

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

BIOINFORMATIKA
**IZGRADNJA FM-INDEKSA KORIŠTENJEM
STABLA VALIĆA**

Danijel Sokač
Pamela Miletić

Zagreb, siječanj 2015.

Sadržaj

Uvod.....	3
Opis algoritma.....	4
FM-indeks.....	4
Sufiksno polje.....	4
Kärkkäinen-Sandersov algoritam.....	5
LF-tablica.....	5
Stablo valića.....	5
PrefixSum-tablica.....	5
2.6 FM indeks i Burrows-Wheeler transformacija.....	5
2.6.1 Burrows-Wheeler transformacija.....	5
Primjer izvođenja algoritma.....	7
Kärkkäinen-Sandersov algoritam.....	7
Konstrukcija BWT iz dobivenog sufiksnog polja.....	7
Stablo valića.....	8
PrefixSum-tablica.....	10
Testiranje.....	11
Zaključak.....	14
Literatura.....	15

1. Uvod

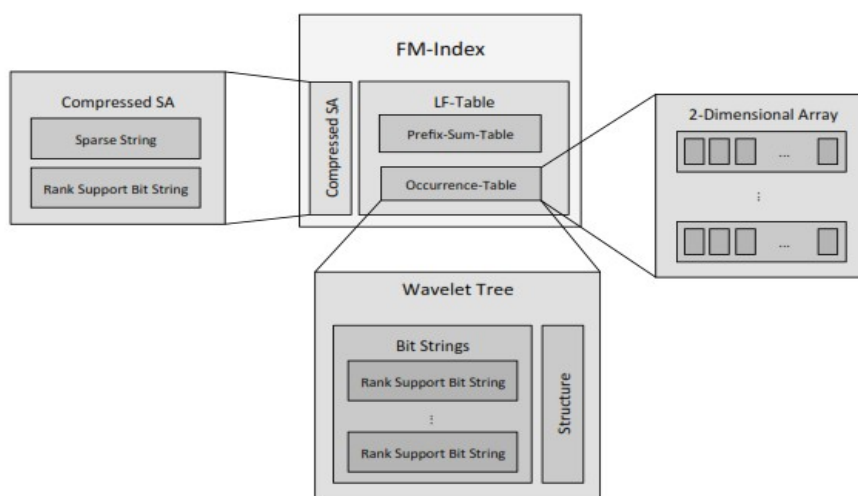
Nova otkrića u svijetu genoma sa sobom povlače i nova pitanja. Proučava se sve veći spektar podataka, a njih je potrebno negdje i pohraniti, ali i na efikasan i brz način doći do željenih informacija. Vrlo često istraživanja zahtijevaju identifikaciju sekvenci u genomima koji su od određenog značaja, odnosno dekodiranje podataka skrivenih u genomu. Određeni pristupi pomoću kojih se ovo pokušavalo postići, trošili su previše memorije budući da je potrebno analizirati veliku količinu podataka. Problem je riješen korištenjem FM indeksa.

FM indeks je memorijski učinkovit i vrlo brzi pristup pretraživanju uzoraka. Može se koristiti i u druge svrhe, ne samo za istraživanje sekvenci u genomima. Ovo je dobro rješenje za bilo koju aplikaciju u kojoj je potrebno pronaći željeni podniz unutar teksta, i to efikasno i u kratkome vremenu.

U ovome radu smo implementirali jedno takvo rješenje koje koristi FM-indeks te ga testirali koristeći genome bakterija. U nastavku je opisan algoritam i prikazani su koraci algoritma na jednostavno primjeru.

2. Opis algoritma

Cilj algoritma je omogućiti brzu pretragu bilo kojeg podniza u tekstu. U tu svrhu koristi se FM-indeks koji se sastoji od LF-tablice i sufiksnog polja. LF-tablica pak sadrži PrefixSum-tablicu i stablo valića (engl. *Wavelet Tree*). Navedeno se može vidjeti na idućoj slici (Sl 2.1).



