

NGS基本データフォーマット と基本ツール

基礎生物学研究所
生物機能解析センター
山口勝司

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

データフォーマットとは？

データを記録するルール

ルールがあれば情報を効率良く正確に共有できる

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 (IE/Chrome/Safari)

が違っても、どんな環境でも同じページを閲覧可能

次世代シーケンサー解析では
様々なフォーマットが使われる
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A : A遺伝子の塩基配列データ見せて ← 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
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からtree/phylipファイルを生成します ← 入力と出力の形式から行った解析がわかる
あなた : 系統解析をするソフトなんだな

研究目的にあわせた解析に必要なだから

フォーマットを知ると、そこから自力で必要な情報を獲得できる
これにより、独自性の高い研究が可能になります

- 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！
(エクセルの"並べ変え"機能でできそうだ！) ← 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

効率良い学習のポイント

Wet 研究者がつまずく点

1: たくさん形式が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
- 各フォーマットが含む生物学的情報や解析で使われる場面に注目しましょう

2: 意味不明な文字がでてる！

- \$や#など“意味不明文字”が頻出しますが、実は重要な情報が含まれています
- 「ヒトとコンピュータ、両方に扱いやすい表記」を考えた開発者の努力の結晶です
- 使い方を理解すれば強力な武器になります。がんばって理解しましょう

以上を踏まえて、各フォーマットを見ていきましょう

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

▪ 配列用

FASTA, FASTQ, SRA

▪ アノテーション用

BED, GFF/GTF, WIG

▪ マッピング(アライメント)用

SAM/BAM

FASTA

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
例	公共DBからの配列情報ダウンロード

○規則

“>”で始まる行がタイトル行、改行後に配列
タイトル行は改行不可 配列中では改行可能

○ファイル例

```
>gi|31342400 Bos taurus crystallin, gamma S (CRYGS), mRNA ←タイトル行
TGCACCAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTTGAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
>gi|31342400 Bos taurus crystallin, gamma S (CRYGS), mRNA
TGCACCAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTTGAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
```

FASTQ

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)
例	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 (省略可)
4行目 : 配列のクオリティ
* 配列とクオリティには基本的に改行を入れない

○ファイル例

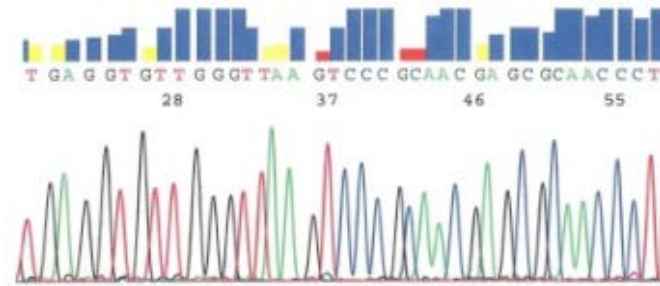
```
@SEQ_ID ←配列ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT ←塩基配列
+
! '*((( (**+))%%++) (%%%) .1***-+*')) **55CCF>>>>>CCCCCCC65 ←クオリティ
```

実習1-1 lessコマンドでEx1_1.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)



```
+  
!''*(((((***+))%%+)) (%%%) .1***-  
+*'')) **55C
```

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)

$QV = 30 \rightarrow p = 0.001$

$QV = 20 \rightarrow p = 0.01$

数値でなく謎の文字が書かれている！

実際のFASTQデータを見ると、

```
@SEQ_ID  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCCTTTGTTCAACTCACAGTTT  
+  
!''*(((((***+))%%+)) (%%%) .1***-+*'')) **55CCF>>>>>CCCCCCC65
```

謎の文字の正体 → “ASCIIコード”を使ってQVを1文字で表したもの

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 (1965年)

0~126の数値に文字を割り当て

A → 65

Apple → 65;112;112;108;101;

FASTQ → ASCIIコードを逆に使って、QV(数値)を文字で表す

65 → A

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:	!	?	@	A	>	=	;	9	7	4	0	,	

文字が各塩基
のQVを表現

QVから文字への変換規則

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

ASCII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	"

