

NGS 基本フォーマットとツール 復習と補足

基礎生物学研究所
ゲノムインフォマティクストレーニングコース
内山 郁夫 (uchiyama@nibb.ac.jp)

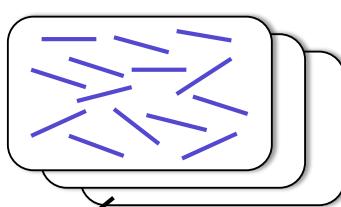
ショートリードのマッピング

ゲノム配列
(リファレンス reference 配列)

形式(配列)

```
>chr
AGCTTTCAATTCTGACTGCAACGGGCAATATGCTT
CTGTGTGGATTAAAAAAAGACTGTCGTAGCAGC
TTCTGAACCTGGTTACCTGCCGTGAGTAATTTAAA
TTTATTGACTTAGGTCACTAAATACTTTAACCAA
TATAGGCCATAGCGCACAGACAGATAAAATACAG
AGTACACAACATCATGAAACGATTAGCACCCACC
ATTACCAACACCATCACCATTACCAACAGGTAACGG
```

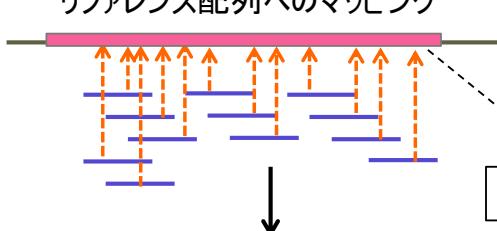
サンプル(ゲノムDNA／RNA)
(リード read 配列)



形式
(配列 + クオリティ値)

```
@SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:
ATCCGGCTGGCCACCGACCTATGTTCCGGCGAATACAAGCTGG
+SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:
@@@AD>DF7DC?FEBF@FII<DF@AA6AEFBDBCA?>A?
@SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:
CACCGTAGTACAGCATCCTGGTACAAATCAGCAATCCAGTC
+SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:
CCCCFPDFHDFPHIIIEGHJJJJCFFHGGHHGGGILJDGLJHH
@SRR1515276.3 HWI-ST808:151:D2D13ACXX:2:1207:
CAGGACATCGCCTTTGATCGGTTCAAGACTTGGACCAACCTGCAT
+SRR1515276.3 HWI-ST808:151:D2D13ACXX:2:1207:
CCCCFFDFAFHFIJGHIIJJIJJHEHIIJGHIFEHIIA@FIFH
```

リファレンス配列へのマッピング



クオリティチェック
アダプター除去

形式(遺伝子アノテーション)

```
chr RefSeq start_codon 190 192 1.000 + . gene_id "b0001
chr RefSeq CDS 190 252 1.000 + 0 gene_id "b0001
chr RefSeq stop_codon 253 255 1.000 + . gene_id "b0001
chr RefSeq exon 190 255 1.000 + . gene_id "b0001
```

形式(マッピング結果)

```
@HD VN:1.0 SO:unsorted
@SQ SN:chr LN:4639675
@PG ID:bowtie2 FN:bowtie2 VN:2.2.4 CL:"/bio/bin/bowtie2-alig
SRR1515276.40 0 chr 4423609 42 51M * 0 0 GGAATTCTCACTGGCA
SRR1515276.158 16 chr 501700 42 51M * 0 0 ACGCACCGACTGCAAAG
SRR1515276.212 4 * 0 0 * * 0 0 GGCCGCTTTCAGCGTGT
SRR1515276.319 0 chr 2922768 42 51M * 0 0 GCTTAAGTGATTAAGG
SRR1515276.367 16 chr 2753873 42 51M * 0 0 GCGTGTCCGTCCGCAGC
SRR1515276.411 0 chr 3440721 42 51M * 0 0 ACGGCATAATTCTTGA
```

復習: cutadaptによるアダプターの除去

実習用ディレクトリ ~/data/IU

入力

- リード配列(FASTQ 形式; paired-end)
etec_1.fq
etec_2.fq

- アダプター配列(それを3'端から除去)

