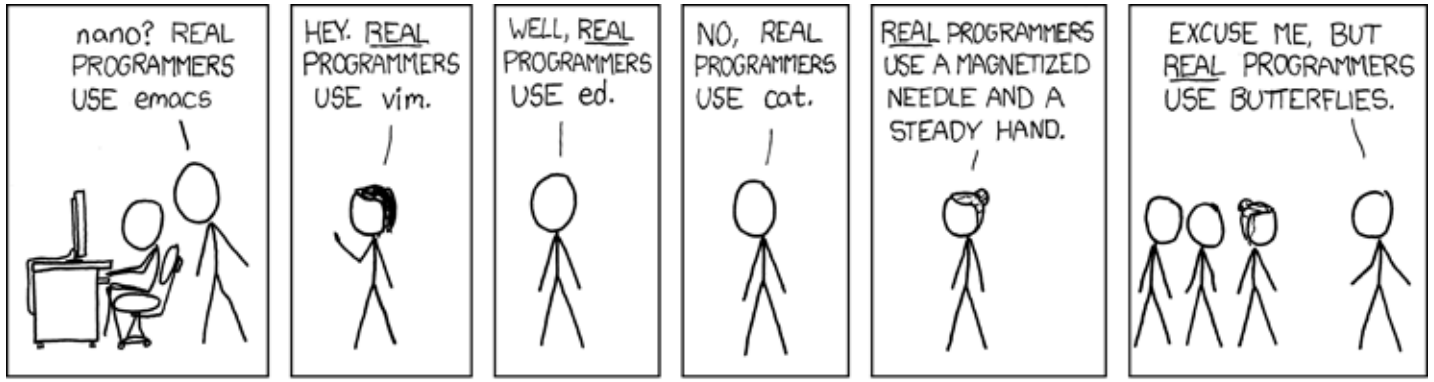
Dr. Emily A. Beck

Dr. Jake Searcy

Learning Objectives

Understand basic structures of DNA-seq file types (fasta, fastq, Sam, VCF)

Learn basic pipelines for getting from raw sequencing files to common desired outcomes (consensus sequences or variant calls)



Fasta files

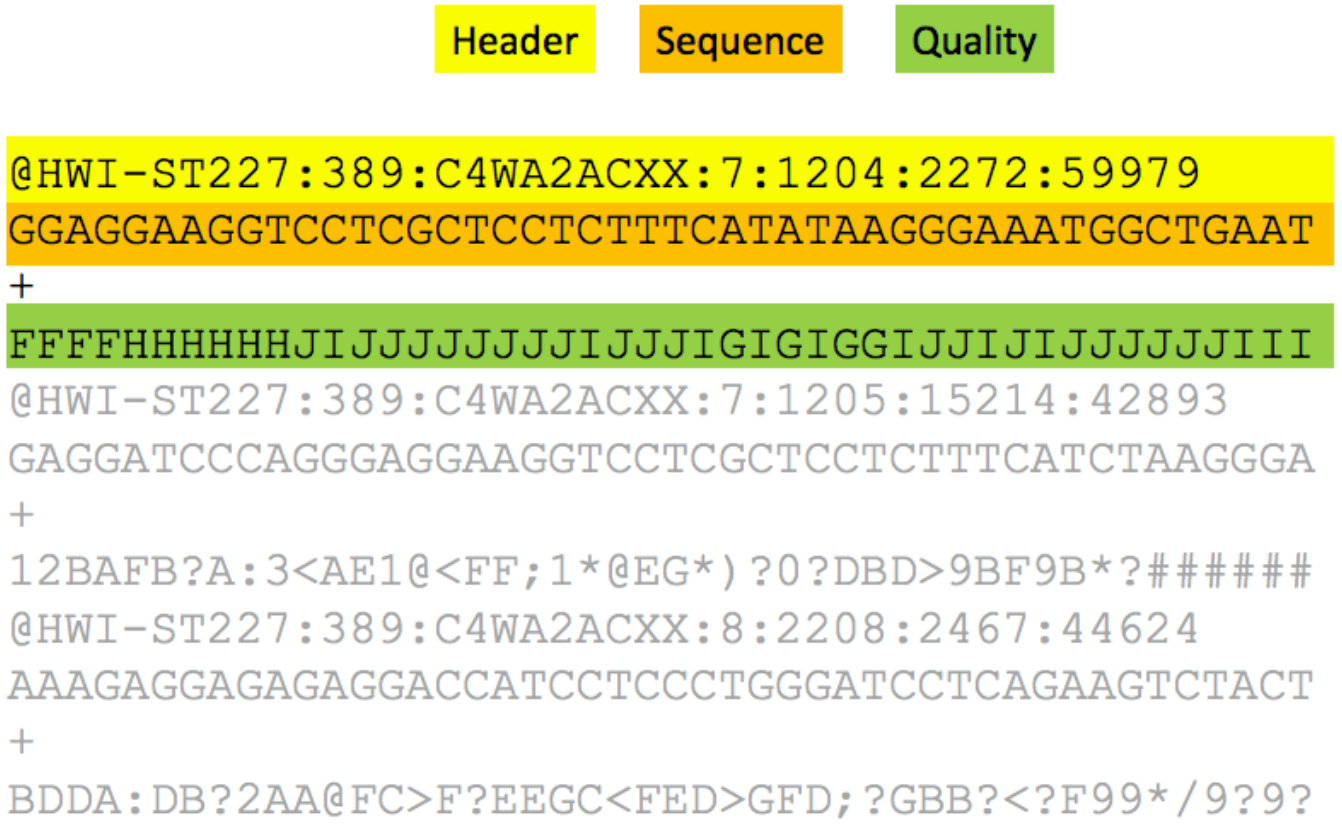
```
>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCTCTTTTCTTATCATTGACATTTAAACTCTGGGGCAGGTCCTCGCGTAGAACGCGGCTGTCAGATCT
GCCACTTCCCCTGCCGAGCGGCGGTGAGAAGTGTGGGAACCGGCGCTGCCAGGCTCACCTGCCTCCCCGC
CCTCCGCTCCCAGGTAACCGCCCGGGCTCCGGCCCCGGCCCGGCTCGGGGCCCGCGGGGCCTCTCCGCTG
CCAGCGACTGCTGTCCCCAAATCAAAGCCCGCCCCAAGTGGCCCCGGGGCTTGATTTTTGCTTTTAAAAG
GAGGCATACAAAGATGGAAGCGAGTTACTGAGGGAGGGATAGGAAGGGGGGTGGAGGAGGGACTTGTCTT
TGCCGAGTGTGCTCTTCTGCAAAAGTAGCAAAATGTTCCACTCCTAAGAGTGGACTTCCAGTCCGGCCCT
GAGCTGGGAGTAGGGGGCGGGAGTCTGCTGCTGCTGTCTGCTAAAGCCACTCGCGACCGCGAAAAATGCA
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCTCTGCATCGCAGTTCACGACATCCACGCTTGGGAAAG
TCCGTACCCGCGCCTGGAGCGCTTAAAGACACCCTGCCGCGGGTCGGGCGAGGTGCAGCAGAAGTTTCCC
GCGGTTGCAAAGTGCAGATGGCTGGACCGCAACAAAGTCTAGAGATGGGGTTCGTTTCTCAGAAAGACGC
```