・ NGSでは10-30を頻用

$$p = 0.001 \rightarrow QV=30$$

・ 妥協案として特定値を加算してから文字に変換
Phred(QV)値 + X = ASCII値とする

・ X値は現在 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)

・ 変換にはコード表と簡単な計算が必要

実習1-2 Ex1_2.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

Ex1_2.fq

```
@SEQ_ID
GATTGGTGAATT
```

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

QV値 + 33 = ASCII値

ASCII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	文 字	10 進	16 進
NUL	0	00	DLE	16	10	SP	32	20	@	64	40	P	80	50	'	96	60	p	112	70
SOH	1	01	DC1	17	11	!	33	21	A	65	41	Q	81	51	a	97	61	q	113	71
STX	2	02	DC2	18	12	"	34	22	B	66	42	R	82	52	b	98	62	r	114	72
ETX	3	03	DC3	19	13	#	35	23	C	67	43	S	83	53	c	99	63	s	115	73
EOT	4	04	DC4	20	14	\$	36	24	D	68	44	T	84	54	d	100	64	t	116	74
ENQ	5	05	NAK	21	15	%	37	25	E	69	45	U	85	55	e	101	65	u	117	75
ACK	6	06	SYN	22	16	&	38	26	F	70	46	V	86	56	f	102	66	v	118	76
BEL	7	07	ETB	23	17	'	39	27	G	71	47	W	87	57	g	103	67	w	119	77
BS	8	08	CAN	24	18	(40	28	H	72	48	X	88	58	h	104	68	x	120	78
HT	9	09	EM	25	19)	41	29	I	73	49	Y	89	59	i	105	69	y	121	79
LF*	10	0a	SUB	26	1a	*	42	2a	J	74	4a	Z	90	5a	j	106	6a	z	122	7a
VT	11	0b	ESC	27	1b	+	43	2b	K	75	4b	[91	5b	k	107	6b	{	123	7b
FF*	12	0c	FS	28	1c	,	44	2c	L	76	4c	\	92	5c	l	108	6c		124	7c
CR	13	0d	GS	29	1d	-	45	2d	M	77	4d]	93	5d	m	109	6d	}	125	7d
SO	14	0e	RS	30	1e	.	46	2e	N	78	4e	^	94	5e	n	110	6e	~	126	7e
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

解説

@SEQ_ID

GATTGGTGAATT

+

??@A>=;9740,

①p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

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

②各文字をコード表から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;58;59;57;55 52;48;44;

QV値: 30;30;31;32;29;25;26;24;22 19;15;11;

QV値 + 33 = ASCII値

ASCII値 - 33 = QV値

fastqファイルを見る上での注意点

- 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!! ←重要

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

$$\text{Phred(QV)値} + X = \text{ASCII値}$$

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: "@" 配列名
- 2行目: 塩基配列
- 3行目: "+" (配列名)
- 4行目: 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

$$QV値 + 33 = \text{ASCII値}$$

fastqの仲間 [SRA \(Sequence Read Archive\)](#)

公共DBへの登録とダウンロードに使用。
バイナリ化(機械語化)された生シーケンスデータ
fastqに変換可能

後ほど詳しく説明

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

▪ 配列用

FASTA, FASTQ, SRA

▪ アノテーション用

BED, GFF/GTF, WIG

▪ マッピング(アライメント)用

SAM, BAM

BED, GFF/GTF

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 エクソン構造
例	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

<3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

BED (Browser Extensible Data format)

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		領域名	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	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

実習1-3 Ex1_3.bedはヒトゲノム(GRCh37)の一部をbed形式にしたものである
lessコマンドで開いてbed形式を確認しよう

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

省略する場合は“-”や“.”を入れる

セミコロンで区切られた タグ
値の対

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 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

GTF (General Transfer Format)

基本的にGFFと同じだが、仕様をより細かく規定

○規則

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 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1-4