Adapter1: AGATCGGAAGAGCGGGTT

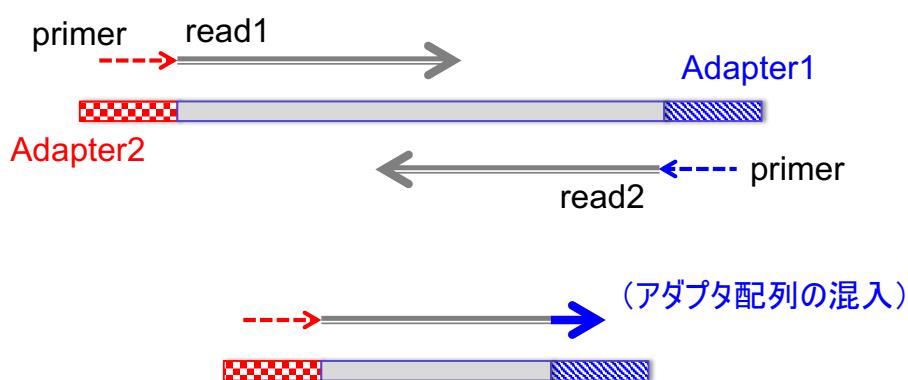
Adapter2: AGATCGGAAGAGCGTCG

◆ アダプター配列除去の実行

除去後のデータ(FASTQ形式)は etec_1.cut.fq, etec_2.cut.fqとする。

```
$ cutadapt  AGATCGGAAGAGCGGGTT  AGATCGGAAGAGCGTCG  
       etec_1.cut.fq  etec_2.cut.fq  
      etec_1.fq   etec_2.fq
```

Illuminaにおけるアダプター配列



Adapter1: AGATCGGAAGAGCACACGTCTGAACCCAGTCAC

Adapter2: AGATCGGAAGAGCGTCGTAGGGAAAGAGTGTA

cutadapt -a (-A) オプションでは、指定した配列とマッチした箇所以降の3'側を切り捨てる所以、アダプタ配列は全長を指定しなくてもよい。

cutadapt その他のオプション

- **-q [5' cutoff,] 3' cutoff** (例 : -q 20)
 - ・クオリティ値が指定したカットオフより低い塩基を3'端から除く(カンマ区切りでカットオフを2つ指定した場合は5'端からも除く)
- **-m min_length** (例 : -m 30)
 - ・アダプター除去後の配列長が指定した長さ以下になつたら配列全体を捨てる。
 - ・ペアエンドの場合、ペアのどちらかが捨てられる場合は両方を捨てる。
→2つのファイルで対応する配列の出現順が揃うようにする。
- **-O overlap_length** (例 : -O 5)
 - ・アダプターとリードとの間で、マッチしたと見なす最低のオーバーラップ長を指定。
デフォルトは3。



復習: bowtie2 用インデックスの作成

実習用ディレクトリ ~/data/IU

入力

- ゲノムデータ (FASTA形式)
`eco_o139.fa` 腸管毒素原性大腸菌(ETEC) O139:H28のゲノム配列

◆ bowtie2用インデックスの作成 (インデックス名は `etec`)

\$ `bowtie2-build`

復習: bowtie2の実行 (paired-end)

実習用ディレクトリ ~/data/IU

入力

- リード配列(FASTQ 形式; paired-end; アダプターを除去したもの)
etec_1.cut.fq
etec_2.cut.fq
- リファレンス配列のインデックス名
etec (先ほど作ったもの)

◆ bowtie2によるマッピングの実行 (出力: etec_bowtie2.sam)

```
$ bowtie2 [ ] etec [ ] etec_1.cut.fq [ ] etec_2.cut.fq  
[ ] etec_bowtie2.sam
```

マッピング結果ファイル(SAMファイル)

ヘッダ(@で始まる)