>NAME_hopefully_informative_and_without_special_characters_you_are_better_than_that
[carriage return] sequence

This is often where we are trying to get!

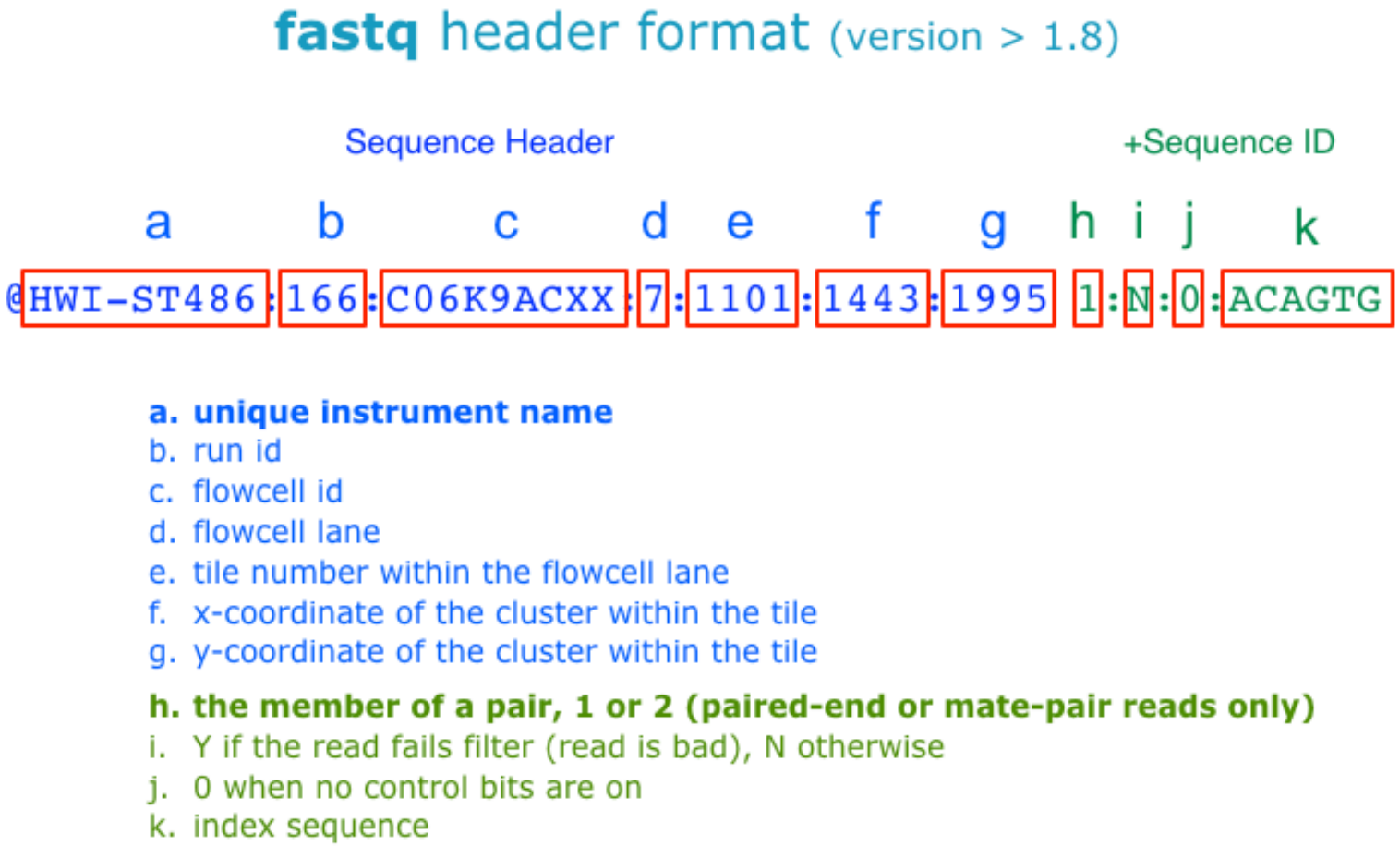
Fasta files can have multiple sequences each denoted with “>”