Ex1_4.gtfは 1_3と同じ領域をgtf形式にしたものである。
lessコマンドで開いてgtf形式を確認しよう

注意 GFF/GTFとBEDでは座標の表現が異なる

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

BED : 開始は0based, 終了は 1-based 座標

具体例

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

BED format : 開始 2, 終了 6 (長さは $6-2=4$)

実習1-5

Ex1_3.bedとEx1_4.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 Format)

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
例	GC含量、発現量などを表す

○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
```

定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

NGS基本データフォーマット

数十以上のフォーマットがあります
頻出フォーマットだけを紹介します

■ 配列用

FASTA, FASTQ, SRA

■ アノテーション用

BED, GFF/GTF, WIG

■ マッピング(アライメント)用

SAM, BAM

SAM (Sequence Alignment/Map format)

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
例	SNP、発現量解析への入力データ形式

○ファイル例

ヘッダー部										マッピング結果	
@HD VN:1.5 SO:coordinate											
@SQ SN:ref LN:45											
r001	163	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATACTG	*	
r002	0	ref	9	30	3S6M1P1i4M	*	0	0	AAAGATAAGGATAT	*	
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	83	ref	37	30	9M	=	7	-39	CAGCGGCAT	*	NM:i:1

実習1-6

Ex1_7.samを開きsam形式を確認しよう

○規則

ヘッダー部

@HD VN:1.5 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

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	マッピング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	163	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATAC TG	*	
r002	0	ref	9	30	3S6M1P1i4M	*	0	0	AAAGATAAGGATAT	*	
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	83	ref	37	30	9M	=	7	-39	CAGCGGCAT	*	NM:i:1

ポイント！ “CIGAR” “FLAG”

SAMのポイント1: CIGAR

数字と文字を組み合わせアライメント状況を示す

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	マッピング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	163	ref	5	30	3M2D2M	=	37	39	GCAAG	44>>>	

3M2D2M

塩基数

状況

3塩基一致、2個挿入、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 挿入を入れている

SAMのポイント2: FLAG リードの状態を示す数値

理解すると「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)	意味
1	ペアリードがある
2	両方適切にマップされている
4	自分がマップされていない
8	ペア相手がマップされていない
16	逆鎖にマップされた (配列も逆鎖で表記)
32	ペア相手は逆鎖にマップされた
64	Read1の配列である
128	Read2の配列である
256	Multiple hitでトップヒットでないアライメント
512	マッピングQVが低い

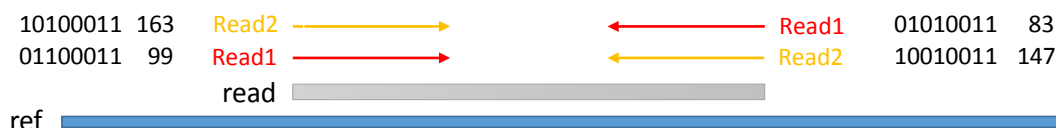
複数の状況に合致する場合は数値を加算

ペアリード, 両方マップされた → $1+2=3$

2進数の個々の有無で評価されている

加算した結果が、ほかの状況と一致しないようになっている

Paired end readでFLAG値の組み合わせを見てみる



	Read1の配列である	Read2の配列である	逆鎖にマップされた	ペア相手がマップされていない	自分がマップされていない	両方適切にマップされている	ペアリードがある	2進数表記	10進数表記
通常のpaired end seqで consistently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	11111111 255
片方しかアラインしていない場合	0	1	0	0	1	0	0	1	01010011 83
	0	1	0	1	0	0	0	1	01100011 99
	1	0	0	1	0	0	1	1	10010011 147
	1	0	1	0	0	0	1	1	10100011 163
どっちもアラインしていない場合	0	1	0	0	1	1	0	1	01001001 73
	0	1	0	1	1	0	0	1	01011001 89
	0	1	0	0	0	1	0	1	01000101 69
	0	1	1	0	0	1	0	1	01100101 101
	1	0	0	0	1	0	0	1	10001001 137
