Common File Formats

A big list and description of file formats commonly used in bioinformatics is located at https://genome.ucsc.edu/FAQ/FAQformat.html. We only go over the most common formats relevant to this workshop here.

bed/bigBed

Description: Contains genome-wide data in a flexible tabular format. Often, ChIP peaks are reported in a bed format. Many of the "data tracks" in the UCSC genome browser are in this format. The `bigBed` format are an index binary format of bed files. Related formats: BED detail format, bedGraph format, bigBed format

Data Type: Genomic features (e.g., transcripts, genes), ChIP peaks, many others.

File Name Conventions: File often end with a `.bed` and are often gzip commressed `.bed.gz`

Example Record:

A very basic bed file:

```
chr1 100100 100101 rs233454
chr1 200100 200101 rs446788
chr1 300100 300101 rs645678
```

Record Components:

Column	Name	Required	Description
1	`chrom`	Y	The name of the chromosome (e.g., chr3, chrY)
2	`chromStart`	Y	The starting position of the feature. The first base in a chromosome is numbered 0
3	`chromEnd`	Y	The ending position of the feature in the chromosome or scaffold. The `chromEnd` base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as `chromStart=0, chromEnd=100`, and span the bases numbered 0-99
4	`name`	N	The name of the feature
5	`score`	N	A score between 0 and 1000
6	`strand`	N	Defines the strand as `+` or `-`
7	`thickStart`	N	The starting position at which the feature is drawn thickly (e.g., start codon)
8	`thickEnd`	N	The ending position at which the feature is drawn thickly (e.g., stop codon)
9	`itemRgb`	N	An RGB value of the form R,G,B (e.g., 255,0,0).
10	`blockCount`	N	The number of blocks (exons) in the BED line
11	`blockSizes`	N	A comma-separated list of block sizes. The number of items in this list should correspond to `blockCount`
12	`blockStarts`	N	A comma-separated list of block starts.

bigWIG/wiggle

Description: The `bigWig` format is the preferred format for dense, continuous data such as GC percent, probability scores, coverage, etc. It is a compressed binary form allowing for fast access. There are two general forms of these files: 1. `variableStep` is for data with irregular intervals between new data points 2. `fixedStep` is for data with regular intervals between new data values.

For more information please see https://genome.ucsc.edu/goldenPath/help/wiggle.html.

Data Type: Continuous genome-wide data. ChIP-seq tools often output normalized data in this format.

File Name Conventions: Files often end with `.bw` or `.bigWig`.

Example Record:

`variableStep`:

```
variableStep chrom=chr2
300701 12.5
300702 12.5
300703 12.5
300704 12.5
300705 12.5
```

`fixedStep`:

```
fixedStep chrom=chr3 start=400601 step=100
11
22
33
```

Record Components:

`variableStep` files start with a declaration line (`variableStep chrom=chr2`) and is followed by two columns containing positions and data values.

`fixedStep` files begins with a declaration line (`fixedStep chrom=chr3 start=400601 step=100`) and is followed by a single column of data values

fasta

Description: Contains sequence data, often the format of a reference genome and used in NCBI BLAST. This can be nucleic acids or amino acids.

Data Type: Sequences

File Name Conventions: Files often end with either a `.fasta` or `.fq` ; however, due to their size, they are often gzip compressed `.fasta.gz` or `.fa.gz` .