FastQ Files



(biocorecrg github)

FastQ Files



(biocorecrg github; GDC 2020)

FastQ Files

Quality Scores

```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
  ............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
  ..............................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
  LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
   |                         |    |        |                              |                     |
 33                        59   64       73                            104                   126

 S - Sanger        Phred+33,  raw reads typically (0, 40)
 X - Solexa        Solexa+64, raw reads typically (-5, 40)
 I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
 J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
 L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

ASCII values 33 - 73 = 0 - 40

'F' = 70 70 - 33 = **37**

(Catchen Evomics 2017)

My sequencing job just finished and I have a giant file…

Step 1: De-mulitplex if there are multiple samples in my sequencing run

Step 2: Quality filter my raw reads (not always needed but usually a good idea)

Step 3: Trim adapters (Not necessary depending on the alignment program)

De-multiplexing is done using the barcodes that were added to your sequencing library

Quality filtering is based on the scores in your fastQ file

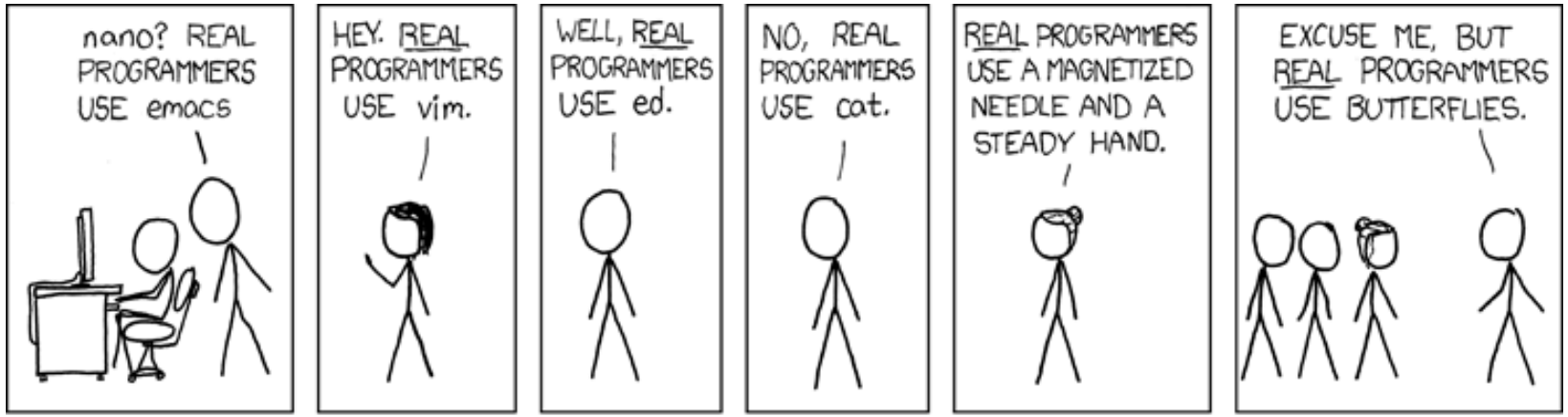
Quality Scores

```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
  ............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
  ..............................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
  LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
   |                         |    |        |                              |                     |
  33                        59   64       73                            104                   126

  S - Sanger        Phred+33,  raw reads typically (0, 40)
  X - Solexa        Solexa+64, raw reads typically (-5, 40)
  I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
  J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
     with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
     (Note: See discussion above).
  L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

ASCII values $33 - 73 = 0 - 40$

'F' = 70 $\quad$ $70 - 33 = \mathbf{37}$

I now have clean/de-multiplexed fastq files and want to align my reads:



Remember that there are a lot of packages out there and a lot of ways to do things.