	1	0	0	1	1	0	0	1	10011001 153
	1	0	0	0	0	1	0	1	10000101 133
	1	0	1	0	0	1	0	1	10100101 165
	0	1	0	0	1	1	0	1	01001101 77
	1	0	0	0	1	1	0	1	10001101 141

自動でflagを計算してくれるサイトがある

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

This utility explains SAM flags in plain English.
It also allows switching easily from a read to its mate.

Flag:

Explanation:

- ☐ 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:

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:	FLAG値	→ リードのマップ状況
	CIGAR値	→ リードのアライメント状況

触れなかった重要点

ペアフラグメント部分の“長さ”列 → フラグメント間距離 + 両リード長

SAM formatの詳細な仕様書

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

BAM

■ 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

後ほど詳しく説明

フォーマット各論まとめ

	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to毎の品質情報	マッピング情報 ペアの状況, 塩基配列
例	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	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 エクソン構造			ゲノム上の座標に対する "数値"情報
例	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	2つの形式 VariableStep/FixedStep

概要

序論

- データフォーマットとは？
- フォーマットを学ぶ理由
- 効率よい学習のポイント

NGS基本データフォーマット

- FASTA FASTQ SRA
- BED GFF/GTF WIG
- SAM/BAM

NGS基本ツール

- SRAtoolkit
- SAMtools
- IGV

NGS基本ツール

実践データ用の専用ソフトの使い方を紹介

実践データはコマンドラインを使った専用ソフトでの操作となる
データ量が多すぎて、マニュアルの編集は不可能 普通のソフトでも困難

各NGSフォーマットを利用したNGS基本ツール

FASTQ : SRAtoolkit, cutadapt, fastQC

BED GFF/GTF : BEDtools

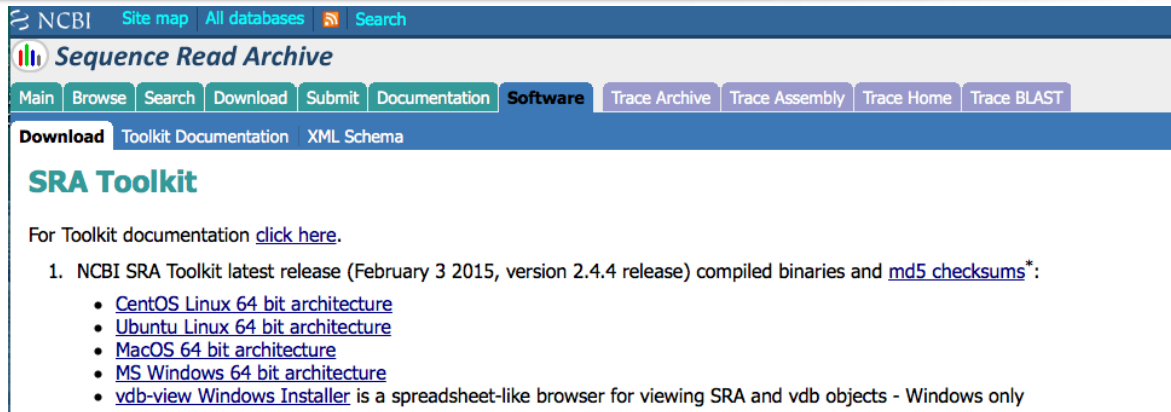
SAM/BAM : SAMtools, Picard

可視化ツール : IGV, JBrowse

NGS解析に特に有用な

SRAtoolkit, SAMtools, IGVに注目して解説します

SRAtoolkit



機能

SRA → fastq の変換

SRA形式の利点

fastqに比べて1/10のファイルサイズ

→保存が楽だし、ダウンロードも早くなる

fastq以外のデータ形式を使うシーケンサーのデータもSRA形式に変換可能

→共通のDB構造を持たせることができ管理が楽

基本事項 toolの呼び出し(SRAtoolkitを例として)

実習2-1 `./sratoolkit/fastq-dump` と打ち

SRAtoolkit の `fastq-dump` を呼び出そう

→`fastq-dump`が呼び出され、使い方が表示された

Usage:

`./sratoolkit/fastq-dump [options] <path> [<path>...]`

`./sratoolkit/fastq-dump [options] <accession>`

Use option `--help` for more information

`./sratoolkit/fastq-dump` : 2.5.2

基本事項 Linuxのマニュアルの見方

[] は、省略可能なオプション (とその引数)
| は選択肢で、列挙されたいずれかを排他的に選択する

Usage:

`./sratoolkit/fastq-dump [options] <path> [<path>...]`

↑
省略可能な
オプション

↗ ↘
変換したいSRAファイル
へのパスを記述

実習2-2

1) Ex2_1.sraをfastq形式に変換しよう。

例) `./sratoolkit/fastq-dump Ex2_1.sra`

2) lessでfastqに変換された事を確認しよう。

例) `less Ex2_1.fastq`

3) lsコマンドでファイルサイズを確認しよう。

例) `ls -l`

SAMtools

Samtools

Home

Download ▾

Workflows ▾

Documentation ▾

Support ▾

Samtools

Samtools is a suite of programs for interacting with high-throughput sequencing data. It consists of three separate repositories:

<http://www.htslib.org>

機能

1, 形式の変換・整理

- 1-1, SAM (人間用) ⇔ BAM (コンピュータ用) の変換
- 1-2, 検索や可視化ソフト用の索引ファイル作成

2, データ抽出

- 2-1, 特定のリードの選出
- 2-2, 統計情報収集 (発現量解析)

SAMtoolsの呼び出し

実習2-3 ./samtools と打ち SAMtools を呼び出そう

→SAMtoolsが呼び出され、使い方が表示された
samtools の後にさらに各command名を打って使う