Slika 2.1. FM-indeks

2.1 FM-indeks

FM-indeks (engl. *The Full-Text-Minut-Space index*) je podatkovna struktura izgrađena nad tekstom T koja podržava efikasno pretraživanje podnizova u T . FM-indeks je također i memorijski učinkovit. Ovaj indeks je zapravo kolekcija tablica s dodatnim funkcionalnostima kao što je prikazano na slici 2.1. FM-indeks sadrži LF-tablicu i sufiksno polje pomoću kojih se može odrediti polazni tekst ili pronaći odgovarajući podniz unutar teksta.

2.2 Sufiksno polje

Sufiksno polje služi za određivanje točne pozicije podniza unutar teksta. Međutim, nije potrebno čuvati pozicije svih sufiksa od T jer LF-mapiranje omogućava rekonstrukciju podniza od T počinjući od bilo koje pozicije.

Sufiksno polje se izgrađuje tako da se na kraj niza dodaje znak $\$$ koji je leksikografski manji od ostalih znakova iz abecede. Zatim se određuju svi sufiksi niza, te se oni poredaju po abecedi.

U najjednostavnijem obliku, sufiksno polje može biti konstruirano za string $S[1..N]$ na način da se [1]:

1. Konstruira polje pokazivača za sve sufikse $S[1..N]$, $S[2..N]$, ..., $S[N..N]$.
2. Sortiraju se pokazivači po leksikografskom redoslijedu sufiksa.

Primjer sortiranja stringa 'mississippi' sa znakom $\$$ pokazano je na idućoj slici (Sl 2.2.1).

SA			SA
1	mississippi\$	12	\$
2	ississippi\$	11	i\$
3	ssissippi\$	8	ippi\$
4	sissippi\$	5	issippi\$
5	issippi\$	2	ississippi\$
6	ssippi\$	1	mississippi\$
7	sippi\$	10	pi\$
8	ippi\$	9	ppi\$
9	ppi\$	7	sippi\$
10	pi\$	4	ssissippi\$
11	i\$	6	ssippi\$
12	\$	3	ssissippi\$

Slika 2.2.1 Sortiranje

U ovome radu smo koristili Kärkkäinen-Sandersov algoritam za izgradnju sufiksnog polja.

2.2.1 Kärkkäinen-Sandersov algoritam

Za neki niz S duljine n nad abecedom Σ provode se idući koraci:

1. Rekurzivno se sortira $2n/3$ sufiksa s_i ($i \bmod 3 \neq 0$, tj. $i = 1, 2$);
2. Sortira se $n/3$ sufiksa s_i ($i \bmod 3 = 0$) koristeći rezultat koraka (1);
3. Napraviti spajanje sortiranih polja dobivenih u koracima (1) i (2).

2.3 LF-tablica

LF-mapiranje (eng. *LF-mapping*; *Last-to-Front mapping*; *Last-to-First mapping*) opisuje vezu, odnosno mapiranje, zadnjeg i prvog stupca u listi leksikografski sortiranih cikličkih rotacija niza S , a temelji se na sljedećem: i -ta pojava znaka c u zadnjem stupcu (stupcu L) leksikografski sortiranih cikličkih rotacija odgovara i -toj pojavi znaka c u prvom stupcu (stupcu F). Dakle, $BWT[i]$ se nalazi u stupcu F na mjestu $LF[i]$.

LF-mapiranjem možemo iz niza B , uz prisutnost stupca F , izgraditi početni niz S . Prvi stupac F sačinjavaju leksikografski sortirani znakovi iz S , a time i B , što znači da nam je samo B dovoljan da bi odredili S . Sama LF-tablica se sastoji od Prefix-Sum tablice i tablice pojava koja može biti izgrađena kao 2D polje ili kao stablo valića.

2.4 Stablo valića

Stablo valića (engl. *Wavelet Tree*) je binarno stablo niza bitova koje predstavlja tekst T . To je posebna struktura podataka koja, kada se sprema korištenjem RRR sekvenci, može riješiti operacije u vremenu $O(\log A)$. A predstavlja veličinu abecede. RRR vrši kompresiju stabla valića.

Stablo valića omogućava dohvat broja pojavljivanja znaka do određenog indeksa.

2.5 PrefixSum-tablica

Svaki element ove tablice predstavlja broj znakova u T koji su leksikografski prije određenog znaka.

