# BTEP course


Alexandra L Emmons Ph.D. & Joe Wu Ph.D.
BTEP/GAU/CCR/NCI/NIH - email ncibtep@mail.nih.gov
Bioinformatics Training and Education Program

# Table of Contents

## NGS FILE FORMATS

# NGS FILE FORMATS

## SEQUENCE FILE FORMATS

{{Sdet}} {{Ssum}}FASTA Format *(https://en.wikipedia.org/wiki/FASTA_format)*

File can contain one or more sequences. The format specifies a single header line which starts with a ">" character, follow by one or more line of sequences data. {{Esum}}

FASTA example

```
>HWI-ST398_0092:1:1:5372:2486#0/1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGA
ACGTAGCAGCAGCATCAGTACGACTACGACGACTAGCACATGCGACGATCGATGCTAGCTGACTATCGA
```

Multiple sequence FASTA example:

```
>Sequence Name 1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGA
ACGTAGCAGCAGCATCAGTACGACTACGACGACTAGCACATGCGACGATCGATGCTAGCTGACTATCGA
>Sequence Name 2
ACGTAGACACGACTAGCATCAGCTACGCATCGATCAGCATCGACTAGCATCACACATCGATCAGCATCA
AGCATCGACTACACTACGACTACGATCCACGTACGACTAGCATGCTAGCGCTAGCTAGCTAGCTAGTCC
AGCTAGCTAGCTAGC
>Sequence Name 3
ACTCAGCATGCATCAGCATCGACTACGACTACGACATCGACTAGCATCAGCAT
```

{{Edet}}

---

{{Sdet}} {{Ssum}} FASTQ Format *(https://en.wikipedia.org/wiki/FASTQ_format)*

Text based format for storing sequence data and corresponding quality scores for each base. To enable a one-one correspondence between the base sequence and the quality score the score is stored as a single one letter/number code using an offset of the standard ASCII code. Quality scores range from 0 to 40 and represent a log10 score for the probability of being wrong. E.g. score of 30 => 1:1000 chance of error.

For paired end reads fastq files come in pairs, typically labelled R1 and R2 (reads are in same order in both files…header often does not distinguish between read1 and read2 {{Esum}}

Each fastq file contain multiple entries and each entry consists of 4 lines:

1. header line beginning with "@" and sequence name
2. sequence line
3. header line beginning with "+" which can have the name but rarely does
4. quality score line

```
@HWI-ST398_0092:6:73:5372:2486#0/1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGA
+HWI-ST398_0092:1:1:5372:2486#0/1
FFFFEEDFCEDFFFFEFFDEFFF_FFFFFDCCFDZDEEADEFECZEDAECDBRDTY^ZYT``_T`_^B(
```

- **6** - Flowcell lane
- **73** - Tile number
- **5372:2486** - 'x','y'-coordinates of the cluster within the tile
- **#0** - index number for a multiplexed sample (0 for no indexing)
- **/1** - the member of a pair, /1 or /2 (paired-end or mate-pair reads only)