```
Program: samtools (Tools for alignments in the SAM format)
Version: 1.0 (using htslib 1.0)

Usage:  samtools <command> [options]

Commands:
  -- indexing
    faidx      index/extract FASTA
    index      index alignment
  -- editing
    calmd      recalculate MD/NM tags and '=' bases
    fixmate    fix mate information
    reheader   replace BAM header
    rmdup      remove PCR duplicates
    targetcut  cut fosmid regions (for fosmid pool only)
  -- file operations
    bamshuf    shuffle and group alignments by name
    cat        concatenate BAMs
    merge      merge sorted alignments
    mpileup    multi-way pileup
    sort       sort alignment file
    split      splits a file by read group
    bam2fq     converts a BAM to a FASTQ
  -- stats
    bedcov     read depth per BED region
    depth      compute the depth
    flagstat   simple stats
    idxstats   BAM index stats
    phase      phase heterozygotes
    stats      generate stats (former bamcheck)
  -- viewing
    flags      explain BAM flags
    tview     text alignment viewer
    view       SAM<->BAM<->CRAM conversion
```

検索高速化 → faidx, index

統計情報 → bedcov, depth, flagstat, idxstats, phase, stats

**SAM ⇄ BAM の変換
リードの選出** → bam2fq, view

SAMtools SAM/BAM変換

実習2-4 ./samtools view と打ちview機能の使い方情報を呼び出そう

```
Usage:  samtools view [options] <in.bam>|<in.sam>|<in.cram>
[region ...]