2.6 FM indeks i Burrows-Wheeler transformacija

FM indeks (engl. *Full-text indeks in Minute space*) se temelji na korištenju Burrows-Wheeler transformacije (BWT) kao podloge za sažimanje i sufiksna polja.

2.6.1 Burrows-Wheeler transformacija

BW transformacija reverzibilno permutira string na način da se znakovi iz ponavljajućih podstringova grupiraju zajedno. Ovaj je postupak koristan zbog brzog sažimanja teksta.

Koraci su idući [1]:

1. Na kraj teksta je potrebno dodati znak \$ koji je leksikografski manji od svakog znaka iz teksta T;
2. Stvori se konceptualna matrica u kojoj je prvi redak jednak T, a svaki idući je ciklički pomak prethodnog za jedno mjesto udesno;
3. U dobivenoj matrici sortirati retke u leksikografskom poretku. Na taj način smo dobili matricu M_T ;
4. Iz posljednjeg stupca M_T se dobiva transformirani tekst L.

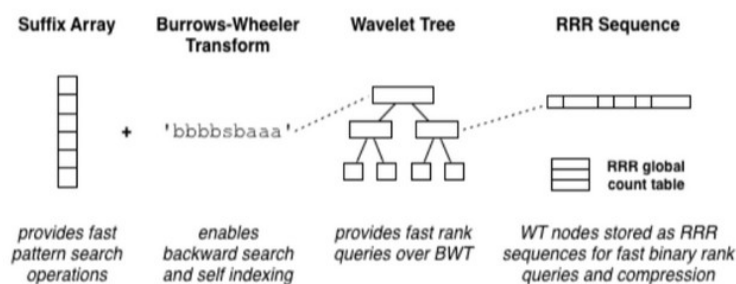
Postupak je povezan sa sufiksnim poljem: $BWT[i] = S[SA[i]-1]$, $BWT[1] = \$$, gdje je S originalni string, sufiksno polje SA i BWT je Burrows Wheeler transformacija.

Zaključno, i-ti simbol BWT je simbol neposredno prije i-tog sufiksa (Sl 3.1.1).

BWT	SA
i	12 \$
p	11 i\$
s	8 ippi\$
s	5 issippi\$
m	2 ississippi\$
\$	1 mississippi\$
p	10 pi\$
i	9 ppi\$
s	7 sippi\$
s	4 sissippi\$
i	6 ssippi\$
i	3 ssissippi\$

Slika 3.1.1 BWT i SA

Ferragina i Manzini su predložili da se kombinira BWT i sufiksno polje čime se kreira FM indeks koji omogućava pretraživanje unazad. Prikaz cjelokupnog postupka je na idućoj slici (Sl 3.1.2).



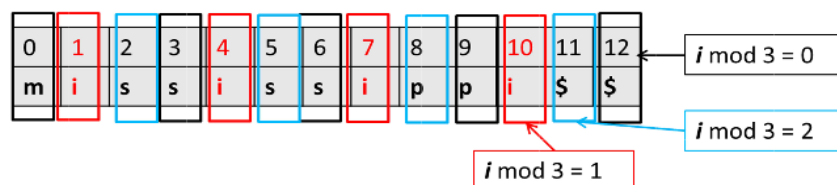
Slika 3.1.2 SA+BWT+Wavelet Tree+RRR

3. Primjer izvođenja algoritma

Kako bismo još bolje objasnili algoritam, prikazat ćemo korak po korak algoritma na jednostavnom primjeru. Uzet ćemo tekst 'mississippi'. Slijede opisi kroz sve faze algoritma [2].

3.1 Kärkkäinen-Sandersov algoritam

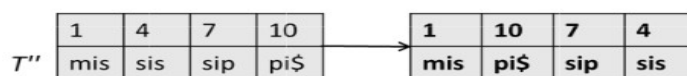
Kako smo rekli, uzimamo za niz $S = \text{mississippi}\$$. Dodali smo dva znaka $\$$ i time je duljina niza višekratnik broja 3. Sada je potrebno iz S stvoriti T' na način da se spajaju podnizovi $S[i, i+2]$, gdje je $i \bmod 3 \neq 0$. T' je $\text{ississippi}\$ \text{ssissippi}$, što se lako može uočiti iz slike niže (Sl 3.1.1).



