

NGS基本データフォーマット

基礎生物学研究所
GITC 2021 夏 NGS解析入門

杉浦宏樹

概要

- はじめに
 - データフォーマットとは
 - フォーマットを学ぶ理由
 - Wet研究者がつまずきやすい点
- NGS基本データフォーマット
 - FASTA, FASTQ, SRA
 - BED, GFF/GTF/GFF3, WIG
 - SAM/BAM

`cd ~/gitc/data/4_format` で作業フォルダに移動

データフォーマットとは

- データを記録する際のルール
ルールがあれば情報を効率よく、正確に共有することができる
- 例：Webページ
- HTMLフォーマットを使用することで
 - ハード(PC/スマートフォン)
 - OS(Windows/Mac)
 - ソフト(Chrome/Safari/Firefox)が違ってても、同じページを閲覧可能

**次世代シーケンサ解析では様々なフォーマットが存在
これらの把握が解析に必須！**

フォーマットを学ぶ理由

- NGS解析の基礎知識だから

研究者間のコミュニケーションや解析方法の理解に必須

- 例 1) 同僚 X : A 遺伝子の塩基配列データを見せて
あなた : 了解です。fasta で送りますね
- 例 2) マニュアル : このソフトは fasta と fastq から BAM ファイルを生成します
あなた : マッピングを行うソフトなんだな
- ← fasta 形式が塩基配列情報を含むことを理解していれば、やりとりがスムーズ
- ← 入力と出力の形式から行う解析がわかる

- 研究目的に合わせた解析に必要なだから

フォーマットを知ること、自力で必要な情報を獲得でき
独自性の高い研究が可能に

- 例 3)
1. 巨大な fasta ファイルから配列名だけ取り出したい
 2. fasta 形式では、配列名の頭に常に ">" がつく
 3. ">" がある行だけ集めれば、配列名のリストができる！
(grep コマンドが使えるぞ！)
- ← 専用のプログラムがなくても自分がほしい結果を得られる

Wet研究者がつかずきやすい点

- 形式がたくさんあって区別がつかない！
 - 実態はなじみ深い生物学的情報です
 - 解析で使われる場面や各フォーマットが含む生物学的情報に注目しましょう
- 「謎の文字」が出てくる！
 - \$, %, #など、「謎の文字」が頻出しますが、重要な情報
 - 「ヒトとコンピュータの両方が扱いやすい表記」を考えた努力の結晶
 - 使い方を理解すれば強力な武器になる

NGS解析の流れ

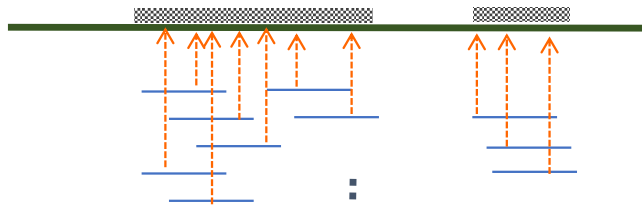
ゲノム（リファレンス）配列

FASTA 形式

```
>chr
ACCTTTTACATTCAGACGACGCGGCAATATGCT
CTGGTGGATTAATAAAGAGTCTGATGACGAC
TTCTGACCTGGTACCTGGGAGGATTAATAA
TTTATTTGACCTAGGACCTAATACCTTACCA
TTTATGCTATGCTGACGACGATTAATAATACG
AGTACGACGATTCGATGAACGATTAAGACGAC
```

インデックス作成

リファレンス配列へのマッピング



マッピング結果 SAM 形式

@HD	VN:1.0	SO:unsorted							
@SQ	SN:chr	IN:4639675							
@PG	ID:bowtie2	PN:bowtie2	VN:2.2.4	CL: "/>bio/bin/bowtie2-align					
SRR1515276.40	0	chr	4423609	42	51M	*	0	0	GGAATTCCTACCTGCA
SRR1515276.158	16	chr	501700	42	51M	*	0	0	ACGACCGAGTGAAG
SRR1515276.212	4	*	0	0	*	*	0	0	GGCGCTTTCAGGCTGT
SRR1515276.319	0	chr	2922768	42	51M	*	0	0	CTTTAGTTCATTTAGG
SRR1515276.367	16	chr	2753873	42	51M	*	0	0	CTGTCCTGCTGCTGAC
SRR1515276.411	0	chr	3440721	42	51M	*	0	0	ACGCAATATTTCTTGA
SRR1515276.434	0	chr	4198737	42	51M	*	0	0	CGCCGATCCCTTCTG

サンプルリード（ゲノム DNA/RNA）

FASTQ 形式（配列+クオリティ）

```
@SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:3625:88631 length=51
ATCCGCTGCGGACCGACCTATGTTCCGCGCAATACAGCTGGGTCAG
+SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:3625:88631 length=51
@@@ND>DDF7DC2FBBBF@FTT<DF@AA6AFFHHHCA?A?B=>B::
@SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:3871:88513 length=51
CACCGTGTAGTACACGATCCGCGTACATCACATCCAGTCCGCTCC
+SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:3871:88513 length=51
CCCFHDFHDFHHHHHHHHUUUGFHGGGGGGGJJJJJJHHGGGHH
```

遺伝子アノテーション GFF(GTF) 形式

```
chr RefSeq start codon 190 192 1.000 + . gene_id "b0001"; transcript_id "b0001";
chr RefSeq CDS 190 252 1.000 + 0 gene_id "b0001"; transcript_id "b0001";
chr RefSeq stop codon 253 255 1.000 + . gene_id "b0001"; transcript_id "b0001";
chr RefSeq exon 190 255 1.000 + . gene_id "b0001"; transcript>
```

コンピュータが
扱いやすい
SAM 形式

BAM 形式

並べ替え
検索
ゲノムブラウザへ

NGS 基本データフォーマット

数十以上のフォーマットが存在しますが、
今回は頻出フォーマットに絞って紹介します

- 配列情報

FASTA, FASTQ, SRA

- アノテーション

BED, GFF/GTF/GFF3, WIG

- マッピング(アライメント)

