Introduction
- Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

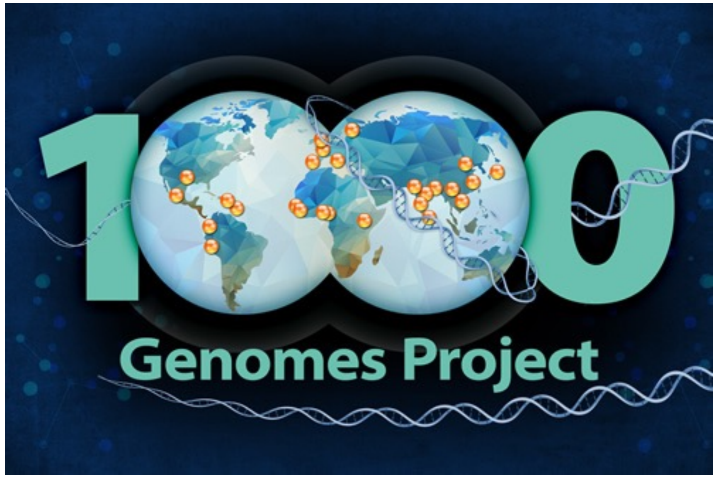
Recursive Prefix-Free Parsing for Building Big BWTs

Marco Oliva[1], Travis Gagie[2] and Christina Boucher[1]

[1] Department of Computer and Information Science, University of Florida

[2] Faculty of Computer Science, Dalhousie University

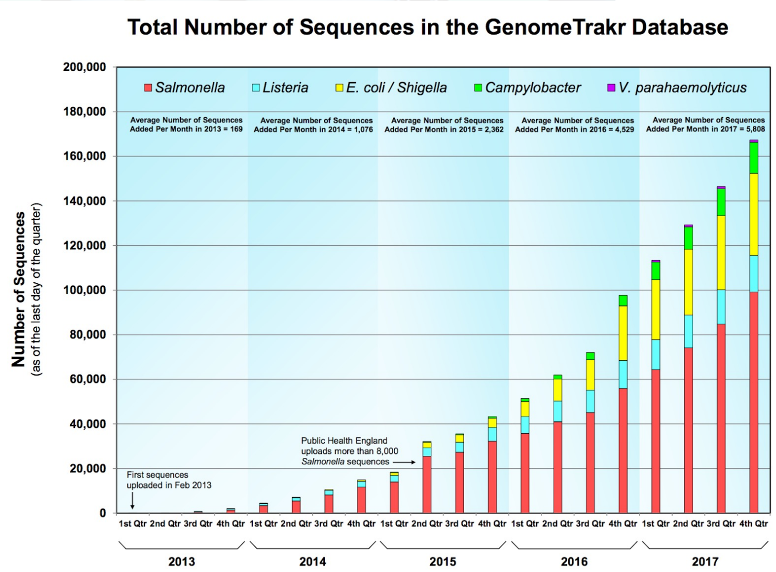
Motivation



2008 - 2015
More than 3000 individual sequenced

Motivation

2008- 2015
More than 3000 individual sequenced



2012- present
More than 580,000 isolates sequenced

Introduction
- Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Motivation

1000 Genomes Project
Total Number of Sequences in the GenomeTrakr Database
Salmonella
Listeria
E. coli / Shigella
Campylobacter
V. parahaemolyticus
Average Number of Sequences Added Per Month in 2013 = 169
Average Number of Sequences Added Per Month in 2014 = 1,076
Average Number of Sequences Added Per Month in 2015 = 2,362
Average Number of Sequences Added Per Month in 2016 = 4,529
Average Number of Sequences Added Per Month in 2017 = 5,808
200,000
180,000
160,000
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
Number of Sequences
(as of the last day of the quarter)
First sequences uploaded in Feb 2
1st Qtr 2nd Qtr 3rd
2013
The 100,000 Genomes Project