Slika 3.1.1 Stvaranje T' iz početnog niza S

S' se tvori iz T' tako da se svaki podniz $T' [3i-2, 3i]$ zamijeni jedinstvenim leksikografskim imenom $S'[i]$ uz uvjet: $S'[i] \leq S'[j]$, ako i samo ako $T'[i, i+2] \leq T'[j, j+2]$.

Korištenjem sortiranih sufiksa niza S' se određuje redoslijed sufiksa S'' . Potrebno je prvo iz S oblikovati T'' koji čine trojke $S[i, i+2]$, $i \bmod 3 = 0$. T'' ispada $\text{mississippi}\$$ (Sl 3.1.2). Provođi se sortiranje trojki iz T'' .



Slika 3.1.2 Dobivanje T''



Slika 3.1.3 Spajanje S' i S'' i određivanje sufiksnog polja

Sufiksno polje se određuje prema sortiranim sufiksima u T' (tj. S') i T'' (tj. S''). Sufiksno polje je prikazano na gornjoj slici (Sl 3.1.3).

3.2 Konstrukcija BWT iz dobivenog sufiksnog polja

Konstruirat ćemo niz BWT za niz $S = \text{mississippi}\$$ korištenjem $SA=[12,11,8,5,2,1,10,9,7,4,6,3]$.

i	$SA[i]$	Abecedno poredani sufiksi od S	$B[i]$
0	12	\$	i
1	11	i\$	p
2	8	ippi\$	s
3	5	issippi\$	s
4	2	ississippi\$	m

5	1	mississippi\$	\$
6	10	pi\$	p
7	9	ppi\$	i
8	7	sippi\$	s
9	4	ssissippi\$	s
10	6	ssippi\$	i
11	3	ssissippi\$	i

$B[i]$ smo dobili prema pravilu da je $B[i] = \$$ kada je $SA[i] = 1$, a inače je $B[i] = S[SA[i] - 1]$ [3].

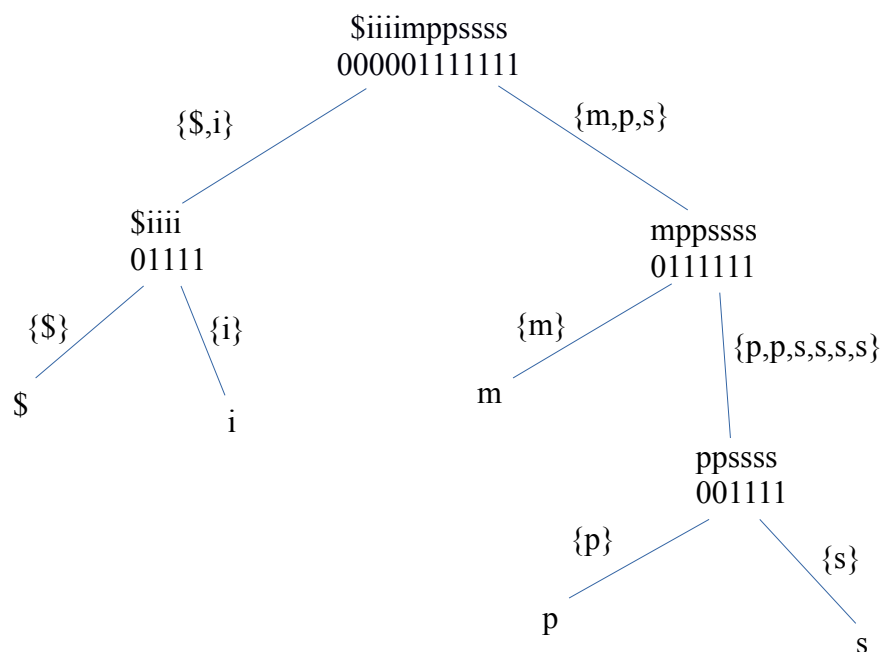
Zadnji stupac, odnosno $S_z = ipssm\$pissii$, jeste BWT. U idućim koracima osim zadnjeg stupca trebat će nam i prvi, $S_p = \$iiiiimppssss$.