SAM/BAM

FASTA(.fasta, .fa, .mfa)

概要	配列情報の標準フォーマット
内容	塩基配列 アミノ酸配列
	公共 DB から得られる配列情報

○規則 タイトル行：">" で始まる行

>配列ID 説明(スペース区切り)

タイトル行は改行不可

配列：タイトル行の改行後に記載

塩基配列

配列中は改行可能

>pETEC_80 Escherichia coli E24377A

配列ID

説明 (スペース区切り)

○ファイル例

```
>ETEC_chr Escherichia coli E24377A, complete genome
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAA
GAGTGTCTGATAGCAGCTTCTGAACCTGGTTACCTGCCGTGAGTAAATTAAAAT
TTTATTGACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCGCACA
>pETEC_80 Escherichia coli E24377A plasmid
TTCAGATTAAACACTCCAACATCACCGCGGGCAACTTTGCGCTGAATGCGACA
GTGGCCGGCTCTGAAATCAGCAATACCACGCTGACGGCCACCACCAACATCAA
CCTGACGGCTAAGACGAACAGTGCGAGTTCTGGTGTTTACCTGAAAGAT
```

← タイトル行

← 配列

参考) https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp

FASTQ(.fastq, .fq) FASTA+Qualityの意味

概要	NGS 結果データの実質的な標準形式
内容	塩基配列、一塩基ごとの品質情報 (Quality value)
	マッピング、アセンブルでの入力データ形式

○規則

- 1 行目 : "@" の後にタイトル (配列IDや説明)
 - 2 行目 : 塩基配列
 - 3 行目 : "+" の後にタイトル (タイトルは省略可)
 - 4 行目 : 塩基配列のクオリティ (Quality value)
- * fastaとは異なり塩基配列やクオリティにも改行を入れない

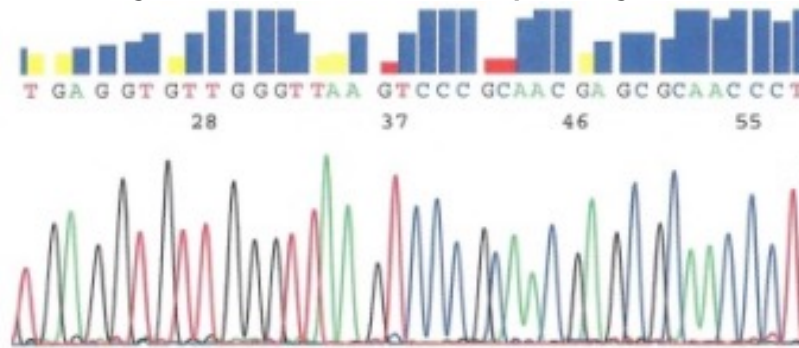
○ファイル例

```
@SEQ_ID ← 配列 ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTTT ← 塩基配列
+
      ← タイトル (今回は省略)
! '*((( (***) )%%%++) (%%%) .1***-+*') ) **55CCF>>>>>CCCCCCC65 ← クオリティ (QV)
```

[実習 1] less コマンドで ex1.fq の中身を見て、fastq 形式を確認しよう

FASTQ のポイント

塩基配列の信頼性も示せる
Quality Value (Phred quality score)



ABI キャピラリーシーケンサーで
この部分で表されていた値

$QV = -10 \log_{10} p$ (p : 間違った塩基決定である確率)

$QV = 30 \rightarrow p = 0.001$ (エラー率 0.1% = 塩基の信頼性 99.9%)

$QV = 20 \rightarrow p = 0.01$ (エラー率 1.0% = 塩基の信頼性 99.0%)

実際の FASTQ データをみると、 数値でなく、英数字や記号が書かれている！

@SEQ_ID

GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCCTTTGTTCAACTCACAGTTT

+

! ' ' * ((((* * * +)) % % % + +) (% % %) . 1 * * * - + * ' ') * * 55 C C F > > > > > C C C C C C C C 65

← 塩基配列

← QV

英数字や記号の正体

→ "ASCII 文字" を使って QV を 1 文字で表したものの

ASCII : American Standard Code for Information Interchange

コンピュータでは文字を数値で表す
通信のため文字と数値の対応関係を規定
0 ~ 127 の数値に文字を割り当てる

A \longleftrightarrow 65 (10 進数) APPLE \longleftrightarrow 65;112;112;108;101 (10 進数)

FASTQ では ASCII 文字を使って、QV (数値) を文字で表す

利点 : 10 進数表記よりもファイルサイズを減らせる
(字数が半分、区切り文字も不要)

塩基:	G	A	T	T	G	G	T	G	A	A	T	T	文字が各塩基 の QV を表現
文字:	?	?	@	A	>	=	;	9	7	4	0	,	

QV から文字への変換規則

問題点：ASCII コードでは 0 - 32 はコンピュータ用の特殊文字に割り当てられている

ASCII 文字コード表

数値	文字
0	Null 文字
1	SOH (ヘッダ開始)
2	STX (テキスト開始)
3	ETX (テキスト終了)
4	EOT (転送終了)
.....
30	RS (レコード区切り)
31	US (ユニット区切り)
32	(スペース)
33	!
34	"

特殊文字
コンピュータ用

- ・ NGS では 10 - 30 を頻用
 $p = 0.001 \rightarrow QV = 30$
…ASCIIコードではレコード区切りを意味
 - ・ 妥協案として特定値を加算してから文字に変換
 $QV \text{ 値} + X = \text{ASCII 値}$ とする
 - ・ X は現在 $X = 33$ でほぼ統一
- 例) QV 30 を表す場合
 $30 + 33 = 63$
→ ASCII コードで 63 に該当する
文字を当てる ("?" が該当)
- ・ 変換には ASCII 文字コード表と簡単な計算が必要