Quality Scores

```
Quality (Q) = -10log10P
```

Quality Score => Probabiliy that the base has been called incorrectly

- 10 => 1 in 10
- 20 => 1 in 100
- 30 => 1 in 1,000
- 40 => 1 in 10,000

**ASCII code table:** Showing the ASCII code, the Quality Score (ASCII code -33), and the Symbol used in the fastq files

| ASCII | Symbol | ASCII | Q score | Symbol | ASCII | Q score | Symbol | ASCII | Q score | Symbol |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | [NULL] | 32 | | [SPACE] | 64 | 31 | @ | 96 | 63 | ` |
| 1 | [START OF HEADING] | 33 | 0 | ! | 65 | 32 | A | 97 | 64 | a |
| 2 | [START OF TEXT] | 34 | 1 | " | 66 | 33 | B | 98 | 65 | b |
| 3 | [END OF TEXT] | 35 | 2 | # | 67 | 34 | C | 99 | 66 | c |
| 4 | [END OF TRANSMISSION] | 36 | 3 | $ | 68 | 35 | D | 100 | 67 | d |
| 5 | [ENQUIRY] | 37 | 4 | % | 69 | 36 | E | 101 | 68 | e |
| 6 | [ACKNOWLEDGE] | 38 | 5 | & | 70 | 37 | F | 102 | 69 | f |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | [BELL] | 39 | 6 | ‘ | 71 | 38 | G | 103 | 70 | g |
| 8 | [BACKSPACE] | 40 | 7 | ( | 72 | 39 | H | 104 | 71 | h |
| 9 | [HORIZONTAL TAB] | 41 | 8 | ) | 73 | 40 | I | 105 | 72 | I |
| 10 | [LINE FEED] | 42 | 9 | * | 74 | 41 | J | 106 | 73 | j |
| 11 | [VERTICAL TAB] | 43 | 10 | + | 75 | 42 | K | 107 | 74 | k |
| 12 | [FORM FEED] | 44 | 11 | , | 76 | 43 | L | 108 | 75 | l |
| 13 | [CARRIAGE RETURN] | 45 | 12 | - | 77 | 44 | M | 109 | 76 | m |
| 14 | [SHIFT OUT] | 46 | 13 | . | 78 | 45 | N | 110 | 77 | n |
| 15 | [SHIFT IN] | 47 | 14 | / | 79 | 46 | O | 111 | 78 | o |
| 16 | [DATA LINK ESCAPE] | 48 | 15 | 0 | 80 | 47 | P | 112 | 79 | p |
| 17 | [DEVICE CONTROL 1] | 49 | 16 | 1 | 81 | 48 | Q | 113 | 80 | q |
| 18 | [DEVICE CONTROL 2] | 50 | 17 | 2 | 82 | 49 | R | 114 | 81 | r |
| 19 | [DEVICE CONTROL 3] | 51 | 18 | 3 | 83 | 50 | S | 115 | 82 | s |
| 20 | [DEVICE CONTROL 4] | 52 | 19 | 4 | 84 | 51 | T | 116 | 83 | t |
| 21 | [NEGATIVE ACKNOWLEDGE] | 53 | 20 | 5 | 85 | 52 | U | 117 | 84 | u |
| 22 | [SYNCHRONOUS IDLE] | 54 | 21 | 6 | 86 | 53 | V | 118 | 85 | v |
| 23 | [ENG OF TRANS. BLOCK] | 55 | 22 | 7 | 87 | 54 | W | 119 | 86 | w |
| 24 | [CANCEL] | 56 | 23 | 8 | 88 | 55 | X | 120 | 87 | x |
| 25 | [END OF MEDIUM] | 57 | 24 | 9 | 89 | 56 | Y | 121 | 88 | Y |
| 26 | [SUBSTITUTE] | 58 | 25 | : | 90 | 57 | Z | 122 | 89 | z |
| 27 | [ESCAPE] | 59 | 26 | ; | 91 | 58 | [ | 123 | 90 | { |
| 28 | [FILE SEPARATOR] | 60 | 27 | < | 92 | 59 | \ | 124 | 91 | \| |
| 29 | [GROUP SEPARATOR] | 61 | 28 | | 93 | 60 | ] | 125 | 92 | } |
| 30 | | 62 | 29 | > | 94 | 61 | ^ | 126 | 93 | ~ |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | [RECORD SEPARATOR] | | | | | | | | | |
| 31 | [UNIT SEPARATOR] | 63 | 30 | ? | 95 | 62 | _ | 127 | | [DEL] |

{{Edet}}

# ALIGNMENT FILE FORMATS

{{Sdet}} {{Ssum}} SAM Format *(https://en.wikipedia.org/wiki/SAM_(file_format))*

The SAM Format (Sequence Alignment/Map) is a text format for storing sequence alignment data in a series of tab delimited ASCII columns. The first base in a reference sequence has coordinate 1. {{Esum}}

The file has two parts:

1. Header - Each line starts with a "@". @HD, @SQ, @RG, @PG
2. Alignments - One line for each entry.

Header Example

```
@HD VN:1.0 SO:unsorted
@SQ SN:chr1 LN:195471971
@SQ SN:chr2 LN:182113224
@SQ SN:chr3 LN:160039680
@SQ SN:chr4 LN:156508116
@SQ SN:chr5 LN:151834684
@SQ SN:chr6 LN:149736546
@SQ SN:chr7 LN:145441459
@SQ SN:chr8 LN:129401213
@SQ SN:chr9 LN:124595110
@SQ SN:chr10 LN:130694993
@SQ SN:chr11 LN:122082543
@SQ SN:chr12 LN:120129022
@SQ SN:chr13 LN:120421639
@SQ SN:chr14 LN:124902244
@SQ SN:chr15 LN:104043685
@SQ SN:chr16 LN:98207768
@SQ SN:chr17 LN:94987271
@SQ SN:chr18 LN:90702639
@SQ SN:chr19 LN:61431566
@SQ SN:chrX LN:171031299
@SQ SN:chrY LN:91744698
@SQ SN:chrM LN:16299
```

```
@PG ID:bowtie2 PN:bowtie2 VN:2.2.9 CL:"/usr/local/apps/bowtie/2-2.2.9
2.DELETE/mm10 -q jun_minus_dex_rep1a -S jun_minus_dex_rep1a_mm10.sam
```