Example Record:

```
>gi|301173067|ref|NR_036051.1| Homo sapiens microRNA 1302-2 (MIR1302-2), microRNA
GGATGCCCAGCTAGTTTGAATTTTAGATAAACAACGAATAATTTCGTAGCATAAATATGTCCCAAGCTTA
GTTTGGGACATACTTATGCTAAAAAACATTATTGGTTGTTTATCTGAGATTCAGAATTAAGCATTTTA
>gi|223555930|ref|NR_026818.1| Homo sapiens family with sequence similarity 138, member A (FAM138A), non-coding RNA
CACACAACGGGGTTTCGGGGCTGTGGACCCTGTGCCAGGAAAGGAAGGGCGCAGCTCCTGCAATGCGGAG
CAGCCAGGGCAGTGGGCACCAGGCTTTAGCCTCCCTTTCTCACCCTACAGAGGGCAGGCCCTTCAGCTCC
ATTCTCCTCCAAGGCTGCAGAGGGGGCAGGAATTGGGGGTGACAGGAGAGCTGTAAGGTCTCCAGTGGGT
CATTCTGGGCCCAGAGATGGGTGCTGAAGCTCCCACGCCTGCCTGTGAAAATGGAGTCCTCTCTCACCTG
GGAGAGCCAGGTGCTGCCCCGAGAAGGATGCATTTATGGCTTCGTGAAGTCTTTCCTGACCCCCGATGCT
GCTGACTATAGAGACAAAGTCTCACTATGTTGCTCAGGCTGGTCTTGAACTCCTGGCCTCAAGCGATCCT
CCCACCTCAGCCTCCCAAAGTGTTGGGATTATAGACATGAGCCACTGCACCTGGCCGACCTTGGGCAAGT
TCTTAAACCCTTCAAAGCCTCATTTTTCTCCAATCACAAAAGGGAAAGATGGTAATATTTTCCCCACCAA
ATTCTTGTCGGATGCCCTCACAGAATTGAGATTATGTACGTAAAACACCAGGTGCCTAACCCGGCACAGA
GCAGGAGGGCTAAGCGTGACATCCAGCACGTGGTCAGTGGAATCCAGTATTCCTACCCACCTCTCTAGTC
TCCCCTCCACCCCTCTCCCTTTCAGAGGCACCAAGCTGCTTGTGGTCTTGTCTATTCCCACTCCCTGCCT
GACTGAACATTTTCTCCACCTCCTGATCATCAGCAGCAGAAACTGGCTGCTCTTCCTCCTGGGTAGACAG
CCAGACTGTATTTCCCAGCTGCCCCTGCAGTGAGATGTGGCCATCGGAGCCAGCATTGGCCAATGGACTC
TGCATGGGAGTGACGCATGCTGCCTCCAGGCTTGTCCCTAAAACCTCCCACGTGTCCTCCGCCTGCTCTT
CCCACCTCCAAGGAGCACGGCAATTGTGGAAGACCCAGATTAGTGATGGCAGAACCATAGATGGGAGGAA
CCTGGGTCCCTGACTTAAAGTATCATGGATTTGGATGTTCCCTTAGTGAGAAATAAACTTCCATTGTGTT
TAAGCCTTTA
```

Record Components:

A sequence in FASTA format begins with a single-line description, called the defline, and always starts with a greater-than (`>`) symbol. In the example above, there are two records:

- `>gi|301173067|ref|NR_036051.1| Homo sapiens microRNA 1302-2 (MIR1302-2), microRNA`
- `>gi|223555930|ref|NR_026818.1| Homo sapiens family with sequence similarity 138, member A (FAM138A), non-coding RNA`

The word directly after the `>` symbol is the identifier of the sequence and the rest of the line is the optional description. After the defline there are at least one line of sequence. Often, each sequence line has a maximum length to assist in readability of the file. Therefore, a single sequence may span several lines like in the examples above.

fastq

Description: Contains sequence data (reads) and is often the format returned to you by a sequencing facility.

Data Type: Unaligned sequences

File Name Conventions: Files often end with either a `.fastq` or `.fq` ; however, due to their size, they are often gzip compressed `.fastq.gz` or `.fq.gz` . In addition, usually a `R1` or `R2` are present in the file name indicating which mate the file responds to if paired-end reads (e.g.,

`sample01_R1.fq.gz` and `sample01_R2.fq.gz` represent a set of paired-end reads).

Example Record:

```
@SRR1523824.1 GA002-SR-R00025:7:1:1824:771/1
AATCTCACCAGATGTCTTGGGTGTTTGTATTACTTGG
+
>>>>4>>>>>>>>>>>>>>>>>>>>1>:>:::::4.
```

Record Components:

- Read ID: `@SRR1523824.1 GA002-SR-R00025:7:1:1824:771/1`
- Sequence: `AATCTCACCAGATGTCTTGGGTGTTTGTATTACTTGG`
- Secondary ID (rarely used): `+`
- Quality (ASCII): `>>>>4>>>>>>>>>>>>>>>>>>>>1>:>:::::4.`

Other Information:

- Each read should have a unique ID
- The sequence length and quality length should match
- To convert quality scores to their Phred-scaled value, you first lookup the decimal value for the ASCII character (e.g., http://www.asciitable.com). Then, you have to subtract the platform-dependent offset (see https://en.wikipedia.org/wiki/FASTQ_format).

narrowPeak

Description: This format is used to provide called peaks of signal enrichment based on pooled, normalized data. It is a special kind of BED format.

