2015年度第一回
バイオインフォマティクス実習

Bowtieを用いたシーケンスデータのマッピング

Cygwin

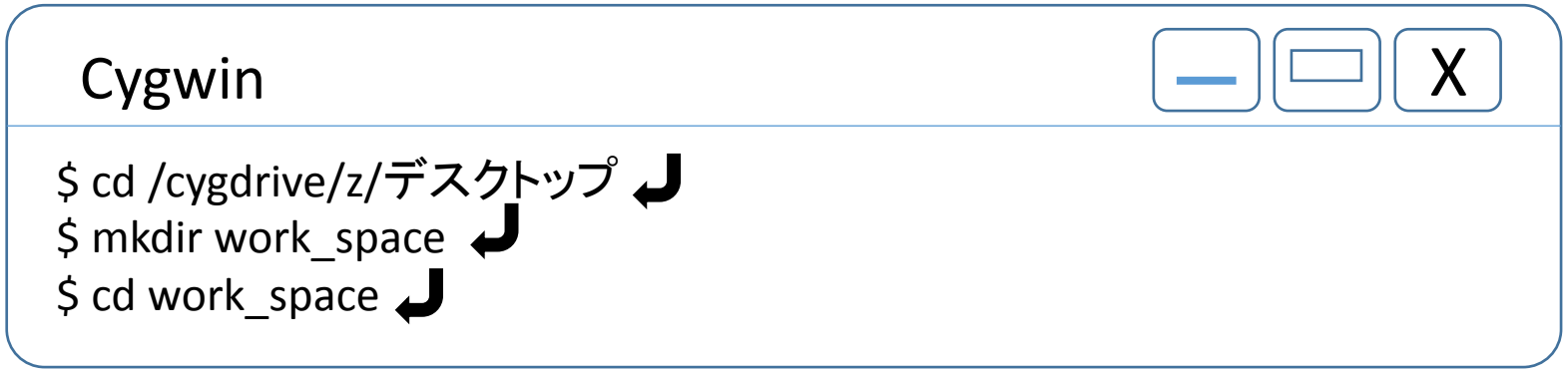
- cygwin
  windows上で動作するUnix環境の一つ

- www.cygwin.comで配布しているsetup.exeをダウンロード

- 実行してインストール

- Cygwinターミナル（端末）を起動
  スタートメニュー　→　2.ネットワークツール　→　仮想UNIX端末（cygwin64)

作業ディレクトリ作成



ファイルの取得

- Index用ゲノムreference
  マウス染色体1番
  http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz

- シーケンスデータ
  GEO データベースaccession number GSE65976
  SRR1508230.sra ,SRR1508234.sra
  SRR1805875.sra, SRR1805876.sra

課題配布フォルダ → bioinfojisyu → chr1.fa
part_SRR1805875.fastq

Bowtie

- マッピングツールの一つ
- Burrows Wheeler transformを利用している
- 高速である
- メモリの消費は少ない
- 並列化に対応している