3.3 Stablo valića

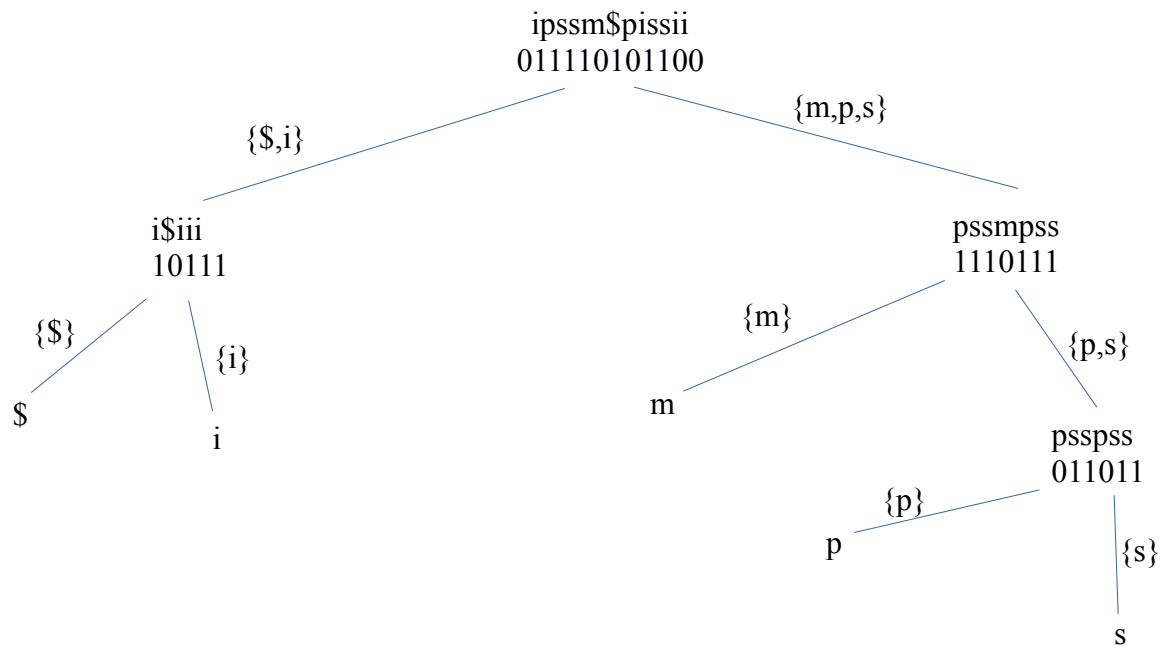
Prvi i zadnji stupac dobivene u prethodnom koraku treba spremiti kao stablo valića (Wavelet Tree).

Krenimo sa prvim stupcem $S_p = \$iiiiimppssss$.

Abecedno poredano, niz se sastoji od slova \$imps. Podijelimo ih u dvije grupe, grupi [\$,i] dajemo vrijednost 0, a grupi [m,p,s] vrijednost 1. Dodijelimo te vrijednosti svim znakovima u nizu, podijelimo one sa vrijednošću 0 u jedan niz, one sa vrijednošću 1 u drugi. Nastavljamo na taj način dok ne dođemo do toga da se svaki niz sastoji od jednog znaka – tada je kreiranje stabla valića završeno.



Isto ćemo napraviti i za zadnji stupac $S_z = \text{ipssm\$pissii}$.



3.4 PrefixSum-tablica

U ovoj tablici se za svaki znak u nizu S čuva broj leksikografski manjih znakova.

U nizu $S = \text{mississippi\$}$ imamo znakove $\$, i, m, p, s$.

U idućoj tablici je za svaki od tih znakova prikazan broj znakova koji su leksikografski prije tog znaka.

Znak i	Broj znakova koji su leksikografski prije znaka i
\$	0
i	1
m	5
p	6
s	8

4. Testiranje

Za testiranje smo koristili genome bakterija, a podaci se nalaze u datotekama FASTA formata. Rezultati su prikazani u nastavku, a mogu se naći i na github repozitoriju u folderu 'test'.