Read Alignment Example

```
8_100_10000_12419 163 chr7 271183 255 40M = 271294 151 TGGTGTA
TGGTGTATTATACGCTACCGTGCGGTGCCGGGGGCAACCG
BBBABBBBBBBBBBBBBBBCBBBBCBBBBBBBBBBBBBB XA:i:0 MD:Z:40 NM:i:0
```

Read Alignment with headers Example

| QNAME | FLAG | RNAME | POS | MAPQ | CIGAR | MRNM | MPOS | TLEN | SEQ | QUAL | OPT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8_100_10000_12419 | 163 | chrVII | 271183 | 255 | 40M | = | 271294 | 151 | TTA... | BBB... | XA:i:0 MD:Z:40 NM:i:0 |

| Column | NAME | Field Description |
|---|---|---|
| 1 | QNAME | Query template/pair NAME |
| 2 | FLAG | bitwise FLAG |
| 3 | RNAME | Reference sequence NAME |
| 4 | POS | 1-based leftmost POSition/coordinate of clipped sequence |
| 5 | MAPQ | MAPping Quality (Phred-scaled) |
| 6 | CIGAR | extended CIGAR string |
| 7 | MRNM | Mate Reference sequence NaMe (‘=’ if same as RNAME) |
| 8 | MPOS | 1-based Mate POSistion |
| 9 | TLEN | inferred Template LENgth (insert size) |
| 10 | SEQ | query SEQuence on the same strand as the reference |
| 11 | QUAL | query QUALity (ASCII-33 gives the Phred base quality) |
| 12+ | OPT | variable OPTional fields in the format TAG:VTYPE:VALUE |

Understanding Flag codes

http://broadinstitute.github.io/picard/explain-flags.html

| Code | Description |
|---|---|
| 1 | read paired |
| 2 | read mapped in proper pair |

| Code | Description |
|---|---|
| 4 | read unmapped |
| 8 | mate unmapped |
| 16 | read reverse strand |
| 32 | mate reverse strand |
| 64 | first in pair |
| 128 | second in pair |
| 256 | not primary alignment |
| 512 | read fails platform/vendor quality checks |
| 1024 | read is PCR or optical duplicate |
| 2048 | supplementary alignment |

{{Edet}}

---

{{Sdet}} {{Ssum}} BAM/CRAM FORMAT

BAM *(https://en.wikipedia.org/wiki/Binary_Alignment_Map)* (*.bam) *is the compressed binary version of the Sequence Alignment/Map (SAM) format, a compact and index-able representation of nucleotide sequence alignments. BAM is compressed in the BGZF format that supports random access through the BAM file index (*.bam.bai). {{Esum}}

HINT: Filename.bam and filename.bai always go together

The major advantage of BAM/CRAM vs SAM format is that the former are compress (use much less disk space), and, by virtue of their indexing, it is possible to load the file in pieces and rapidly jump to any location in the file (i.e the specific coordinates on a specific chromosome).

CRAM *(https://en.wikipedia.org/wiki/CRAM_(file_format))* (*.cram) - newer implementation of BAM-like binary data.

1. Significantly better lossless compression than BAM
2. Full compatibility with BAM
3. Effortless transition to CRAM from using BAM files
4. Support for controlled loss of BAM data

{{Edet}}

---

# ANNOTATION FILE FORMATS

{{Sdet}} {{Ssum}} BED Format *(https://genome.ucsc.edu/FAQ/FAQformat.html#format1)*

These files contain limited annontation based on gene coordinates. It is a line based format (no header). Is has a minimal data requirement (bed 6) but and be expanded with additoinal information (bed 12). This format is zero-based for the coordinate start and one-based for the coordinate endBased. Thus the first base in a sequence would have start value=0 and an end value of 1.