[実習 2] ex2.fq の QV 値を求め、すべての配列の p 値（エラー確率）が 0.01 以下となるように 3' 側をトリミングしよう

ex2.fq

```
@SEQ_ID
GATTGGTGAATT
+
??@A>=;9740,
```

QV 値 + 33 = ASCII 値

ASCII 文字コード表

文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進
NUL	0	00	DLE	16	10	SP	32	20	@	64	40	P	80	50	`	96	60
SOH	1	01	DC1	17	11	!	33	21	A	65	41	Q	81	51	a	97	61
STX	2	02	DC2	18	12	"	34	22	B	66	42	R	82	52	b	98	62
ETX	3	03	DC3	19	13	#	35	23	C	67	43	S	83	53	c	99	63
EOT	4	04	DC4	20	14	\$	36	24	D	68	44	T	84	54	d	100	64
ENQ	5	05	NAK	21	15	%	37	25	E	69	45	U	85	55	e	101	65
ACK	6	06	SYN	22	16	&	38	26	F	70	46	V	86	56	f	102	66
BEL	7	07	ETB	23	17	'	39	27	G	71	47	W	87	57	g	103	67
BS	8	08	CAN	24	18	(40	28	H	72	48	X	88	58	h	104	68
HT	9	09	EM	25	19)	41	29	I	73	49	Y	89	59	i	105	69
LF*	10	0a	SUB	26	1a	*	42	2a	J	74	4a	Z	90	5a	j	106	6a
VT	11	0b	ESC	27	1b	+	43	2b	K	75	4b	[91	5b	k	107	6b
FF*	12	0c	FS	28	1c	,	44	2c	L	76	4c	\	92	5c	l	108	6c
CR	13	0d	GS	29	1d	-	45	2d	M	77	4d]	93	5d	m	109	6d
SO	14	0e	RS	30	1e	.	46	2e	N	78	4e	^	94	5e	n	110	6e
SI	15	0f	US	31	1f	/	47	2f	O	79	4f	_	95	5f	o	111	6f
			DEL	127	7f												

* LFはNL、FFはNPと呼ばれることもある。

<http://e-words.jp/p/r-ascii.html>

* 赤字は制御文字、SPは空白文字(スペース)、黒字と緑字は図形文字。

* 緑字はISO 646で割り当ての変更が認められており、例えば日本ではバックスラッシュが円記号になっている

[実習 2] ex2.fq の QV 値を求め、すべての配列の p 値（エラー確率）が 0.01 以下となるように 3' 側をトリミングしよう

解説

@SEQ_ID ← 配列 ID
 GATTGGTGAATT ← 塩基配列
 + ← 配列 ID (省略)
 ??@A>=;9740, ← QV

① p 値が 0.01 の時の QV 値を求める

$$\begin{aligned} QV &= -10 \log_{10} p \\ &= -10 \log_{10} 0.01 \\ &= -10 (-2) \\ &= 20 \end{aligned}$$

QV < 20 部分をトリムすればよい

② 各文字を ASCII 値になおし、33 を引いて QV 値にする

塩基:	G	A	T	T	G	G	T	G	A	A	T	T
文字:	?	?	@	A	>	=	;	9	7	4	0	,
ASCII値:	63	63	64	65	62	61	59	57	55	52	48	44
QV値:	30	30	31	32	29	28	26	24	22	19	15	11

$$\begin{aligned} QV \text{ 値} + 33 &= \text{ASCII 値} \\ QV \text{ 値} &= \text{ASCII 値} - 33 \end{aligned}$$

文 字	10 進	16 進	文 字	10 進	16 進	文 字	10 進	16 進
SP	32	20	0	48	30	@	64	40
!	33	21	1	49	31	A	65	41
"	34	22	2	50	32	B	66	42
#	35	23	3	51	33	C	67	43
\$	36	24	4	52	34	D	68	44
%	37	25	5	53	35	E	69	45
&	38	26	6	54	36	F	70	46
'	39	27	7	55	37	G	71	47
(40	28	8	56	38	H	72	48
)	41	29	9	57	39	I	73	49
*	42	2a	:	58	3a	J	74	4a
+	43	2b	;	59	3b	K	75	4b
,	44	2c	<	60	3c	L	76	4c
-	45	2d	=	61	3d	M	77	4d
.	46	2e	>	62	3e	N	78	4e
/	47	2f	?	63	3f	O	79	4f

(参考)古いFASTQ ファイルを見る上での注意

1. QV 値はあくまでシーケンサーによる推定値 目安として利用
2. 古い Solexa / Illumina データでは規格が乱立！！ ←注意

解析ソフト ver. (CASAVA)	~1.3	1.3~1.5	1.5~1.8	1.8~
参考使用時期	~2009	2009~2010	2010~2012	2012~
QV 値算出法	Solexa	Phred	Phred	Phred
X 値	64	64	64	33
QV range	-5~40	0~40	3~40 (2=end of read)	0~40

QV値 + X = ASCII 値

自分のデータがどのバージョン由来か確認し
解析ソフトの設定を補正する必要がある

FASTQ のまとめ

概要： 塩基配列情報と各塩基の信頼性を表現する

規則： 1 行目： "@" 配列IDやタイトル
2 行目： 塩基配列
3 行目： "+" (配列名)
4 行目： 塩基配列のクオリティ

ポイント： クオリティは ASCII 文字で表現されている

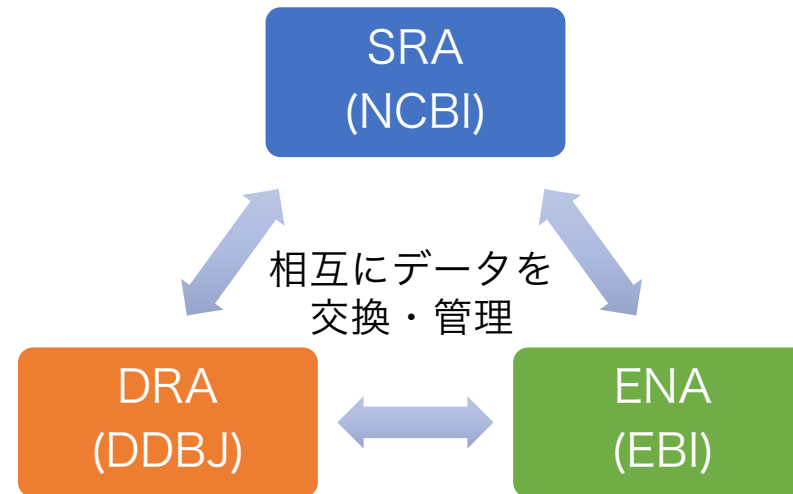
$QV \text{ 値} + 33 = \text{ASCII 値}$

FASTA/FASTQ を扱う際に便利なツール

Seqkit : <https://github.com/shenwei356/seqkit>

SRA(Sequence Read Archive)