BWA

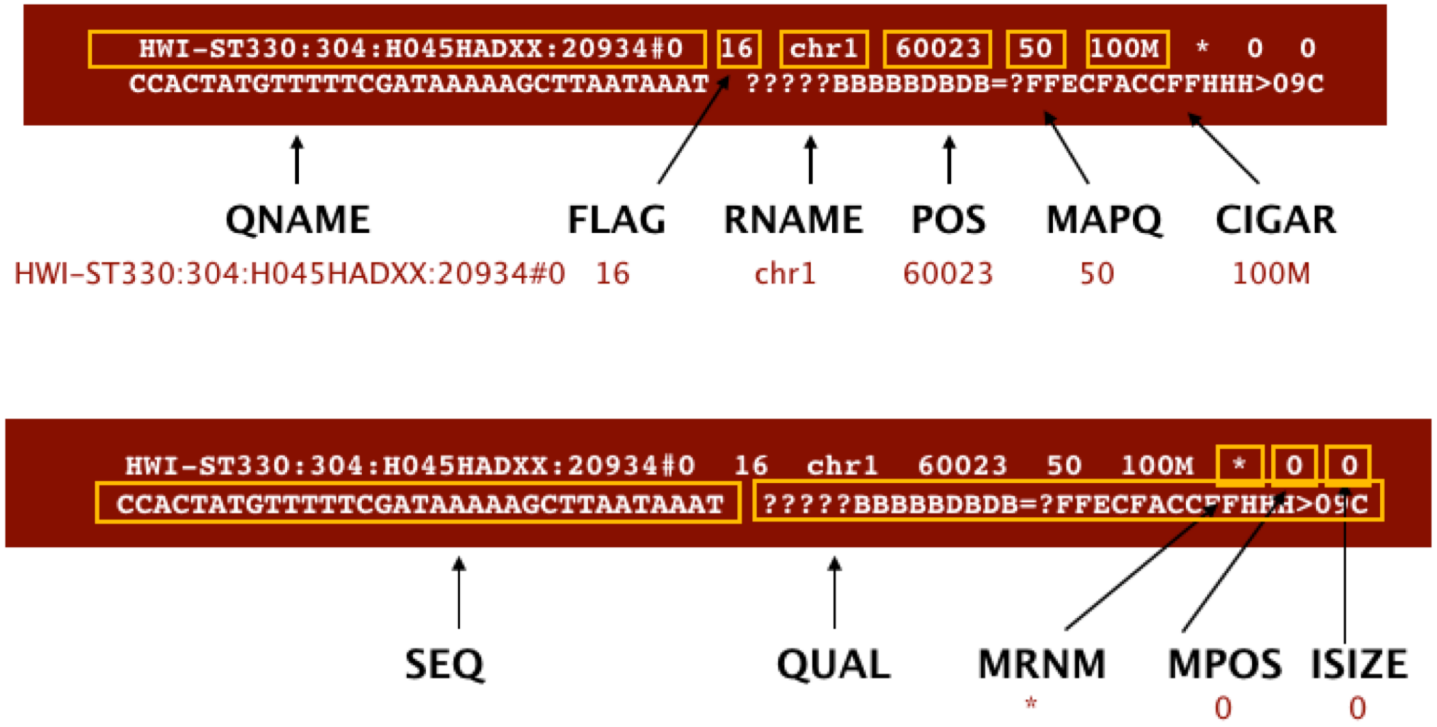
GSNAP

STAR

SAM file (Sequence Alignment Map)

.tsv file

```
@HD  The header line
VN: format version
SO: Sorting order of alignments

@SQ  Reference sequence dictionary
SN: reference sequence name
LN: reference sequence length
SP: species

@PG  Program
PN: program name
VN: program version
```



(Rkhetani Github)

SAM file (Sequence Alignment Map)

.tsv file

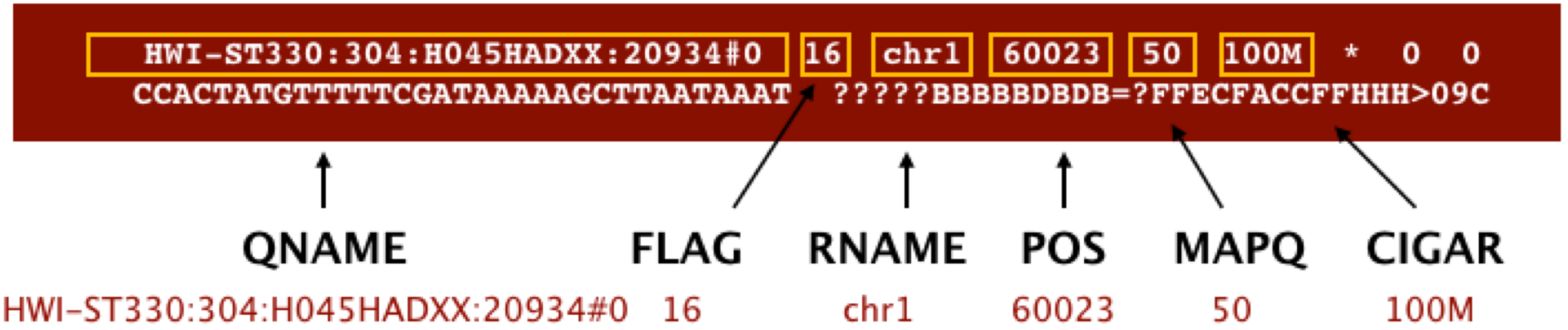
```
@HD  The header line
VN: format version
SO: Sorting order of alignments

@SQ  Reference sequence dictionary
SN: reference sequence name
LN: reference sequence length
SP: species

@PG  Program
PN: program name
VN: program version
```

(Rkhetani Github)

SAM file (Sequence Alignment Map)

.tsv file

Shown as two lines but is actually represented as a single line in the SAM file



QNAME: Read name

FLAG: (Next slide)

RNAME: Chromosome

POS: Nucleotide position

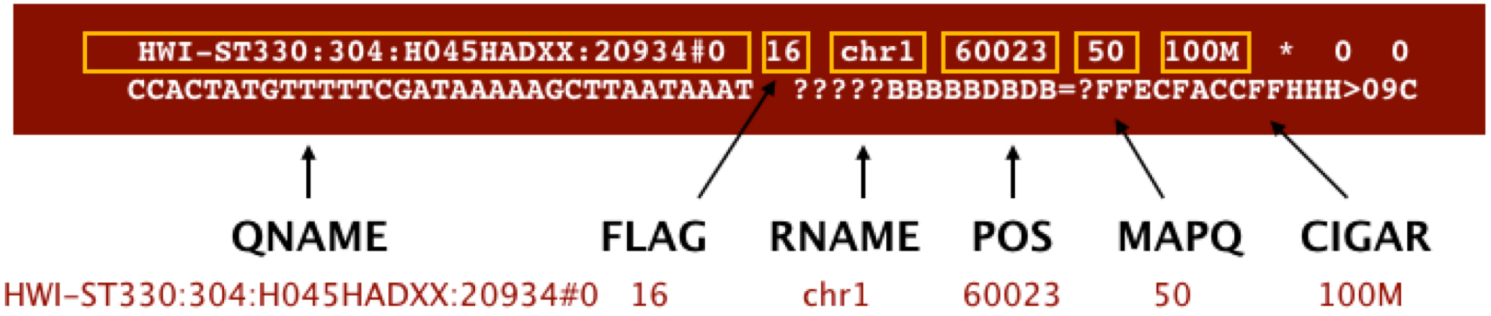
MAPQ: Quality Score

CIGAR: (Coming up)

