基生研ゲノムインフォマティックス・トレーニングコース　2017春
RNA-seq入門　- NGSの基礎から*de novo*解析まで -
入門編
2017.02.23-2017.02.24

NGS基本データフォーマットとクオリティコントロール

基礎生物学研究所
生物機能解析センター
山口勝司

NGS基本データフォーマット

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

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

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
あなた　：了解です。fastaで送りますね

← 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

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
あなた　：系統解析をするソフトなんだな

← 入力と出力の形式から行った解析がわかる

研究目的にあわせた解析に必要だから

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くる！

- ＄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
TGCACCAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTTGAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
>gi|31342400 Bos taurus crystallin, gamma S (CRYGS), mRNA
TGCACCAAACATGTCTAAAGCTGGAACCAAAATTACTTTCTTTGAAGACAAAA
ACTTTCAAGGCCGCCACTATGACAGCGATTGCGACTGTGCAGATTTCCACATG
TACCTGAGCCGCTGCAACTCCATCAGAGTGGAAGGAGGCACCTGGGCTGTGTA
TGAAAGGCCCA
```

←タイトル行

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

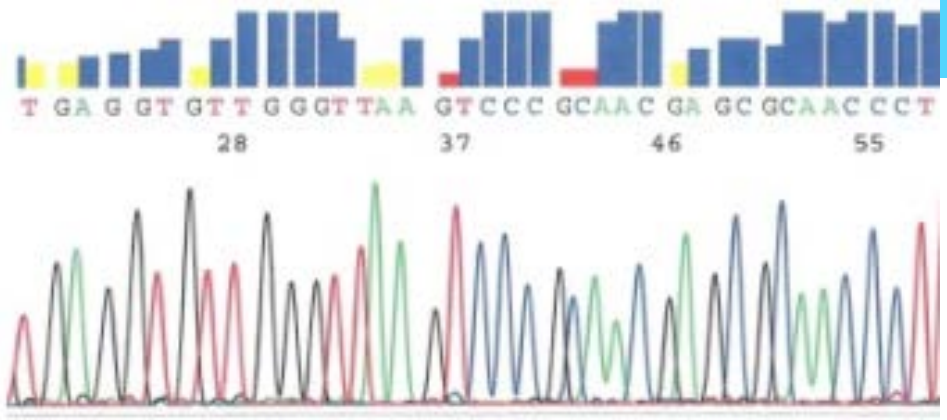
```
@SEQ_ID ←配列ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT ←塩基配列
+ ←配列ID（省略）
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65 ←クオリティ
```

実習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 → p = 0.001

QV = 20 → p = 0.01

数値でなく謎の文字が書かれている！

実際のFASTQデータをみると、

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).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 → 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	文字	10進	16進
NUL	0	00	DLE	16	10	SP	32	20	0	48	30	@	64	40	P	80	50	`	96	60	p	112	70
SOH	1	01	DC1	17	11	!	33	21	1	49	31	A	65	41	Q	81	51	a	97	61	q	113	71
STX	2	02	DC2	18	12	"	34	22	2	50	32	B	66	42	R	82	52	b	98	62	r	114	72
ETX	3	03	DC3	19	13	#	35	23	3	51	33	C	67	43	S	83	53	c	99	63	s	115	73
EOT	4	04	DC4	20	14	$	36	24	4	52	34	D	68	44	T	84	54	d	100	64	t	116	74
ENQ	5	05	NAK	21	15	%	37	25	5	53	35	E	69	45	U	85	55	e	101	65	u	117	75
ACK	6	06	SYN	22	16	&	38	26	6	54	36	F	70	46	V	86	56	f	102	66	v	118	76
BEL	7	07	ETB	23	17	'	39	27	7	55	37	G	71	47	W	87	57	g	103	67	w	119	77
BS	8	08	CAN	24	18	(	40	28	8	56	38	H	72	48	X	88	58	h	104	68	x	120	78
HT	9	09	EM	25	19	)	41	29	9	57	39	I	73	49	Y	89	59	i	105	69	y	121	79
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Z	90	5a	j	106	6a	z	122	7a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	K	75	4b	[	91	5b	k	107	6b	{	123	7b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	\¥	92	5c	l	108	6c	|	124	7c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	M	77	4d	]	93	5d	m	109	6d	}	125	7d
SO	14	0e	RS	30	1e	.	46	2e	>	62	3e	N	78	4e	^	94	5e	n	110	6e	~	126	7e
SI	15	0f	US	31	1f	/	47	2f	?	63	3f	O	79	4f	_	95	5f	o	111	6f	DEL	127	7f

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

* LFはNL、FFはNPと呼ばれることもある。
* 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

$$QV = -10 \log_{10} p$$
$$= -10 \log_{10} 0.01$$
$$= -10 (-2)$$
$$= 20$$

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

Phred(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 ＝ ASCII値

fastqの仲間　SRA（Sequence Read Archve）

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

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

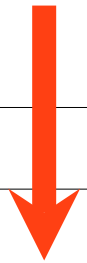
ゲノムアノテーションの標準的形式

○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



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

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

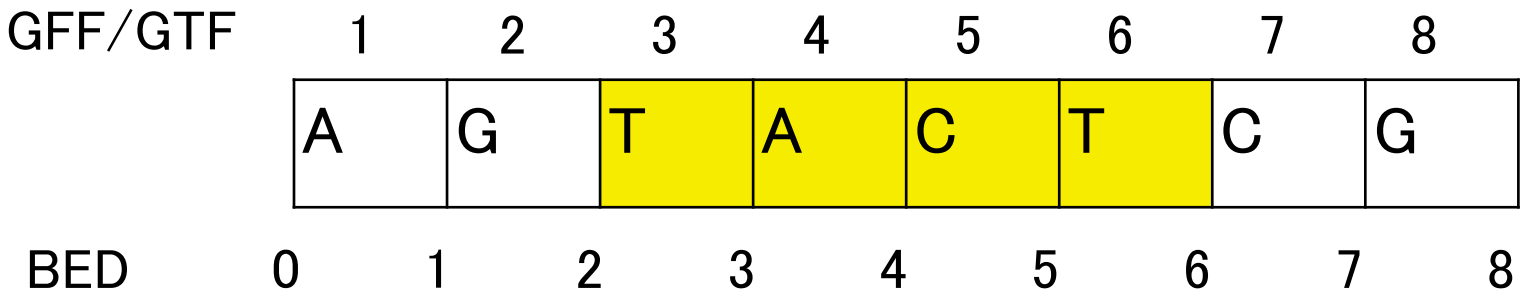
実習4　Ex1_4.gtfは 1_3と同じ領域をgtf形式にしたものである。
lessコマンドで開いてgtf形式を確認しよう

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Obased, 終了は 1-based 座標

具体例



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

〇規則　　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

