

# NGS 基本フォーマットとツール 補足と復習

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](mailto:uchiyama@nibb.ac.jp))

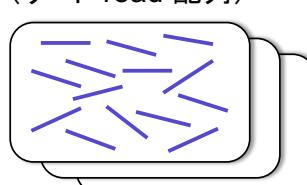
## ショートリードのマッピング

ゲノム配列  
(リファレンス reference 配列)

形式(配列)  

```
>chr
AGCTTTTATTCCTGACTGCAACGGGCAATATGCT
CTGGTGGATTAAAAAAAAGAGTGCTGTAGCAGC
TTCTGAACCTGGTTACCTGCCGTGAGTAATTTAA
TTTATTGACTTAGGTCACTAAATCTTTAACCAA
TATAGGCATAGCCCACAGACAGATAAAAATACAG
AGTACACAAACATCCATGAAACGCCATTAGCACCCAC
ATTACCAACCATCACCATCACACAGGTAAACCG
```

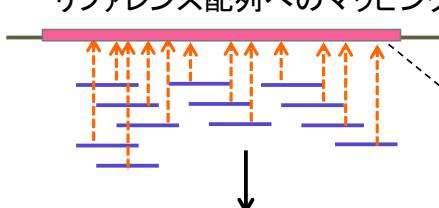
サンプル(ゲノムDNA／RNA)  
(リード read 配列)



形式  
(配列＋クオリティ値)

```
@SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:
ATCCGGCTGGCGCACCGACCTATGTCGGCGGAATACAAGCTGG
+SRR1515276.1 HWI-ST808:151:D2D13ACXX:2:1207:
@@AD>DDF7DC?FFEBP@DFII<DF@AAA6AEFBDBDC@?A?
@SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:
CACCGTGTAGTACAGCAGATCCCTGGTACAATCAGCAATCCAGTC
+SRR1515276.2 HWI-ST808:151:D2D13ACXX:2:1207:
CCCCFPDPHDFPHIIIEGINJJJGFHGHHGGIIJDGIJHH
@SRR1515276.3 HWI-ST808:151:D2D13ACXX:2:1207:
CAGGACATCGCCCTTGTATGGTCAGACTCTGGACCACTGCAI
+SRR1515276.3 HWI-ST808:151:D2D13ACXX:2:1207:
CCCCFFDPAPHFHIIJGHIIJJIJHEHIIJGHIFEHIIIA@FIF
```

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
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 PN:bowtie2 VN:2.2.4 CL:"/bio/bin/bowtie2-align
SRR1515276.40 0 chr 4423609 42 51M * 0 0 GGAATTCTCACTGCCA
SRR1515276.158 16 chr 501700 42 51M * 0 0 ACCCACCGACTGCAAG
SRR1515276.212 4 * 0 0 * 0 0 GGCCGCTTTCAGCGTGT
SRR1515276.319 0 chr 2922768 42 51M * 0 0 GCTTAAGTTGATTTAAGG
SRR1515276.367 16 chr 2753873 42 51M * 0 0 GCTGTCCGTCCGGCAGC
SRR1515276.411 0 chr 3440721 42 51M * 0 0 ACGGCATATACTTCTTGA
```

# 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: AGATCGGAAGAGCGGTT

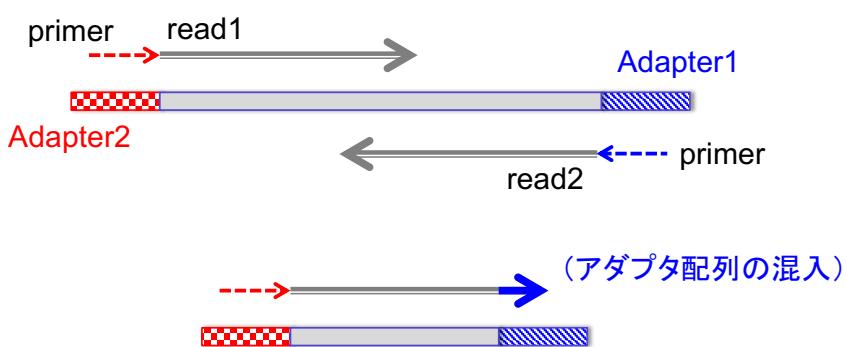
Adapter2: AGATCGGAAGAGCGTCG

## ◆ アダプター除去の実行

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

\$ `cutadapt` [redacted]

# Illuminaにおけるアダプター配列



Adapter1: AGATCGGAAGAGCACACGTCTGAAC~~TCCAGTCAC~~

Adapter2: AGATCGGAAGAGCGTCGTAGGGAAAGAGTGT

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'側を切り捨てるので、アダプター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っ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

输入

- リード配列(FASTQ 形式; paired-end; アダプター除去後)

etec\_1.cut.fq  
etec\_2.cut.fq

- リファレンス配列のインデックス名  
etc (先ほど作ったもの)

#### ◆ bowtie2の実行 (出力: etec bowtie2.sam)