NGS データを登録するデータベース



配列データにはそれぞれ **SRR**, **DRR**, **ERR** で始まるアクセッション番号が付けられている。

ex) DRR140361

論文等でこの番号が記載されていれば、これを使いデータのダウンロードが可能である。

次世代シーケンサを使った研究ではFASTQファイルをSRAに登録し、
そのアクセッション番号を明記することが求められる。

SRA format(.sra)

- SRA で使用されている圧縮（バイナリ*）形式 * 機械語
- SRA への NGS データの登録とダウンロードのためだけの専用の形式
- FASTQ に変換可能

SRA を扱う際に便利なツール

SRA toolkit : <https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>

SRA Toolkit 使用例

fastq-dump

…SRA 形式のファイルからFASTQ ファイルを抽出するコマンド

➤ シングルエンドリードの場合（オプションなしで実行する）

```
$ fastq-dump hoge.sra
```

➤ ペアエンドリードの場合（ファイルが分割されるように指示する必要がある）

```
$ fastq-dump --split-files hoge.sra
```

[実習 3]

DRR140361.sra はナミテントウの RAD-seq 解析結果のデータ (paired-end) である。SRA Toolkit の fastq-dump コマンドを使用して、sra 形式のファイルから fastq ファイルを抽出しよう。また ls コマンドで両ファイルのファイルサイズを確認しよう。

```
$ fastq-dump --split-files DRR140361.sra
```

DRR140361_1.fastq, DRR140361_2.fastq と分割された fastq ファイルが生成されていることを確認する。それぞれ forward と reverse に対応する。

```
$ ls -lh
```

sra 形式のファイルの方がサイズが小さいことを確認する。

NGS 基本データフォーマット

数十以上のフォーマットが存在しますが、
今回は頻出フォーマットに絞って紹介します

- 配列情報

FASTA, FASTQ, SRA

- アノテーション

BED, GFF/GTF/GFF3, WIG

- マッピング(アライメント)

SAM/BAM

BED (.bed), GFF/GTF/GFF3(.gff/gtf/gff3)

概要	ゲノム上の特徴配列を表現する（アノテーション情報）
内容	遺伝子名 染色体上の位置 向き エキソン構造
	公共 DB からアノテーション情報をダウンロード 解析したい領域の指定 アノテーション作業 遺伝子構造予測ソフトの結果出力

<4 形式の違い>

BED	ブラウザでの描画情報（色など）を記録可能
GFF	拡張性が高く様々な特徴情報を記録可能
GTF	GFF の厳格化版 一貫した規則で特徴情報を記録可能
GFF3	GTF（GFF version2）の改良版

BED (Browser Extensible Data)

ブラウザでの描画情報（色など）を記録可能

○規則

項目数 3 - 12 タブ区切り

省略する場合は何も書かない（タブを 2 個連続させる）

染色体/ Scaffold 名	指定領域		遺伝子名	スコ ア/ 表記 の濃 淡	ス ト ラ ン ド	太線表示		表示色 赤, 緑, 青 の強度 (0 - 255)	ブロック (exon等) の情報 コンマ区切りで表記		
	開始 位置	終了 位置				開始 位置	終了 位置		個 数	サイズ	開始 位置
chr22	1000	5000	cloneA	960	+	1000	5000	255, 0, 0	2	567,488,	0,3512
chr22	2000	6000	cloneB	900	-	2000	6000	0, 0, 255	2	433,399,	0,3601

1 - 3 項目は必須

4 - 12 項目は省略可

領域開始位置 = 0 とした位置

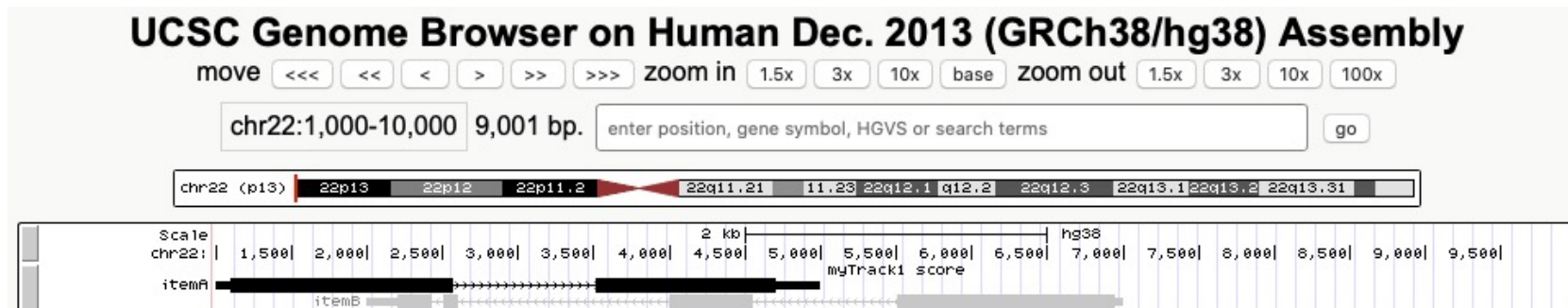
BED フォーマットを扱う際に便利なツール

bedtools : <http://bedtools.readthedocs.io/en/latest/>

[実習 4] ex4.bed はヒトゲノム (GRCh37) の一部を bed 形式にしたものである
less コマンドで bed 形式を確認しよう

BED format ブラウザ表示例

染色体/ Scaffold d名	指定領域		遺伝子名	スコ ア/ 表記 の濃 淡	ス ト ラ ン ド	太線表示		表示色 赤, 緑, 青 の強度 (0 - 255)	ブロック (exon等) の情報 コンマ区切りで表記		
	開始 位置	終止 位置				開始 位置	終了 位置		個 数	サイズ	開始 位置
chr22	1000	5000	itemA	960	+	1100	4700	0	2	1567,1488,	0,2512
chr22	2000	7000	itemB	200	-	2200	6950	0	4	433,100,550,1500,	0,500,2000,3500