```
@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
```

ヘッダー部

マッピング結果

実習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	TTAGATAAAGGATACTG	*	
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

	Read2の配列である	Read1の配列である	ペア相手は逆鎖にマップされた	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	samファイルの記載は10進数表記
	1	1	1	1	1	1	1	1	11111111	255
通常のpaired end seqで	0	1	0	1	0	0	1	1	01010011	83
consistentにアラインしていれば	0	1	1	0	0	0	1	1	01100011	99
この4通りになる	1	0	0	1	0	0	1	1	10010011	147
	1	0	1	0	0	0	1	1	10100011	163
片方しかアラインしてない場合	0	1	0	0	1	0	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
どっちもアラインしていない場合	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: [ ] Explain

Switch to mate

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

クオリティーコントロール

NGSデータ解析におけるクオリティーコントロールの重要性

- 得られるdataのクオリティーは通常同一にはならない
  機器の調子　エアーかみ
  作製ライブラリーのサイズ分布
  機器間の性能差

- 作製したライブラリーに問題はなかったか
  コンタミ配列の有無
  短いライブラリーほどクラスター増幅されやすい
  GC率に偏りがあるものは増幅されにくい
  PCR増幅の適性度

シーケンスのクオリティーcheckツール　FASTQC

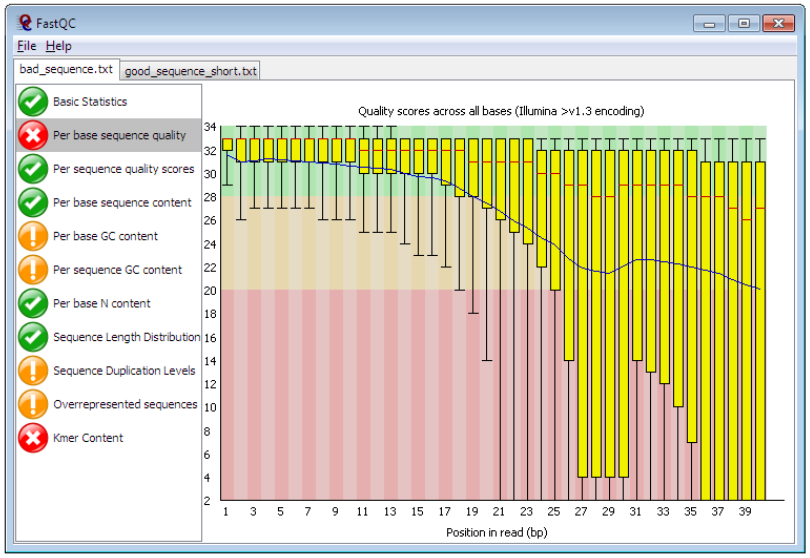
Babraham Bioinformatics

About | People | Services | Projects | Training | Publications

FastQC

Function	A quality control tool for high throughput sequence data.
Language	Java
Requirements	A suitable Java Runtime Environment The Picard BAM/SAM Libraries (included in download)
Code Maturity	Stable. Mature code, but feedback is appreciated.
Code Released	Yes, under GPL v3 or later.
Initial Contact	Simon Andrews

Download Now



Documentation

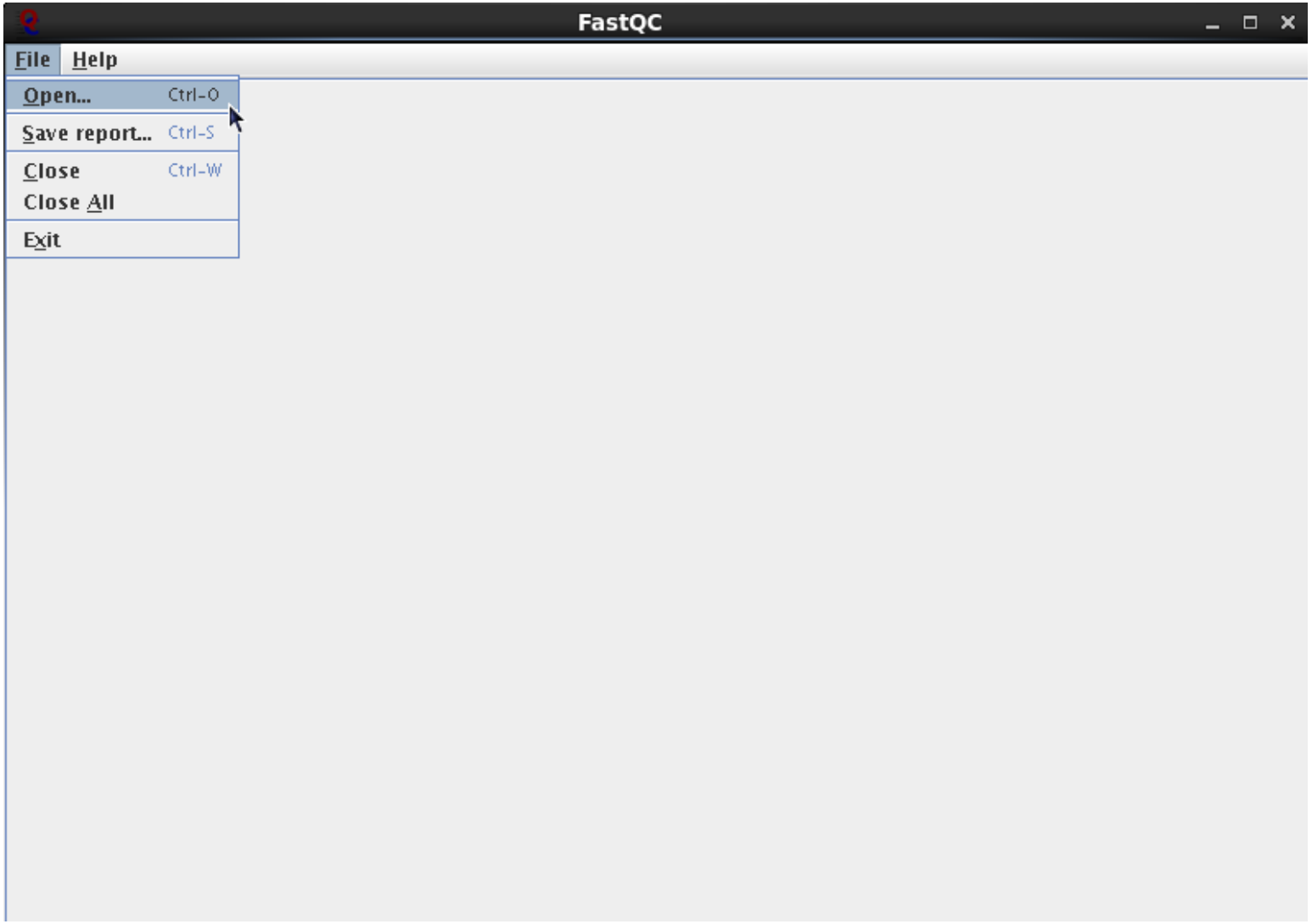
A copy of the FastQC documentation is available for you to try before you buy (well download..).

Example Reports

- Good Illumina Data
- Bad Illumina Data
- Adapter dimer contaminated run
- Small RNA with read-through adapter
- Reduced Representation BS-Seq
- PacBio
- 454

FASTQC使用法

GUI　　java jdkを予めインストールしておく必要がある。



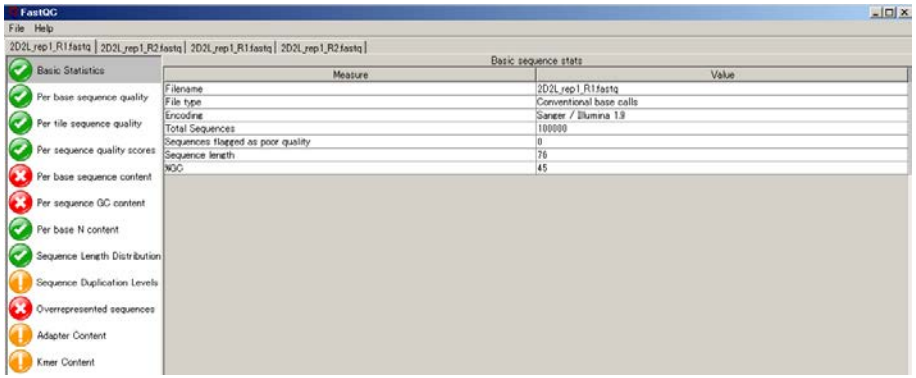
CUI

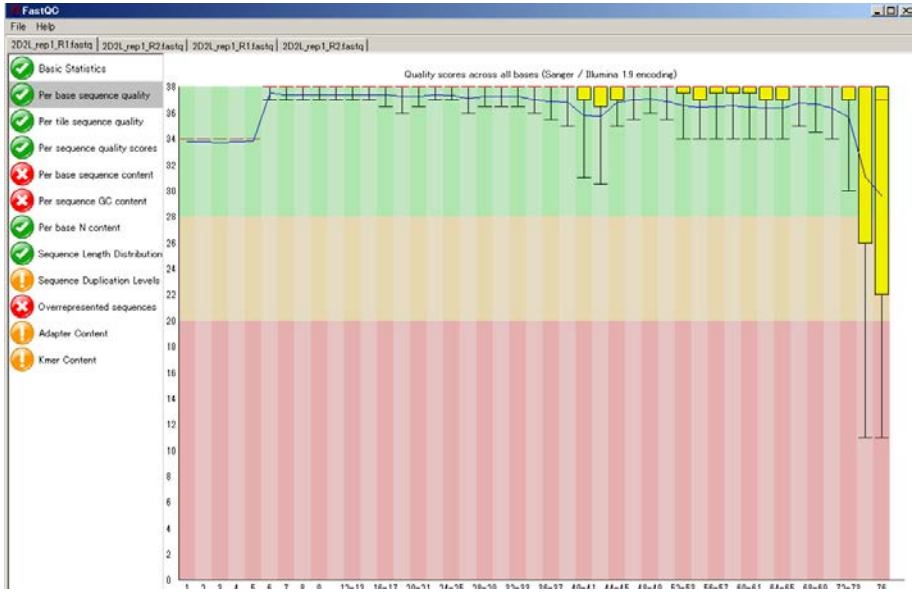
```
$ fastqc -h