@HD VN:1.0 SO:unsorted	VN:2.3.0 CL:"/bio/bin/bowtie2-align-s --wrapper basic-0 -x etec -S etec_bowtie2.sam -l etec_1.cut.fq -2 etec_2.cut.fq	AS:i:-1 XS:i:-1 XN:i:0		
@SQ SN:ETEC_chr LN:4979619	49M = 3758170 0 ACACGGCCATGGCTG... #####ED>EBBDBDE,E...	##?ED>EBBDBDE,E...		
@SQ SN:PETEC_80 LN:79237	*	AS:i:-1 XS:i:-1 XN:i:0		
@SQ SN:PETEC_35 LN:34367	49M = 4361458 0 CAACCGTTAACCGAA...	#####H#H#H#...		
@SQ SN:PETEC_73 LN:70609	*	YT:Z:UP YF:Z:NS		
@SQ SN:PETEC_6 LN:6199	49M = 4361458 0 NNNNNNNNNNNNNNN...	AS:i:0 XS:i:0 XN:i:0		
@SQ SN:PETEC_74 LN:74224	*	#####H#H#H#...		
@SQ SN:PETEC_5 LN:5033	49M = 4362922 0 CGGTGGATGCCCTGGC...	YT:Z:UP YF:Z:NS		
@PG ID:bowtie2 PN:bowtie2 VN:2.3.0 CL:"/bio/bin/bowtie2-align-s --wrapper basic-0 -x etec -S etec_bowtie2.sam -l etec_1.cut.fq -2 etec_2.cut.fq	49M = 4362922 0 NNNNNNTNNNTTCGG...	AS:i:-2 XS:i:-2 XN:i:0		
SRR345261.25 89 ETEC_chr 3758170 1 49M = 3758170 0 ACACGGCCATGGCTG... #####ED>EBBDBDE,E...	49M = 4362922 0 NNNNNNTNNNTTCGG...	YT:Z:UP YF:Z:NS		
SRR345261.25 133 ETEC_chr 3758170 0 *	49M = 3758170 0 NNNNNNNNNNNNNNN...	AS:i:0 XS:i:0 XN:i:0		
SRR345261.50 73 ETEC_chr 4361458 1 49M = 4361458 0 CAACCGTTAACCGAA...	*	#####H#H#H#...		
SRR345261.50 133 ETEC_chr 4361458 0 *	49M = 4361458 0 NNNNNNNNNNNNNNN...	YT:Z:UP YF:Z:NS		
SRR345261.75 73 ETEC_chr 4362922 1 49M = 4362922 0 CGGTGGATGCCCTGGC...	*	AS:i:-2 XS:i:-2 XN:i:0		
SRR345261.75 133 ETEC_chr 4362922 0 *	49M = 4362922 0 NNNNNNTNNNTTCGG...	YT:Z:UP YF:Z:NS		
SRR345261.100 73 ETEC_chr 679991 42 49M = 679991 0 GTGCTTTAATGACTCC...	49M = 679991 0 GTGCTTTAATGACTCC...	AS:i:0 XS:i:0 XM:i:0		
SRR345261.100 133 ETEC_chr 679991 0 *	49M = 679991 0 NNNNNNCACCGNTAGT...	YT:Z:UP YF:Z:NS		
SRR345261.125 73 ETEC_chr 4376280 42 49M = 4376280 0 CTCAGGATGAGGGTCA...	*	AS:i:0 XS:i:0 XM:i:0		
SRR345261.125 133 ETEC_chr 4376280 0 *	49M = 4376280 0 NNNNNNTTTCCNTAG...	YT:Z:UP YF:Z:NS		
SRR345261.125 133 ETEC_chr 779844 42 49M = 779844 0 TTCAGAAACCCCTGAA...	*	AS:i:-5 XS:i:0 XM:i:1		
SRR345261.150 89 ETEC_chr 779844 0 *	49M = 779844 0 CNCGAGTACNTTGAA...	YT:Z:UP YF:Z:NS		
SRR345261.175 83 ETEC_chr 3605306 42 49M = 3605113 -242 CCCGTTGGCGGGCCA...	*	AS:i:0 XS:i:0 XM:i:0		
SRR345261.175 163 ETEC_chr 3605113 42 49M = 3605306 242 CCCGGTTCTGCTGTGG...	49M = 3605306 242 CCCGGTTCTGCTGTGG...	AS:i:-3 XS:i:0 XM:i:3		
SRR345261.200 77 *	*	AS:i:0 XS:i:0 XM:i:0		
SRR345261.200 141 *	0 0 *	YT:Z:UP		
SRR345261.225 83 ETEC_chr 2879707 1 49M = 2879600 -156 CACACACGAGCTGAC...	0 0 *	YT:Z:UP		
SRR345261.225 163 ETEC_chr 2879600 1 49M = 2879707 156 CCCACCTCTCTCAGT...	0 0 *	AS:i:0 XS:i:0 XM:i:0		
SRR345261.250 99 ETEC_chr 4361346 1 49M = 4361525 228 GTACTTTACGGGGGA...	0 0 *	AS:i:-1 XS:i:-1 XN:i:0		
SRR345261.250 147 ETEC_chr 4361525 1 49M = 4361346 -228 CCCGGCTAACCTGGG...	0 0 *	AS:i:0 XS:i:0 XM:i:0		
FLAG	マップされた染色体と位置 (* はマップされなかった)	リード配列	配列クリティカル	オプション
MAPQ	ペアの相手がマップされた染色体(同じなら=)と位置、フラグメントの長さ(右側のリードは負値)	リード配列	テイ	AS アライメントスコア
CIGAR	同じ名前のリード =ペアエンドのリード対	配列クリティカル	ティ	XS 他の位置でのベストスコア
		リード配列	テイ	YF リードがfiltering outされた理由

復習: SAMからBAMへの変換

実習用ディレクトリ ~/data/IU

入力

- SAMファイル (さきほどbowtie2によって作成されたもの)
`etec_bowtie2.sam`

- ◆ SAMからBAMへ変換する (出力ファイル名: `etec_bowtie2.bam`)

```
$ samtools [ ] [ ] etec_bowtie2.sam [ ]  
etec_bowtie2.bam
```

- ◆ 作成したBAMファイルをヘッダ付きでSAMに変換してlessで表示する

```
$ samtools view -h etec_bowtie2.bam [ ] less
```