Bowtie

- http://bowtie-bio.sourceforge.net/index.shtml

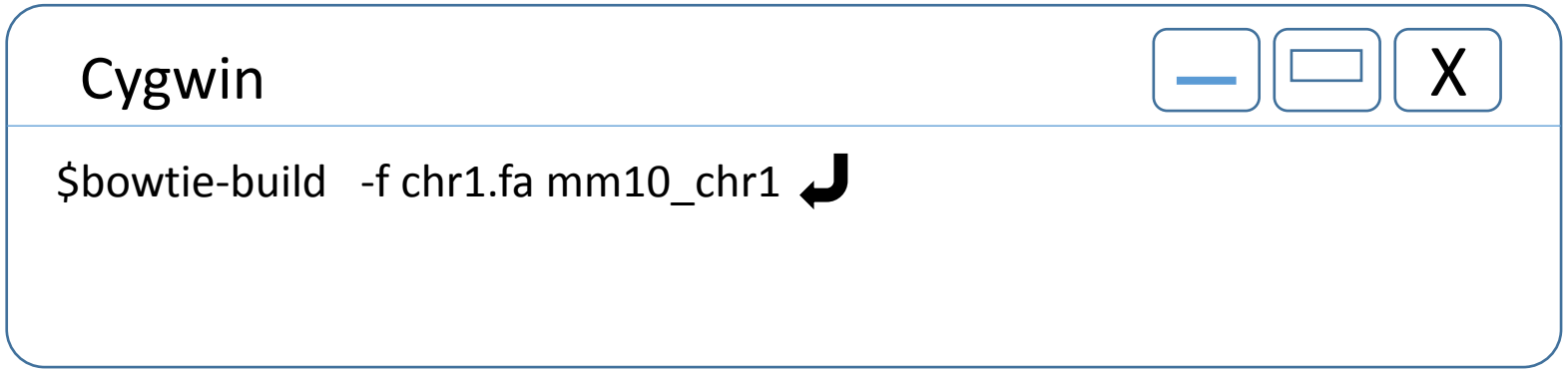
  Latest releaseから最新版をダウンロード

  bowtie-1.1.1-mingw.x86_64.zip

  展開して使用

  課題配布フォルダのbowtie-1.1.1フォルダを各自のデスクトップにコピー

Bowtieのコマンド1
Index作成



bowtie-build -f リファレンスファイル名 インデックス名
referenceのゲノム配列をBurrows-Wheeler変換を使ってインデックス化する
mm10_chr1.1.ebwt
mm10_chr1.2.ebwt
mm10_chr1.3.ebwt
mm10_chr1.4.ebwt
mm10_chr1.rev.1.ebwt
mm10_chr1.rev.2.ebwt
6個のファイルが作成される

Burrows Wheeler transform

- The BWT applies a reversible transformation to a block of input text. The transformation does not itself compress the data, but reorders it to make it easy to compress with simple algorithms such as move-to-front coding.

Burrows M, Wheeler DJ (1994) A block-sorting lossless data

compression algorithm. Technical report 124. Palo Alto, CA:

Digital Equipment Corporation.

Burrows Wheeler transform

original
X = abracadabra
末尾に＄を付ける
左に一文字ずつシフトして
ローテーション

a	b	r	a	c	a	d	a	b	r	a	$
b	r	a	c	a	d	a	b	r	a	$	a
r	a	c	a	d	a	b	r	a	$	a	b
a	c	a	d	a	b	r	a	$	a	b	r
c	a	d	a	b	r	a	$	a	b	r	a
a	d	a	b	r	a	$	a	b	r	a	c
d	a	b	r	a	$	a	b	r	a	c	a
a	b	r	a	$	a	b	r	a	c	a	d
b	r	a	$	a	b	r	a	c	a	d	a
r	a	$	a	b	r	a	c	a	d	a	b
a	$	a	b	r	a	c	a	d	a	b	r
$	a	b	r	a	c	a	d	a	b	r	a

Burrows Wheeler transform

アルファベット順にソート
＄はaよりも先

BWT(X) = ard$rcaaaabb

F											L
$	a	b	r	a	c	a	d	a	b	r	a
a	$	a	b	r	a	c	a	d	a	b	r
a	b	r	a	$	a	b	r	a	c	a	d
a	b	r	a	c	a	d	a	b	r	a	$
a	c	a	d	a	b	r	a	$	a	b	r
a	d	a	b	r	a	$	a	b	r	a	c
b	r	a	$	a	b	r	a	c	a	d	a
b	r	a	c	a	d	a	b	r	a	$	a
c	a	d	a	b	r	a	$	a	b	r	a
d	a	b	r	a	$	a	b	r	a	c	a
r	a	$	a	b	r	a	c	a	d	a	b
r	a	c	a	d	a	b	r	a	$	a	b

