Christina Boucher, Travis Gagie, Tomohiro I, Dominik Köppl, Ben Langmead, Giovanni Manzini, Gonzalo Navarro, Alejandro Pacheco, Massimiliano Rossi

PHONI
Streamed Matching Statistics with Multi-Genome References
matching statistics (MS)

how fits banana into bandana?

why? MS $\Rightarrow$ maximal exact matches (MEMs) $\Rightarrow$ seed and extend $\Rightarrow$ read alignment
matching statistics (MS)

- text $T$
- pattern $P$

matching statistics ($R$, $L$) is

- $P[i .. i + L[i]-1] = T[R[i]..R[i]+L[i]-1]$
- $P[i .. i + L[i]]$ does not occur

there is no $P[2..5] = anan$ in $T$
matching statistics (MS)

T = b a n d a n a

P = b a n a n a

longest prefix of $P[1..]$ occurring in $T$

$T = b a n d a n a$

$P = b a n a n a$

$(1,3)$
matching statistics (MS)

\[ T = \text{banana} \]

\[ P = \text{banana} \]

- \((1,3)\) directed from 1 to 3
- \((5,3)\) directed from 5 to 3
- \((6,2)\) directed from 6 to 2
- \((2,1)\) directed from 2 to 1
matching statistics (MS)

\[ T = \text{banana} \]
\[ P = \text{banana} \]
\[ R = 1 \ 5 \ 6 \ 5 \ 6 \ 2 \]
\[ L = 3 \ 3 \ 2 \ 3 \ 2 \ 1 \]

obtain MS
matching statistics (MS)

\[ T = \text{banana} \]

\[ P = \text{banana} \]

\[ R \text{ not uniquely defined} \]

(2,1) (5,1) (7,1)

(2,1)
## MS computation

<table>
<thead>
<tr>
<th>used data structure</th>
<th>space in bits</th>
<th>time build</th>
<th>time query</th>
<th>authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>suffix tree (ST)</td>
<td>$O(n \lg n)$</td>
<td>$O(n)$</td>
<td>$O(</td>
<td>P</td>
</tr>
<tr>
<td>compressed ST (CST)</td>
<td>$O(n \lg \sigma)$</td>
<td>$O(n)$</td>
<td>$O(</td>
<td>P</td>
</tr>
<tr>
<td>$r$-index + grammar</td>
<td>$O(r \lg n + \ z \ lg^2 n)$</td>
<td>$O(n \lg r)$</td>
<td>$O(</td>
<td>P</td>
</tr>
</tbody>
</table>

$n = |T|$, $\sigma$: alphabet size, $r$: #runs in BWT, $z$: #LZ77 factors
space important?

construction of CST with

- $T$: up to 1000x Chromosome 19 samples
- 64 GB of RAM available

⇒ can index only 64 sequences!

- Chromosome 19 needs ~ 60 MB in ASCII
- CST implementation: cst_sct3 of sdsl-lite
space important?

MONI [Rossi+ RECOMB '21]:

- $r$-index [Gagie+ '20],
- Big BWT [Boucher+ '19],
- and data structures for MS memory requirement scales roughly logarithmic!

![Graph showing memory requirements](image)
MONI: augmented $r$-index

steps:

- determine $R$ by backward search
- then compute $L$:
  - scan $R$ and $P$ from left to right
  - random access to $T$ for computing $L[i] = \text{LCP}(T[R[i]..], P[i..])$
- needs to store $P$ and $R$
- for large $P$: streaming $P$ and MS becomes interesting

idea of PHONI:
compute $L$ directly with a grammar index
### MS computation

<table>
<thead>
<tr>
<th>BWT</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>$a$</td>
</tr>
<tr>
<td>n</td>
<td>$a$</td>
</tr>
<tr>
<td>d</td>
<td>$a$</td>
</tr>
<tr>
<td>b</td>
<td>andana$</td>
</tr>
<tr>
<td>$</td>
<td>bandana$</td>
</tr>
<tr>
<td>n</td>
<td>dana$</td>
</tr>
<tr>
<td>a</td>
<td>na$</td>
</tr>
<tr>
<td>a</td>
<td>ndana$</td>
</tr>
</tbody>
</table>

For this talk simplified:

- BWT instead of $r$-index
- Only compute $L$
- Compute $R$ with suffix array (SA)

($r$-index: SA entries for each run boundary)
backward steps

\[
\begin{array}{c}
\text{BWT} \\
a \\
n \\
d \\
b \\
$ \\
n \\
a \\
a \\
\end{array}
\quad F
\begin{array}{c}
$ \\
a$ \\
ona$ \\
ana$ \\
andana$ \\
badowa$ \\
andana$ \\
nona$ \\
da$ \\
\end{array}
\]