Fusobacterium_mortiferum_atcc_9817.GCA_000158195.2.24.dna.toplevel

supercont2.4 (Characters : 15949)

Memory: 203902B

Index creation time: 0.038608s

Search time: Count Find Pattern

7.2e-05 0.00044 A

0.006902 0.006902 TGT

0.028897 0.033939 TAAAAAGCATATTATCC

0.114293 0.112843

CATCTTCTCTTTTCATAGTATTACTATATATCTCTTTTCTTGATACATAACACAAGCAC

1.03712 1.03556

CCCAAAATTCATCATTAAATTCGTTTTTACTATTTAGTAATGTCCCATCCATATCAGTAATAATTAGCTTTATCA
TTTCTGCCCTCTCAATTTTTTAATATTATAGCATAAAAAAATTATTTTTTTAAAAGAGGATTATATTCTTCAT
AAAAATATATGAGAATATTAAAGTAATTCCTTCAGTAACAAACATGGTAATCCAAATTCCAGTAGTACCTAA
AAATATAGGAAGTATAAGAATGAATAAGACTAAAAGTAAAATTCCTCTTAGCATAGTAATAAGTAGAGAT
TTTAACCATATCTACAGCTGTAAAATATCCAGCACTAAAAATATTAAATCCACATATTAAGTAAGCAAAGCTA
TAAGTTGTAAGAGCTTTTTTAGTGAGATTAATATCTTCAACATCATTTAAAAATATAGATACAATGTTTGCAG
AAAAAATATTGATTAAAGTGTAAGAATATTCCTAATACTCCAAGGATAAAGAACTATTTTAATTATTTTC
AAAGATTTTTTATGCTCTTTAGCTCCAAGATTAAAACTTAGTATAGGTTGTACCCCTTGATTAAAGCCTATC
ATAGTCATAGTGATAAAAGAGGAAATATATCCAATTATTCCAAA

supercont2.5 (Characters : 250110)

Memory: 1353474B

Index creation time: 0.54234s

Search time(s): Count Find Pattern

3.7e-05 0.005462 A

0.072328 0.073519 TGT

0.432364 0.430373 GGTGTTTTTTCAAAAAAATAAA

13.5519 13.545

TTTCTATCTCAGCATTAACAACCTTTATAGAGGGAAGTAAAATATCTACAAAAGTATCTTTCTTTTTTAATGG
GGATAGTTTGCTTAAATCAATATTAATTTTAGTAAAAACATATAGTTCATCTGAATCTTTCCATTTTTTATATC
TTCTAAACTATTTACCTCTATCTTTTTATACTCAACTTTTGGAATAGAAGATTCTGCATAACTTATAATTTAG
AGTAAAAATAAGAATACAGATAAAAATATTTTTTAATTTTCATAATAAACTCTCCTTAAAAAATAGTCGGTT
AATAGTATATCATTTTATCATAGTAAGTAAAGAATTTTTTATTCATAGAATTTGAAGTAATCGTAATTAATTTT
TATTTCTATAAATAAAGTCCTATAATTAATGTTATATATAAAATATTTTAAGGTGTTTTTTCAAAAAAATAAATC
AAGAATAACTTTTATTTAAAGAATAAAGAGTAGCTAAAGCTTATAAGATATTAACCAAAATCTCTCT
AAGTTATAAAGGAAGTGTTTTTTATATTTATTTTCGCCAAGCT

supercont2.1 (Characters : 196658)

Memory: 1093118B

Index creation time: 0.422395s

Search time(s): Count Find Pattern

7.2e-05 0.004931 A

0.062626 0.085377 TGT

0.486314 0.484155 GTAAATCCATTTTCAGTAGGAGCA

13.2372 13.2193

TAGTATATGTTATAAAGTATGCTAATACTGTAAAAAAGATAAAGAAAAATCAATAGTGGCAGGTATTGAAG
TAAATACAAGAGTAAATGAAGAGGGACATGAATTACCTGAACCTACTAAAAAAGATTTTGGTCGATTATG
GTATTAGTTGGAATAATTATATTACTAATATTAGGTTATATGCCTTGGTATGTAATAGAATTAGATGGTGGAAG
AACATTTCAAGATGTTATAAATGCTCCTATTAATTCATAGCTTCAATACCTATATTAGGAAGTTTAAAGTGGCT
TTAAACATTTCACTCCTTTTGGTGATTGGTATTTTGGAGAATTTGCCTTTGTTTTCTTATTAGGGTCAATATTA
ATTGGATTAATAAATAGAATGCCAGAGGAGCAGTTTGTAAAAGAATTTGCAAATGGTGCTAAGGATTTATTA
GGAGTAATACTAGTACTTTCTGTTGCTAATGGTATATCTGTACTTATGGGAAGTAAAAGTGAAGGAATGTCT
GTAACATTTGTTTATTGGATTCAAACAGCTCTTCAAGGAATACCATCTTGGGCTTTTCAATAGCAACAGTG
GTTGCTTATGTAGGAATAGGATTTTCTTACAATCTAC