(Rkhetani Github)

SAM file (Sequence Alignment Map) Flags

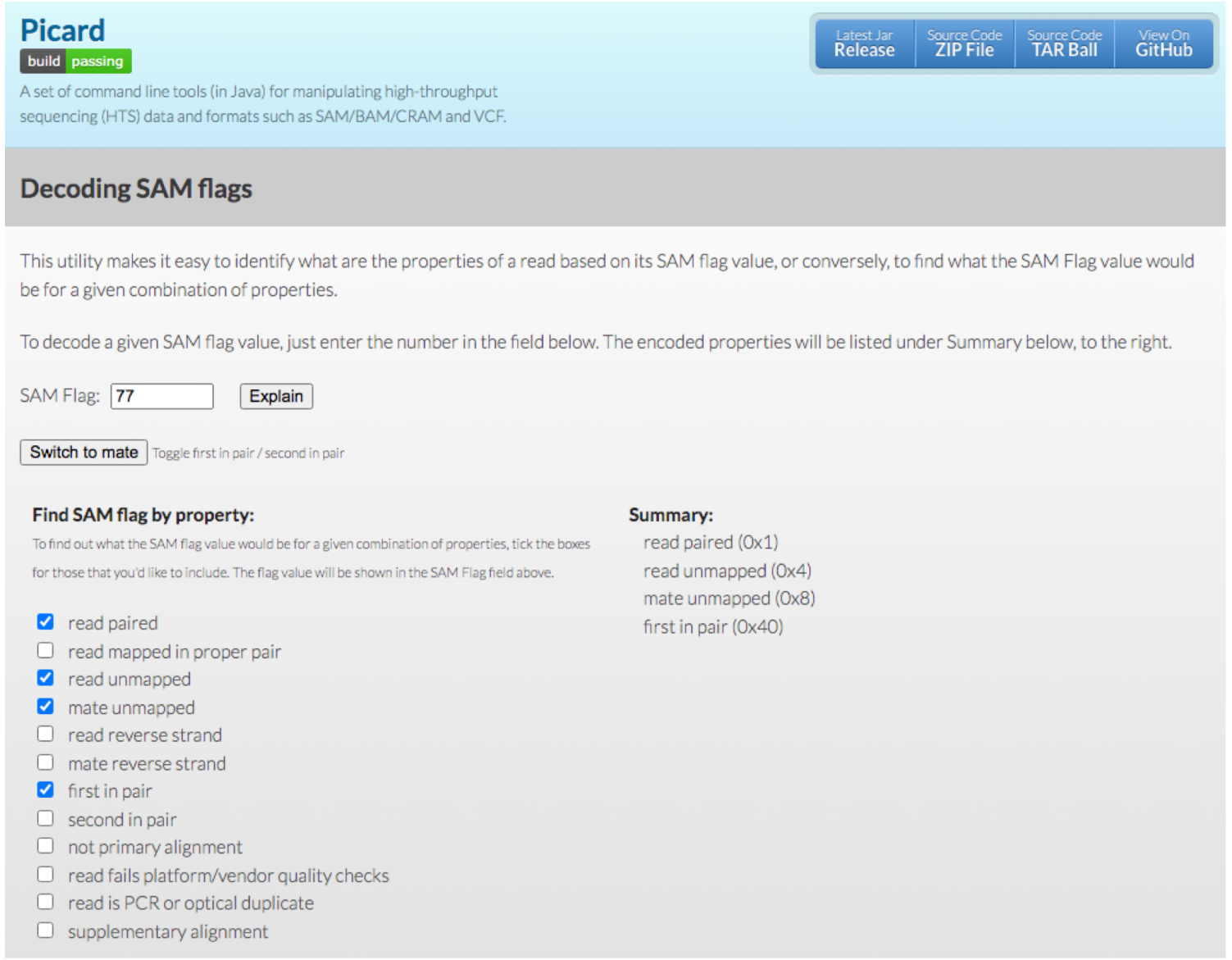
Shown as two lines but is actually represented as a single line in the SAM file



Binary (Decimal)	Hex	Description
00000000001 (1)	0x1	Is the read paired?
00000000010 (2)	0x2	Are both reads in a pair mapped "properly" (i.e., in the correct orientation with respect to one another)?
00000000100 (4)	0x4	Is the read itself unmapped?
00000001000 (8)	0x8	Is the mate read unmapped?
00000010000 (16)	0x10	Has the read been mapped to the reverse strand?
00000100000 (32)	0x20	Has the mate read been mapped to the reverse strand?
00001000000 (64)	0x40	Is the read the first read in a pair?
00010000000 (128)	0x80	Is the read the second read in a pair?
00100000000 (256)	0x100	Is the alignment not primary? (A read with split matches may have multiple primary alignment records.)
01000000000 (512)	0x200	Does the read fail platform/vendor quality checks?
10000000000 (1024)	0x400	Is the read a PCR or optical duplicate?