復習: BAMファイルのインデックスづけと検索

実習用ディレクトリ ~/data/IU

入力

- BAMファイル (さきほどSAMからの変換によって作成されたもの)
`etec_bowtie2.bam`

- ◆ リファレンス配列上の位置の順にソートする
(出力ファイル: `etec_bowtie2_sorted.bam`)

```
$ samtools [ ] etec_bowtie2.bam  
[ ] etec_bowtie2_sorted.bam
```

- ◆ ソートされたBAMファイルに対してインデックスを作成する

```
$ samtools [ ] etec_bowtie2_sorted.bam
```

- ◆ インデックスを使って、リファレンスの染色体配列(染色体名: `ETEC_chr`)の10000-12000 の範囲にマッピングされた結果のみを表示する

```
% samtools [ ] etec_bowtie2_sorted.bam  
[ ]
```

SAM/BAM フォーマット補足

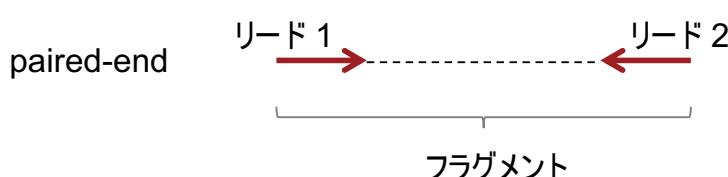
- Bowtie2のデフォルトオプションでマッピングした結果のSAM/BAMファイルは、もとのFASTQファイルに含まれている各リードの配列とクオリティデータをすべて含んでいる。以下のコマンドでSAM/BAMファイルからFASTQファイルを作成できる。

```
$ samtools fastq etec_bowtie2.bam -1 r1.fq -2 r2.fq
```

- 個々のリード配列を記録する代わりに、リファレンス配列を参照して、各リードのリファレンス上の位置とアライメント情報のみを記録することによって、さらに圧縮率を高めたバイナリ形式としてCRAM形式がある。

```
$ samtools view -C etec_bowtie2.sam -T eco_o139.fa  
-o etec_bowtie2.cram
```

Bowtie2のオプション1 ペアエンドリード対の検索



- **-I int** フラグメント長の最小値(default: 0)
- **-X int** フラグメント長の最大値(default: 500)
- **--fr / --rf / --ff** リード1とリード2の相対的な向き (default:fr)

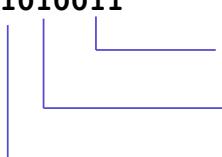
--fr --rf --ff
→ ← ← → → ← → →

- 条件を満たさない(discordant)リード対もデフォルトでは出力される。その際、2カラム目(FLAG)の2ビット目(ペアが正しくアラインされたか?)に0がセットされる。

フラグ(FLAG)

- True/Falseの2状態を1/0で表した変数。複数のフラグをまとめて、2進数の数値で表現される。
- フラグ値は10進数で表示されるが、2進数に変換することで解釈される。

FLAG値

10進数	2進数	解釈
83	01010011	ペアリードである
		各リードが適切にアラインされている
		逆鎖にマップされている
		1番目のリードである

unix コマンドによる 10進数→2進数の変換

```
% echo 'obase=2;83' | bc
```

```
1010011
```

samtools を使ったフラグの解釈

```
% samtools flags 83
0x53    83      PAIRED,PROPER_PAIR,REVERSE,READ1
```

各フラグの説明を表示

```
% samtools flags
```

Paired end readでのFLAG値



通常のpaired end seqで consistentにアラインしていれば この4通りになる	片方しかアラインしていない場合	どっちもアラインしていない場合	2進数表記	samファイルの記載は 10進数表記
0 1 0 1 0 0 1 1	0 1 0 0 1 0 0 1	0 1 0 0 1 1 0 1	01010011	83
0 1 1 0 0 0 1 1	0 1 0 1 1 0 0 1	0 0 1 0 1 1 0 1	01100011	99
1 0 0 0 1 0 0 1	0 1 0 0 0 1 0 1	0 0 1 0 0 1 1 0	10010011	147
1 0 0 1 0 0 0 1	0 1 0 0 0 0 1 1	0 0 1 0 0 0 1 1	10100011	163
0 1 0 0 1 0 0 1	0 1 0 0 1 0 0 1	0 1 0 0 0 0 1 1	01001001	73
0 1 0 1 0 0 0 1	0 1 0 1 0 0 0 1	0 1 0 0 0 1 0 1	01011001	89
0 1 0 0 0 0 1 0	0 1 0 0 0 0 1 0	0 1 0 0 0 1 0 0	01000101	69
0 1 1 1 0 0 0 1	0 1 1 1 0 0 0 1	0 1 1 0 0 0 1 0	01100101	101
1 0 0 0 0 1 0 0	1 0 0 0 0 1 0 0	1 0 0 0 0 0 1 0	10001001	137
1 0 0 0 1 1 0 0	1 0 0 0 1 1 0 0	1 0 0 0 1 0 0 1	10011001	153
1 0 0 0 0 0 1 0	1 0 0 0 0 0 1 0	1 0 0 0 0 0 1 0	10000101	133
1 0 0 1 0 0 0 1	1 0 0 1 0 0 0 1	1 0 0 1 0 0 0 1	10100101	165
0 1 0 0 0 1 1 0	0 1 0 0 0 1 1 0	0 1 0 0 0 1 1 0	01001101	77
1 0 0 0 0 1 1 0	1 0 0 0 0 1 1 0	1 0 0 0 0 1 1 0	10001101	141