Escherichia_coli_07798.GCA_000303655.1.24.dna.toplevel.fa

E07798.contig.252 (Characters : 202)

Memory: 126491B

Index creation time: 0.001829s

Search time: Count Find Pattern

7.2e-05 6.2e-05 A

0.000169 0.000219 TGT

0.001104 0.001187 CGAAGCCGCGACGACACATAATG

0.006351 0.009462

TCGCTGAGATATTTACAGGGAGGCTTTGTAGTCTTCCAGCAACAGTTCAGCAAAGTCCATTTTGAAGCCGCG
ACGACACATAATGCGCATCACCACGATGTCGGTGGCTTCACCGCCGAGAGTGAAGGCCGGAACCTGCCAG
CCGCGCAGACGCAGACGTTTCAGAGAGGTCGTACAGGGTGTATCC

E07798.contig.111 (Characters : 446)

Memory: 127153B

Index creation time: 0.001435s

Search time: Count Find Pattern

7.4e-05 3.2e-05 A

0.000154 0.000336 TGT

0.002365 0.002424 CATCGCCACACCATCAGGAACGGG

0.015606 0.01488

GATTGTTGGTCAGTGCGCCTGACAGGCGGAACCTTACAGGTCAGTCTGCCGCCCTGTGTCAGCAGGGTAAT
TGCGTCATCCACCGGATGCGTCAGGGACCAGGATTTATTGCTCTGCTTGGCGATCTTAAATACACCACCG
ACAACCTGAATTCCGCCATCCTTAATGCTCCAGCCCTGCGCAGC

E07798.contig.94 (Characters : 1994)

Memory: 134657B

Index creation time: 0.006803s

Search time: Count Find Pattern

7.3e-05 8.7e-05 A

0.001278 0.001267 TGT

0.00444 0.004428 GTTATGACAGAACAAACAAAATTGACCTTTACTG

0.00107 0.001953

GTCAGAAAAAGCAGGACATTCTCGAAGCGATTTCGTGACCACCAGGTGGTGATCGTCGCCGGGGAAACGG
GTTCCGGTAAAACGACTCAGCTACCGAAAATCTGTATGGAGCTGGGGCGCGGGATTAAAGGGCTGATCGG
CCATACCCAGCCGCGTCGGCTGGCGGCGCGAACGGTGGCGAACCGTATTGCGGAAGAGCTAAAAACGGA
GCCGGGCGGTTGCATCGGTTACAAAGTGCCTTTCAGCGATCACGTCAGTGATAACACGATGGTCAAGCTGA
TGACCGACGGTATCTGCTGGCAGAGA

Abiotrophia_defectiva_atcc_49176.GCA_000160075.2.24.dna.toplevel.fa

A_defectiva-1.0.1_Cont0.6 (Characters : 329)

Memory: 127399B

Index creation time: 0.001081s

Search time: Count Find Pattern

6.9e-05 5.9e-05 A

0.000265 0.000203 TGT

0.000645 0.001465 GGATGGTAAATCTCAG

0.007814 0.010943

CAATTAATCGCTCAGCAGAAGCGACATTAGGATAGACGCCAATGACCTTATCCCGCCGTCTGATTTACGAT
TCAAACGTTTCGACAGAATTGGTAGTCCGTAAATTGGGATGGTAAATCTCAGGCTCCATCATAAACTGAACG
GCATCTTCAAAGCCCTCATCTAAAACCTTTAAGGCCT

A_defectiva-1.0.1_Cont0.5 (Characters : 2049)