(Rkhetani Github; Olivier-Rue 2021)

Picard Tools is great for decoding Sam flags



Picard

build passing

A set of command line tools (in Java) for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF.

Latest Jar Release | Source Code ZIP File | Source Code TAR Ball | View On GitHub

Decoding SAM flags

This utility makes it easy to identify what are the properties of a read based on its SAM flag value, or conversely, to find what the SAM Flag value would be for a given combination of properties.

To decode a given SAM flag value, just enter the number in the field below. The encoded properties will be listed under Summary below, to the right.

SAM Flag: 77 Explain

Switch to mate Toggle first in pair / second in pair

Find SAM flag by property:

To find out what the SAM flag value would be for a given combination of properties, tick the boxes for those that you'd like to include. The flag value will be shown in the SAM Flag field above.

- [x] read paired
- [ ] read mapped in proper pair
- [x] read unmapped
- [x] mate unmapped
- [ ] read reverse strand
- [ ] mate reverse strand
- [x] first in pair
- [ ] second in pair
- [ ] not primary alignment
- [ ] read fails platform/vendor quality checks
- [ ] read is PCR or optical duplicate
- [ ] supplementary alignment

Summary:

read paired (0x1)
read unmapped (0x4)
mate unmapped (0x8)
first in pair (0x40)

SAM file (Sequence Alignment Map) CIGAR

Shown as two lines but is actually represented as a single line in the SAM file

```
HWI-ST330:304:H045HADXX:20934#0  16  chr1  60023  50  100M  *  0  0
CCACTATGTTTTCGATAAAAAGCTTAATAAAT  ?????BBBBBDBDB=?FFECFACCFFHHH>09C
```

QNAME	FLAG	RNAME	POS	MAPQ	CIGAR
HWI-ST330:304:H045HADXX:20934#0	16	chr1	60023	50	100M

CIGAR (Concise Idiosyncratic Gapped Alignment Report)

Reference sequence with aligned reads	CIGAR string	Explanation
C T G C A T G T T A G A T A A * * G A T A G C T G T G C T A		
A A G G A T A * C T G	1M2I4M1D3M	Insertion & Deletion
G A T A A * G G A T A	5M1P1I4M	Padding & Insertion
T G T T A ▬▬▬▬▬▬▬▬ T G C T A	5M15N5M	Spliced read
a a a C A T G T T A G	3S8M	Soft clipping
A A A C A T G T T A G	3H8M	Hard clipping

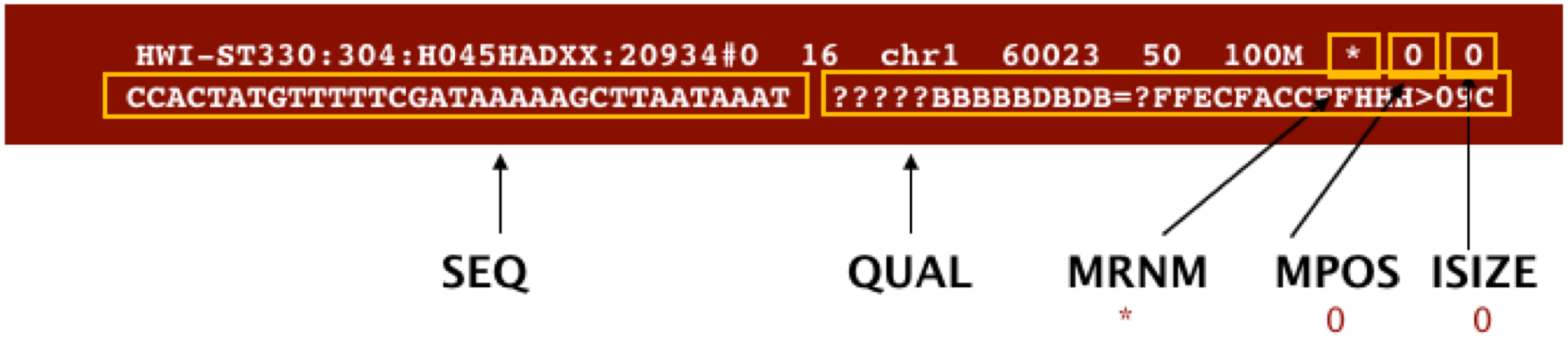
Note: lowercase letters are used to indicate soft clipping

(Rkhetani Github; Olivier-Rue 2021)

SAM file (Sequence Alignment Map)

.tsv file

Shown as two lines but is actually represented as a single line in the SAM file



SEQ: Sequence

QUAL: Quality Score

MRNM: Mate Reference Name

MPOS: Position of Mate

ISIZE: Inferred insert size

} Paired-end data

(Rkhetani Github)

Samtools and Picard Tools

Excellent packages for merging, indexing, filtering, running state, visualizing your alignments

Samtools can easily convert sam to bam formats and quality filter your alignments based on alignment scores and call consensus on your alignment to creast a fasta consensus sequence!