Options: -b      output BAM
         -C      output CRAM (requires -T)
         -l      use fast BAM compression (implies -b)
         -u      uncompressed BAM output (implies -b)
         -h      include header in SAM output
         -H      print SAM header only (no alignments)
         -c      print only the count of matching records
         -o FILE output file name [stdout]
         -U FILE output reads not selected by filters to FILE

[null]
         -t FILE FILE listing reference names and lengths (see
long help) [null]
         -T FILE reference sequence FASTA FILE [null]
         -L FILE only include reads overlapping this BED FILE
[null]
         -r STR  only include reads in read group STR [null]
```

bam → sam 変換のしかた

bamファイルを指定して実行(リターン) 出力ファイルは > で指定
例) samtools view ./in.bam > out.sam
(ヘッダーも出力するときは -h オプションをつける)

sam → bam 変換のしかた

-b で出力をbamにしてほしいことを示す
例) samtools view -b ./in.sam > ./out.bam

SAMtools SAM/BAM変換

実習2-5 sam/bamの違いを実感しよう

- 1) lessコマンドを使ってEx1_7.bamの内容を見てみよう
 - “Ex1_7.bam” may be a binary file. See it anyway? と表示される
 - y と打つ (yes、読みます)
 - 読めない文字が表示される(バイナリ化されている)
- 2) samtools viewを使ってEx1_7.bamの内容をsamに変えて再度中身を確認しよう
例) samtools view Ex1_7.bam > Ex1_7_new.sam
- 3) ls コマンドでEx1_7.sam, Ex1_7.bamのサイズを比較しよう
 - ls -l (ls = ディレクトリ中のファイル情報を表示、-l = long 長い説明で)
 - bamの方がサイズが小さいはず(情報を圧縮できている)

BAM化のメリット: ストレージ領域の節約 データ送付の高速化 処理の高速化
SAM化のメリット: 人が見て理解できる

SAMtools 検索・indexファイル作成1

データをあらかじめ整理しておき検索や可視化を容易にする
以下の2ステップを続けて行う

- Step 1 samtools sort をつかってリードの順番を整理する
Step 2 samtools index で索引を作成し効率よく検索できるようにする

実習2-6 samtools sort を呼び出してみよう

```
Usage: samtools sort [options...] [in.bam] ← input はbam形式
Options:
  -l INT      Set compression level, from 0 to 9 [-1]
  -m INT      Set maximum memory per thread; suffix K/M/G recognized [768M]
  -n          Sort by read name
  -o FILE     Write final output to FILE rather than standard output
  -O FORMAT   Write output as FORMAT ('sam'/'bam'/'cram') (either -O or
  -T PREFIX   Write temporary files to PREFIX.nnnn.bam      -T is required)
```

出力フォーマットを指定

実習2-7 samtools index を呼び出してみよう