2008- 2015
More than 3000 individual sequenced

2012- present
More than 580,000 isolates sequenced

2012-2018
Sequencing rare disease

Introduction

- Motivation

Background

- PFP
- From PFP to BWT

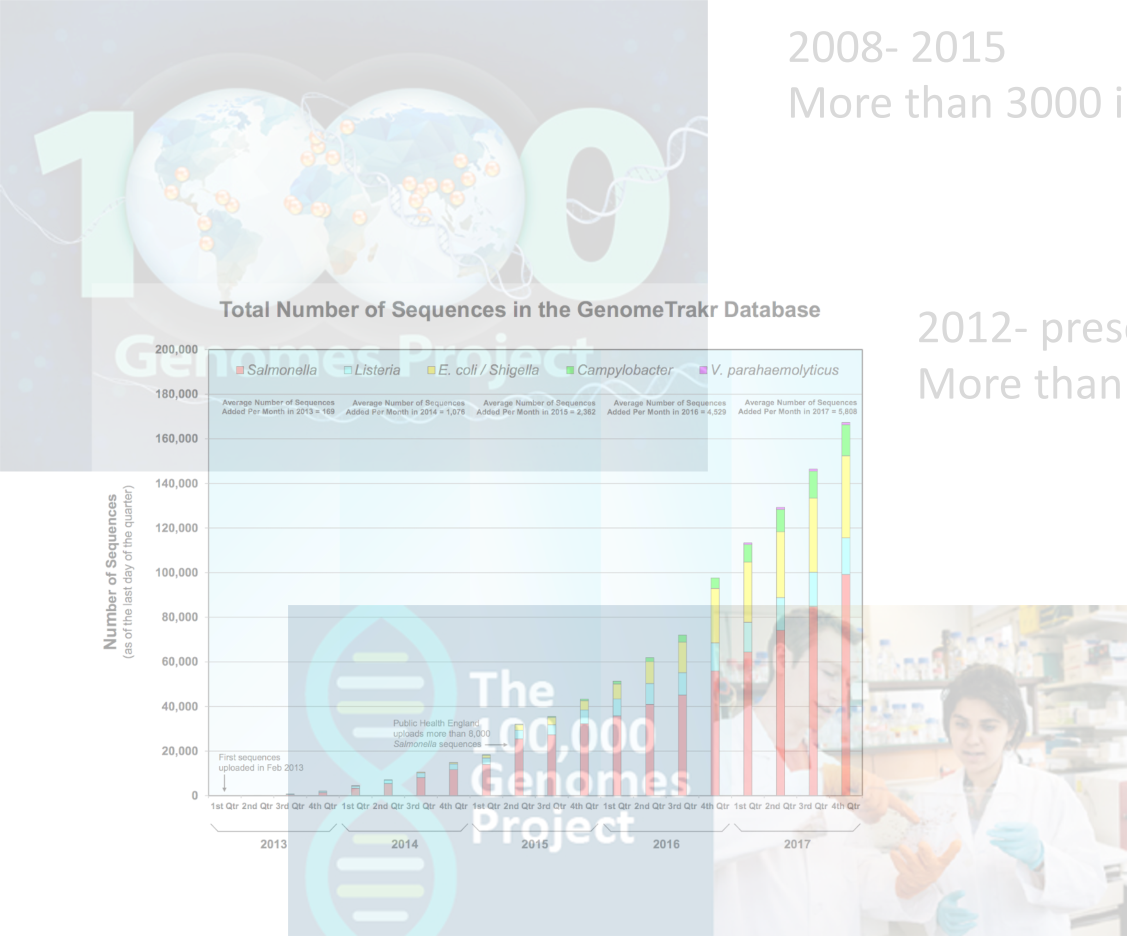
Recursive PFP

- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments

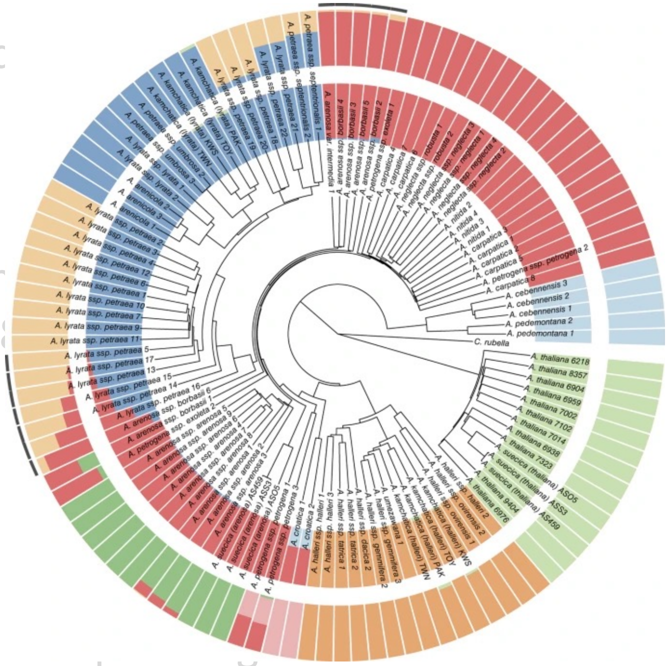
- Chr. 19

Motivation



2008- 2015
More than 3000 inc

2012- preser
More than 5



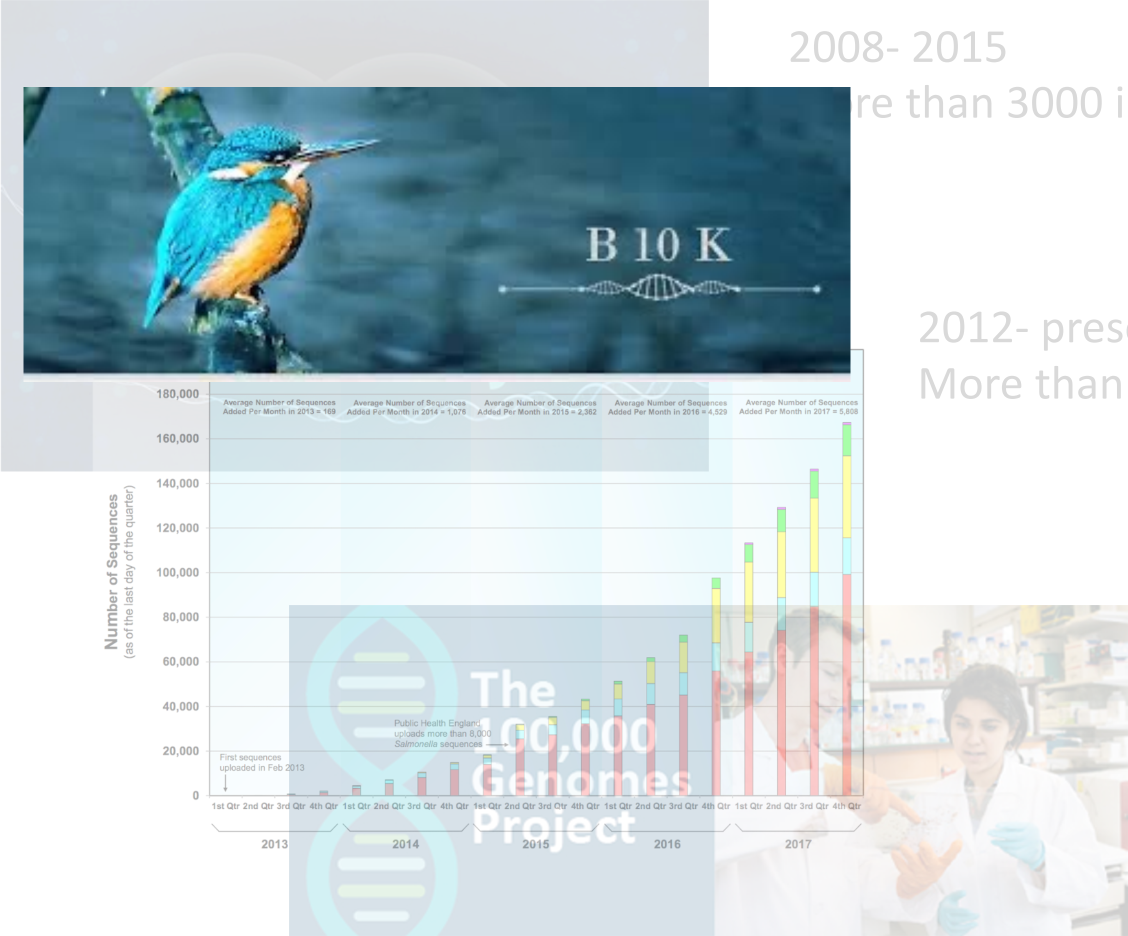
Introduction
- Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Motivation

