
RLPBWT con bitvectors

Algorithm 1 Algoritmo per estrazione simbolo da una run in una colonna

```
1: function GET_SYMBOL( $s, r$ )  $\triangleright s = \top$  iff column start with 0,  $r$  run index
2:   if  $s$  then
3:     if  $r \bmod 2 = 0$  then return '0' else return '1'
4:   else
5:     if  $r \bmod 2 = 0$  then return '1' else return '0'
```

Algorithm 2 Algoritmo per uvtrick

```
1: function UVTRICK( $k, i$ )  $\triangleright k$  is column index,  $i$  row index
2:   if  $i = 0$  then
3:     return (0, 0)
4:    $run \leftarrow rank_h^k(i)$ 
5:   if  $run = 0$  then
6:     if  $rlpbwt[k].start$  then
7:       return ( $index$ , 0)
8:     else
9:       return (0,  $index$ )
10:  else if  $run = 1$  then
11:    if  $rlpbwt[k].start$  then
12:      return ( $select_h^k(run) + 1$ ,  $i - (select_h^k(run) + 1)$ )
13:    else
14:      return ( $i - (select_h^k(run) + 1)$ ,  $select_h^k(run) + 1$ )
15:  else
16:    if  $run \bmod 2 = 0$  then
17:       $pre_u \leftarrow select_u^k(\frac{run}{2}) + 1$ 
18:       $pre_v \leftarrow select_v^k(\frac{run}{2}) + 1$ 
19:       $offset \leftarrow i - (select_h^k(run) + 1)$ 
20:      if  $rlpbwt[k].start$  then
21:        return ( $pre_u + offset$ ,  $pre_v$ )
22:      else
23:        return ( $pre_u$ ,  $pre_v + offset$ )
24:    else
25:       $run_u \leftarrow (\frac{run}{2}) + 1$ 
26:       $run_v \leftarrow \frac{run}{2}$ 
27:      if  $\neg rlpbwt[k].start$  then
28:         $swap(run_u, run_v)$ 
29:       $pre_u \leftarrow select_u^k(run_u) + 1$ 
30:       $pre_v \leftarrow select_v^k(run_v) + 1$ 
31:       $offset \leftarrow i - (select_h^k(run) + 1)$ 
32:      if  $rlpbwt[k].start$  then
33:        return ( $pre_u$ ,  $pre_v + offset$ )
34:      else
35:        return ( $pre_u + offset$ ,  $pre_v$ )
```

Algorithm 3 Algoritmo per lf-mapping

```
1: function LF( $k, i, s$ )  $\triangleright k$  is column index,  $i$  row index,  $s$  symbol
2:    $c \leftarrow rlpbwt[k].c$ 
3:    $(u, v) \leftarrow uvtrick(k, i)$ 
4:   if  $s = 0$  then
5:     return  $u$ 
6:   else
7:     return  $c + v$ 
```

Algorithm 4 Algoritmo per lf-mapping inverso

```
1: function REVERSE_LF( $k, i$ )  $\triangleright k$  is column index,  $i$  row index
2:   if  $k = 0$  then  $\triangleright$  by design
3:     return 0
4:    $k \leftarrow k - 1$ 
5:    $c \leftarrow rlpbwt[k].c$ 
6:   if  $i < c$  then
7:     if  $rlpbwt[k].start$  then
8:        $run \leftarrow rank_u^k(i) \cdot 2$ 
9:     else
10:       $run \leftarrow rank_u^k(i) \cdot 2 + 1$ 
11:      $i_{run} \leftarrow 0$ 
12:     if  $run \neq 0$  then
13:        $i_{run} \leftarrow select_h^k(run) + 1$ 
14:      $(prev_0, \_) \leftarrow uvtrick(k, i_{run})$ 
15:     return  $i_{run} + (i - prev_0)$ 
16:   else
17:     if  $rlpbwt[k].start$  then
18:        $run \leftarrow rank_v^k(i) \cdot 2 + 1$ 
19:     else
20:        $run \leftarrow rank_v^k(i) \cdot 2$ 
21:      $i_{run} \leftarrow 0$ 
22:     if  $run \neq 0$  then
23:        $i_{run} \leftarrow select_h^k(run) + 1$ 
24:      $(\_, prev_1) \leftarrow uvtrick(k, i_{run})$ 
25:     return  $i_{run} + (i - (c + prev_1))$ 
```