```
Usage: samtools index [-bc] [-m INT] <in.bam> [out.index]
Options:
  -b          Generate BAI-format index for BAM files [default]
  -c          Generate CSI-format index for BAM files
  -m INT      Set minimum interval size for CSI indices to 2^INT [14]
```

SAMtools 検索・indexファイル作成2

Step 1 samtools sort をつかってリードの順番を整理する

実習2-8 samtools sort を使ってEx1_7.bamの内容をソートしよう

(出力ファイル名はEx1_7_s.bam)

例) samtools sort -O bam -T test Ex1_7.bam > Ex1_7_s.bam

実習2-9 sortの効果を実感しよう

1) samtools viewを使ってEx1_7_s.bamをsamに変えて保存しよう

例) samtools view Ex1_7_s.bam > Ex1_7_s.sam

samtools view -h Ex1_7_s.bam > Ex1_7_s_h.sam

2) テキストの頭や末尾を表示するhead や tail コマンドで結果を見てみよう

例) head -n 10 Ex1_7.sam

head -n 10 Ex1_7_s.sam

head -n 10 Ex1_7_s_h.sam

tail -n 10 Ex1_7.sam

tail -n 10 Ex1_7_s.sam

Ex1_7_s.sam ではリファレンス配列上の位置順にリードの順番が整理されている。

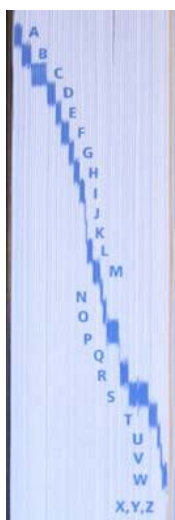
SAMtools 検索・indexファイル作成3

Step 2 samtools index で索引を作成し、効率よく検索できるようにする

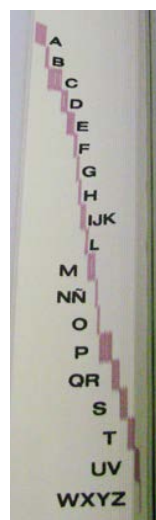
実習2-10 samtools index を使ってEx1_7_s.bamのインデックスファイル
(Ex1_7_s.bam.bai)を作ろう

例) samtools index Ex1_7_s.bam

<インデックスファイルとは？>



英語辞書



スペイン語辞書

辞書の小口印刷のような物

索引をつけることで直接見たい場所(の近く)から
検索を始められる

データの偏りによって適切なインデックス区分は変わる
(英語→Wで始まる語は多いので独自の区分を与える。西語→WとXYZは同じ区分)

↓
各ファイルにあわせて個別に
インデックスを作る必要がある

SAMtools

Samtools

Home

Download ▾

Workflows ▾

Documentation ▾

Support ▾

Samtools

Samtools is a suite of programs for interacting with high-throughput sequencing data. It consists of three separate repositories:

<http://www.htslib.org>

機能

1, 形式の変換・整理

- 1-1, SAM (人間用) ⇔ BAM (コンピュータ用) の変換
- 1-2, 検索や可視化ソフト用の索引ファイル作成

2, データ抽出

- 2-1, 特定のリードの選出
- 2-2, 統計情報収集(発現量解析!)

SAMtools 2, データ抽出

2-1 特定のリードの選出

方法1) samtools view を使ったマップ位置による選出

indexファイルが作成されていれば、viewをつかって
特定のリファレンス部分にマップされたリードを選び出せる

例) ./samtools view Ex1_7_s.bam cp1:1000-2000

リファレンス名 位置

方法2) samtools view -f/-F を使ったマップ状況 (flag値)による選出
-f/-F オプションを使うことで、特定のマップ状況にあるリードを
選び出すことができる。

-f 該当するflag値をもつリードを抽出

-F 該当するflag値をもつリード”以外”を抽出

例) ./samtools view -f 83 Ex1_7_s.bam

SAMtools 2, データ抽出

2-1 特定のリードの選出

実習2-11

Ex1_7_s.bamから以下の遺伝子にマップされたリードを取り出し、数を比較しよう

染色体名

遺伝子名

位置

cp1

16SrRNA

27762-29094

cp1

rbcL

60184-61611

cp1

rpoC

68739-75880

例) ./samtools view Ex1_7_s.bam cp1:1000-2000 |wc -l

↑
行数を数えるコマンド

実習2-12

前頁の例(-f 83) はどんなペアリード関係を指定しているのだろうか？
例) ./samtools flags 83