FastQC - A high throughput sequence QC analysis tool

SYNOPSIS

        fastqc seqfile1 seqfile2 .. seqfileN

fastqc [-o output dir] [--(no)extract] [-f fastq|bam|sam]
           [-c contaminant file] seqfile1 .. seqfileN
```

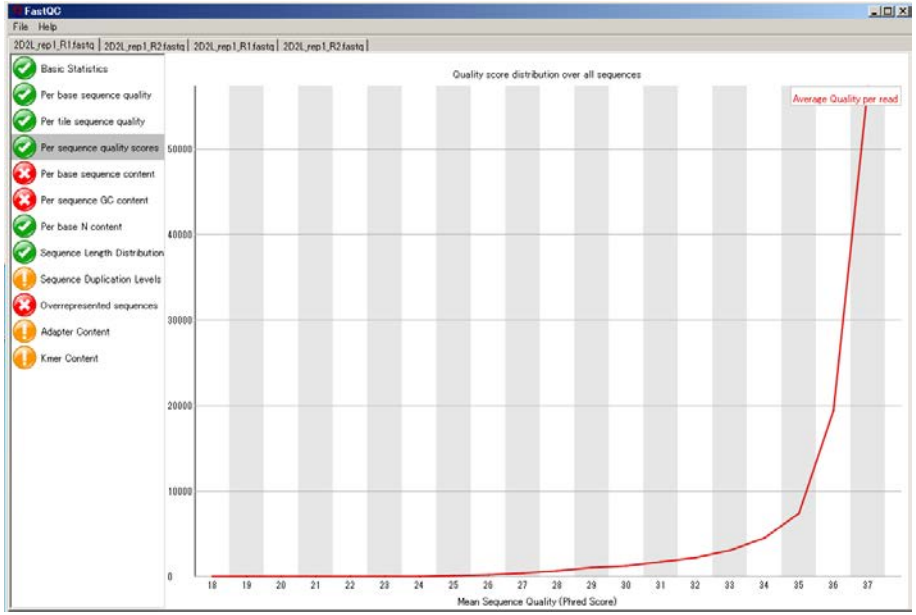
gzファイルなら
--extract

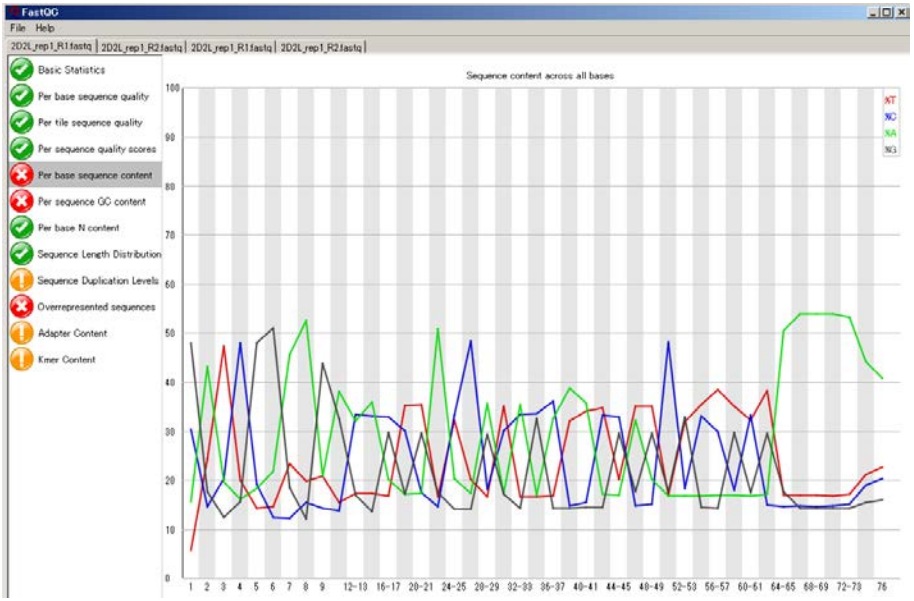
Linux版を利用

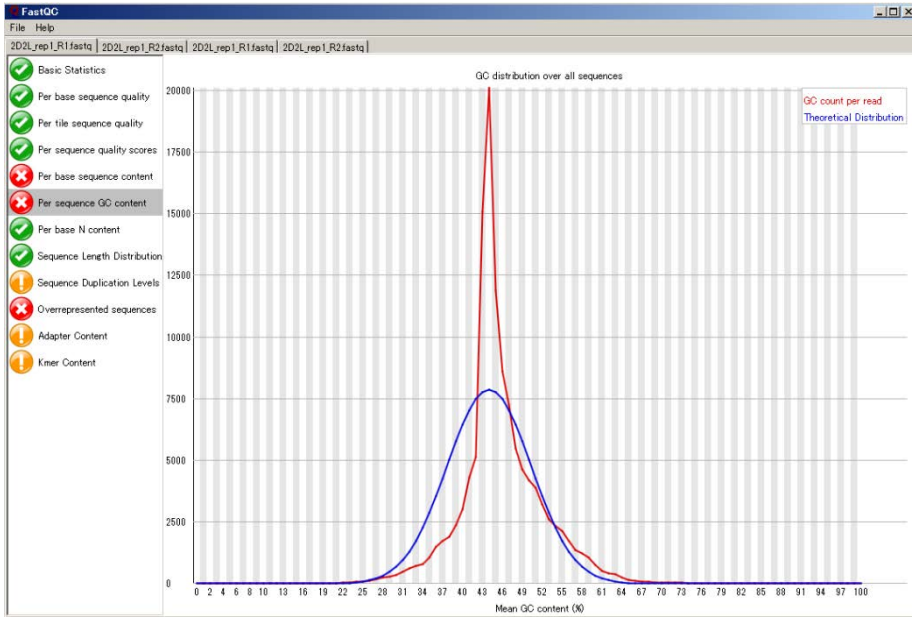
概説

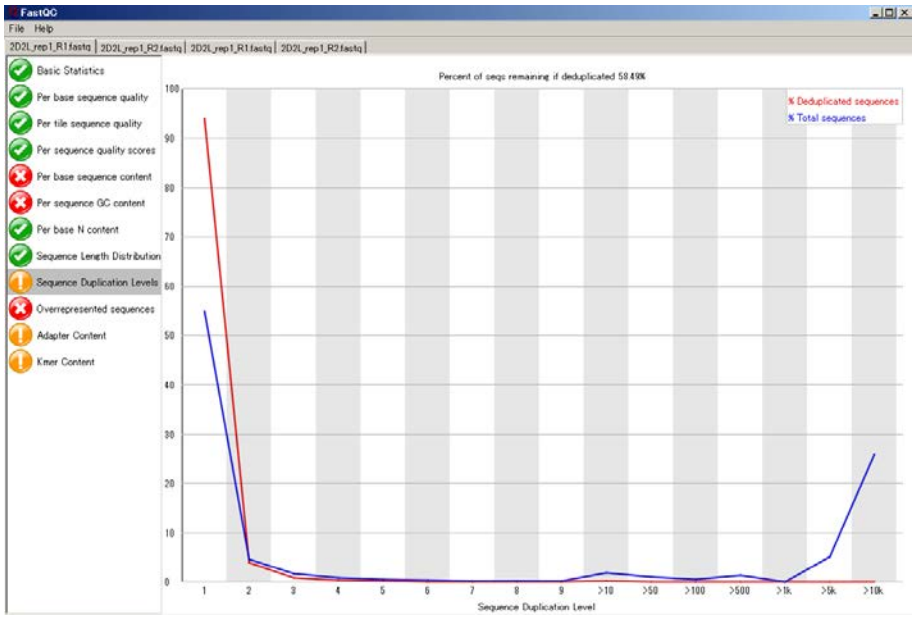

FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Basic sequence stats
Measure
Value
Filename
2D2L_rep1_R1.fastq
File type
Conventional base calls
Encoding
Sanger / Illumina 1.9
Total Sequences
100000
Sequences flagged as poor quality
0
Sequence length
76
%GC
45

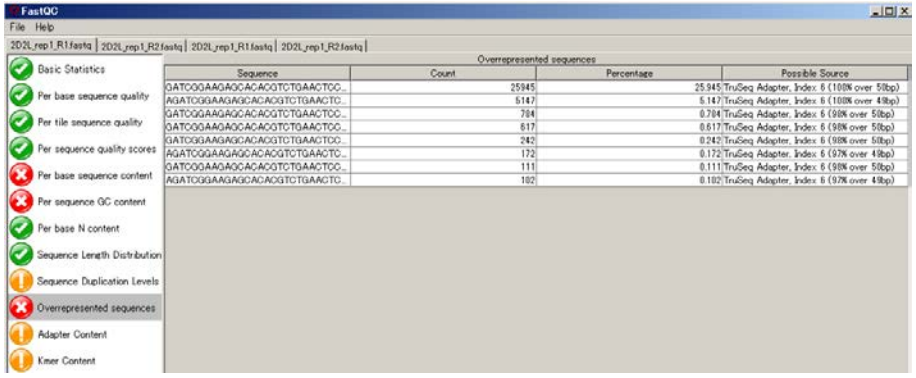

FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Quality scores across all bases (Sanger / Illumina 1.9 encoding)

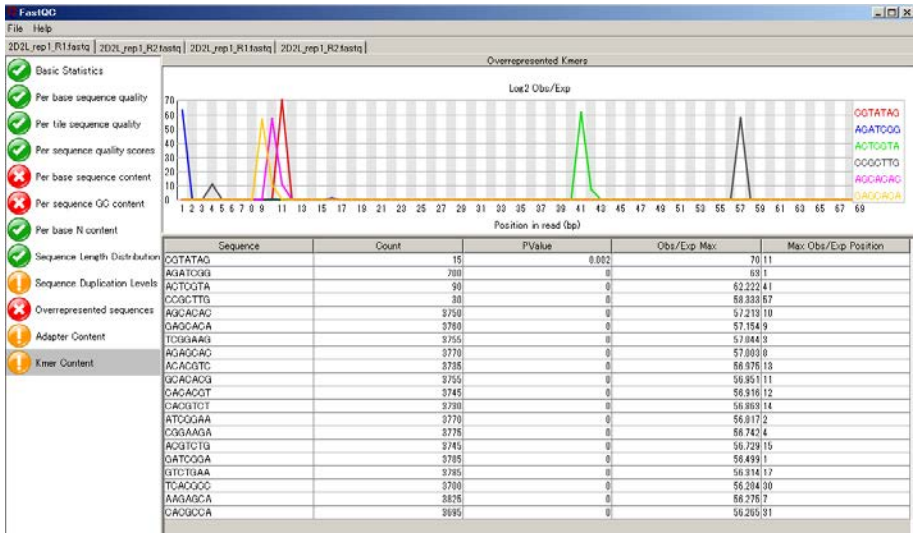

FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Quality score distribution over all sequences
Average Quality per read
Mean Sequence Quality (Phred Score)


FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Sequence content across all bases