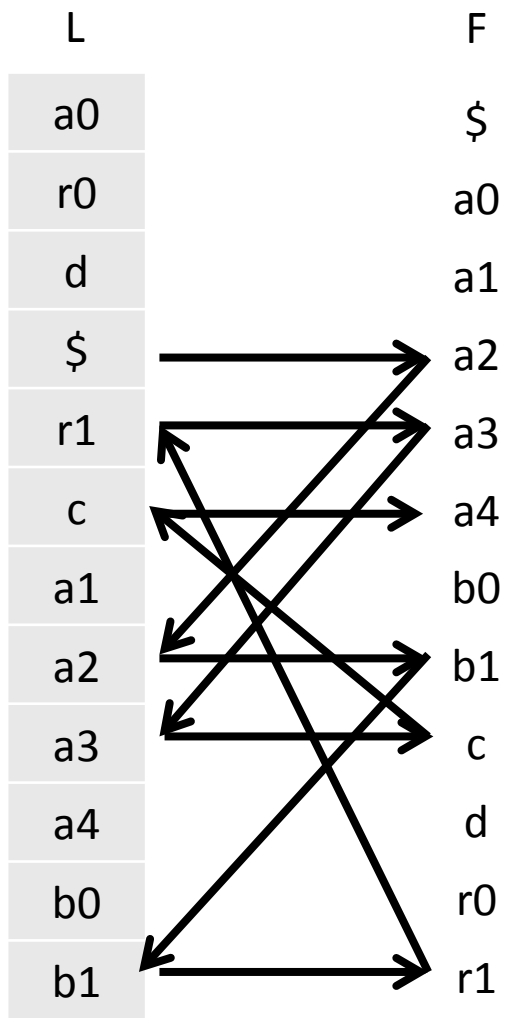
Burrows Wheeler 逆変換

（複数個ある
文字には番号を
付けてある）

L
a0
r0
d
$
r1
c
a1
a2
a3
a4
b0
b1

辞書式に
並べ替え

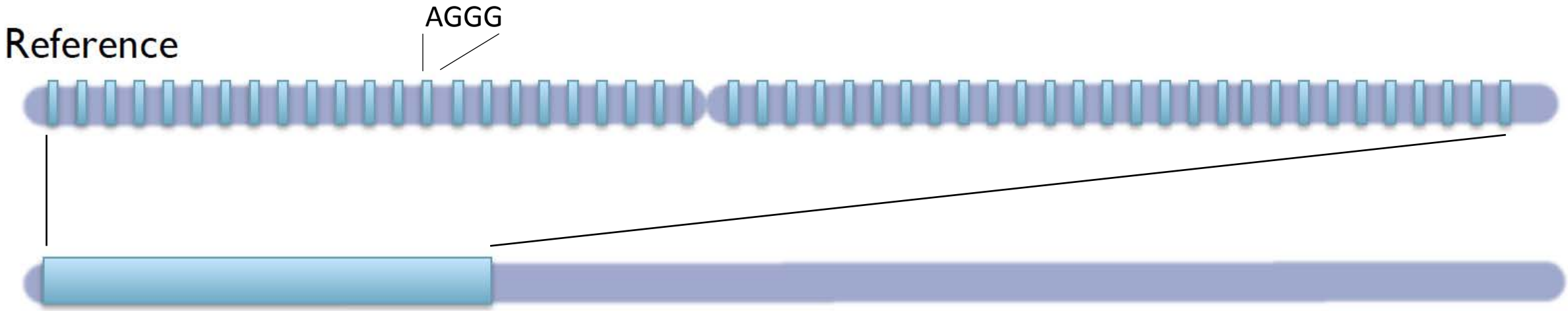
F
$
a0
a1
a2
a3
a4
b0
b1
c
d
r0
r1



LはFの一文字前
＄から出発して
L→F→L→L…と
たどっていくと
オリジナルの文字列を
復元できる

a2b1r1a3ca4…

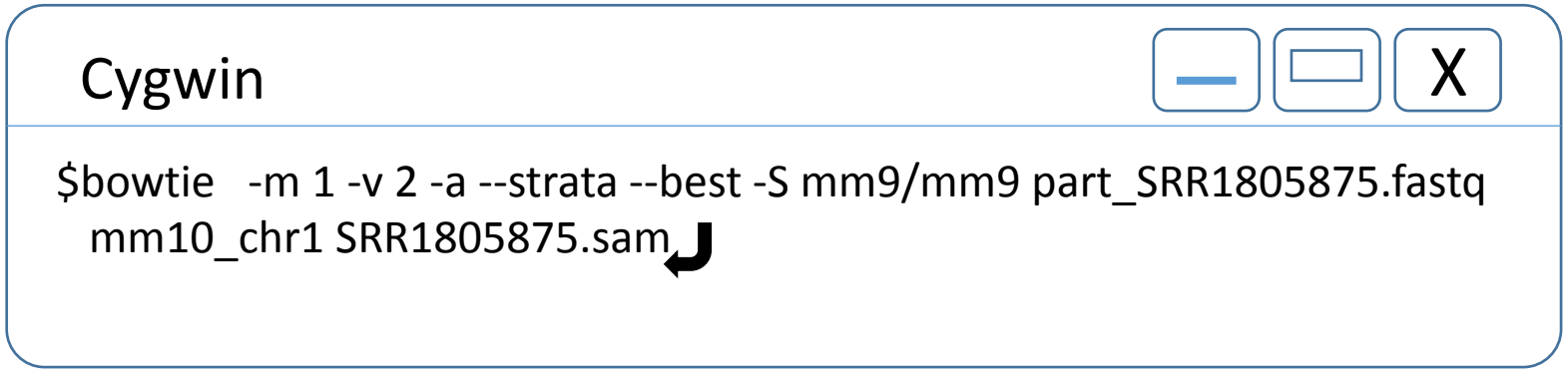
BWTの利点



同じ文字が固まる傾向がある
検索しやすい
簡単に復元できる

F L
GGGA ~ A
GGGC ~ A
GGGT ~ A
…

マッピング



bowtie (option) 参照インデックス名 fastqファイル名 出力ファイル名

-m 1　　: 1リードを1か所にマッピングする
-v 2　　: ミスマッチを2個まで許容する
-a　　:候補の配列を全て列挙する
--best　:ベストマッチの場所にマッピング
--strata :
-S　　: 結果をsamファイル形式で出力

Burrows Wheeler transform

original
X **= ctgaaactggt**

0	c	t	g	a	a	a	c	t	g	g	t	$
1	t	g	a	a	a	c	t	g	g	t	$	c
2	G	a	a	a	c	t	g	g	t	$	c	t
3	a	a	a	c	t	g	g	t	$	c	t	g
4	a	a	c	t	g	g	t	$	c	t	g	a
5	a	c	t	g	g	t	$	c	t	g	a	a
6	c	t	g	g	t	$	c	t	g	a	a	a
7	t	g	g	t	$	c	t	g	a	a	a	c
8	g	g	t	$	c	t	g	a	a	a	c	t
9	g	t	$	c	t	g	a	a	a	c	t	g
10	t	$	c	t	g	a	a	a	c	t	g	g
11	$	c	t	g	a	a	a	c	t	g	g	t

Burrows Wheeler transform

X **= ctgaaactggt$**
BWT(X) **= tgaa$attggcc**

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

Backward Search

ctgaaactggt$から
配列“ctgg”を検索

$\underline{R}(W)$: 文字Wが出現する最初の列

$\overline{R}(W)$: 文字Wが出現する最後の列

$\underline{R}(g) = 6$

$\overline{R}(g) = 8$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
➡ 6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
➡ 8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