Samtoolsを用いた フラグによるフィルタリング

● samtools view -f フラグ値 BAMファイル

指定したフラグ値中で1であるフラグが、BAMファイル中のフラグ値でもすべて1になっている行のみを抜き出す。

例) ペアリードでかつ両方が適切にアラインされている行のみを抜き出す

```
% samtools view -f 3 etec_bowtie2_sorted.bam
```

3は2進数で 11 だから、1番目と2番目のフラグが1である行を抜き出す(それ以外のフラグは無視する)。

● samtools view -F フラグ値 BAMファイル

指定したフラグ値中で1であるフラグが、BAMファイル中のフラグ値ではすべて0になっている行のみを抜き出す。

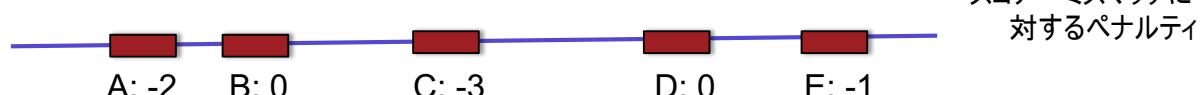
例) ペアリードの両方が適切にアラインされていない行のみを抜き出す

```
% samtools view -F 2 etec_bowtie2_sorted.bam
```

2番目のフラグが0である行を抜き出す。

Bowtie2のオプション2 アライメント出力のモード

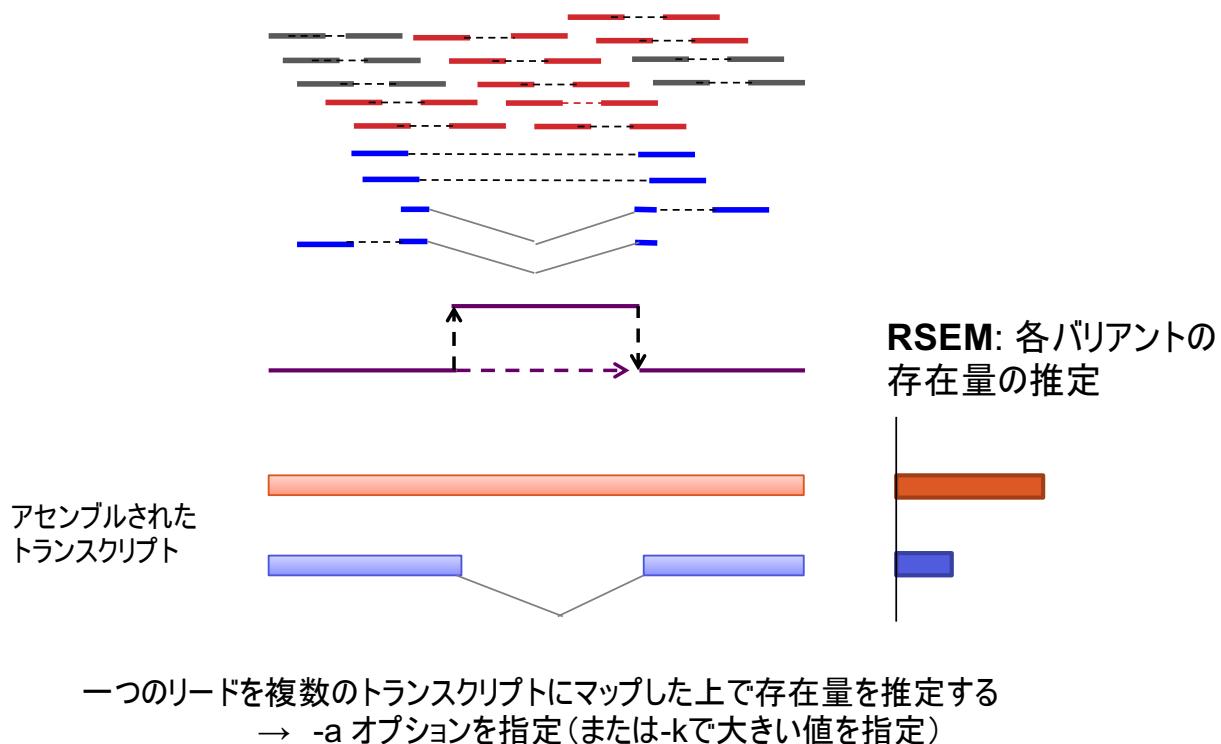
- 一般に、1つのリードは複数の箇所にマップされる。



- default (best one mode)
条件を満たすアライメントを検索し、最高スコアのものを1つ出力
(ただし、検索は完全でないので、最高スコアを取りこぼす可能性はある)
上記の例では、BまたはD (どちらかがランダムに選ばれる)
- a
条件を満たすアライメントをすべて出力
上記の例では、A,B,C,D,E
- k <int>
条件を満たすアライメントを、見つかった順に指定した数だけ出力
上記の例で、-k 2 のとき、左から順に見つかるとすると、AとB
(実際には位置の順に見つかるわけではない)
- a や -k を指定したとき、最高スコアでないアライメントには9番目のフラグ (secondary alignment) に1がセットされる

(参考)デノボ・アセンブルによるRNA-Seq解析

デノボ・アセンブルによる転写配列の構築



マッピングクオリティ(MAPQ)

- マッピングクオリティ(MAPQ)値は以下の式で計算される。

$$\text{MAPQ} = -10 \log_{10}(P_e)$$

ただし、 P_e はリードが間違った位置にマップされている確率の推定値。

- MAPQは、リードがその位置にどの程度ユニークにマップされたかを示す指標であり、その位置でのアライメントスコアが、他のすべての位置におけるスコアよりもずっと大きいときに大きくなる。
- Bowtie2のデフォルトでは同じスコアのアライメントが複数の位置で得られた場合、ランダムに一つの位置を出力し、MAPQに低い値を設定する。