表示の濃淡

shade	score in range	≤ 166	167-277	278-388	389-499	500-611	612-722	723-833	834-944	≥ 945
-------	----------------	-------	---------	---------	---------	---------	---------	---------	---------	-------

(参考)

- <https://genome.ucsc.edu/FAQ/FAQformat.html#format1>
- <https://genome-asia.ucsc.edu/goldenPath/help/hgTracksHelp.html> Example #3A

GFF (General Feature Format / Gene Finding Format)

拡張性が高く様々な特徴情報を記録可能

ゲノムアノテーションの標準的形式

○規則

項目数 5 - 9 タブ区切り

セミコロンで区切られたタグ=値の対

省略する場合は“-”や“.”を入れる

染色体/ Scaffold 名	予測ソフト 名等	領域の 種類	指定領域		スコア	ストラン ド	読 み 枠	属性
			開始 位置	終止 位置				
chr22	Manual	exon	1001	5000	960	+	0	.

chr22	Manual	exon	2001	6000	900	-	0	NAME "pol1";
-------	--------	------	------	------	-----	---	---	--------------

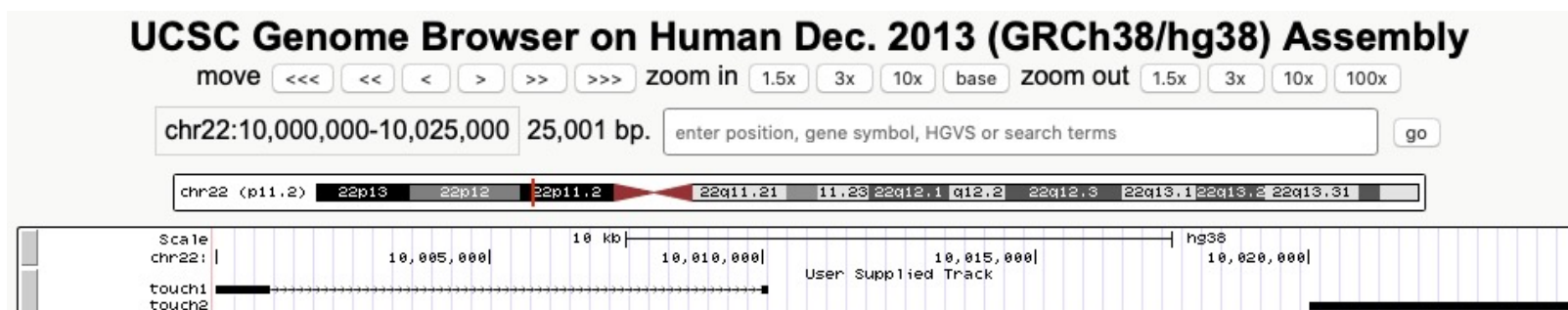
必須

省略可

属性カラムに様々な情報を追加できる → 拡張性高

GFF format ブラウザ表示例

染色体/ Scaffold 名	予測ソフト 名等	領域の種類	指定領域			スト ラン ド	読 み 枠	属性
			開始位置	終止位置	スコア			
chr22	TeleGene	enhancer	10000000	10001000	500	+	.	touch1
chr22	TeleGene	promoter	10010000	10010100	900	+	.	touch1
chr22	TeleGene	promoter	10020000	10025000	800	-	.	touch2



(参考) <https://genome.ucsc.edu/FAQ/FAQformat.html#format3>

GTF (General Transfer Format)

○規則 基本的に GFF と同じだが、仕様をより細かく規定

染色体/ Scaffold 名	予測ソフト 名等	領域の 種類	指定領域		スコア	スト ランド	読み 枠	属性
			開始 位置	終止 位置				
chr22	Twinscan	CDS	380	401	.	+	0	gene_id "001"; transcript_id "001.1";
chr22	Twinscan	CDS	501	650	.	+	2	gene_id "001"; transcript_id "001.1";
chr22	Twinscan	CDS	700	707	.	+	2	gene_id "001"; transcript_id "001.1";
chr22	Twinscan	start_codon	380	382	.	+	0	gene_id "001"; transcript_id "001.1";
chr22	Twinscan	stop_codon	708	710	.	+	0	gene_id "001"; transcript_id "001.1";

必須：CDS, start_codon, stop_codon

任意：5UTR, 3UTR, inter, inter CNS, intron_CNS, exon

それ以外は無効

遺伝子と転写産物の ID を表記する

[実習 5] ex5.gtf は ex4.bed と同じ領域を gtf 形式にしたものである
less コマンドで gtf 形式を確認しよう

参考) <http://genome.ucsc.edu/FAQ/FAQformat.html#format4>

GFF3 (General Feature Format version3)

○規則

GTF (GFF version2) の改良版
いくつかのカラムでその値の制約が厳しくなっている
項目数 9 タブ区切り

染色体/ Scaffold 名	予測 ソフト 名等	領域の 種類	指定領域		スコア	ストランド	読み枠	属性
			開始位置	終止位置				

##gff-version 3

ctg123	.	exon	1300	1500	.	+	.	ID=exon00001
ctg123	.	exon	1050	1500	.	+	.	ID=exon00002
ctg123	.	exon	3000	3902	.	+	.	ID=exon00003
ctg123	.	exon	5000	5500	.	+	.	ID=exon00004
ctg123	.	exon	7000	9000	.	+	.	ID=exon00005