Data Type: ChIP peaks

File Name Conventions: Files often end with `.narrowPeak`

Example Record:

```
chr1    9356548 9356648 .       0       .       182     5.0945  -1  50
chr1    9358722 9358822 .       0       .       91      4.6052  -1  40
chr1    9361082 9361182 .       0       .       182     9.2103  -1  75
```

Record Components:

Column	Name	Description
1	`chrom`	The name of the chromosome (e.g., chr3, chrY)
2	`chromStart`	The starting position of the feature. The first base in a chromosome is numbered 0
3	`chromEnd`	The ending position of the feature in the chromosome or scaffold. The `chromEnd` base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as `chromStart=0, chromEnd=100`, and span the bases numbered 0-99
4	`name`	The name of the feature, preferably unique. Use `.` if no name is assigned
5	`score`	A score between 0 and 1000
6	`strand`	Defines the strand as `+` or `-`. Use `.` if not orientation is assigned
7	`signalValue`	Measurement of overall enrichment for the region
8	`pValue`	Measurement of statistical significance (-log10). Used -1 if no `pValue` is assigned
9	`qValue`	Measurement of statistical significance using false discovery rate (-log10). Use -1 if no `qValue` is assigned
10	`peak`	Point-source called for this peak; 0-based offsten from `chromStart`, Use -1 if no point-source is called

SAM/BAM

Description: Most often holds the alignment information based on your reads (e.g., fastq) and your reference assembly. The SAM format is in ASCII text while the BAM format is a binary, compressed, and often indexed form allowing for less disk space and fast, random access.

Data Type: Sequence alignments

File Name Conventions: `.sam`, `.bam`, and `.bai` for bam index

Example Record:

Header:

```
@HD VN:1.3  SO:coordinate
@SQ SN:chr10    LN:129993255
@SQ SN:chr11    LN:121843856
...
@SQ SN:chrX LN:166650296
@SQ SN:chrY LN:15902555
@RG ID:SRR1523824   LB:SRR1523824.GSM1234472    PL:illumina SM:spleen.P.1
@PG ID:bwa  PN:bwa  CL:bwa samse -r @RG\tID:SRR1523824\tSM:spleen.P.1\tLB:SRR1523824.GSM1234472\tPL:illumina /group/refere
nceFiles/Mus_musculus/UCSC/mm9/Sequence/WholeGenomeFasta/genome.fa - /group/biocore-analysis/khernandez/CRI-Workshop-Test/
run/01-Trimmomatic/SRR1523824.IP.spleen.P.1.clipped.R1.fq.gz    VN:0.7.12-r1039
```

Alignment:

```
SRR1523824.26438991 16  chr10   3000654 37  36M *   0   0   CCCCCGTAGAGCACACAGGTGACAGCTGCTATCTTC    3,<>>A9@BB??<<936BBBBB
BBBBBBBBBBBB?B    RG:Z:SRR1523824 XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:36
```

Record Components:

The official standard format definitions: http://www.htslib.org/doc/sam.html

SAM/BAM files consist of a header section followed by alignment records. The header section contains metadata about the genome you aligned to, the sample information, and possible command-line arguments used to generate the SAM/BAM file. You can read more about the header elements at http://samtools.github.io/hts-specs/SAMv1.pdf.

Each alignment record follows this format:

Column	Name	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

The flag field is defined as:

Bit	String Representation	Description
`0x0001`	p	the read is paired in sequencing
`0x0002`	P	the read is mapped in a proper pair
`0x0004`	u	the query sequence itself is unmapped
`0x0008`	U	the mate is unmapped
`0x0010`	r	strand of the query (1 for reverse)
`0x0020`	R	strand of the mate
`0x0040`	1	the read is the first read in a pair
`0x0080`	2	the read is the second read in a pair
`0x0100`	s	the alignment is not primary
`0x0200`	f	the read fails platform/vendor quality checks
`0x0400`	d	the read is either a PCR or an optical duplicate
`0x0800`	S	the alignment is supplementary

Excellent tool for looking up the bitwise flag information: https://broadinstitute.github.io/picard/explain-flags.html