RLPBWT con bitvectors

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

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

Algorithm 2 Algoritmo per uvtrick

```
function UVTRICK($k, i$)    ▷ $k$ is column index, $i$ row index
    if $i = 0$ then
        return $(0,\ 0)$
    $run \leftarrow rank_h^k(i)$
    if $run = 0$ then
        if $rlpbwt[k].start$ then
            return $(index,\ 0)$
        else
            return $(0,\ index)$
    else if $run = 1$ then
        if $rlpbwt[k].start$ then
            return $(select_h^k(run) + 1,\ i - (select_h^k(run) + 1))$
        else
            return $(i - (select_h^k(run) + 1),\ select_h^k(run) + 1)$
    else
        if $run \bmod 2 = 0$ then
            $pre_u \leftarrow select_u^k(\frac{run}{2}) + 1$
            $pre_v \leftarrow select_v^k(\frac{run}{2}) + 1$
            $offset \leftarrow i - (select_h^k(run) + 1)$
            if $rlpbwt[k].start$ then
                return $(pre_u + offset,\ pre_v)$
            else
                return $(pre_u,\ pre_v + offset)$
        else
            $run_u \leftarrow (\frac{run}{2}) + 1$
            $run_v \leftarrow \frac{run}{2}$
            if $\neg rlpbwt[k].start$ then
                $swap(run_u, run_v)$
            $pre_u \leftarrow select_u^k(run_u) + 1$
            $pre_v \leftarrow select_v^k(run_v) + 1$
            $offset \leftarrow i - (select_h^k(run) + 1)$
            if $rlpbwt[k].start$ then
                return $(pre_u,\ pre_v + offset)$
            else
                return $(pre_u + offset,\ pre_v)$
```

Algorithm 3 Algoritmo per lf-mapping

```
function LF($k, i, s$)                              ▷ $k$ is column index, $i$ row index, $s$ symbol
    $c \leftarrow rlpbwt[k].c$
    $(u, v) \leftarrow uvtrick(k,\ i)$
    if $s = 0$ then
        return $u$
    else
        return $c + v$
```

Algorithm 4 Algoritmo per lf-mapping inverso

```
function REVERSE_LF($k, i$)                         ▷ $k$ is column index, $i$ row index
    if $k = 0$ then                                 ▷ by design
        return 0
    $k \leftarrow k - 1$
    $c \leftarrow rlpbwt[k].c$
    if $i < c$ then
        if $rlpbwt[k].start$ then
            $run \leftarrow rank_u^k(i) \cdot 2$
        else
            $run \leftarrow rank_u^k(i) \cdot 2 + 1$
        $i_{run} \leftarrow 0$
        if $run \neq 0$ then
            $i_{run} \leftarrow select_h^k(run) + 1$
        $(prev_0,\ \_) \leftarrow uvtrick(k,\ i_{run})$
        return $i_{run} + (i - prev_0)$
    else
        if $rlpbwt[k].start$ then
            $run \leftarrow rank_v^k(i) \cdot 2 + 1$
        else
            $run \leftarrow rank_v^k(i) \cdot 2$
        $i_{run} \leftarrow 0$
        if $run \neq 0$ then
            $i_{run} \leftarrow select_h^k(run) + 1$
        $(\_,\ prev_1) \leftarrow uvtrick(k,\ i_{run})$
        return $i_{run} + (i - (c + prev_1))$
```

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

```
function EXTERNAL_MATCHES($z$)                              ▷ assuming $|z| = rlpbwt.width$
    $f \leftarrow 0,\ f_{run} \leftarrow 0,\ f' \leftarrow 0$
    $g \leftarrow 0,\ g_{run} \leftarrow 0,\ g' \leftarrow 0$
    $e \leftarrow 0,\ l \leftarrow 0$
    for $every\ k \in [0,\ |z|)$ do
        $f_{run} \leftarrow rank_h^k(f),\ g_{run} \leftarrow rank_h^k(g)$
        $f' \leftarrow lf(k,\ f,\ z[k]),\ g' \leftarrow lf(k,\ g,\ z[k])$
        $l \leftarrow g - f$
        if $f' < g'$ then
            $f \leftarrow f',\ g \leftarrow g'$
        else
            if $k \neq 0$ then
                report $matches\ in\ [e,\ k-1]\ with\ l$ haplotypes
            if $f' = |lcp^{k+1}|$ then
                $e \leftarrow k + 1$
            else
                $e \leftarrow lcp^{k+1}[f']$
            if $(z[e] = 0 \land f' > 0) \lor f' = height$ then
                $f' \leftarrow g' - 1$
                if $e \geq 1$ then
                    $f_{rev} \leftarrow f',\ k' \leftarrow k + 1$
                    while $k' \neq e - 1$ do
                        $f_{rev} \leftarrow reverse\_lf(k',\ f_{rev}),\ k' \leftarrow k' - 1$
                    $run \leftarrow rank_h^{k'}(f_{rev}),\ symb \leftarrow get\_symbol(start^{k'}, run)$
                    while $e > 0 \land z[e-1] = symb$ do
                        $f_{rev} \leftarrow reverse\_lf(e,\ f_{rev})$
                        $run \leftarrow rank_h^{e-1}(f_{rev})$
                        $symb \leftarrow get\_symbol(start^{e-1}, run)$
                while $f' > 0 \land (k+1) - lcp^{k+1}[f] \leq e$ do $e \leftarrow e - 1$
                $f \leftarrow f',\ g \leftarrow g'$
            else
                $g' \leftarrow f' - 1$
                if $e \geq 1$ then
                    $f_{rev} \leftarrow f',\ k' \leftarrow k + 1$
                    while $k' \neq e - 1$ do
                        $f_{rev} \leftarrow reverse\_lf(k',\ f_{rev}),\ k' \leftarrow k' - 1$
                    $run \leftarrow rank_h^{k'}(f_{rev}),\ symb \leftarrow get\_symbol(start^{k'}, run)$
                    while $e > 0 \land z[e-1] = symb$ do
                        $f_{rev} \leftarrow reverse\_lf(e,\ f_{rev})$
                        $run \leftarrow rank_h^{e-1}(f_{rev})$
                        $symb \leftarrow get\_symbol(start^{e-1}, run)$
                while $e < height \land (k+1) - lcp^{k+1}[e] \leq e$ do $e \leftarrow e + 1$
                $f \leftarrow f',\ g \leftarrow g'$
    if $f < g$ then
        $l \leftarrow g - f$
        report $matches\ in\ [e,\ |z|-1]\ with\ l$ haplotypes
```