NGS FILE FORMATS

Peter FitzGerald, PhD

Head Genome Analysis Unit, CCR, NCI

DIFFERENT FILE FORMATS

Sequence Data

- FASTA
- FASTQ
- SRA

Alignment Data

- SAM
- BAM
- CRAM

Variant Data

- VCF

Annotation Data

- BED
- bigBED
- GFF
- GTF

Graphing Data

- bedGraph
- WIG
- bigWIG

DIFFERENT FILE FORMATS

COMPRESSED VARIANTS

*.gz. - gzip compression
*.zip - zip compression and or archive
*.tar - archive of files
*.tar.gz. - gzip compressed archive

SEQUENCE FILE FORMATS

FASTA FORMAT

FASTA

Standard text based format for storing simple sequence data.

Each entry consists of a header line that begins with a “>” followed by one or more lines of sequence data.

The format allows for multiple sequences in a single file.

SEQUENCE FILE FORMATS

FASTA FORMAT

FASTA

Single sequence example:

```
>HWI-ST398_0092:1:1:5372:2486#0/1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGAT
ACGTAGCAGCAGCATCAGTACGACTACGACGACTAGCACATGCGACGATCGATGCTAGCTGACTATCGATG
```

Multiple sequence example:

```
>Sequence Name 1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGAT
ACGTAGCAGCAGCATCAGTACGACTACGACGACTAGCACATGCGACGATCGATGCTAGCTGACTATCGATG
>Sequence Name 2
ACGTAGACACGACTAGCATCAGCTACGCATCGATCAGCATCGACTAGCATCACACATCGATCAGCATCACGACTAGCAT
AGCATCGACTACACTACGACTACGATCCACGTACGACTAGCATGCTAGCGCTAGCTAGCTAGCTAGTCGATCGATGAGT
AGCTAGCTAGCTAGC
>Sequence Name 3
ACTCAGCATGCATCAGCATCGACTACGACTACGACATCGACTAGCATCAGCAT
```

SEQUENCE FILE FORMATS

FASTQ FORMAT

FASTQ

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 $\log^{10}$ score for the probability of being wrong.
E.g. score of 30 => 1:1000 chance of error

SEQUENCE FILE FORMATS

FASTQ FORMAT

FASTQ

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

SEQUENCE FILE FORMATS

FASTQ FORMAT

FASTQ

```
@HWI-ST398_0092:6:73:5372:2486#0/1
TTTTTCGTTCTTTTCATGTACCGCTTTTTGTTCGGTTAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGAT
+HWI-ST398_0092:1:1:5372:2486#0/1
ffffeedfcedffffeffdefff_fffffdccfdZdeeadefecZedaecdbRdTY^ZYT``_T`_^bc_Wceaa[
```

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)

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

SEQUENCE FILE FORMATS

QUALITY SCORES

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

SEQUENCE FILE FORMATS

QUALITY SCORES

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

ASCII_BASE=64 Old Illumina

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [	38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

SEQUENCE FILE FORMATS

SRA FORMAT

SRA (**S**equence **R**ead **A**rchive) is a binary archive, used by NCBI for distributing data from it SRA database.

1. Archive format that can hold many different types of data (reads and or alignments etc)
2. Requires use of one or more of the programs in the SRA toolkit to extract usable data.
3. When used with NGS data the most useful tool is probably **fastq-dump**
4. Its challenging to know what data is in the archive

ALIGNMENT FILE FORMATS

SAM FORMAT

```
8_100_10000_12419      163     chrVII  271183  255     40M     =       271294  151
TGGTGTATTATACGCTACCGTGCGGTGCCGGGGGCAACCG
bbbabbbbbbbbbbbbbbbcbbbbcbbbbbbbbbbbbbb       XA:i:0  MD:Z:40 NM:i:0
```

The **SAM Format** (**S**equence **A**lignment/**M**ap) is a text format for storing sequence alignment data in a series of tab delimited ASCII columns.

The file has two parts:

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

ALIGNMENT FILE FORMATS

SAM FORMAT

Example of SAM Header

@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/bowtie2-align-s --wrapper basic-0 -x /fdb/bowtie2.DELETE/mm10 -q jun_minus_dex_rep1a -S jun_minus_dex_rep1a_mm10.sam -p8"

ALIGNMENT FILE FORMATS

SAM FORMAT

```
8_100_10000_12419    163    chrVII  271183  255    40M    =    271294  151
TGGTGTATTATACGCTACCGTGCGGTGCCGGGGGCAACCG
bbbabbbbbbbbbbbbbbbcbbbbcbbbbbbbbbbbbbb    XA:i:0  MD:Z:40 NM:i:0
```

8_100_10000_12	163	chr7	271183	255	40M	=	271294	151	TGGTGTA TTATACG	bbbabbbb bbbbbbbb	XA:i:0 MD:Z:40
QNAME	**FLAG**	**RNAME**	**POS**	**MAPQ**	**CIGAR**	**MRNM**	**MPOS**	**TLEN**	**SEQ**	**QUAL**	**OPT**

Col	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

ALIGNMENT FILE FORMATS

SAM FORMAT

8_100_10000_12419 **163** chrVII 271183 255 40M = 271294 151
TGGTGTATTATACGCTACCGTGCGGTGCCGGGGGCAACCG
bbbabbbbbbbbbbbbbbbcbbbbcbbbbbbbbbbbbbb XA:i:0 MD:Z:40 NM:i:0

Understanding Flag codes

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

1	read paired
2	read mapped in proper pair
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

ALIGNMENT FILE FORMATS

BAM/CRAM FORMAT

BAM (*.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).