%T
%C
%A
%G


FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
GC distribution over all sequences
GC count per read
Theoretical Distribution
Mean GC content (%)

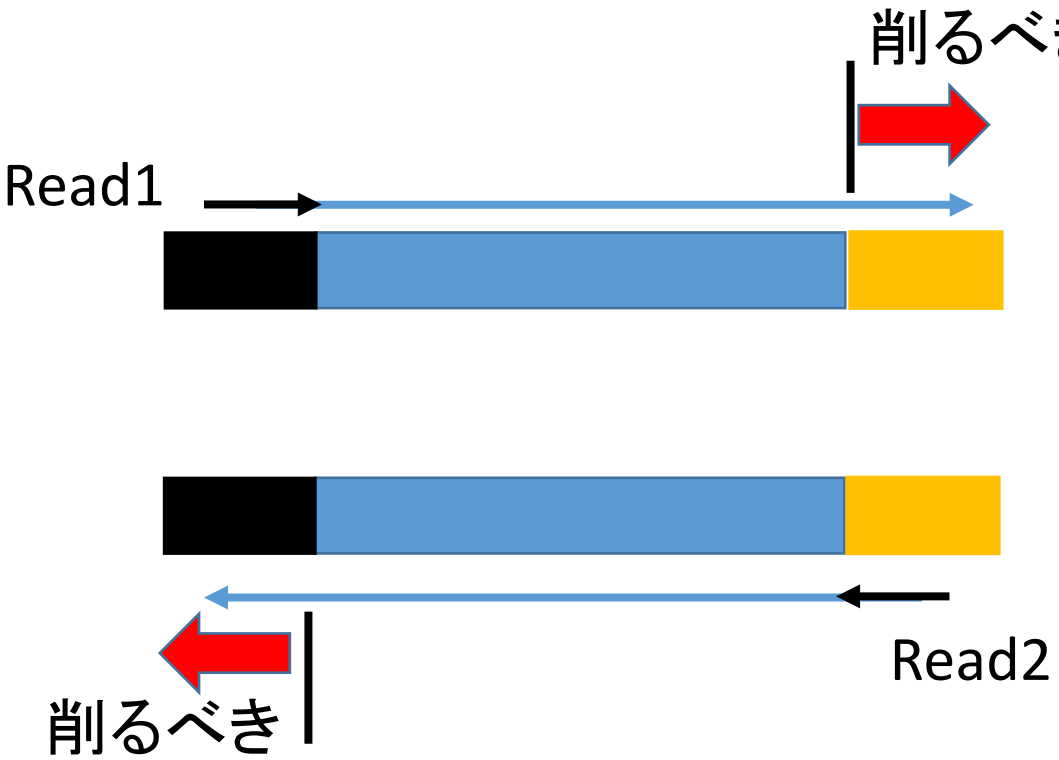

FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Percent of seqs remaining if deduplicated 59.49%
% Deduplicated sequences
% Total sequences
Sequence Duplication Level

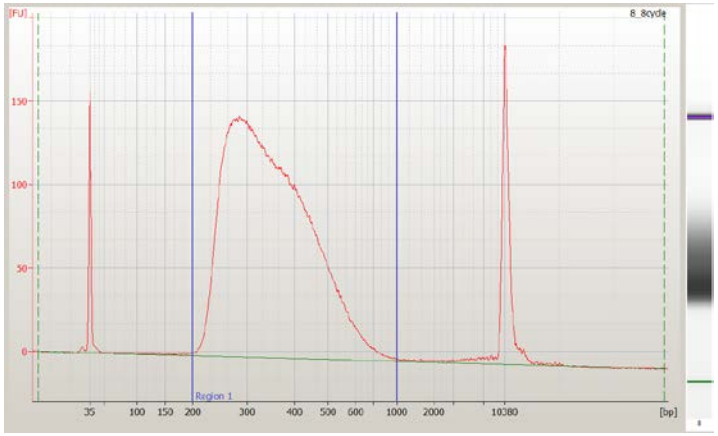

FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Overrepresented sequences
Sequence
Count
Percentage
Possible Source
GATCGGAAGAGCACACGTCTGAACTCC... 25945 25.945 TruSeq Adapter, Index: 6 (100% over 50bp)
AGATCGGAAGAGCACACGTCTGAACTC... 5147 5.147 TruSeq Adapter, Index: 6 (100% over 49bp)
GATCGGAAGAGCACACGTCTGAACTCC... 784 0.784 TruSeq Adapter, Index: 6 (98% over 50bp)
GATCGGAAGAGCACACGTCTGAACTCC... 617 0.617 TruSeq Adapter, Index: 6 (98% over 50bp)
GATCGGAAGAGCACACGTCTGAACTCC... 242 0.242 TruSeq Adapter, Index: 6 (98% over 50bp)
AGATCGGAAGAGCACACGTCTGAACTC... 172 0.172 TruSeq Adapter, Index: 6 (97% over 49bp)
GATCGGAAGAGCACACGTCTGAACTCC... 111 0.111 TruSeq Adapter, Index: 6 (98% over 50bp)