実習2-13

ペアが両方マップされていないリードを抽出し、数を調べよう
例) ./samtools view -f 12 Ex1_7_s.bam |wc -l

マッピングソフトによって
区分が異なる場合もあるので注意

アライメント結果を目視しながら
確認してから使用すること

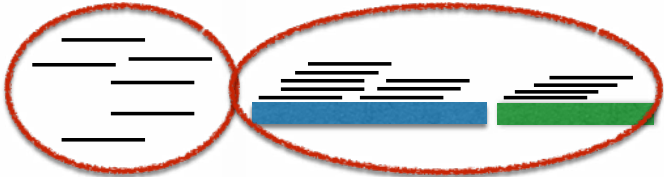
数値 (16進数)	数値 (10進数)	意味
1	1	ペアリードがある
2	2	両方適切にマップされている
4	4	自分がマップされていない
8	8	ペア相手がマップされていない
10	16	逆鎖にマップされた (配列も逆鎖で表記)
20	32	ペア相手は逆鎖にマップされた
40	64	ペアリードの 1 番目である
80	128	ペアリードの 2 番目である
100	256	Multiple hitでトップヒットでないアライメント
200	512	マッピングQVが低い

SAMtools 2, データ抽出

2-2 統計情報収集

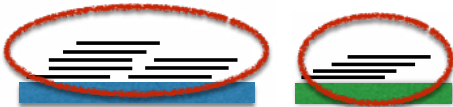
方法 1 samtools flagstat を使ってマッピング結果全体の簡単な情報を得る

n本マップされm本マップされなかった



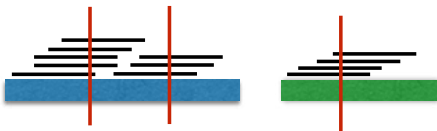
方法 2 samtools idxstats でRef配列毎にマップされたリード数を得る

RefA にM本 RefBにN本マップされた



方法 3 samtools depth で塩基毎に深度 (読まれた回数) を得る

RefAのX塩基目はY回読まれた



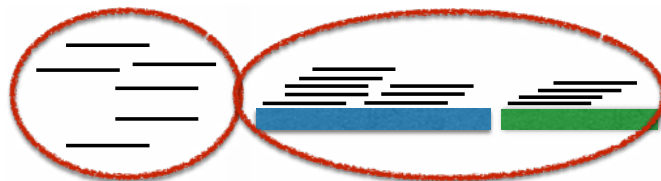
SAMtools 2, データ抽出 2-2 統計情報収集

方法1 samtools flagstat を使ってマッピング結果の簡単な統計情報を得る

Usage: samtools flagstat <in.bam>

例) samtools flagstat Ex1_7_s.bam

全リード数 → 200000 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 duplicates
マップされたリード → 48481 + 0 mapped (24.24%:nan%)
200000 + 0 paired in sequencing
ペアであるリード数 → 100000 + 0 read1
100000 + 0 read2
適切にペアになったリード(向きなど) → 47050 + 0 properly paired (23.52%:nan%)
47846 + 0 with itself and mate mapped
635 + 0 singletons (0.32%:nan%)
ペアの両方がマップされたリード → 0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
自分しかマップされていないリード
異なったリファレンス配列にマップ



各行の詳細な意味は使ったマッピングソフトの出力形式に影響される

SAMtools 2, データ抽出 2-2 統計情報収集

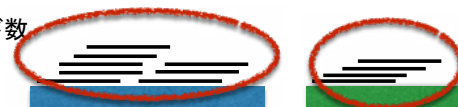
方法2 samtools idxstats でRef配列毎にマップされたリード数を得る

Usage: samtools idxstats <in.bam>

例) samtools idxstats Ex1_7_s.bam

Ref. 名	Ref. 配列長	マップされたリード数	マップされなかったリード数
cp1	86483	48481	635
*	0	0	150884

ペアなのにマップされなかったリード数

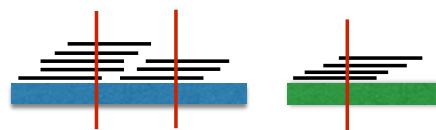


方法3 samtools depth を使って深度(読まれた回数)の統計情報を得る

Usage: samtools depth [options] in1.bam [in2.bam [...]]

例) samtools depth Ex1_7_s.bam

Ref. 名	位置	深度 読まれた回数
Cp1	3313	83
Cp1	3314	120
Cp1	3315	144
Cp1	3316	148




SAMtoolsのまとめ

マッピングデータの整理から、実験結果解析まで
様々な場面で活躍する必須ツール

samtools

view ← SAM/BAM変換 特定リード抽出

sort
index  ← 検索高速化

flagstat
idxstats
depth  ← 統計情報抽出

その他のSAMtoolsの機能

mpileup	SNP検出などで活躍
tview	簡便なアライメントビューワー
merge	複数のBAMファイルを結合する

まとめ

最初は意味がわからないフォーマットでも、読み解けば
日々の実験で接する生物学的情報を表しているにすぎない

解説しなかったフォーマット、ツールも類似の構造をもつ
仕様書をよめば恐るるにたらず！

“Practice makes perfect！”

ぜひ自分のデータも読み解いてみてください。