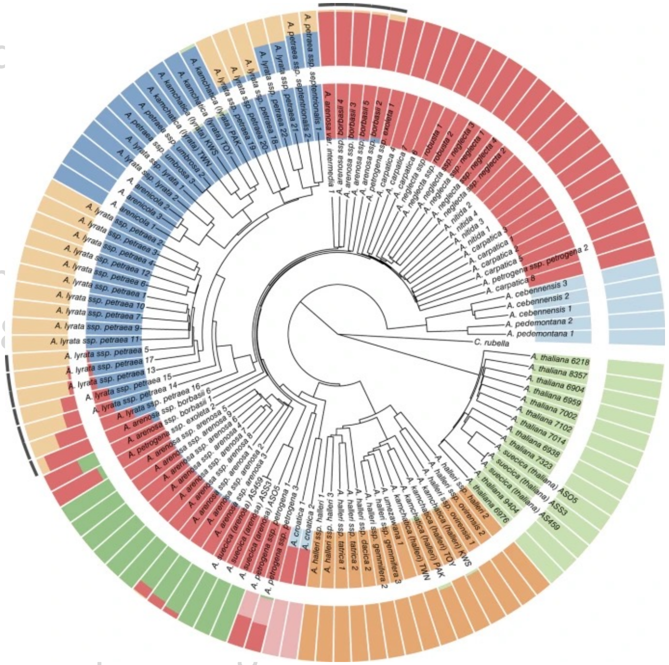


2008- 2015

re than 3000 inc

2012- preser

More than 5



Introduction
- Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

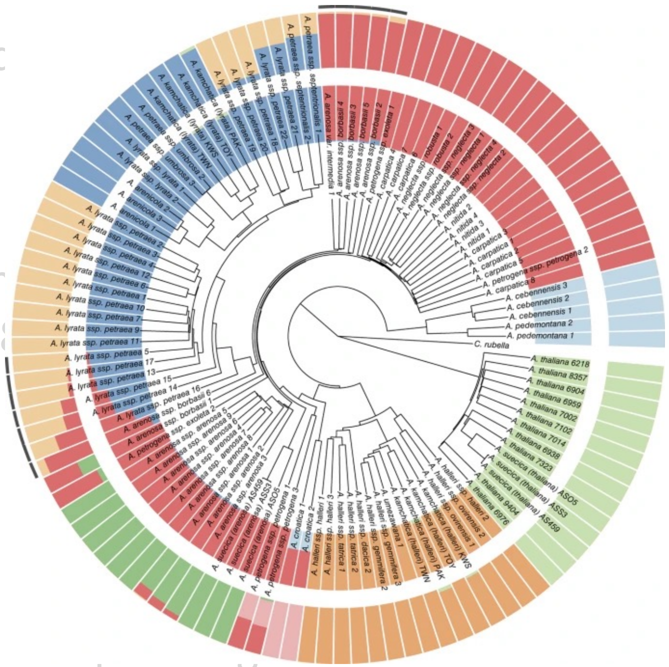
Experiments
- Chr. 19

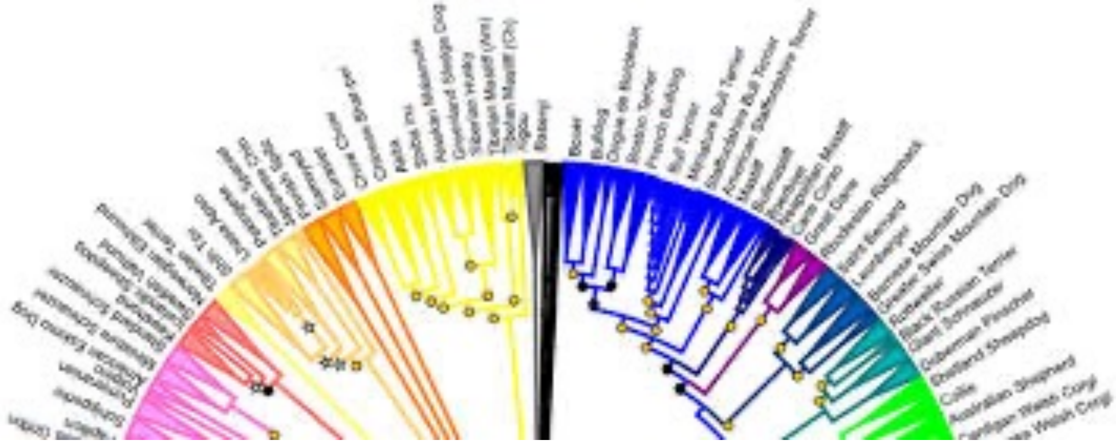
Motivation

B 10 K
180,000
160,000
140,000
120,000
100,000
80,000
60,000
Number of Sequences
(as of the last day of the quarter)
Average Number of Sequences Added Per Month in 2013 = 169
Average Number of Sequences Added Per Month in 2014 = 1,076
Average Number of Sequences Added Per Month in 2015 = 2,362
Average Number of Sequences Added Per Month in 2016 = 4,529
Average Number of Sequences Added Per Month in 2017 = 5,608
The

2008- 2015
re than 3000 inc

2012- preser
More than 5





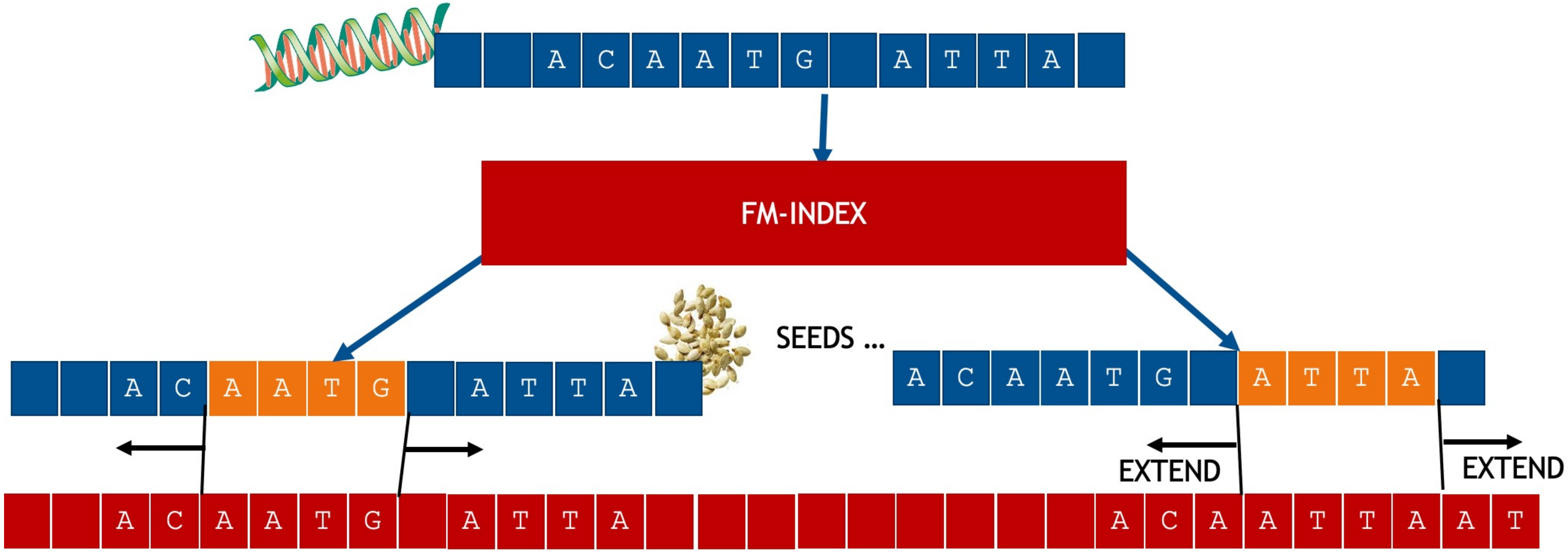
Introduction
- Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

FM-index, BWT



Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

```
S:      AAGGTNTATACCTTCCCAGGTAACAAACCAACCAA$
        AAGGTNTATACCTTCCCAGGTAAAAAACCAACCAA$
        AAGGTNTATACCTTCCCAGGTAATAAACCAACCAA$
```

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

```
S:   ##AAGGTNTATACCTTCCCAGGTAACAAACCAACCAA$
       AAGGTNTATACCTTCCCAGGTAAAAAACCAACCAA$
       AAGGTNTATACCTTCCCAGGTAATAAACCAACCAA$##
```

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

```
S:   ##AAGGTNTATACCTTCCCAGGTAACAAACCAACCAA$
       AAGGTNTATACCTTCCCAGGTAAAAAACCAACCAA$
       AAGGTNTATACCTTCCCAGGTAATAAACCAACCAA$##


W:     2
TS:    {NT, AG, A$, ##}
```

UF Herbert Wertheim College of Engineering
Department of Computer & Information Science & Engineering
UNIVERSITY *of* FLORIDA

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

Prefix Free Parsing

```
S:    ##AAGGTNTATACCTTCCCAGGTAACAAACCAACCAA$
        AAGGTNTATACCTTCCCAGGTAAAAAACCAACCAA$
        AAGGTNTATACCTTCCCAGGTAATAAACCAACCAA$##


W:      2
TS:     {NT, AG, A$, ##}
```

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**

A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**

A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2

TS: {NT, AG, A$, ##}

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

D: {##AAG}
P: [h(##AAG)]

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

D: {##AAG, AGGTNT}
P: [h(##AAG), h(AGGTNT)]

Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing

S: **##**A**AG**GT**NT**ATACCTTCCC**AG**GTAACAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAAAAAACCAACCA**A$**
A**AG**GT**NT**ATACCTTCCC**AG**GTAATAAACCAACCA**A$##**

W: 2
TS: {NT, AG, A$, ##}

D: {##AAG, A$##, A$AAG, AGGTAAAAAACCAACCAA$,
AGGTAACAAACCAACCAA$, AGGTAATAAACCAACCAA$,
AGGTNT, NTATACCTTCCCAG}
P: [1, 7, 8, 5, 3, 7, 8, 4, 3, 7, 8, 6, 2]