(参考) <http://gmod.org/wiki/GFF3>

GFF/GTF/GFF3とBEDでは座標表現が異なる

GFF/GTF/GFF3 : 開始、終了ともに 1-based (1 から始まる) 座標

BED : 開始は 0-based, 終了は 1-based 座標

具体例

GFF/GTF/GFF3	1	2	3	4	5	6	7	8	
	A	G	T	A	C	T	C	G	
BED	0	1	2	3	4	5	6	7	8

黄色部分を示す時

GFF/GTF/GFF3 : 開始 3, 終了 6 (長さは $6 - 3 + 1 = 4$)

BED : 開始 2, 終了 6 (長さは $6 - 2 = 4$)

[実習 6] ex4.bed と ex5.gtf を開き、実際に座標がずれていることを確認しよう

WIG (wiggle)

概要	ゲノム上の量的特徴を表現するための形式
内容	ゲノム上の座標に対する "数値" 情報
	GC 含量、発現量などを表す
座標	開始、終了ともに 1-based (1 から始まる)

○規則 2 形式から選べる

1) VariableStep 柔軟な指定が可能

variableStep chrom=chr2

300601 22.5

300701 30.5

300751 28.2

位置と値の組で領域を指定するため
間隔は位置ごとに変更可能

2) FixedStep コンパクトな表現が可能

fixedStep chrom=chr3 start=300601 step=100

22.5

30.5

25.8

間隔は固定で、開始位置と
間隔は先頭行で指定し、
後は値のみを示していく

参考) <http://genome.ucsc.edu/goldenPath/help/wiggle.html>

WIG format ブラウザ表示例

VariableStep

variableStep chrom=chr19 span=150

49304701	10.0
49304901	12.5
49305401	15.0
49305601	17.5
49305901	20.0
49306081	17.5
49306301	15.0
49306691	12.5
49307871	10.0

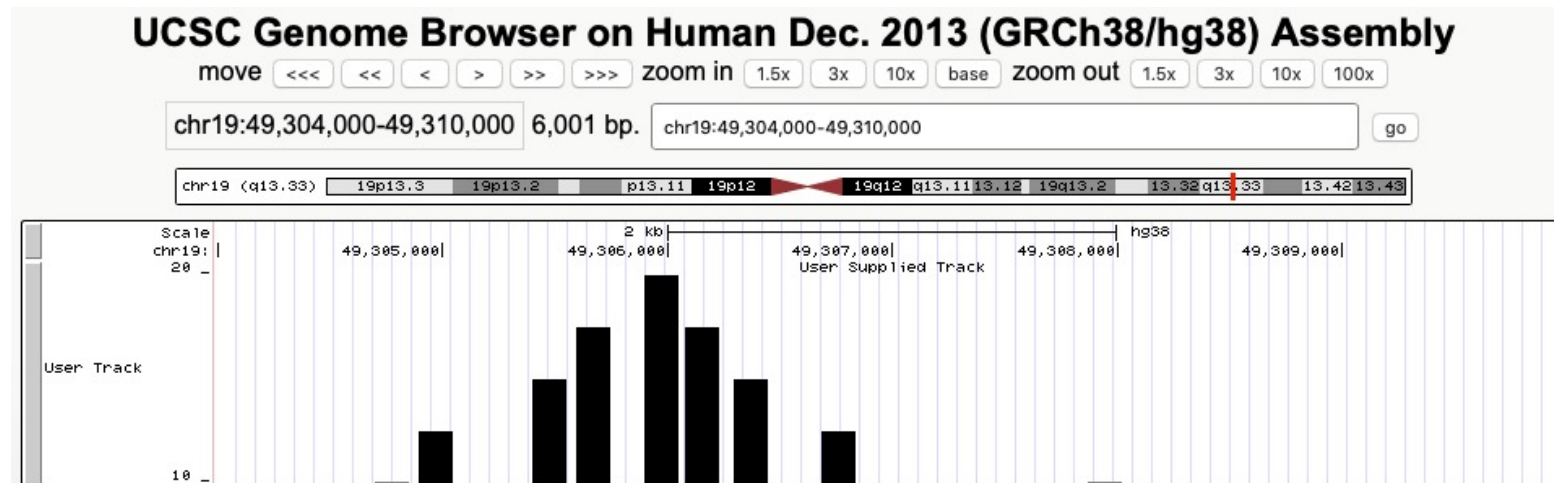
位置と値の組で
領域を指定するため
間隔は位置ごとに
変更可能

FixedStep

fixedStep chrom=chr19 start=49307401 step=300 span=200

1000
900
800
700
600
500
400
300
200
100

間隔は固定で、
開始位置と間隔は
先頭行で指定し、
後は値のみを示していく



(参考) <https://genome.ucsc.edu/goldenpath/help/wiggle.html>

NGS 基本データフォーマット

数十以上のフォーマットが存在しますが、
今回は頻出フォーマットに絞って紹介します

- 配列情報

FASTA, FASTQ, SRA

- アノテーション

BED, GFF/GTF/GFF3, WIG

- マッピング(アライメント)

SAM/BAM

SAM(Sequence Alignment Map)

概要	マッピング（アライメント）結果を表現
内容	マッピング情報（位置, インデル, ミスマッチ） ペアフラグメントの状況, 塩基配列
	SNP、発現量解析への入力データ形式
座標	開始、終了ともに 1-based (1 から始まる)

○ファイル例

```
@HD VN:1.6 SO:coordinate
```

ヘッダー部

```
@SQ SN:ref LN:45
```

```
r001 99 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCCTAAGCTAA * SA:Z:ref,29,-,6H5M
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M
r001 147 ref 37 30 9M = 7 -39 CAGCGGCAT * NM:i:1
```

マッピング結果

[実習 7] ex7.sam を開き sam 形式を確認しよう

参考) <https://samtools.github.io/hts-specs/SAMv1.pdf>

○規則

ヘッダー部

@HD VN:1.6 SO:coordinate

@SQ SN:ref LN:45

"@"で開始

@HD VN: (バージョン) SO: (ソート状況)