VCF (Variant Calling Format) file

##fileformat=VCFv4.2
##fileDate=20180727
##source="Stacks v1.46"
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=AD,Number=1,Type=Integer,Description="Allele Depth">
##FORMAT=<ID=GL,Number=.,Type=Float,Description="Genotype Likelihood">
##INFO=<ID=locori,Number=1,Type=Character,Description="Orientation the corresponding Stacks locus aligns in">

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	3063_001	3063_002	3063_003	3063_004	3063_005	3063_006	3063_007	3063_008	3063_009	3063_01	3063_011	3063_012
1	11734	8621_83	A	G	.	PASS	NS=178;AF=(	GT:DP:AD	0/1:34:23,11	0/0:41:41,0	0/1:56:36,20	0/1:32:22,10	0/0:66:66,0	0/1:54:12,42	0/1:43:24,19	0/0:48:48,0	0/0:35:35,0	0/0:75:75,0	0/1:22:7,15	0/1:56:22,34
1	11833	8620_19	C	G	.	PASS	NS=137;AF=(	GT:DP:AD	1/1:34:0,34	./.:0:.,.	./.:0:.,.	0/0:22:22,0	1/1:12:0,12	0/0:16:16,0	1/1:41:0,41	1/1:20:0,20	./.:0:.,.	0/0:24:24,0	1/1:22:0,22	1/1:18:0,18
1	11864	8620_50	T	C	.	PASS	NS=136;AF=(	GT:DP:AD	0/0:34:34,0	./.:0:.,.	./.:0:.,.	1/1:22:0,22	0/0:12:12,0	1/1:16:0,16	0/0:41:41,0	0/0:20:20,0	./.:0:.,.	1/1:24:0,24	0/0:22:22,0	0/0:18:18,0
1	11866	8620_52	T	C	.	PASS	NS=136;AF=(	GT:DP:AD	0/0:34:34,0	./.:0:.,.	./.:0:.,.	1/1:22:0,22	0/0:12:12,0	1/1:16:0,16	0/0:41:41,0	0/0:20:20,0	./.:0:.,.	1/1:24:0,24	0/0:22:22,0	0/0:18:18,0
1	11867	8620_53	T	A	.	PASS	NS=137;AF=(	GT:DP:AD	0/0:34:34,0	./.:0:.,.	./.:0:.,.	1/1:22:0,22	0/0:12:12,0	1/1:16:0,16	0/0:41:41,0	0/0:20:20,0	./.:0:.,.	1/1:24:0,24	0/0:22:22,0	0/0:18:18,0
1	11919	8620_105	C	T	.	PASS	NS=138;AF=(	GT:DP:AD	1/1:34:0,34	./.:0:.,.	./.:0:.,.	0/0:22:22,0	1/1:12:0,12	0/0:16:16,0	1/1:41:0,41	1/1:20:0,20	./.:0:.,.	0/0:24:24,0	1/1:22:0,22	1/1:18:0,18
1	11931	8620_117	A	G	.	PASS	NS=132;AF=(	GT:DP:AD	0/0:34:34,0	./.:0:.,.	./.:0:.,.	0/0:22:22,0	0/0:12:12,0	0/0:16:16,0	0/0:41:41,0	0/0:20:20,0	./.:0:.,.	0/0:24:24,0	0/0:22:22,0	0/0:18:18,0
1	49662	10510_50	T	C	.	PASS	NS=180;AF=(	GT:DP:AD	0/0:55:55,0	0/1:51:22,29	0/1:37:21,16	0/1:27:6,21	0/0:41:41,0	0/1:34:18,16	0/0:47:47,0	0/0:63:63,0	0/1:19:11,8	0/1:60:33,27	0/0:27:27,0	0/0:70:70,0
1	49679	10510_33	G	A	.	PASS	NS=180;AF=(	GT:DP:AD	0/0:55:55,0	0/1:51:29,22	0/0:37:37,0	0/0:27:27,0	0/1:41:15,26	0/0:34:34,0	0/0:47:47,0	0/1:63:32,31	0/1:19:12,7	0/1:60:27,33	0/0:27:27,0	0/0:70:70,0
1	49790	10509_81	T	C	.	PASS	NS=185;AF=(	GT:DP:AD	0/0:34:34,0	0/1:68:42,26	0/0:92:92,0	0/0:19:19,0	0/1:55:16,39	0/0:83:83,0	0/0:74:74,0	0/1:45:15,30	0/1:24:13,11	0/1:58:15,43	0/0:76:76,0	0/0:72:72,0
1	89330	10938_81	T	C	.	PASS	NS=182;AF=(	GT:DP:AD	1/1:22:0,22	0/0:24:24,0	0/1:74:29,45	0/1:59:19,40	0/1:53:16,37	0/1:49:19,30	1/1:45:0,45	0/1:56:33,23	0/0:14:14,0	0/0:50:50,0	1/1:25:0,25	1/1:36:0,36
1	89378	10938_129	C	G	.	PASS	NS=182;AF=(	GT:DP:AD	1/1:22:0,22	0/0:24:24,0	0/1:74:29,45	0/1:59:19,40	0/1:53:16,37	0/1:49:19,30	1/1:45:0,45	0/1:56:33,23	0/0:14:14,0	0/0:50:50,0	1/1:25:0,25	1/1:36:0,36
1	92076	10974_10	G	A	.	PASS	NS=189;AF=(	GT:DP:AD	0/0:35:35,0	0/1:80:32,48	0/0:40:40,0	0/0:82:82,0	0/1:75:45,30	0/0:101:101,	0/0:81:81,0	0/1:55:36,19	0/1:15:6,9	0/1:71:29,42	0/0:58:58,0	0/0:53:53,0