$P = \text{banana} \text{ ana} \text{ana} \text{ana}$

$L = \text{stop because } d \neq n$
matching pair

\[
P = \text{b a } \boxed{n} \text{ a n a}
\]

\[
LCP(P[i..], T[q'..]) \text{ and } LCP(P[i..], T[q''..])
\]

which is longer?

\[
\Rightarrow \text{continue with } q'
\]

continue with \(q'\) or \(q''\):

closest neighbors of \(q\) in BWT with letter \(n\)

\[
LCP: \ 0
\]

\[
LCP: \ 1
\]
continue backward steps

BWT
and
a
b
$
n
a
a

F
$
a$
ana$
andana$
bandana$
dana$
a$
da$
ndana$

LCP: 1

P = b a n a n a

L = 2 3 2 1
continue backward steps

BWT

\[
\begin{array}{c}
\text{a} \\
\text{nd} \\
\text{b} \\
\text{$} \\
\text{n} \\
\text{a} \\
\text{a}
\end{array}
\quad F
\begin{array}{c}
\text{$} \\
\text{a$} \\
\text{ana$} \\
\text{andana$} \\
\text{dana$} \\
\text{na$} \\
\text{ndana$}
\end{array}
\]

\[P = \text{bananaanaa}\]
\[L = \text{2 3 2 1}\]
find continuation again

BWT
- db
- a
- anda$a$
- bandana$
- dana$
- na$
- ndana$

\(F\)
- $
- a$
- a$
- ana$
- andana$
- bandana$
- dana$
- na$
- ndana$

\(P = b\ a\ n\ a\ n\ a\)

\(L = 3\ 3\ 2\ 3\ 2\ 1\)

LCP: 2

we want to stream \(P\), so we have not \(P\) for LCP queries!
from LCP to LCE

\[
P = \text{bananaana}
\]

\[
L = 3 \quad 3 \quad 2 \quad 3 \quad 2 \quad 1
\]

actually: can use previous BWT position for LCP computation

⇒ LCE queries
LCE grammar

grammar answering longest common extension (LCE) queries

- use RePair + prefix free parsing [Gagie+ ’19]
- random access on SLP [Gagie+ ’20]
  SLP = straight line program (special kind of grammar)
- already used in MONI for random access on $T$
prefix free parsing (PFP)

- factorize $T$ context-sensitively
- same substrings have nearly same factorization
prefix free parsing (PFP)

- build grammar on each factor $F_x$ independently
- build grammar on roots

$T =$

\[
\begin{array}{ccccc}
F_1 & F_2 & F_3 & F_4 & F_5 \\
\end{array}
\]
LCE(\(p_1, p_2\)) = LCP(T [\(p_1..\), T [\(p_2..\)])

- traverse from root down
- compare character-wise
but this is slow

- slower than MONI
- the larger $T$ the faster the execution of PHONI

why is the latter?
- the larger $T$ the less likely backward search fails

$P = \text{one of 10x Chromosome 19 sequences not in } T$
faster LCE queries

- character-wise comparison will hit factor boundary at the same time

⇒ ascend and compare node by node!
with faster LCEs ...

PHONI faster than MONI at $T = 1000$ sequences!

(y axis is closer zoomed)
MONI / PHONI : build dependencies
index construction

\( T \) consists of multiple Chromosome 19 sequences

gap to MONI due to thresholds

# sequences in \( T \)

time

total time [min]

CST
MONI
PHONI

space

memory [GB]

# sequences in \( T \)

MONI
PHONI
maximal RAM usage during queries

MONI additionally needs

- thresholds,
- each pattern and its $R$

stored in RAM

\[ P = \text{one of 10x Chromosome 19 sequences not in } T \]
maximal RAM usage during queries

- fix $T = 64$ sequences
- let $P = (P_1, \ldots, P_{10})$
- compute MS for the prefix of $P_i$ covering $x\%$ of $P_i$
what is PHONI?

• computation of matching statistics for highly repetitive $T$ (e.g. $T =$ pan-genome)

• stands on the shoulders of giants:
  - $r$-index [Gagie+ '20] [Bannai+ '20]
  - Big BWT [Boucher+ '19]
  - PFP grammar [Gagie+ '20]

our contribution:

• LCE queries on PFP grammars

• theoretically inferior to MONI, but practically competitive if
  - $P$ is large : since we can stream $P$, and
  - large parts of $P$ occur in $T$ ⇒ only few LCE queries