@SQ SN: (リファレンス名) LN: (リファレンスの長さ)

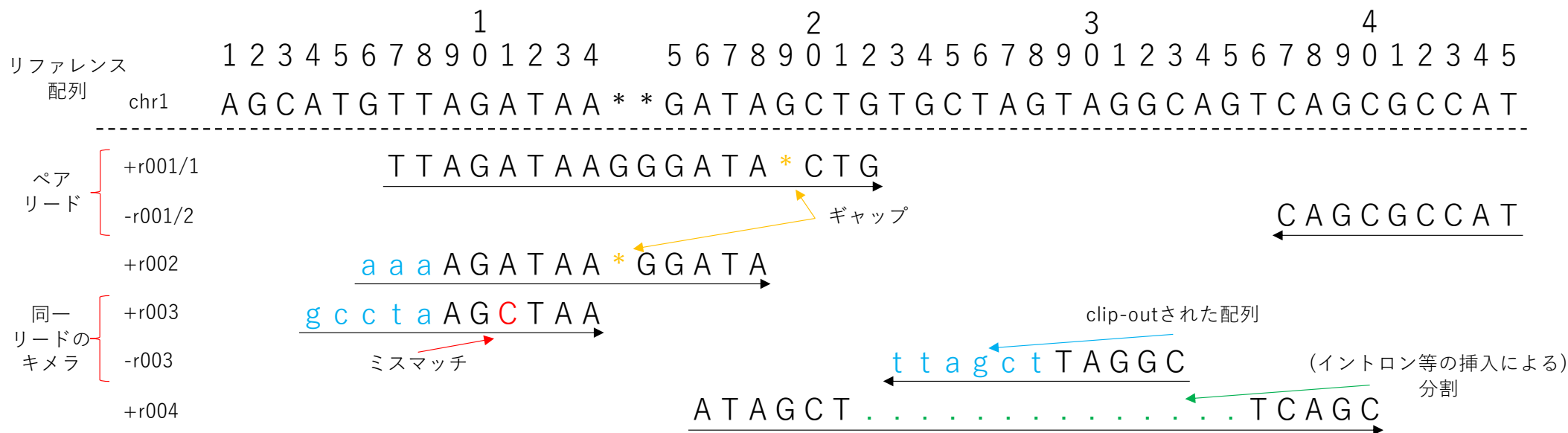
マッピング結果部分 項目間はタブで区切る

クエリ 配列名	FLAG	リファ レンス 配列名	アライ メント 開始位 置	マッ ピン グ QV	CIGAR	ペアフラグメン トの場所				配列	配 列 Q V	オプション
						Ref 名	開 始	長 さ				
r001	99	ref	7	30	8M2I4M1D3M	=	37	39		TTAGATAAAGGATACTG	*	
r002	0	ref	9	30	3S6M1P1I4M	*	0	0		AAAAGATAAGGATA	*	
r003	0	ref	9	30	5S6M	*	0	0		GCCTAAGCTAA	*	SA:Z:ref,29
r004	0	ref	16	30	6M14N5M	*	0	0		ATAGCTTCAGC	*	
r003	2064	ref	29	17	6H5M	*	0	0		TAGGC	*	SA:Z:ref,9,
r001	147	ref	37	30	9M	=	7	-39		CAGCGGCAT	*	NM:i:1

参考: <https://samtools.github.io/hts-specs/SAMv1.pdf>

クエリ 配列名	FLAG	リファ レンス 配列名	アライ メント 開始位 置	マッ ピン グ QV	CIGAR	ペアフラグメン トの場所			配列	配 列 Q V	オプション
						Ref 名	開 始	長 さ			
r001	99	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATACTG	*	
r002	0	ref	9	30	3S6M1P1I4M	*	0	0	AAAAGATAAGGATA	*	
r003	0	ref	9	30	5S6M	*	0	0	GCCTAAGCTAA	*	SA:Z:ref,29
r004	0	ref	16	30	6M14N5M	*	0	0	ATAGCTTCAGC	*	
r003	2064	ref	29	17	6H5M	*	0	0	TAGGC	*	SA:Z:ref,9,
r001	147	ref	37	30	9M	=	7	-39	CAGCGGCAT	*	NM:i:1

ペア
リード



SAMのポイント1：CIGAR

アライメント状況を数字と文字を組み合わせて示す

フラグメント名	FLAG	リファレンス配列名	アライメント開始位置	マップングQV	CIGAR	ペアフラグメントの場所			配列	配列QV	オプション
						Ref名	開始	長さ			
r001	99	ref	5	30	3M2D2M	=	37	39	GCAAG	44>>>	

3M2D2M

塩基数

状況

3 塩基一致、2 塩基欠失、2 塩基一致

ref : ATGCGCATTAGCCTAA
read : GCA--AG

記号	状況
M	一致
I	挿入
D	欠失
N	イントロン (RNAvsDNAのみ)
S	クリップ (塩基情報残す)
H	クリップ (塩基情報削除)
P	他リードが挿入されている

SAMのポイント2：FLAG

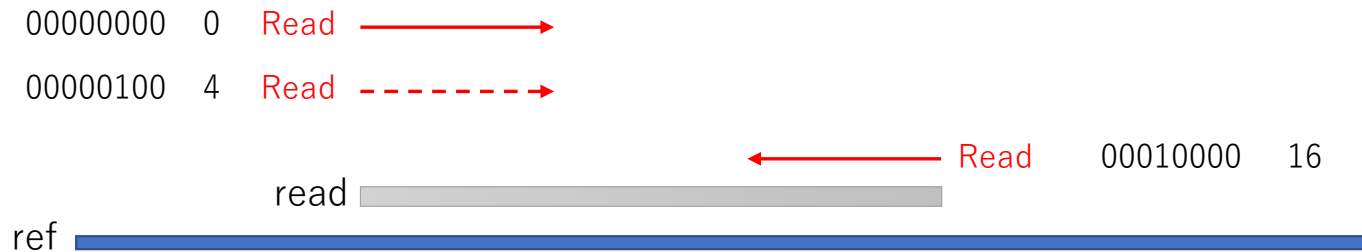
フラグとはある状態についての有無を0 or 1 で表したもの
ここではアライメントの状態を合わせて一つの整数値として記載
理解できると「マップされなかったリードだけ選ぶ」などの操作が可能になる