- MAPQが低いアライメントの位置は信用できないので、下流の解析の際には捨てた方が良い場合もある。

Samtoolsを用いた MAPQによるフィルタリング

- samtools view -q 閾値 BAMファイル名

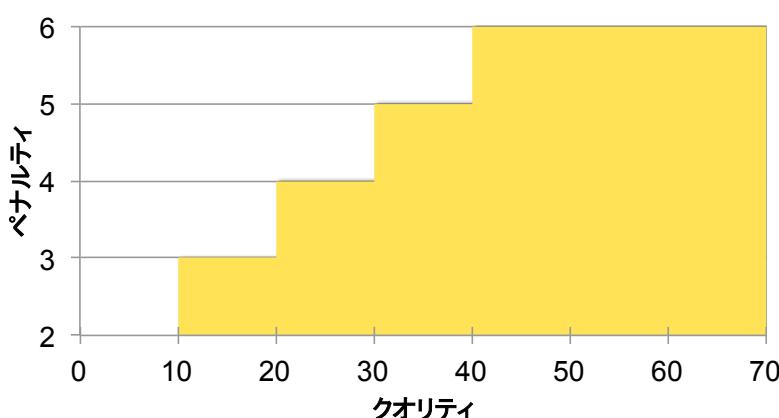
MAPQの値が閾値より小さい行を除く

例) MAPQが20以上の行のみを出力

```
$ samtools view -q 20 etec_bowtie2.bam
```

(参考) Bowtie2における アライメントスコア

- マッチは0で、ミスマッチにマイナスのペナルティ(最高スコアが0点)
- ミスマッチペナルティは、クオリティ値に応じて -2 から -6 の値をとる(下図)
- あいまい塩基(N)のペナルティは -1
- ギャップペナルティは、ギャップの長さ n に対して $-(5 + 3n)$
- スコアのカットオフは、長さ L に対して $-0.6(L+1)$



Bowtie2のオプション3 アライメントのモード

- **--end-to-end** リード配列全長に渡るアライメント(default)

```
Read:      GACTGGGCGATCTGACTTCG
           |||||   |||||||||  ||
Reference: GACTG--CGATCTGACATCG
```

- **--local** リード配列のうち、類似度の高い一部の領域のみを抜き出してアラインしたもの

```
Read:      ACGGTTGCCTTAA-TCCGCCACG
           |||||||||  |||||
Reference: TAACTTGCCTTAAATCCGCCCTGG
```

CIGAR文字列

- リードとリファレンス配列とのアライメントの詳細を表す。
- ギャップなしでアラインされている場合、 nM (n はリード配列の長さ)となる。
- ギャップが入っている場合、 nD (欠失)または nI (挿入) (n は挿入・欠失の長さ)が入る。

5M2D4M1I5M

```
ref  AGACGAGATTA-GCATG
     ::::: ::::: :::::
read ACACG--ATTAGGCTTG
```

- ローカルアライメントのとき、両端の除かれる部分は nS で、また TopHatなどのスプライシングを考慮するアライメントにおいて、イントロンとしてスキップされるリファレンス配列上の領域は nN で表される。

5S4M1I5M

```
ref  ACGGCTGATTA-GCATG
     ::::: :::::
read  taaccATTAGGCTTG
```

インデックスを使った高速検索 ハッシュテーブル

ゲノム配列

ACACGTTACGGT.....