Backward Search

$$\underline{R}(aW) = C(a) + O(a, \underline{R}(W) - 1) + 1$$

$$\overline{R}(aW) = C(a) + O(a, \overline{R}(W))$$

$C(a)$: aよりもアルファベット的に小さい文字数

$O(a, i)$: BWT列内でi列目までのaの数

$$\begin{aligned}\underline{R}(gg) &= C(g) + O(g, \underline{R}(g) - 1) + 1 \\ &= 5 + O(g, 6 - 1) + 1 \\ &= 5 + 1 + 1 \\ &= 7\end{aligned}$$

$$\begin{aligned}\overline{R}(gg) &= C(g) + O(g, \overline{R}(g)) \\ &= 5 + O(g, 8) \\ &= 5 + 2 \\ &= 7\end{aligned}$$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

backward search

$$\underline{R}(tgg) = C(t) + O(t, \underline{R}(gg) - 1) + 1$$
$$= 8 + O(t, 7 - 1) + 1$$
$$= 8 + 2 + 1$$
$$= 11$$

$$\overline{R}(tgg) = C(t) + O(t, \overline{R}(gg))$$
$$= 8 + O(t, 8)$$
$$= 8 + 3$$
$$= 11$$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

backward search

$$\underline{R}(ctgg) = C(c) + O(c, \underline{R}(tgg) - 1) + 1$$
$$= 3 + O(c, 11 - 1) + 1$$
$$= 3 + 1 + 1$$
$$= 5$$

$$\overline{R}(ctgg) = C(c) + O(c, \overline{R}(tgg))$$
$$= 3 + O(c, 11)$$
$$= 3 + 2$$
$$= 5$$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

backward search

$$\underline{R}(ttgg) = C(t) + O(t, \underline{R}(tgg) - 1) + 1$$
$$= 8 + O(t, 11 - 1) + 1$$
$$= 8 + 3 + 1$$
$$= 12$$

$$\overline{R}(ttgg) = C(t) + O(t, \overline{R}(tgg))$$
$$= 8 + O(t, 11)$$
$$= 8 + 2$$
$$= 11$$

$\underline{R}(aW) \le \overline{R}(aW)$: 配列aWが存在する条件

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

Backward Search

$$\underline{R}(ct) = C(c) + O(c, \underline{R}(t) - 1) + 1$$
$$= 3 + O(c, 9 - 1) + 1$$
$$= 3 + 0 + 1$$
$$= 4$$

$$\overline{R}(ct) = C(c) + O(c, \overline{R}(t))$$
$$= 3 + O(c, 11)$$
$$= 3 + 2$$
$$= 5$$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

Backward Search

$$\underline{R}(act) = C(a) + O(a, \underline{R}(ct) - 1) + 1$$
$$= 0 + O(a, 4 - 1) + 1$$
$$= 0 + 2 + 1$$
$$= 3$$

$$\overline{R}(act) = C(a) + O(a, \overline{R}(ct))$$
$$= 0 + O(a, 5)$$
$$= 0 + 3$$
$$= 3$$

0	$	c	t	g	a	a	a	c	t	g	g	t	11
1	a	a	a	c	t	g	g	t	$	c	t	g	3
2	a	a	c	t	g	g	t	$	c	t	g	a	4
3	a	c	t	g	g	t	$	c	t	g	a	a	5
4	c	t	g	a	a	a	c	t	g	g	t	$	0
5	c	t	g	g	t	$	c	t	g	a	a	a	6
6	g	a	a	a	c	t	g	g	t	$	c	t	2
7	g	g	t	$	c	t	g	a	a	a	c	t	8
8	g	t	$	c	t	g	a	a	a	c	t	g	9
9	t	$	c	t	g	a	a	a	c	t	g	g	10
10	t	g	a	a	a	c	t	g	g	t	$	c	1
11	t	g	g	t	$	c	t	g	a	a	a	c	7

第2回 予定
Integrative Genomics Viewer(IGV)による可視化

- samtoolsでファイル変換
  １）sam→bam変換
  ２）bamファイルを染色体順に並べ替え
  ３）indexファイルの作成

- IGVへアップロード、表示