AGATCGGAAGAGCACACGTCTGAACTC... 102 0.102 TruSeq Adapter, Index: 6 (97% over 49bp)


FastQC
File Help
2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq | 2D2L_rep1_R1.fastq | 2D2L_rep1_R2.fastq
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Kmer Content
Overrepresented Kmers
Log2 Obs/Exp
CGTATAG
AGATCGG
ACTCGTA
CCGCTTG
AGCACAC
CACACAG
Position in read (bp)
Sequence
Count
PValue
Obs/Exp Max
Max Obs/Exp Position
CGTATAG 15 0.002 70 11
AGATCGG 788 0 69.1
ACTCGTA 90 0 62.222 41
CCGCTTG 38 0 58.333 57
AGCACAC 3750 0 57.219 10
CACACAG 3760 0 57.154 9
TCGGAAG 3755 0 57.044 3
AGAGCAC 3770 0 57.003 8
ACACGTC 3735 0 56.975 13
GCACACG 3755 0 56.951 11
CACACGT 3745 0 56.916 12
CACGTCT 3730 0 56.869 14
ATCGGAA 3770 0 56.817 2
CGGAAGA 3775 0 56.742 4
ACGTCTG 3745 0 56.729 15
GATCGGA 3765 0 56.499 1
GTCTGAA 3785 0 56.314 17
TCACGCC 3760 0 56.204 30
AAGAGCA 3825 0 56.176 7
CACGCCA 3695 0 56.065 31

RNA-SeqにおけるPreprocessingの必要性

RNA-Seq解析においてmappingはglobal alignmentが
用いられることが多い。
・Global matchにおいて末端に余計な配列があるとmapしない





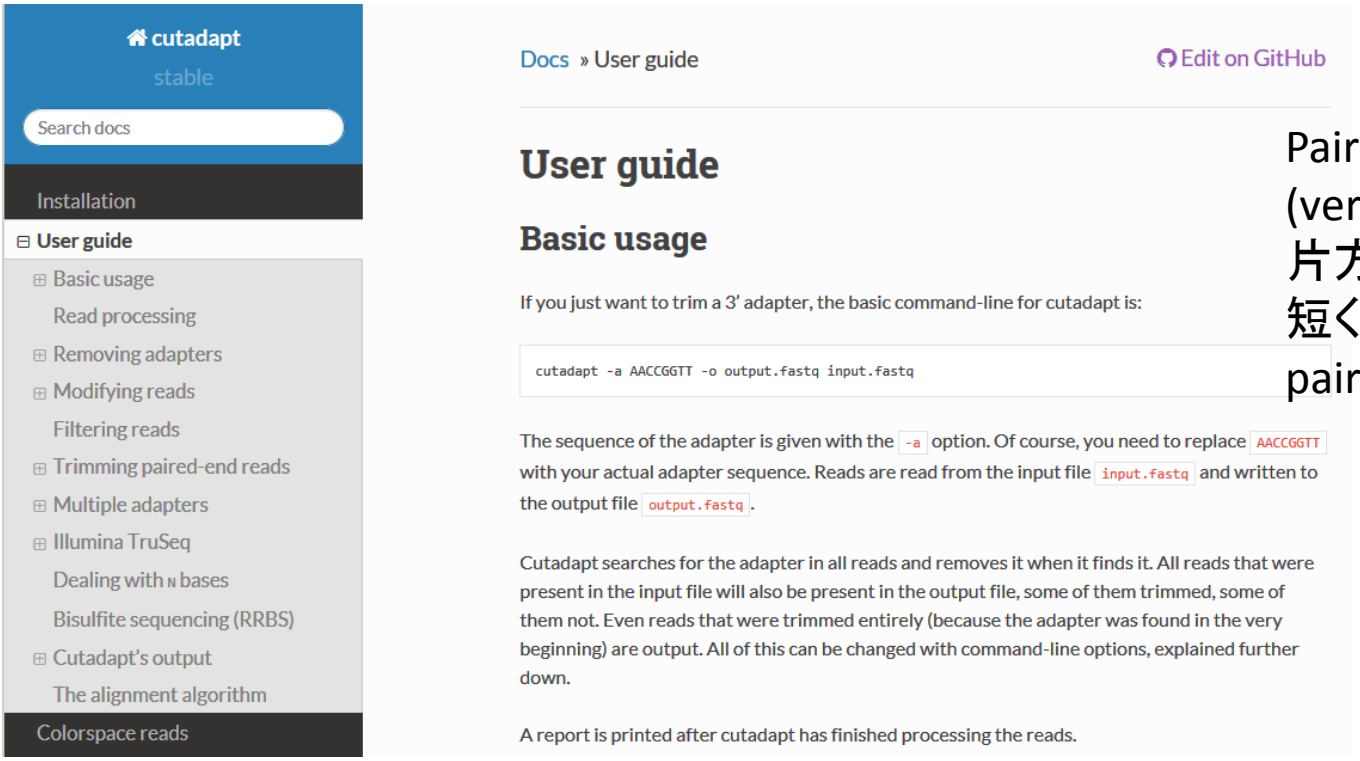
通常イルミナRNA-Seqライブラリーは
200baseくらいの長さから存在する