Introduction

✓ Motivation

Background

- PFP
- From PFP to BWT

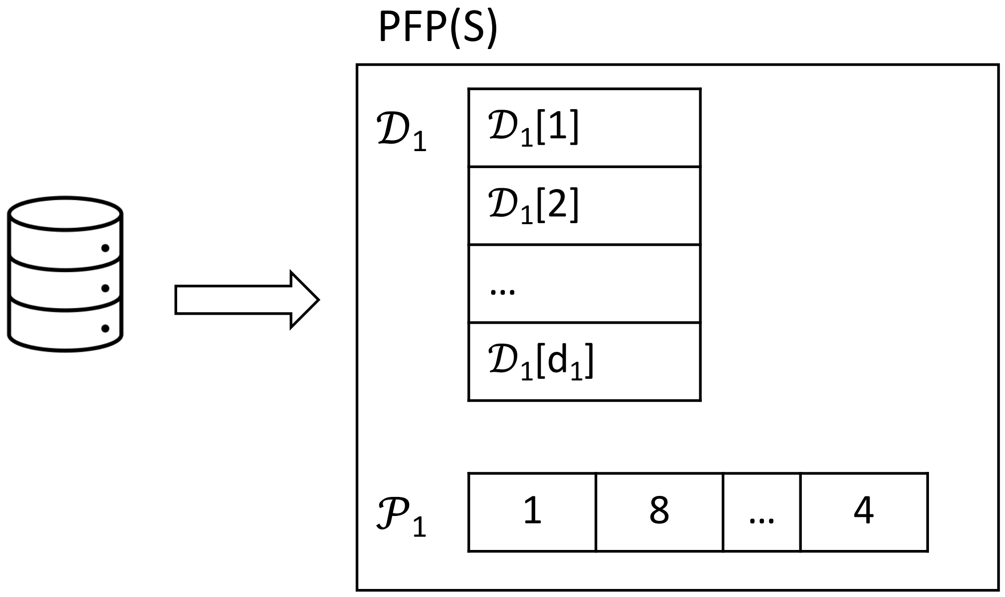
Recursive PFP

- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments

- Chr. 19

Prefix Free Parsing



PFP(S)

$\mathcal{D}_1$

$\mathcal{D}_1[1]$
$\mathcal{D}_1[2]$
…
$\mathcal{D}_1[d_1]$

$\mathcal{P}_1$

1	8	…	4

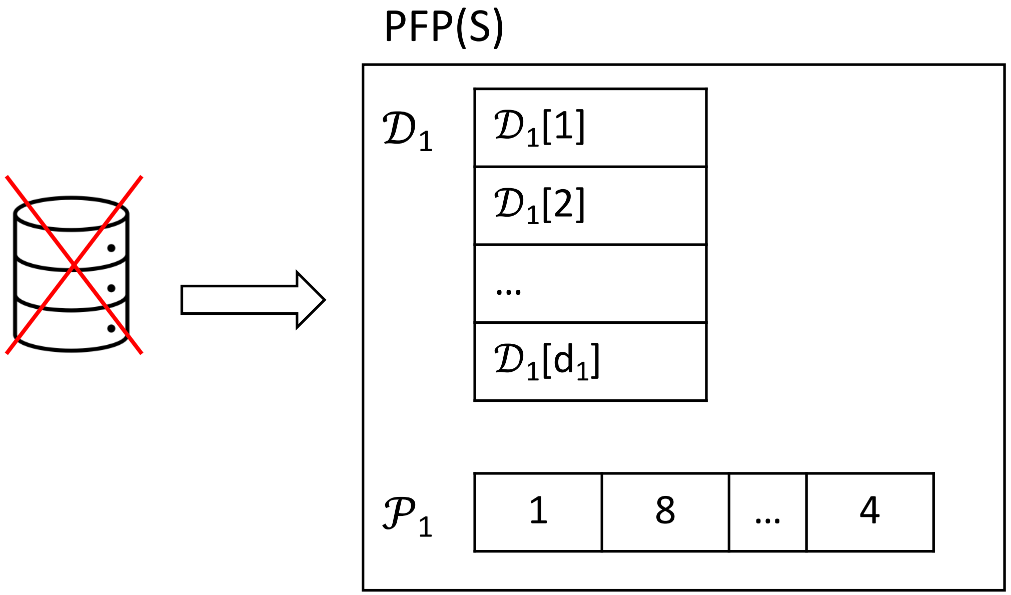
Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing



PFP(S)

$\mathcal{D}_1$

$\mathcal{D}_1[1]$

$\mathcal{D}_1[2]$

…

$\mathcal{D}_1[d_1]$

$\mathcal{P}_1$

1	8	…	4

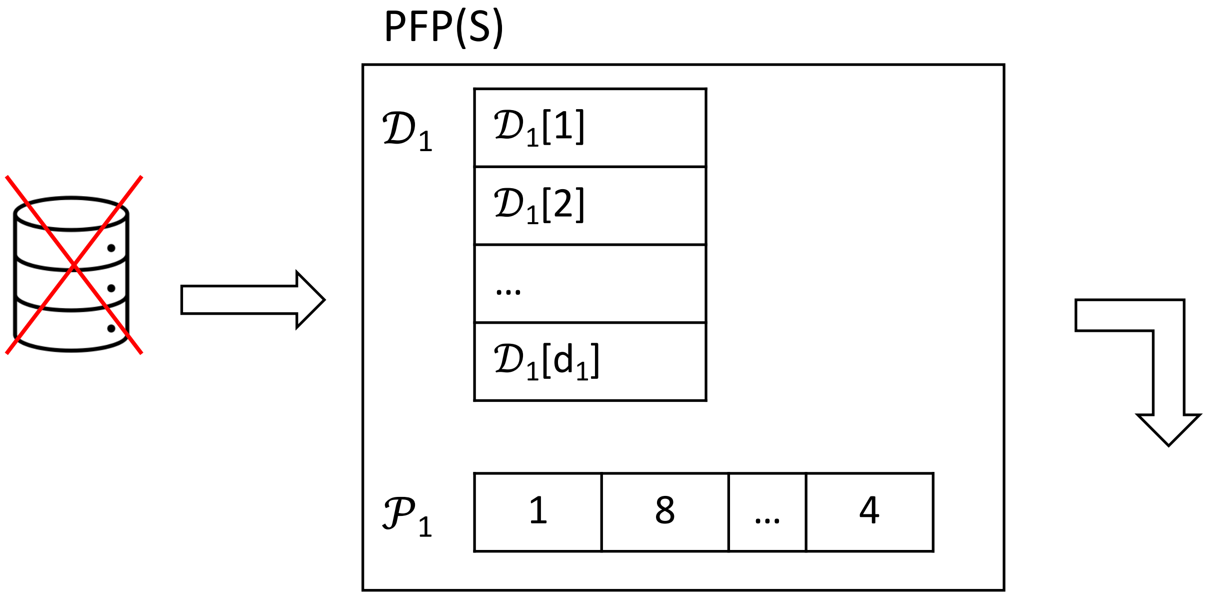
Introduction
✓ Motivation

Background
- PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Prefix Free Parsing



PFP(S)

$\mathcal{D}_1$

$\mathcal{D}_1[1]$

$\mathcal{D}_1[2]$

…

$\mathcal{D}_1[d_1]$

$\mathcal{P}_1$

1	8	…	4

```
CCCCC$CCCTTTTC$$$GGTATAATTACCC$$
$$$GGGAGCAAAGGTTTTTGGGTTTAAAAAAA
AAAGGTGTTTTTTTTTAAAATTTTTCCCCCCC
CTTTAAAAAAACCCAAAACAAGGGGGTGGGGG
CCCCCGGGGGGGGCCCCCCCAAGCCAAAAAC
GAAAAAACCCCCGTCCTTTTCACCCCACGGGT
GGTGTGTTTTTTTTGGGGGTGGGGCCCGCGGG
```

From Prefix Free Parsing to the BWT

Lemma

No phrase suffix of length greater than w is a proper prefix of any phrase suffix.

Proof sketch

Every phrase starts with a trigger strings and ends with a trigger string and contains no other trigger string. If a phrase suffix α with $|\alpha| > w$ were a proper prefix of a phrase suffix β then β would contain a trigger string $|\alpha| - w$ characters from its start and $|\beta| - |\alpha|$ characters from its end.