```
$ bowtie2
```

# マッピング結果ファイル(SAMファイル)

ヘッダ(@で始まる)

## 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  
etec\_bowtie2.sam

◆ SAMからBAMへの変換 (出力ファイル名: etec\_bowtie2.bam)

```
$ samtools
```

◆ 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

```
$ samtools
```

## 復習: BAMファイルのインデックスづけ

実習用ディレクトリ ~/data/IU

入力

- BAMファイル  
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
```

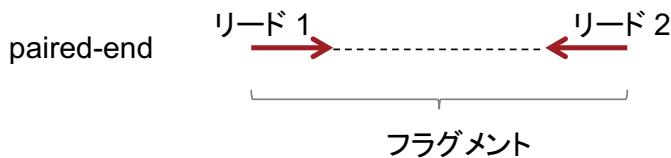
◆ ソートされたBAMファイルに対してインデックスを作成する

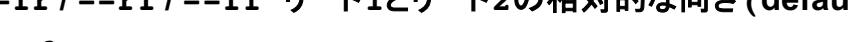
```
$ samtools
```

◆ 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
```

## 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

# 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值

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

```
# 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

Diagram illustrating Paired end read alignment:

- Read1:** Forward read, aligned to the left.
- Read2:** Reverse-complement read, aligned to the right.
- ref:** Reference sequence.

FLAG values are represented as binary numbers:

FLAG Value (Binary)	Description
01010011 83	Both ends mapped correctly (0x40 + 0x01)
10010011 147	Forward end mapped, reverse end unmapped (0x80 + 0x01)
01100011 99	Reverse end mapped, forward end unmapped (0x40 + 0x03)
10100011 163	Both ends unmapped (0x80 + 0x03)

Legend for FLAG bits:

- 0: ペアリードがある
- 1: ベア相手がマップされていない
- 2: 自分がマップされていない
- 4: 両方適切にマップされている
- 8: ベア相手は逆順にマップされた
- 16: 逆順にマップされた
- 32: ベア相手は逆順にマップされた
- 64: Read1の配列である
- 128: Read2の配列である

2進数表記 samファイルの記載は  
10進数表記

## 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。

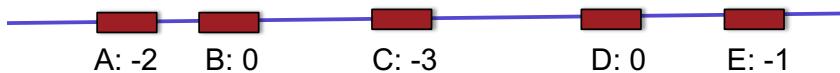
例) ペアリードの両方が適切にアラインされていない行のみを抜き出す

```
% 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

- 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)

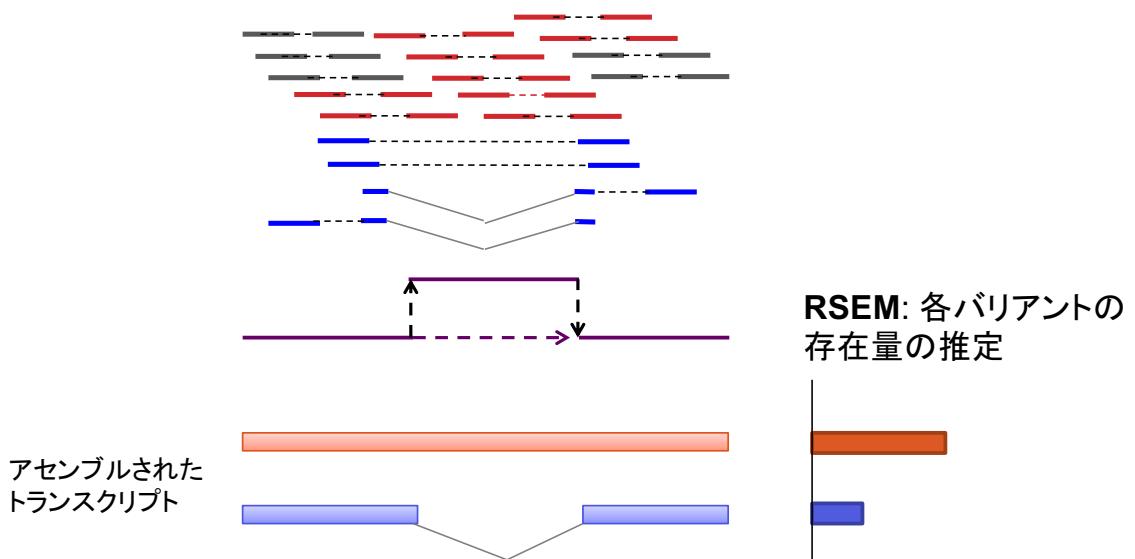
- -a

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 や -a 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

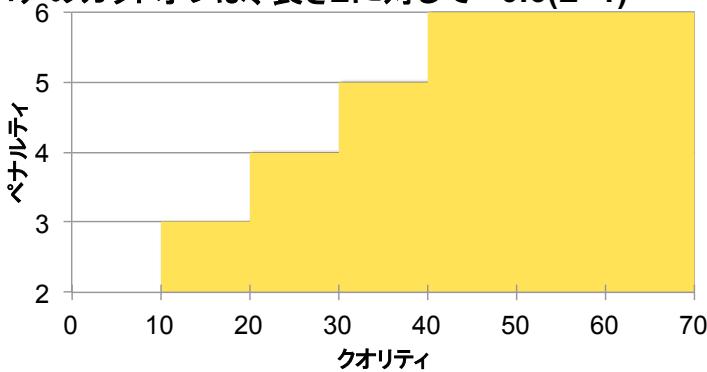
## (参考) デノボ・アセンブルによるRNA-Seq解析

### デノボ・アセンブルによる転写配列の構築



## (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  $-0.6(L+1)$



## 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
```