1	92141	10974_75	G	A	.	PASS	NS=187;AF=(	GT:DP:AD	1/1:35:0,35	0/0:80:80,0	0/1:40:12,28	0/1:82:49,33	0/1:75:30,45	0/1:101:48,5	1/1:81:0,81	0/1:55:19,36	0/0:15:15,0	0/0:71:71,0	1/1:58:0,58	1/1:53:0,53
1	92177	10974_111	T	C	.	PASS	NS=187;AF=(	GT:DP:AD	1/1:35:0,35	0/0:80:80,0	0/1:40:12,28	0/1:82:49,33	0/1:75:30,45	0/1:101:48,5	1/1:81:0,81	0/1:55:19,36	0/0:15:15,0	0/0:71:71,0	1/1:58:0,58	1/1:53:0,53
1	92208	10974_142	G	T	.	PASS	NS=187;AF=(	GT:DP:AD	1/1:35:0,35	0/0:80:80,0	0/1:40:12,28	0/1:82:49,33	0/1:75:30,45	0/1:101:48,5	1/1:81:0,81	0/1:55:19,36	0/0:15:15,0	0/0:71:71,0	1/1:58:0,58	1/1:53:0,53
1	103372	8501_6	T	A	.	PASS	NS=151;AF=(	GT:DP:AD	./.:0:.,.	1/1:40:0,40	0/1:106:70,3	1/1:21:0,21	0/0:29:29,0	0/1:49:22,27	0/0:54:54,0	0/0:71:71,0	0/1:11:4,7	1/1:37:0,37	0/0:24:24,0	0/0:12:12,0
1	103391	8501_25	C	A	.	PASS	NS=151;AF=(	GT:DP:AD	./.:0:.,.	0/0:40:40,0	0/1:106:81,2	0/0:21:21,0	0/0:29:29,0	0/0:49:49,0	0/1:54:43,11	0/1:71:34,37	0/0:11:11,0	0/0:37:37,0	0/0:24:24,0	0/0:12:12,0
1	103392	8501_26	A	G	.	PASS	NS=150;AF=(	GT:DP:AD	./.:0:.,.	0/0:40:40,0	0/1:106:81,2	0/0:21:21,0	0/0:29:29,0	0/0:49:49,0	0/1:54:43,11	0/1:71:34,37	0/0:11:11,0	0/0:37:37,0	0/0:24:24,0	0/0:12:12,0
1	103425	8501_59	G	A	.	PASS	NS=147;AF=(	GT:DP:AD	./.:0:.,.	1/1:40:0,40	0/1:106:70,3	./.:21:.,.	0/0:29:29,0	0/1:49:22,27	0/0:54:54,0	0/0:71:71,0	0/1:11:4,7	1/1:37:0,37	0/0:24:24,0	0/0:12:12,0
1	114651	8597_50	A	G	.	PASS	NS=180;AF=(	GT:DP:AD	0/1:56:22,34	0/1:49:25,24	0/1:79:44,35	0/1:59:21,38	0/1:62:28,34	0/0:59:59,0	0/1:67:30,37	0/1:69:49,20	./.:39:.,.	0/1:61:28,33	0/1:49:36,13	0/1:61:41,20
1	129639	8750_100	C	G	.	PASS	NS=182;AF=(	GT:DP:AD	0/0:39:39,0	0/1:48:29,19	0/0:76:76,0	0/0:37:37,0	0/1:106:81,2	0/0:96:96,0	0/0:72:72,0	0/1:46:27,19	0/1:13:6,7	0/1:80:47,33	0/0:42:42,0	0/0:27:27,0
1	136323	8801_51	A	G	.	PASS	NS=161;AF=(	GT:DP:AD	0/1:14:4,10	0/0:27:27,0	0/0:37:37,0	./.:0:.,.	0/1:33:18,15	0/0:34:34,0	0/1:30:22,8	0/1:27:11,16	0/0:14:14,0	0/0:42:42,0	0/0:26:26,0	0/1:34:10,24
1	136362	8801_12	A	T	.	PASS	NS=162;AF=(	GT:DP:AD	0/1:14:10,4	0/1:27:13,14	1/1:37:0,37	./.:0:.,.	0/0:33:33,0	1/1:34:0,34	0/1:30:8,22	0/0:27:27,0	0/1:14:4,10	0/1:42:20,22	1/1:26:0,26	0/1:34:24,10
1	136395	8800_24	G	A	.	PASS	NS=184;AF=(	GT:DP:AD	0/1:70:51,19	0/1:39:12,27	0/0:59:59,0	0/0:27:27,0	0/1:93:33,60	0/0:35:35,0	0/1:54:35,19	0/1:85:30,55	0/1:12:8,4	0/1:62:29,33	0/1:59:35,24	0/0:83:83,0
1	161184	9026_50	T	C	.	PASS	NS=182;AF=(	GT:DP:AD	1/1:70:0,70	0/0:80:80,0	0/1:69:45,24	0/1:38:15,23	0/1:71:43,28	0/1:42:14,28	1/1:99:0,99	0/1:79:52,27	0/0:20:20,0	0/0:40:40,0	1/1:52:0,52	1/1:70:0,70
1	161356	9025_125	C	T	.	PASS	NS=187;AF=(	GT:DP:AD	0/0:55:55,0	0/1:37:27,10	0/1:90:44,46	0/1:31:20,11	0/0:55:55,0	0/1:80:41,39	0/0:59:59,0	0/0:82:82,0	0/1:14:5,9	0/1:62:38,24	0/0:46:46,0	0/0:43:43,0
1	186810	9283_103	T	C	.	PASS	NS=151;AF=(	GT:DP:AD	1/1:20:0,20	0/0:15:15,0	1/1:20:0,20	0/1:30:15,15	0/0:35:35,0	0/1:34:12,22	0/1:21:11,10	0/0:32:32,0	0/0:10:10,0	0/0:35:35,0	0/1:30:11,19	0/1:28:14,14

Questions about these file types or your project? Come to office hours and talk with me!