Memory: 134957B

Index creation time: 0.008828s

Search time: Count Find Pattern

7e-05 9.1e-05 A

0.000548 0.001342 TGT

0.010878 0.010915 TAATCGCTCAGCAGAAGCGACATTA

0.087846 0.084705

CAAGAAGGCTTGTTTCGTTGCGCTGATAGCGTTCAAAGAGCTCGGTTTTGAACTCGCCAGAGCGCGTGCGT
GGGACTTGTAATTTAAGGGTGCCGATGGATAGCGTCCATTGGCGTTCATAGTAGCCGTTGCGATAATCTTGG
CGCTGGTCGGAGCGTTTCATGGGGTGCGGCTTGGATGAACTGGTGGCGTTCTTCTTCCATGTAGGCATTAAA
GACGGTAATGGCCAGCGTCTTCATGAGCGTGTTTCATGTCACTGTTTAAACTGCCTCTGTTAATTCTTCAAG
TTTTAGGTAAATATTAAGGTAAGTCATGGTAATCTCCTCTTTCAATGTTTTCGGACAACACTACATTGTACCAGA
GATT

A_defectiva-1.0.1_Cont0.7 (Characters : 34347)

Memory: 292709B

Index creation time: 0.073271s

Search time: Count Find Pattern

7.1e-05 0.000613 A

0.012522 0.0012826 TGT

0.064146 0.075257 TTTCCGGAGTGCAAGCCCCG

0.687067 0.694946

TTTCCGACAACACTACATTGTACCAGAGATTACCTGGCTTTTTCTTTTACACAATTATAGGGACTTAATCCCGA
AAGCCCCGAAAAGCCGTGAGCCAGTGGTCCACTCTTTTCGAGTGGACCACTAGCTCCTGGGAAAAGTGCG
GAAACAGGGTCAAAGCCCCGAAAAGTCGTGAGCCAGTGGACCACTCCGTTCGAGT

Plasmodium_falciparum.ASM276v1.24.dna.chromosome.13.fa

13 (Characters : 2837504)

Memory: 14310094B

Index creation time: 6.61836s

Search time: Count Find Pattern

7.3e-05 0.035017 A

0.736482 0.731513 TGT

7.71857 7.72394 GAAGATAAGAGTAAGTAAGTAAGAGTAAGTAAGT

Najmanji broj znakova za koji se provelo testiranje je 202 i u tom slučaju je memorija 126491B, a vrijeme kreiranja indeksa 0.001829s.

Najveći broj znakova za koji smo proveli testiranje je 2837504 i memorija je tada 14310094B, dok je vrijeme stvaranja indeksa 6.61836s.

5. Zaključak

Istraživanje u području genoma je prouzročilo stvaranje velikog broja podataka kojeg je trebalo nekako pohraniti te iz toga izvući samo potrebne dijelove. Osmišljen je vrlo djelotvoran postupak koji omogućava upravo pretraživanje informacija od interesa među velikim skupom podataka, odnosno FM-indeks.

Cilj ovoga projekta je bio implementirati FM-indeks te zadovoljiti zahtjeve za efikasno i brzo pretraživanje, kao i korištenje što manje memorije. FM-indeks sadrži sufiksno polje koje kreiramo koristeći Kärkkäinen-Sandersov algoritam koji se pokazao kao dobar odabir budući da prilično brzo obavi posao. FM-indeks još sadrži i LF-tablicu koja se sastoji od Prefix-Sum tablice i stabla valića.

Provedeno je nekoliko testiranja na temelju kojih smo zaključili da smo zadovoljili zahtjeve za brzinom i memorijskom učinkovitošću, a također smo provjerili i ispravnost programa.

6. Literatura

- [1] *FM-Indexes and Backwards Search*: <http://alexbowe.com/fm-index>. Kolovoz 2011.
- [2] ŠIKIĆ, M. DOMAZET-LOŠO, M. *Bioinformatika*:
http://www.fer.unizg.hr/download/repository/bioinformatika_skripta_v1.2.pdf. Zagreb, prosinac 2013.
- [3] DOMAZET-LOŠO, M. Samostojni indeksi.
http://www.fer.unizg.hr/download/repository/Bioinformatika_-_7._predavanje_%28Samostojni_indeksi%29.pdf. Zagreb, prosinac 2014.