Herbert Wertheim
College of Engineering
Department of Computer & Information Science & Engineering
UNIVERSITY *of* FLORIDA

Introduction
- ✓ Motivation

Background
- ✓ PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

From Prefix Free Parsing to the BWT

Corollary

If two characters $S[i]$ and $S[j]$ are followed by different phrase suffixes α and β with $|\alpha|, |\beta| \geq w$, then $S[i]$ precedes $S[j]$ in BWT if and only if $\alpha \prec \beta$.

Proof sketch

Suppose $|\alpha| \leq |\beta|$. Since $\alpha \neq \beta[0..|\alpha| - 1]$, either $\alpha \prec \beta$ or $\beta \prec \alpha$. In the first case $S[i]$ precedes $S[j]$ in BWT and in the second case $S[j]$ precedes $S[i]$.

From Prefix Free Parsing to the BWT

Lemma

If two characters $S[i]$ and $S[j]$ are followed by the same phrase suffix α with |α| ≥ w, then $S[i]$ precedes $S[j]$ in BWT if and only if $S[i + |\alpha| - w + 1] \prec S[j + |\alpha| - w + 1]$.

Corollary

If two characters $S[i]$ and $S[j]$ in phrases $P[i']$ and $P[j']$ are followed by the same phrase suffix α with $|\alpha| \geq w$, then $S[i]$ precedes $S[j]$ in $\mathrm{BWT}(S)$ if and only if $P[i']$ precedes $P[j']$ in $\mathrm{BWT}(P)$.

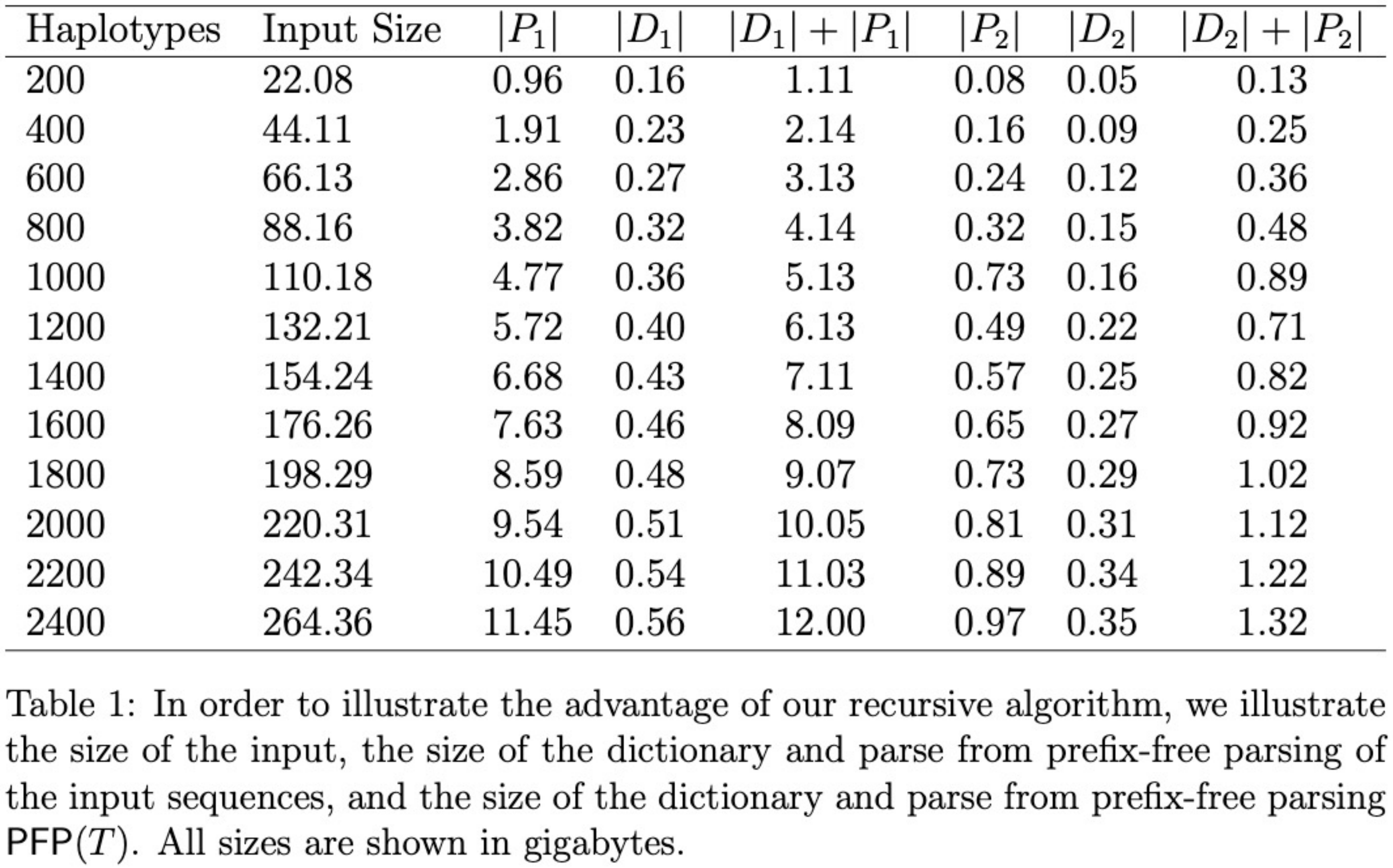
Introduction
- ✓ Motivation

Background
- ✓ PFP
- From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Scaling PFP



Haplotypes	Input Size	$\lvert P_1\rvert$	$\lvert D_1\rvert$	$\lvert D_1\rvert + \lvert P_1\rvert$	$\lvert P_2\rvert$	$\lvert D_2\rvert$	$\lvert D_2\rvert + \lvert P_2\rvert$
200	22.08	0.96	0.16	1.11	0.08	0.05	0.13
400	44.11	1.91	0.23	2.14	0.16	0.09	0.25
600	66.13	2.86	0.27	3.13	0.24	0.12	0.36
800	88.16	3.82	0.32	4.14	0.32	0.15	0.48
1000	110.18	4.77	0.36	5.13	0.73	0.16	0.89
1200	132.21	5.72	0.40	6.13	0.49	0.22	0.71
1400	154.24	6.68	0.43	7.11	0.57	0.25	0.82
1600	176.26	7.63	0.46	8.09	0.65	0.27	0.92
1800	198.29	8.59	0.48	9.07	0.73	0.29	1.02
2000	220.31	9.54	0.51	10.05	0.81	0.31	1.12
2200	242.34	10.49	0.54	11.03	0.89	0.34	1.22
2400	264.36	11.45	0.56	12.00	0.97	0.35	1.32

Table 1: In order to illustrate the advantage of our recursive algorithm, we illustrate the size of the input, the size of the dictionary and parse from prefix-free parsing of the input sequences, and the size of the dictionary and parse from prefix-free parsing PFP(T). All sizes are shown in gigabytes.

Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Recursive PFP

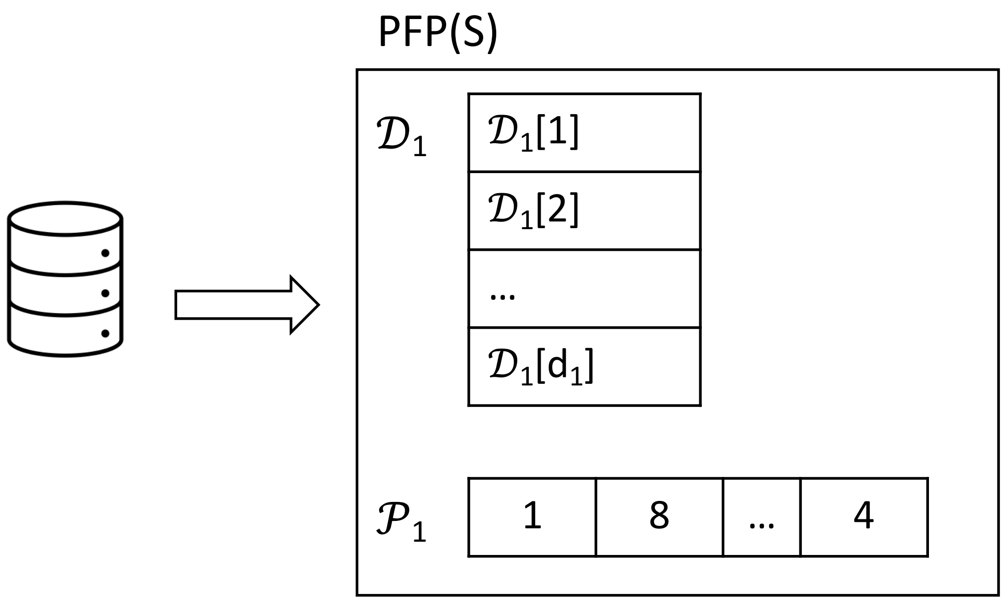
Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Algorithm Overview



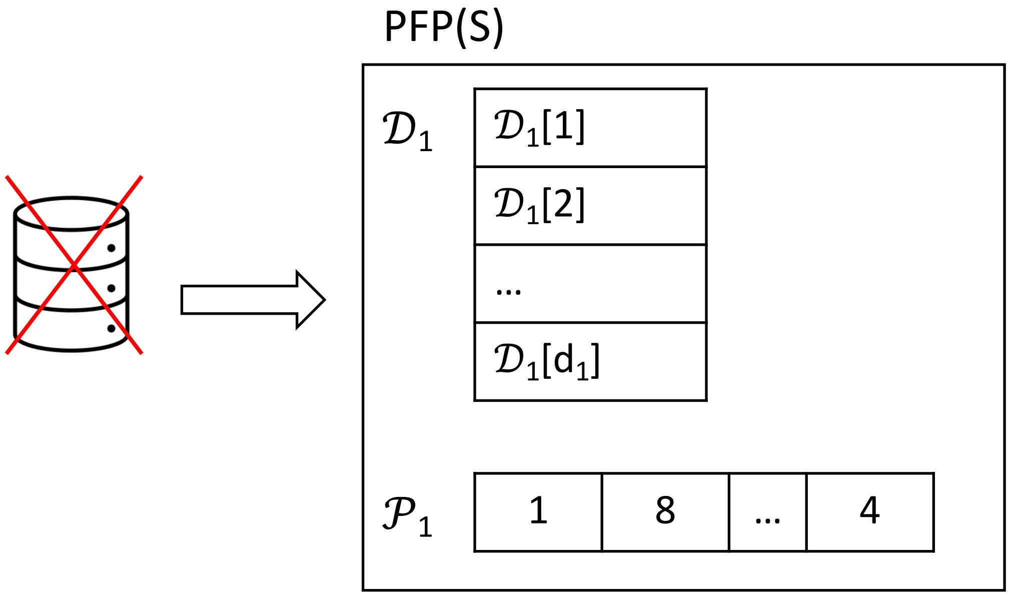
Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Algorithm Overview



PFP(S)

$\mathcal{D}_1$

$\mathcal{D}_1[1]$

$\mathcal{D}_1[2]$

…

$\mathcal{D}_1[d_1]$

$\mathcal{P}_1$

1	8	…	4

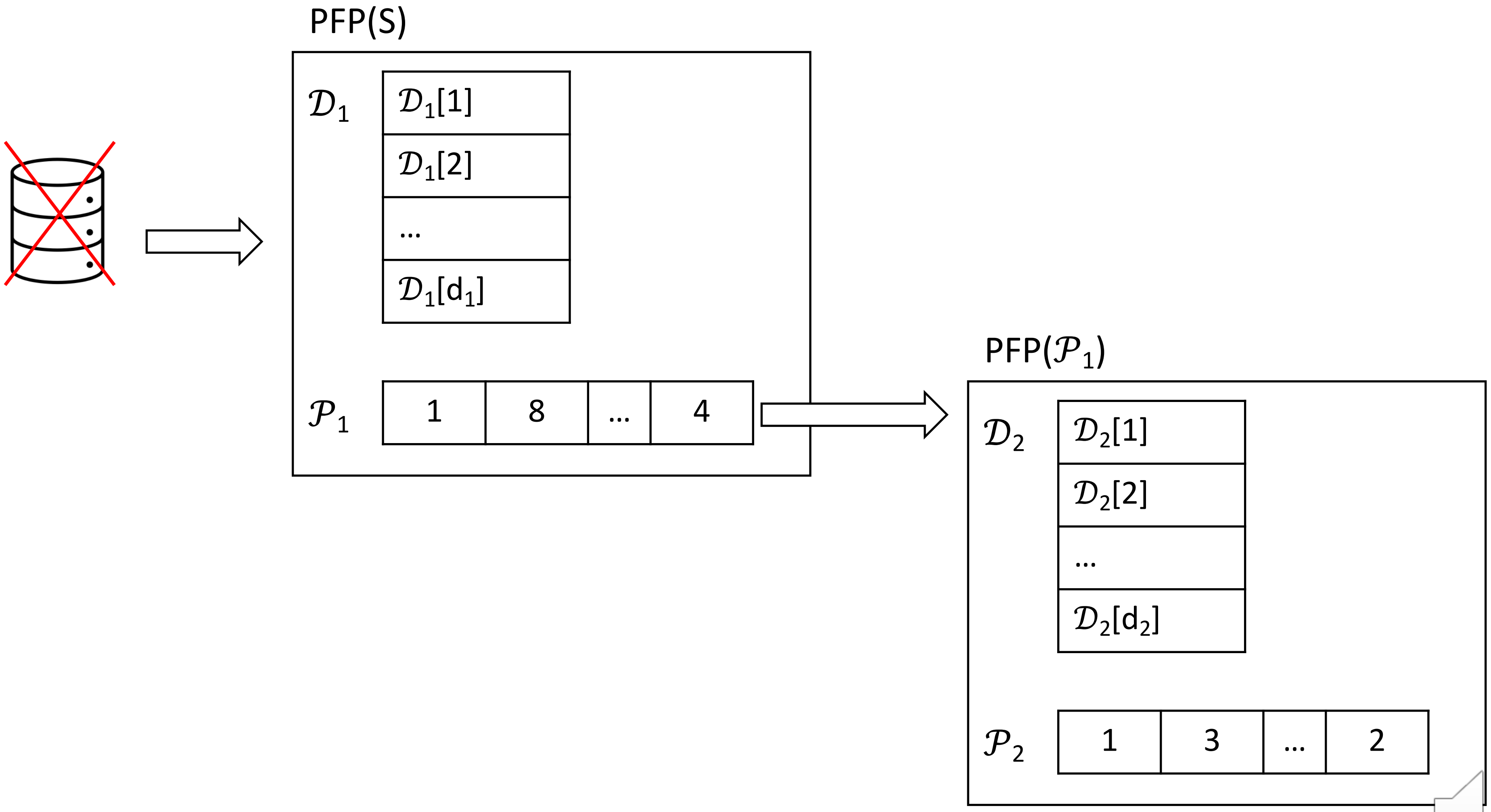
Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Algorithm Overview



PFP(S)

$\mathcal{D}_1$: $\mathcal{D}_1[1]$, $\mathcal{D}_1[2]$, …, $\mathcal{D}_1[d_1]$

$\mathcal{P}_1$: 1, 8, …, 4

PFP($\mathcal{P}_1$)

$\mathcal{D}_2$: $\mathcal{D}_2[1]$, $\mathcal{D}_2[2]$, …, $\mathcal{D}_2[d_2]$

$\mathcal{P}_2$: 1, 3, …, 2

UF Herbert Wertheim College of Engineering
Department of Computer & Information Science & Engineering
UNIVERSITY *of* FLORIDA