数値(10進数)	数値(2進数)	意味
1	000000000001	ペアリードがある
2	000000000010	両方適切にマップされている
4	000000000100	自分がマップされていない
8	000000001000	ペア相手がマップされていない
16	000000010000	逆鎖にマップされた（配列も逆鎖で表記）
32	000000100000	ペア相手は逆鎖にマップされた
64	000001000000	Read 1 の配列である
128	000010000000	Read 2 の配列である
256	000100000000	Multiple hit でトップヒットでないアライメント
512	001000000000	マッピング QV が低い
1024	010000000000	PCR あるいは光学的重複
2048	100000000000	キメラ検出された場合の補足的アライメント

複数の状況に合致する場合は数値を加算

(例) ペアリードがあり、, 両方マップされた → $1 + 2 = 3$

single end readの場合



数値(10進数)	数値(2進数)	意味
1	000000000001	ペアリードがある
4	000000000100	自分がマップされていない
16	000000010000	逆鎖にマップされた(配列も逆鎖で表記)

FLAG = 0 : すべてのビット値が0になっている
正常にマップされており、順鎖に対してマップされている

FLAG = 4 : 正常にマップされなかった

FLAG = 16 : 正常にマップされており、逆鎖に対してマップされている

Paired end readでFLAG値の組合せを見る



	ペアリードがある 両方適切にマッピングされている 自分がマッピングされていない ペア相手がマッピングされていない 逆鎖にマッピングされた ペア相手は逆鎖にマッピングされた Read1の配列である Read2の配列である	2進数表記	SAM ファイルの記載 10進数表記
	1 1 1 1 1 1 1 1	11111111	255
通常の paired end read で consistent にアラインしていれば この4通りになる	0 1 0 1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 1 0 0 1 1 1 0 1 0 0 0 1 1	01010011 01100011 10010011 10100011	83 99 147 163
片方しかアラインしていない場合	0 1 0 0 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1	01001001 01011001 01000101 01100101 10001001 10011001 10000101 10100101	73 89 69 101 137 153 133 165
どちらもアラインしていない場合	0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1	01001101 10001101	77 141

自動で FLAG を計算してくれるサイト

The screenshot shows the 'Picard' web interface for 'Decoding SAM flags'. At the top, there's a header with the Picard logo, a 'build passing' status, and links for 'Latest Jar Release', 'Source Code ZIP File', 'Source Code TAR Ball', and 'View On GitHub'. Below the header, the title 'Decoding SAM flags' is displayed. The main content area explains the utility's purpose: to identify read properties from SAM flag values or vice versa. It includes a text input field for 'SAM Flag:' with an 'Explain' button. A 'Switch to mate' toggle is also present. The 'Find SAM flag by property:' section contains a list of 13 properties with checkboxes, such as 'read paired', 'read mapped in proper pair', 'read unmapped', 'mate unmapped', 'read reverse strand', 'mate reverse strand', 'first in pair', 'second in pair', 'not primary alignment', 'read fails platform/vendor quality checks', 'read is PCR or optical duplicate', and 'supplementary alignment'. A 'Summary:' section is located to the right of the properties. The footer indicates the project is maintained by 'broadinstitute' and hosted on GitHub Pages with a theme by 'orderedlist'.

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: [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.

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

Summary:

Project maintained by [broadinstitute](#) | Hosted on GitHub Pages — Theme by [orderedlist](#)

<https://broadinstitute.github.io/picard/explain-flags.html>

SAMのまとめ

概要：各リードがマップされた場所と状態を表す

規則：ヘッダ部とアライメント部からなる タブ区切り

ポイント

CIGAR 値 → アライメント状況を数字と文字を組み合わせで示す

FLAG 値 → リードのマップ状況を数値で示す

SAM format の詳細な仕様書

<http://samtools.github.io/hts-specs/SAMv1.pdf>

BAM format

- BAM

SAM をバイナリ（機械語）化したもの

容量が小さくなるが、人には理解できない

SAM に戻すことも可能なので必要に応じて変換

- BAM indexing file

BAM ファイルに対して作られる検索用ファイル

高速検索や可視化ソフトなどに必要

SAM/BAM format を扱う際に便利なツール

- **Samtools** : <http://www.htslib.org/>
- **Picard** : <https://broadinstitute.github.io/picard/index.html>

NGS 基本データフォーマットまとめ

	FASTA	FASTQ	SAM
概要	配列情報の標準形式	NGS 結果の標準形式	マッピング結果を示す
内容	塩基配列 アミノ酸配列	塩基配列と 一塩基毎の品質情報	マッピング情報 ペアの状況, 塩基配列
	公共 DB からの 配列情報ダウンロード	マッピング、アセンブル解析 での入力データ形式	マップ結果の閲覧、集計 SNP、発現量解析への入力
特徴		QV 値は ASCII 文字で表現 SRA から変換可能	CIGAR, FLAG 値を利用 バイナリ化したのが BAM

	BED	GFF	GTF	GFF3	WIG
概要	ゲノム上の特徴配列を表現する				ゲノム上の量的特徴を表現
内容	遺伝子名 染色体上の位置 向き エキソン構造				ゲノム上の座標に対する “数値”情報
	公共 DB からアノテーション情報をダウンロード 解析したい領域の指定 アノテーション作業 遺伝子構造予測ソフトの結果出力				GC 含量、発現量などを表す
特徴	ブラウザでの 描画情報を記録	拡張性高	GFF の厳格化版 一貫した規則	GTF の 改良版	2 つの形式 VariableStep/FixedStep