{{Esum}}

1. chrom - name of the chromosome
2. chromStart - Start of feature (0-based)
3. chromEnd - End of the feature (not included in display) + 9 optional columns - most common are:
4. name - a label for the feature
5. score - a score (0-1000)
6. strand - which strand the feature on (+/-)

---

1. thickStart - The starting position at which the feature is drawn thickly (for example, the start codon in gene displays). When there is no thick part, thickStart and thickEnd are usually set to the chromStart position.
2. thickEnd - The ending position at which the feature is drawn thickly (for example the stop codon in gene displays).
3. itemRgb - An RGB value of the form R,G,B (e.g. 255,0,0). If the track line itemRgb attribute is set to "On", this RBG value will determine the display color of the data contained in this BED line. NOTE: It is recommended that a simple color scheme (eight colors or less) be used with this attribute to avoid overwhelming the color resources of the Genome Browser and your Internet browser.
4. blockCount - The number of blocks (exons) in the BED line.
5. blockSizes - A comma-separated list of the block sizes. The number of items in this list should correspond to blockCount.
6. blockStarts - A comma-separated list of block starts. All of the blockStart positions should be calculated relative

Example - Bed 6

| Chromosome | Start | End | name | score | strand |
|---|---|---|---|---|---|
| chr1 | 15000 | 20000 | gene1 | 50 | + |
| chr2 | 106000 | 108000 | gene2 | 400 | - |

{{Edet}}

---

{{Sdet}} {{Ssum}} GFF FORMAT *(https://genome.ucsc.edu/FAQ/FAQformat.html#format3)*

GFF (General Feature Format) GFF lines have nine required fields that must be tab-separated [GFF2 - UCSC & GFF3 - EMBL] {{Esum}}

1. squid - The name of the chromosome or scaffold.
2. source - The program that generated this feature.
3. feature - The name of this type of feature. Some examples of standard feature types are "CDS" "start_codon" "stop_codon" and "exon"
4. start - The starting position of the feature in the sequence. The first base is numbered 1.
5. end - The ending position of the feature (inclusive).
6. score - floating point value
7. strand - Valid entries include "+", "-", or "." (for don't know/don't care).
8. phase - If the feature is a coding exon, frame should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be ".".
9. attributes- A list of feature attributes in the format tag=value pairs separated by ";"

GFF2 http://genome.ucsc.edu/FAQ/FAQformat.html#format3

GFF3 https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md

http://useast.ensembl.org/info/website/upload/gff3.html *(http://useast.ensembl.org/info/website/upload/gff3.html)*

GFF Example

```
0 ##gff-version 3.2.1
1 ##sequence-region ctg123 1 1497228
2 ctg123 . gene 1000 9000 . + . ID=gene00001;Name=EDEN
3 ctg123 . TF_binding_site 1000 1012 . + . ID=tfbs00001;Parent=gene0(
4 ctg123 . mRNA 1050 9000 . + . ID=mRNA00001;Parent=gene00001;Name=EI
5 ctg123 . mRNA 1050 9000 . + . ID=mRNA00002;Parent=gene00001;Name=EI
6 ctg123 . mRNA 1300 9000 . + . ID=mRNA00003;Parent=gene00001;Name=EI
7 ctg123 . exon 1300 1500 . + . ID=exon00001;Parent=mRNA00003
8 ctg123 . exon 1050 1500 . + . ID=exon00002;Parent=mRNA00001,mRNA00(
9 ctg123 . exon 3000 3902 . + . ID=exon00003;Parent=mRNA00001,mRNA00(
10 ctg123 . exon 5000 5500 . + . ID=exon00004;Parent=mRNA00001,mRNA0(
11 ctg123 . exon 7000 9000 . + . ID=exon00005;Parent=mRNA00001,mRNA0(
12 ctg123 . CDS 1201 1500 . + 0 ID=cds00001;Parent=mRNA00001;Name=ede
13 ctg123 . CDS 3000 3902 . + 0 ID=cds00001;Parent=mRNA00001;Name=ede
14 ctg123 . CDS 5000 5500 . + 0 ID=cds00001;Parent=mRNA00001;Name=ede
15 ctg123 . CDS 7000 7600 . + 0 ID=cds00001;Parent=mRNA00001;Name=ede
16 ctg123 . CDS 1201 1500 . + 0 ID=cds00002;Parent=mRNA00002;Name=ede
17 ctg123 . CDS 5000 5500 . + 0 ID=cds00002;Parent=mRNA00002;Name=ede
18 ctg123 . CDS 7000 7600 . + 0 ID=cds00002;Parent=mRNA00002;Name=ede
19 ctg123 . CDS 3301 3902 . + 0 ID=cds00003;Parent=mRNA00003;Name=ede
20 ctg123 . CDS 5000 5500 . + 1 ID=cds00003;Parent=mRNA00003;Name=ede
```

```
21 ctg123 . CDS 7000 7600 . + 1 ID=cds00003;Parent=mRNA00003;Name=ede
22 ctg123 . CDS 3391 3902 . + 0 ID=cds00004;Parent=mRNA00003;Name=ede
23 ctg123 . CDS 5000 5500 . + 1 ID=cds00004;Parent=mRNA00003;Name=ede
24 ctg123 . CDS 7000 7600 . + 1 ID=cds00004;Parent=mRNA00003;Name=ede
```