うち両端にアダプター63baseずつ

Preprocessing tools

現行では以下の2ツールが有名
- Cutadapt
- Trimmomatic

- adapter配列を除去
- 一定クオリティー以下の部位を除去
- 任意の配列部位を除去

生データを処理することで一定のクオリティーを確保したデータとなる



Paired end readに対応
(ver. 1.8以降)
片方のreadが非常に
短くしか残らない場合、
pair read両方とも除去する。

http://cutadapt.readthedocs.org/en/stable/guide.html

MacOSXでのcutadaptのインストール

Cutadapt install手順

Cython をダウンロード
https://pypi.python.org/pypi/Cython/
からCython-0.25.2.tar.gzをダウンロード

cd Cython-0.25.2
sudo python setup.py install

cd ..
git clone
https://github.com/marcelm/cutadapt
cd cutadapt
sudo python setup.py install

現状最新はver. 1.12
http://cutadapt.readthedocs.io/en/stable/installation.html

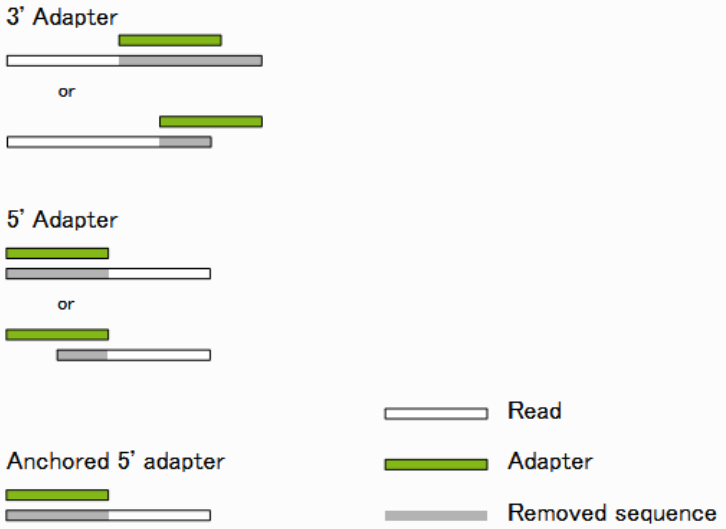
Cutadapt

Removing adapters

Cutadapt supports trimming of multiple types of adapters:

Adapter type	Command-line option
3' adapter	`-a ADAPTER`
5' adapter	`-g ADAPTER`
Anchored 3' adapter	`-a ADAPTER$`
Anchored 5' adapter	`-g ^ADAPTER`
5' or 3' (both possible)	`-b ADAPTER`

Here is an illustration of the allowed adapter locations relative to the read and depending on the adapter type:



Cutしたいアダプター配列の
位置関係など詳細に指定可能

fastqファイルはgz圧縮してあってもよい
fastaファイルも可