# 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:      GACTGGCGATCTGACTTCG
           |||||   |||||||||   ||
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: TAACTTGCCTTAAATCCGCCTGG
```

# 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



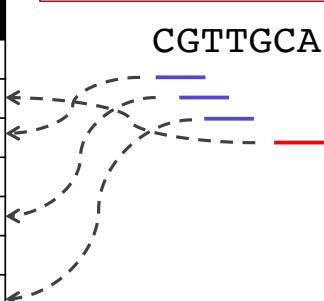
### ①インデックス作成

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

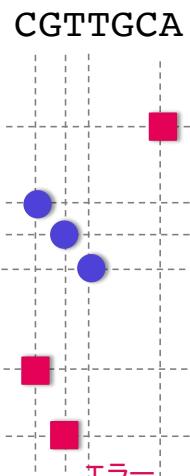
### ②インデックスを使った初期検索(seed検索)

CGTTGCA

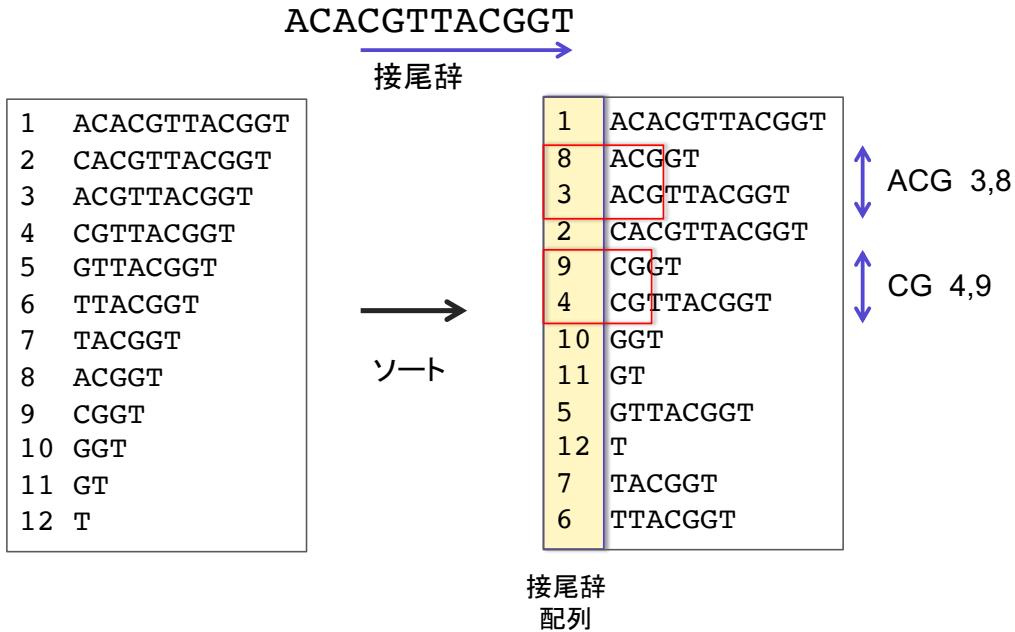


### ③見つかったseedを延長してアライメント

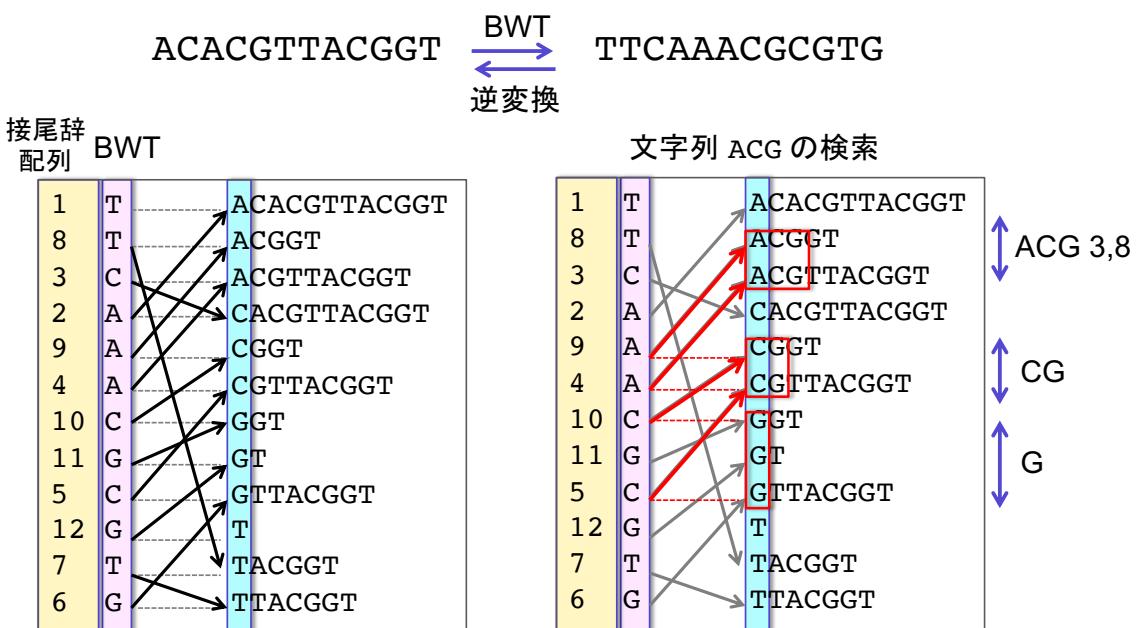
ACACGTTACGGT.....  
CGTT**GCA**



# インデックスを使った高速検索 接尾辞配列(suffix array)



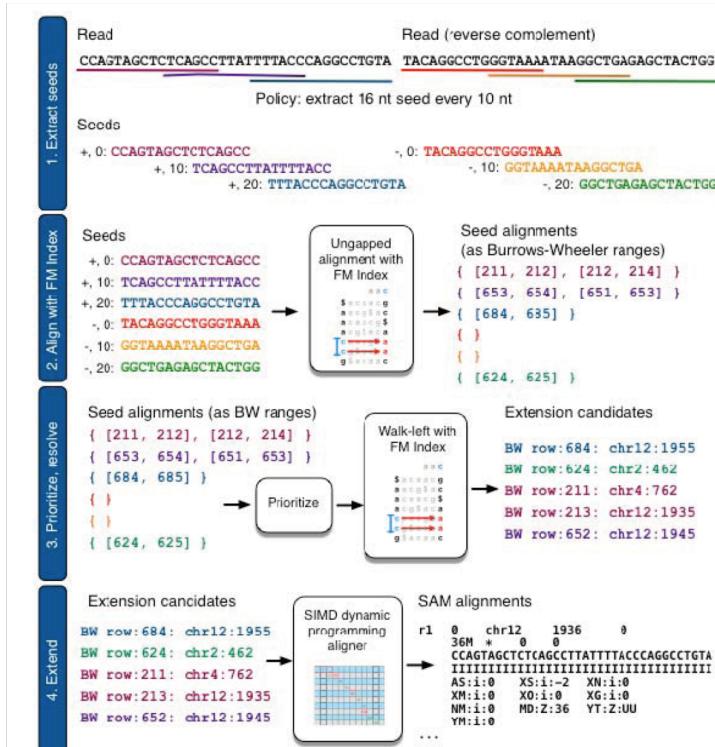
## Burrows-Wheeler 变換 (BWT)に基づく インデックス(FM-Index)



矢印(LF mapping)を辿って元の配列を再構築できる(逆変換)。

メモリ使用量、計算量とも効率のよい検索の実現

# 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、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**