{{Edet}}

---

{{Sdet}} {{Ssum}} GTF Format *(https://genome.ucsc.edu/FAQ/FAQformat.html#format4)*

GTF (Gene Transfer Format) is a refined form of the GFF with group attributes - essentially the same as GFF2 {{Esum}}

1. seqname - The name of the sequence. Must be a chromosome or scaffold. (chr1 or 1)
2. source - The program that generated this feature.
3. feature - The name of this type of feature. Some examples of standard feature types are "CDS" "start_codon" "stop_codon" and "exon"li>
4. start - The starting position of the feature in the sequence. The first base is numbered 1.
5. end - The ending position of the feature (inclusive).
6. score - A score between 0 and 1000 (UCSC) OR floating point value
7. strand - Valid entries include "+", "-", or "." (for don't know/don't care).
8. frame - If the feature is a coding exon, frame should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be ".".
9. attributes/group - A list of feature attributes in the format tag=value pairs separated by “;”

GTF/GFF2 http://useast.ensembl.org/info/website/upload/gff.html

{{Edet}}

---

# GRAPHING FILE FORMATS

{{Sdet}} {{Ssum}} WIG *(https://genome.ucsc.edu/goldenPath/help/wiggle.html)* Wig files were designed to plot quantitative data,for either equally spaced data, or variable spaced data. As such there are two variations of wig files. For both formats the first line is a descriptor with the following lines representing data in a tab separated columns. Both formats use 1-start, fully-closed" coordinates, meaning the first position is 1 and the last position is N=chromosome of length {{Esum}}

Examples

1. Fixed Step A definition line which indicates the wig-type, the chromosome, the start base and the step - distance between each value (bases). (note the positions are not provide in the file, but are inferred from the start and step values) - values must be continuous.

In the example the data represents values for postions 3001,3001,3002 on chromosome 1.

| **fixedStep chrom=chr1 start=3001 step=1** |
|---|
| 24 |
| 56 |
| 100 |

1. **Variable Step** A definition line which indicates the wig-type and the chromosome. Each positons contain a location value and as such and have discontinuous values.

Example (continuous data - like the fixstep example)

| **variableStep chrom=chr1** | |
|---|---|
| 3001 | 24 |
| 3002 | 56 |
| 3003 1 | 00 |

Example (discontinuous data)

| **variableStep chrom=chr1** | |
|---|---|
| 3001 | 24 |
| 3003 | 100 |
| 3010 | 20 |

{{Edet}}

---

{{Sdet}} {{Ssum}} BEDGRAPH *(https://genome.ucsc.edu/goldenPath/help/bedgraph.html)* This is another format used for plotting data, it does not have a header. The chromosome coordinates are zero-based, half-open. This means that the first chromosome position is 0,and the last position in a chromosome of length N would be N - 1. {{Esum}}

1. **chrom** - name of the chromosome
2. **chromStart** - Start of feature (0-based)
3. **chromEnd** - End of the feature (not included in display)
4. **score** - a score (integer or real positive /negative number)

Example

| **Chromosome** | **Start** | **End** | **Score** |
|---|---|---|---|
| chr1 | 15000 | 20000 | 1 |
| chr2 | 106000 | 108000 | 0.75 |

{{Edet}}

**Indexed binary file formats** of WIGs and BEDs (e.g. bigBED, bigWIG) also exist and these are much more efficient. Only the portions of the files needed for the region currently being processed or visualized are transferred/loaded as needed. Thus for large data sets they are considerably faster than regular files.