リード配列

CGTTGCA

①インデックス作成

ハッシュテーブル
各2-merの出現位置を記録

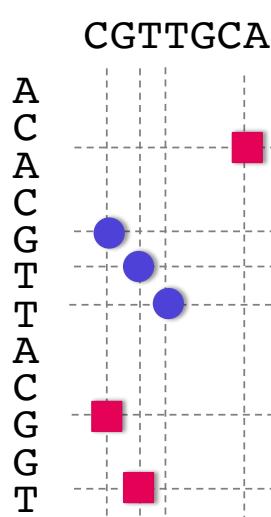
2-mer	positions
AC	1, 3, 8
CA	2
CG	4, 9
GG	10
GT	5, 11
TA	7
TT	6

② インデックスを使った初期検索(seed検索)

CGTTGCA

③ 見つかったseedを延長してアライメント

ACACGTTACGGT.....
CGTT**GCA**



インデックスを使った高速検索 接尾辞配列(suffix array)

ACACGTTACGGT

接尾辞

1	ACACGTTACGGT
2	CACGTTACGGT
3	ACGTTACGGT
4	CGTTACGGT
5	GTTACGGT
6	TTACGGT
7	TACGGT
8	ACGGT
9	CGGT
10	GGT
11	GT
12	T

ソート

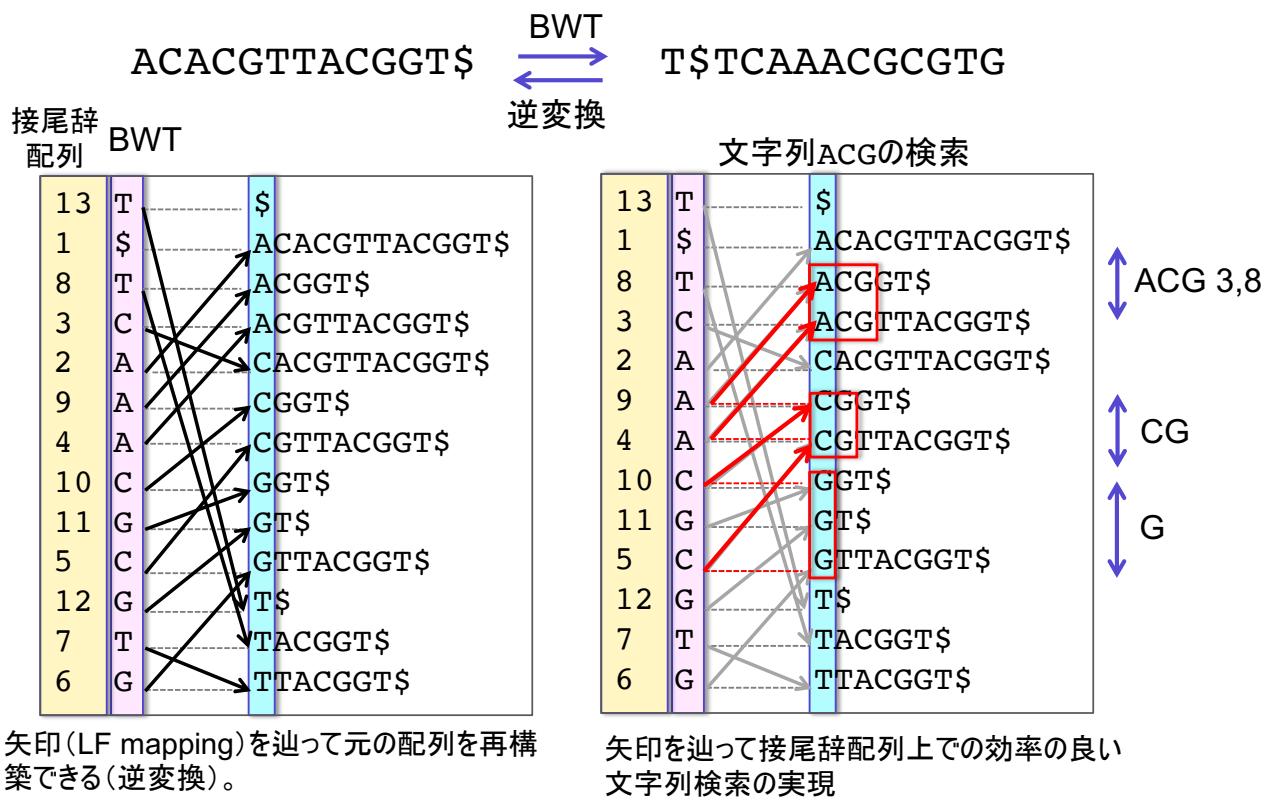
1	ACACGTTACGGT
8	ACGGT
3	ACGTTACGGT
2	CACGTTACGGT
9	CGGT
4	CGTTACGGT
10	GGT
11	GT
5	GTTACGGT
12	T
7	TACGGT
6	TTACGGT