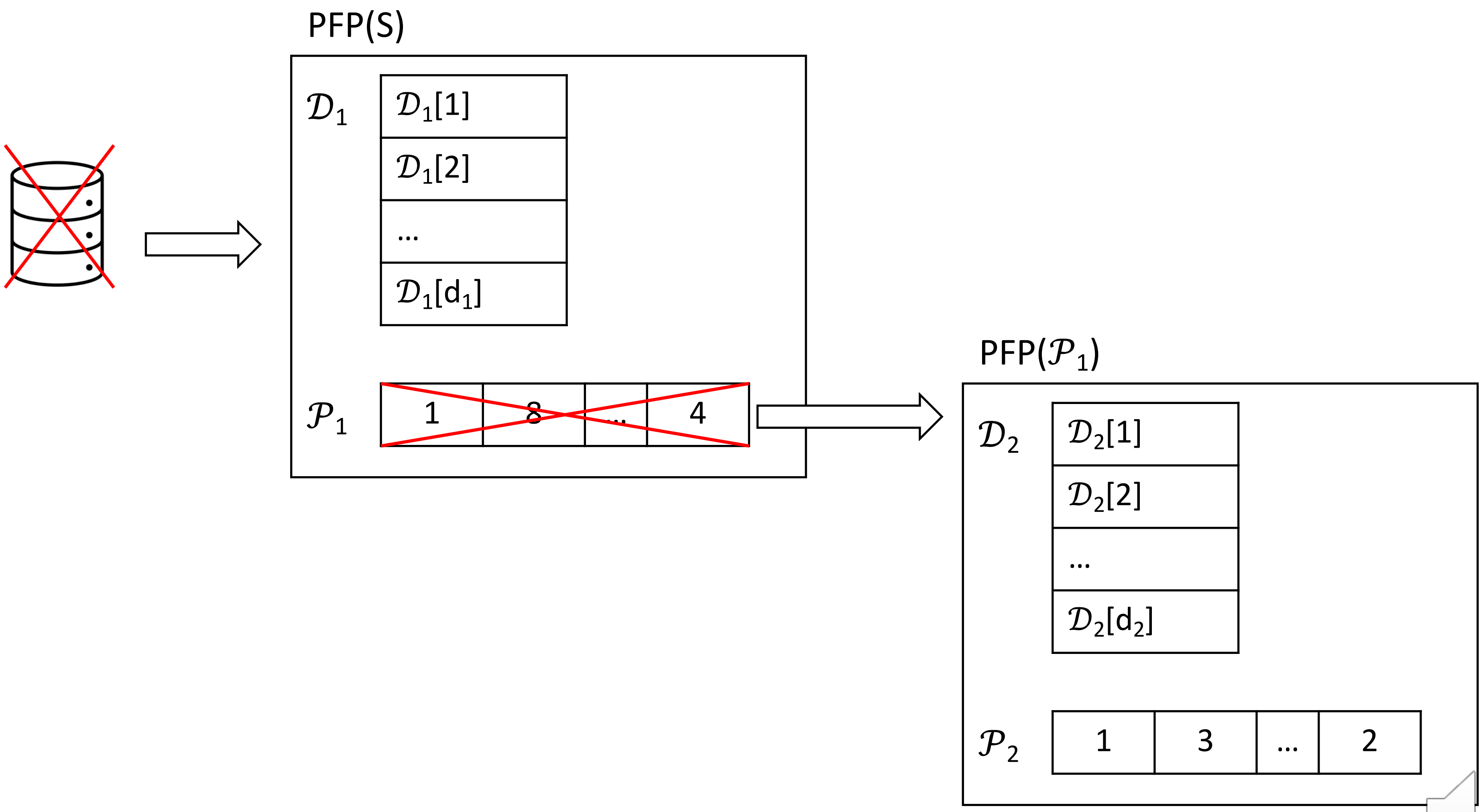
Introduction
✓ Motivation

Background
✓ PFP
✓ From PFP to BWT

Recursive PFP
- Motivation
- Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Algorithm Overview



PFP(S)

$\mathcal{D}_1$

$\mathcal{D}_1[1]$

$\mathcal{D}_1[2]$

…

$\mathcal{D}_1[d_1]$

$\mathcal{P}_1$

1	8	…	4

PFP($\mathcal{P}_1$)

$\mathcal{D}_2$

$\mathcal{D}_2[1]$

$\mathcal{D}_2[2]$

…

$\mathcal{D}_2[d_2]$

$\mathcal{P}_2$

1	3	…	2

Extending Previous Results

Lemma

If two characters $T[i]$ and $T[j]$ in phrases $P_1[i']$ and $P_1[j']$ are followed by the same phrase suffix $\alpha \in S_1$, then $T[i]$ precedes $T[j]$ in the BWT of T if one of the following two conditions is true: (a) $P_1[i']$ and $P_1[j']$ precede two different phrase suffixes $\alpha', \beta' \in S_2$ with $\alpha' \prec \beta'$; or (b) the phrase $P_2[k']$ containing $P_1[i']$ precedes the phrase $P_2[l']$ containing $P_1[j']$ in the BWT of P_2.

Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- ✓ Motivation
- ✓ Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Data Structures

Definition

We define a table T_1 containing $O(|S_1|)$ rows and $O(1)$ columns, such that for each, $\alpha \in S_1$ we store in T_1 its range in $\mathrm{BWT}(T)$ along with the co-lexicographic sub-range of the elements of D_1 which store the occurrence of α.

		co-lex sub-ranges				
phrase suffix	BWT range	$	A	C	G	T
$$	0-0			0-0		
$AAC	1-3			3-3		
$AC	4-8	1-1		2-2		
AAC	9-16	3-3		4-4		5-7
AATAGT	17-18				21-21	
AC	19-43	1-2	3-7	8-9	10-11	12-13
AGGT	44-45				25-25	

…

Introduction
✓ Motivation

Background
✓ PFP
✓ From PFP to BWT

Recursive PFP
✓ Motivation
✓ Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Data Structures

Definition

We define a table T_2 containing $O(|S_2|)$ rows and $O(1)$ columns, such that for each $\alpha' \in S_2$, we store in T_2 the co-lexicographic range of the phrases of D_2 that contain α' along with the meta-characters that precede α' in P_1.

phrase suffix	co-lex range	preceding meta-character
\$ \$	0--0	\$
1 7 17	4--4	\$
…		
5 20 15 18 3 10	2--2	19
5 20 16 23 \$ \$ \$	0--0	12
6 12 5 20 16 23 \$ \$ \$	0--0	4
7 17	4--8	1,14,4,4
8 4 6 12 5 20 16 23 \$ \$ \$	0--0	14
…		

Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

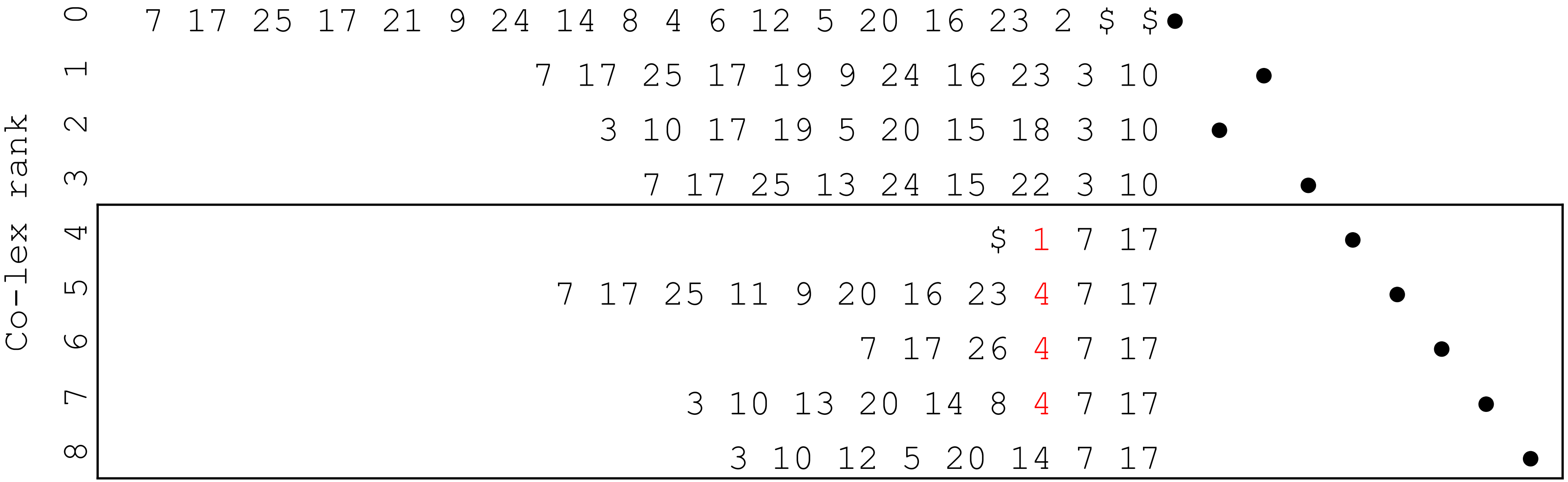
Recursive PFP
- ✓ Motivation
- ✓ Recursive Step
- Data Structures
- Algorithm

Experiments
- Chr. 19

Data Structures

Definition

We define the grid G_2 containing $O(|P_2|)$ rows and $O(|D_2|)$ columns, such that for each element l of D_2, G_2 stores the positions in the BWT of P_2 where l appears.



Co-lex rank

```
0  7 17 25 17 21 9 24 14 8 4 6 12 5 20 16 23 2 $ $●
1                    7 17 25 17 19 9 24 16 23 3 10    ●
2                       3 10 17 19 5 20 15 18 3 10  ●
3                         7 17 25 13 24 15 22 3 10      ●
4                                         $ 1 7 17        ●
5                    7 17 25 11 9 20 16 23 4 7 17          ●
6                                   7 17 26 4 7 17            ●
7                           3 10 13 20 14 8 4 7 17              ●
8                             3 10 12 5 20 14 7 17                ●
```

Introduction
- ✓ Motivation

Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- ✓ Motivation
- ✓ Recursive Step
- ✓ Data Structures
- Algorithm

Experiments
- Chr. 19

Algorithm

Given a string T , the dictionary D_1 and the parse P_1 obtained by running PFP on T, and the dictionary D_2 and the parse P_2 obtained by running PFP on P_1, we can compute the BWT of T from D_1, D_2 and P_2 using $O(|D_1| + |D_2| + |P_2|)$ workspace.

```
for each $\alpha$ in table $T_1$ :
        if $\alpha$ is preceded by only one character:
                output corresponding BWT range
        else:
                for each $\alpha'$ in $T_2$ preceded by $D_1(\alpha)$:
                        if $\alpha'$ is preceded by only one element of $D_1(\alpha)$:
                                output corresponding BWT range
                        else:
                                for each occurrence of $d \in D_1(\alpha)$ in $G_2$:
                                        output corresponding BWT character
```

Herbert Wertheim College of Engineering
Department of Computer & Information Science & Engineering
UNIVERSITY *of* FLORIDA

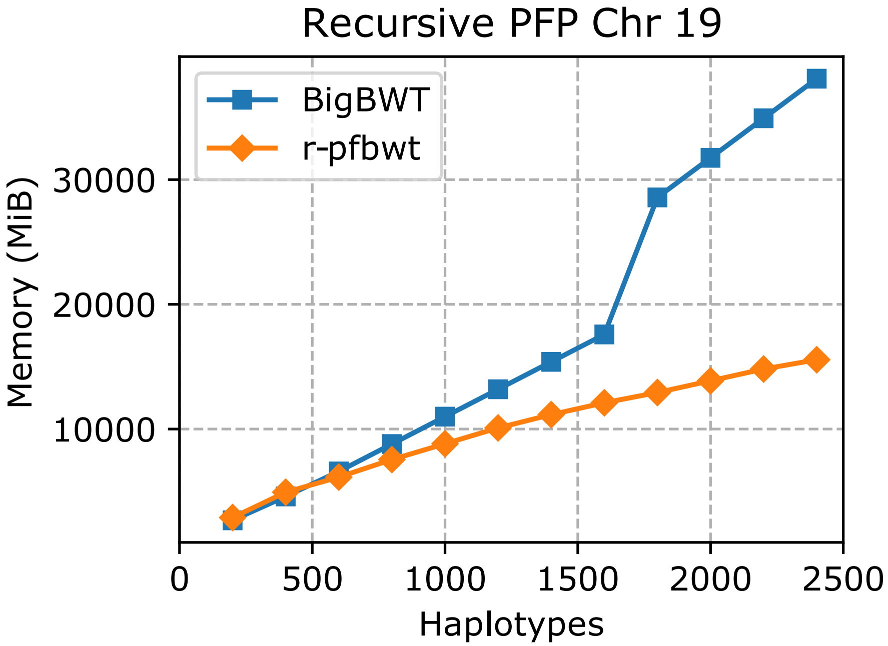
Introduction
- ✓ Motivation

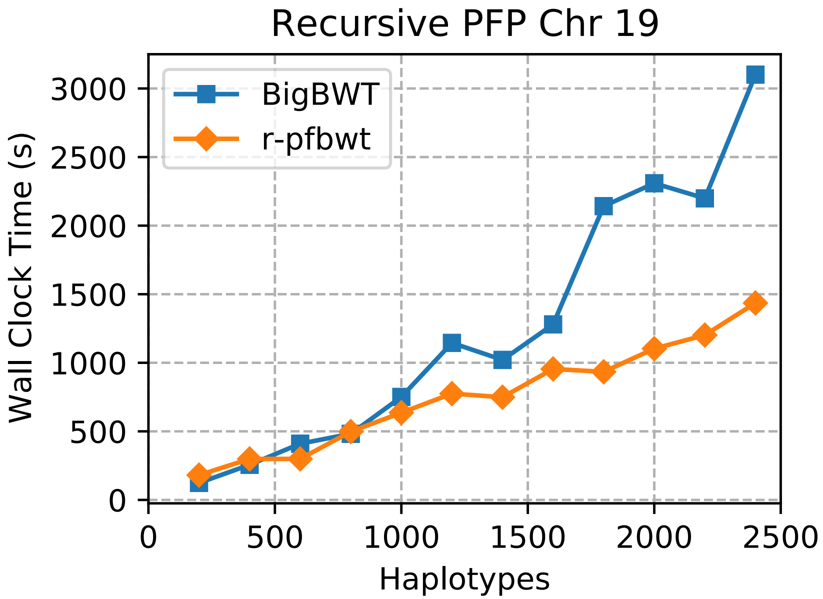
Background
- ✓ PFP
- ✓ From PFP to BWT

Recursive PFP
- ✓ Motivation
- ✓ Recursive Step
- ✓ Data Structures
- ✓ Algorithm

Experiments
- Chr. 19

Results on Chromosome 19





Introduction
✓ Motivation

Background
✓ PFP
✓ From PFP to BWT

Recursive PFP
✓ Motivation
✓ Recursive Step
✓ Data Structures
✓ Algorithm

Experiments
✓ Chr. 19

Thank you!

Source Code: https://github.com/marco-oliva/r-pfbwt

E-mail: marco.oliva@ufl.edu

Funded By

- National Science Foundation NSF SCH: INT (Grant No. 2013998)
- NSF IIBR (Grant No. 2029552)
- National Institutes of Health (NIH) NIAID (Grant No. HG011392 and R01AI141810)
- NSERC Discovery Grant RGPIN-07185-2020.

NIH National Institutes of Health

NSF

NSERC CRSNG