

{ Exact matching 精确匹配 : backward search

{ Inexact matching 非精确匹配 : bounded traversal / backtracking

0 googol\$	0	6	\$googol	l	
1 oogol\$g	1	3	gol\$goo	0	$x = \text{googol\$}$
2 ugol\$go	2	0	googol\$	\$	$\downarrow \text{BWT}$
3 gol\$goo	3	5	l\$googo	0	$\Rightarrow B = lo\$oogg$
4 ol\$goog	4	2	ogol\$go	0	$\uparrow$
5 l\$googo	5	4	ol\$goog	g	$\text{BTi}]$
6 \$googol	6	1	oogol\$g	g	

$\Downarrow$  后缀字典序数组

the positions of the first symbols form the suffix array

得到  $S = [6, 3, 0, 5, 2, 4, 1]$

$\uparrow S(i)$

$\diagup SA$

同时我们可以得到一些结论  $BTi] = \$$ , 则  $STi] = 0$

$BTi] = X[S(i) - 1], BTi] \neq \$$

定义：后缀数组的上、下界；W 是 X 的子串

$R(W) = \min \{k : W \text{ is the prefix of } X[S(k)]\}$

$\bar{R}(W) = \max \{k : W \text{ is the prefix of } X[S(k)]\}$

举例： $W = go$   $R(go) = \min \{k : "go" \text{ is the prefix of } X[S(k)]\}$

$\bar{R}(go) = \max \{k : "go" \text{ is the prefix of } X[S(k)]\}$

$X_0 = googol \$$      $X_3 = gol \$$

$\Downarrow$

$\Downarrow$

$\Rightarrow SA \text{ interval } [1, 2]$

$S(k) = 0, k=2$      $S(k) = 3, k=1$      $R(go) = 1, \bar{R}(go) = 2$

针对空串， $L(w) = 0$ ,  $\bar{R}(w) = n-1$

对于  $x$  的空串，其 SA interval 为  $[0, b]$

知道 SA intervals 可以得到位于  $x$  的位置信息。因此，

序对比对等价于搜寻满足 query 的  $x$  的子串的 SA intervals

## 辅助数据

$C(a)$  be the number of symbols in  $x_{[0, n-2]}$  that are lexicographically smaller than  $a \in \Sigma$

$a$	$g$	$l$	$o$
$C(a)$	0	2	3

$O(a, i)$  the number of occurrences of  $a$  in  $B_{[0, i]}$ .

$O(a, i)$

		$g$	$l$	$o$
0	6 \$googol	1	0	0
1	3 gol\$go	0	0	1
2	0 googol\$	\$	0	1
3	5 l\$googo	⇒ 0	0	2
4	2 ogol\$go	0	0	3
5	4 ol\$goog	9	1	3
6	1 oogol\$g	9	2	3

另有  $O(a, -1) = 0$

精确匹配: backward search

Ferragina 和 Manzini 证明了

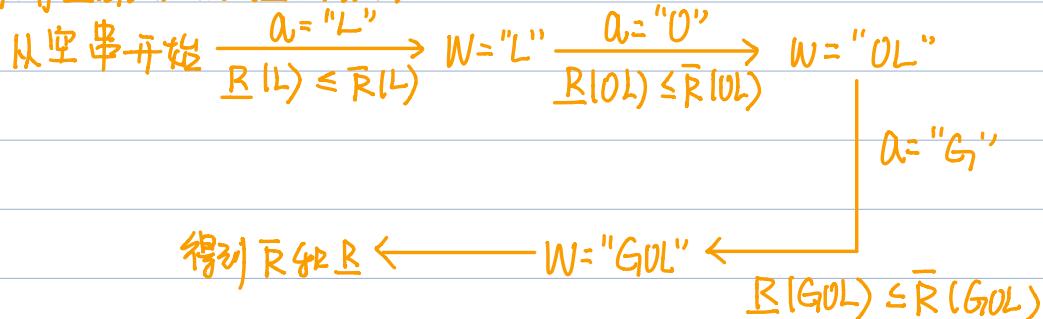
$$R(a) = (l(a) + O(a, R(w)) - 1) + 1$$

$$\bar{R}(a) = (l(a) + O(a, \bar{R}(w)))$$

对  $x$  的子串  $w$ , 如果  $R(a) \leq \bar{R}(aw)$ , 则  $aw$  也是  $x$  的子串  
以上的证明是实现精确匹配的前提

考虑到先前提到的 short-read / query: GOL.

精确匹配的流程大致如下:



非精确匹配: bounded traversal / backtracking

Precalculation:

Calculate BWT string for reference string  $x$

Calculate array  $C(\cdot)$  and  $O(\cdot, \cdot)$  from  $B$

Calculate BWT string  $B'$  for the reverse reference  $x'$

Calculate array  $O'(\cdot, \cdot)$  from  $B'$

非精确匹配的前提准备是计算  $O(\cdot, \cdot)$  和  $O'(\cdot, \cdot)$

并需要存储在内存当中, 供遍历查询

INEXACT SEARCH ( $w, z$ ) # $z$  表示最大差异数, 反 SNP 和 indel

CALCULATED ( $w$ ) # $D(i)$  是  $w[i : i]$  最大差异数的下边界

return INEX RECUR ( $w, |w|-1, z, 0, |x|-1$ )

CALCULATED( $W$ )

```
k ← 0  
l ← |X| - 1  
z ← 0  
for  $i = 0$  to  $|W| - 1$  do  
     $k \leftarrow C(W[i]) + O'(W[i], k - 1) + 1$   
     $l \leftarrow C(W[i]) + O'(W[i], l)$   
    if  $k > l$  then  
         $k \leftarrow 0$   
         $l \leftarrow |X| - 1$   
         $z \leftarrow z + 1$   
     $D(i) \leftarrow z$ 
```



CALCULATED( $W$ )

```
z ← 0  
j ← 0  
for  $i = 0$  to  $|W| - 1$  do  
    if  $W[j, i]$  is not a substring of  $X$  then  
         $z \leftarrow z + 1$   
         $j \leftarrow i + 1$   
     $D(i) \leftarrow z$ 
```

⇒ 当不使用  $D(i)$  时，即不对历史遍空间

$D(i)$  的存在使得索引空间得到优化，进行裁剪，令  $D(i) = 0$

以达到更佳的效率

```
INEXRECUR( $W, i, z, k, l$ )  
    if  $z < D(i)$  then  
        return  $\emptyset$   
    if  $i < 0$  then  
        return  $\{[k, l]\}$   
     $I \leftarrow \emptyset$  insertion  
    *  $I \leftarrow I \cup \text{INEXRECUR}(W, i - 1, z - 1, k, l)$   
    * for each  $b \in \{A, C, G, T\}$  do  
         $k \leftarrow C(b) + O(b, k - 1) + 1$   
         $l \leftarrow C(b) + O(b, l)$   
        if  $k \leq l$  then deletion  
             $I \leftarrow I \cup \text{INEXRECUR}(W, i, z - 1, k, l)$   
            if  $b = W[i]$  then  
                 $I \leftarrow I \cup \text{INEXRECUR}(W, i - 1, z, k, l)$   
            else  
                 $I \leftarrow I \cup \text{INEXRECUR}(W, i - 1, z - 1, k, l)$   
return  $I$ 
```

两个中止条件

stop 1：最大允许的差异数  $Z$  小于下边界

不存在可能的匹配

stop 2：短序列都已被遍历

当不考虑 星号两行的时候，仅仅实现

错配即 SNP