Algorithm 5 Algoritmo per match con aplotipo esterno con panel $width \times height$

```

1: function EXTERNAL_MATCHES( $z$ ) ▷ assuming  $|z| = rlpbwt.width$ 
2:    $f \leftarrow 0, f_{run} \leftarrow 0, f' \leftarrow 0$ 
3:    $g \leftarrow 0, g_{run} \leftarrow 0, g' \leftarrow 0$ 
4:    $e \leftarrow 0, l \leftarrow 0$ 
5:   for every  $k \in [0, |z|)$  do
6:      $f_{run} \leftarrow rank_h^k(f), g_{run} \leftarrow rank_h^k(g)$ 
7:      $f' \leftarrow lf(k, f, z[k]), g' \leftarrow lf(k, g, z[k])$ 
8:      $l \leftarrow g - f$ 
9:     if  $f' < g'$  then
10:       $f \leftarrow f', g \leftarrow g'$ 
11:     else
12:       if  $k \neq 0$  then
13:         report matches in  $[e, k - 1]$  with  $l$  haplotypes
14:       if  $f' = |lcp^{k+1}|$  then
15:          $e \leftarrow k + 1$ 
16:       else
17:          $e \leftarrow lcp^{k+1}[f']$ 
18:       if  $(z[e] = 0 \wedge f' > 0) \vee f' = height$  then
19:          $f' \leftarrow g' - 1$ 
20:         if  $e \geq 1$  then
21:            $f_{rev} \leftarrow f', k' \leftarrow k + 1$ 
22:           while  $k' \neq e - 1$  do
23:              $f_{rev} \leftarrow reverse\_lf(k', f_{rev}), k' \leftarrow k' - 1$ 
24:            $run \leftarrow rank_h^{k'}(f_{rev}), symb \leftarrow get\_symbol(start^{k'}, run)$ 
25:           while  $e > 0 \wedge z[e - 1] = symb$  do
26:              $f_{rev} \leftarrow reverse\_lf(e, f_{rev})$ 
27:              $run \leftarrow rank_h^{e-1}(f_{rev})$ 
28:              $symb \leftarrow get\_symbol(start^{e-1}, run)$ 
29:           while  $f' > 0 \wedge (k + 1) - lcp^{k+1}[f] \leq e$  do  $e \leftarrow e - 1$ 
30:            $f \leftarrow f', g \leftarrow g'$ 
31:       else
32:          $g' \leftarrow f' - 1$ 
33:         if  $e \geq 1$  then
34:            $f_{rev} \leftarrow f', k' \leftarrow k + 1$ 
35:           while  $k' \neq e - 1$  do
36:              $f_{rev} \leftarrow reverse\_lf(k', f_{rev}), k' \leftarrow k' - 1$ 
37:            $run \leftarrow rank_h^{k'}(f_{rev}), symb \leftarrow get\_symbol(start^{k'}, run)$ 
38:           while  $e > 0 \wedge z[e - 1] = symb$  do
39:              $f_{rev} \leftarrow reverse\_lf(e, f_{rev})$ 
40:              $run \leftarrow rank_h^{e-1}(f_{rev})$ 
41:              $symb \leftarrow get\_symbol(start^{e-1}, run)$ 
42:           while  $e < height \wedge (k + 1) - lcp^{k+1}[e] \leq e$  do  $e \leftarrow e + 1$ 
43:            $f \leftarrow f', g \leftarrow g'$ 
44:   if  $f < g$  then
45:      $l \leftarrow g - f$ 
46:   report matches in  $[e, |z| - 1]$  with  $l$  haplotypes

```