ACG 3,8

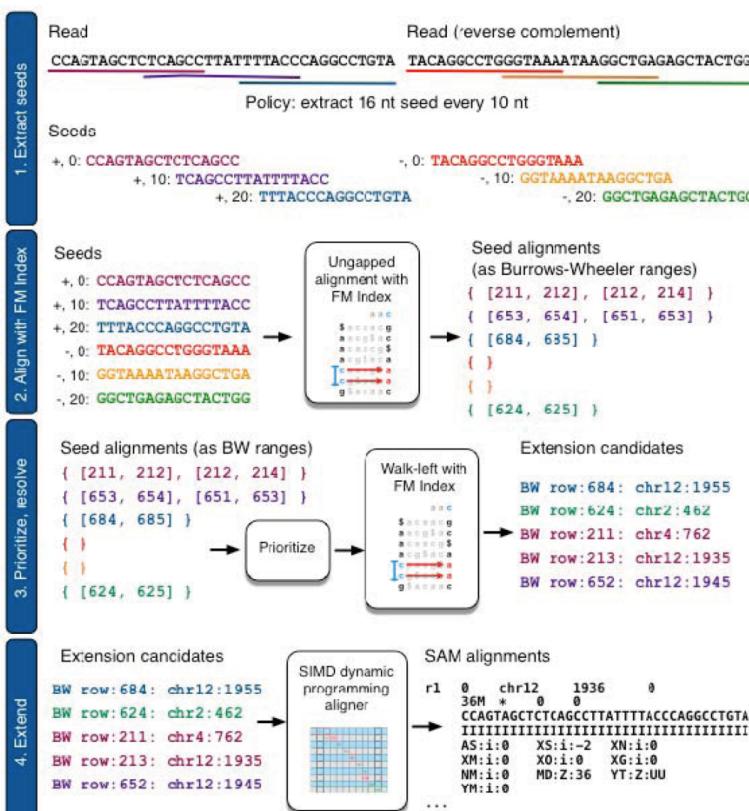
CG 4,9

接尾辞
配列

Burrows-Wheeler 変換 (BWT)に基づくインデックス(FM-Index)



Bowtie2 アルゴリズムの詳細



1. Seed 配列の抽出

各リード配列およびその相補配列から i 塩基ごとに L 塩基の配列を抽出して seed 配列とする(図では $i=10, L=16$)。

2. FM index を用いた検索

各seed配列がゲノム上に出現する位置が BW range として得られる。最大1つのミスマッチを考慮した検索が可能。

3. ヒットの優先付け、位置の取得

BW range の幅が小さいヒットに高い優先度をつけて、ランダムに候補をピックアップしゲノム上の位置を取得。

4. アライメントの計算

得られた位置の周辺で、ギャップ入りのアライメントスコアを計算。これを各候補位置について繰り返して、最高スコアを与えるゲノム上の位置を出力。

Bowtie2のオプション4 検索の精度と速度に関するオプション

- `-N int` `seed` 検索時にミスマッチを許す数(0 or 1)
- `-L int` `seed` の長さ
- `-i func` `seed` をとる間隔(リード長を基に決める式を指定)
- `-D int` 最高スコアが更新されないときアライメント計算を打ち切るまでの回数
- `-R int` リードが高反復の`seed`をもつときにre-seedを行う最大回数

上記のオプションを同時に設定するpreset optionがある。高速(低感度)→高感度(低速)の順に4段階のオプションが用意されている。

- end-to-endモードの場合 (default: `sensitive`)
`--very-fast / --fast / --sensitive / --very-sensitive`
- localモードの場合 (default: `sensitive-local`)
`--very-fast-local / --fast-local / --sensitive-local / --very-sensitive-local`

(参考) HISAT2 スプライシングを考慮した高速マッピングツール

- スプライシングを考慮して、一つのリードをゲノム上で離れた箇所にまたがってマッピングする。
- global とlocalの2つのインデックスを2段階で用いることにより、高速かつ正確にスプライスされたアライメントを実現。