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

CRAM (*.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

ANNOTATION FILE FORMATS

BED FORMAT

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 (+/-)

chr1	15000	20000	gene1	50	+
chr2	106000	108000	gene2	400	-

ANNOTATION FILE FORMATS

BED FORMAT

7. **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.
8. **thickEnd** - The ending position at which the feature is drawn thickly (for example the stop codon in gene displays).
9. **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.
10. **blockCount** - The number of blocks (exons) in the BED line.
11. **blockSizes** - A comma-separated list of the block sizes. The number of items in this list should correspond to blockCount.
12. **blockStarts** - A comma-separated list of block starts. All of the blockStart positions should be calculated relative to chromStart. The number of items in this list should correspond to blockCount.

ANNOTATION FILE FORMATS

GFF FORMAT

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

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"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** - 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

ANNOTATION FILE FORMATS

GFF FORMAT

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=gene00001
4  ctg123 . mRNA             1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=EDEN.1
5  ctg123 . mRNA             1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001;Name=EDEN.2
6  ctg123 . mRNA             1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001;Name=EDEN.3
7  ctg123 . exon             1300  1500  .  +  .  ID=exon00001;Parent=mRNA00003
8  ctg123 . exon             1050  1500  .  +  .  ID=exon00002;Parent=mRNA00001,mRNA00002
9  ctg123 . exon             3000  3902  .  +  .  ID=exon00003;Parent=mRNA00001,mRNA00003
10 ctg123 . exon             5000  5500  .  +  .  ID=exon00004;Parent=mRNA00001,mRNA00002,mRNA00003
11 ctg123 . exon             7000  9000  .  +  .  ID=exon00005;Parent=mRNA00001,mRNA00002,mRNA00003
12 ctg123 . CDS              1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
13 ctg123 . CDS              3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
14 ctg123 . CDS              5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
15 ctg123 . CDS              7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
16 ctg123 . CDS              1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
17 ctg123 . CDS              5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
18 ctg123 . CDS              7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
19 ctg123 . CDS              3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
20 ctg123 . CDS              5000  5500  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
21 ctg123 . CDS              7000  7600  .  +  1  ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
22 ctg123 . CDS              3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
23 ctg123 . CDS              5000  5500  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
24 ctg123 . CDS              7000  7600  .  +  1  ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
```

ANNOTATION FILE FORMATS

GTF FORMAT

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

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

GRAPHING FILE FORMATS

WIG (BIGWIG) FORMAT

1) FixedStep

fixedStep	chrom=chr1 start=3001 step=1
24	
56	
100	

2) VariableStep

variableStep	chrom=chr1
3001	24
3002	56
3003	100

variableStep	chrom=chr1
3001	24
3003	56
3010	100

GRAPHING FILE FORMATS

BEDGRAPH FORMAT

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)

chr1	15000	20000	1
chr2	106000	108000	0.75

Format Conversion Utilities

- Galaxy (http://galaxy.psu.edu/ - http://galaxy.cit.nih.gov/)
    - Galaxy is an open, web-based platform for data intensive biomedical research. Whether on the free public server or your own instance, you can perform, reproduce, and share complete analyses.
- Samtools (http://samtools.sourceforge.net)
    - SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format. Also, note TABIX for indexing generic tab delimited files.
- Picard (http://picard.sourceforge.net/)
    - Picard comprises Java-based command-line utilities that manipulate SAM files, and a Java API (SAM-JDK) for creating new programs that read and write SAM files. Both SAM text format and SAM binary (BAM) format are supported.
- UCSC Utilities (http://hgdownload.cse.ucsc.edu/admin/exe/)

Format Conversion Utilities

- Bamtools -(https://github.com/pezmaster31/bamtools)
    - BamTools provides both a programmer's API and an end-user's toolkit for handling BAM files.
- Bedtools (http://bedtools.readthedocs.io/en/latest/)
    - Collectively, the bedtools utilities are a swiss-army knife of tools for a wide-range of genomics analysis tasks. The most widely-used tools enable genome arithmetic: that is, set theory on the genome. For example, bedtools allows one to intersect, merge, count, complement, and shuffle genomic intervals from multiple files in widely-used genomic file formats such as BAM, BED, GFF/GTF, VCF. While each individual tool is designed to do a relatively simple task (e.g., intersect two interval files), quite sophisticated analyses can be conducted by combining multiple bedtools operations on the UNIX command line.
- FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit//)
    - The FASTX-Toolkit is a collection of command line tools for Short-Reads FASTA/FASTQ files preprocessing.
- SRA ToolKit (https://github.com/ncbi/sra-tools)
    - The SRA Toolkit and SDK from NCBI is a collection of tools and libraries for using data in the INSDC Sequence Read Archives.

Binary Formats & Indices

Indexed binary file formats are much more efficient.

Only the portions of the files needed for the region currently being processed or visualized are transferred and loaded as needed. Thus for large data sets they are considerably faster than regular files.
(e.g. bigBED, bigWIG, BAMindexed)

THE END