```
$ cutadapt
cutadapt version 1.12
Copyright (C) 2010-2016 Marcel Martin <marcel.martin@scilifelab.se>

cutadapt removes adapter sequences from high-throughput sequencing reads.

Usage:
   cutadapt -a ADAPTER [options] [-o output.fastq] input.fastq

For paired-end reads:
   cutadapt -a ADAPT1 -A ADAPT2 [options] -o out1.fastq -p out2.fastq in1.fastq in2.fastq
```

最適な
QV値
minimum-length値
O値
を設定して行う。

crude_fastqフォルダーに生シーケンス配列
trim_fastqフォルダーにcutadaptにかけた配列
を用意してあります

Single readの場合

```
$ cutadapt ¥
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC ¥
-o hoge_read1.cut.fastq ¥
hoge_read1.fastq
```

Paired end readの場合

```
$ cutadapt ¥
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC  ¥
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATC
-o hoge_read1.cut.fastq ¥
-p hoge_read2.cut.fastq ¥
hoge_read1.fastq ¥
hoge_read2.fastq
```

復習問題

以下~data/KYにある2D2L_rep1_R1.fastqと2D2L_rep1_R2.fastqファイルは
アラビドプシスの発芽・緑化後の芽生えをサンプリング、ライブラリー作製した
Paired-end read(76base x2)のRNA-Seqの生リードのfastqファイルである。
これを用いて、
以下のパラメータを参考にし、paired-endでのcutadaptにかけよ。

-qv 30
-O 7
-mincut 50
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATC

Q1. Cutadaptのlogを見て、passしたpair数、quality trimされたbase数を調べよ。

Q2. Cutadapt処理前後のfastqファイルをlessコマンド等で見比べよ

Q3. wcコマンドでcutadapt前後のread数を調べよ。

Q4. Cutadapt処理前後のfastqファイルをfastqcにかけ、cutadapt処理による、低品質配列が除かれていることを確認せよ。

A1. Pairs written(passing filters) : 60,885(60.9%)
Quality-trimmed 824,581(5.4%)
A2. lessコマンドでファイルを見る
A3. trim前400,000なのでreadとしては4で割って、100k read
trim後243,540なのでread数は60,885となり、logの値と一致している。
A4. Per base sequence qualityのタブを見る