Stanford University

GTRAC

Fast Retrieval from Compressed Collections of Genomic Variants

Kedar Tatwawadi
Mikel Hernaez
Idoia Ochoa
Tsachy Weissman

Overview

- Introduction
- Results
- Algorithm Details
- Summary & Further Work

Introduction

GTRAC

(GENOTYPE RANDOM ACCESS COMPRESSOR)

Introduction

- The dramatic increase in the generation of sequenced data, and thus need of a compressed representation
- Sequenced data is processed and stored in the form of a variants, which constitutes the variant dataset.
- As **Variant datasets (VCF format)** are constantly queried, we need a way to efficiently access data

GTRAC aims at achieving a **compressed representation** for the variant dataset, while supporting **fast access** to parts of the dataset.

Sample VCF File

#CHROM	POS	REF	ALT	SAMP1	SAMP2	SAMP3
20	14370	G	A	1|0	1|1	0|0
20	17340	T	TA	0|1	1|0	1|1
20	112500	A	.	0|1	0|1	0|0
20	112800	A	G	0|1	0|1	0|1
20	201215	T	.	0|1	1|0	1|1
20	251513	T	G	1|1	0|0	0|0

Per-Variant Random Access

#CHROM	POS	REF	ALT	SAMP1	SAMP2	SAMP3
20	14370	G	A	1|0	1|1	0|0
20	17340	T	A	0|1	1|0	1|1
20	112500	A	.	0|1	0|1	0|0
20	112800	A	G	0|1	0|1	0|1
20	201215	T	.	0|1	1|0	1|1
20	251513	T	G	1|1	0|0	0|0

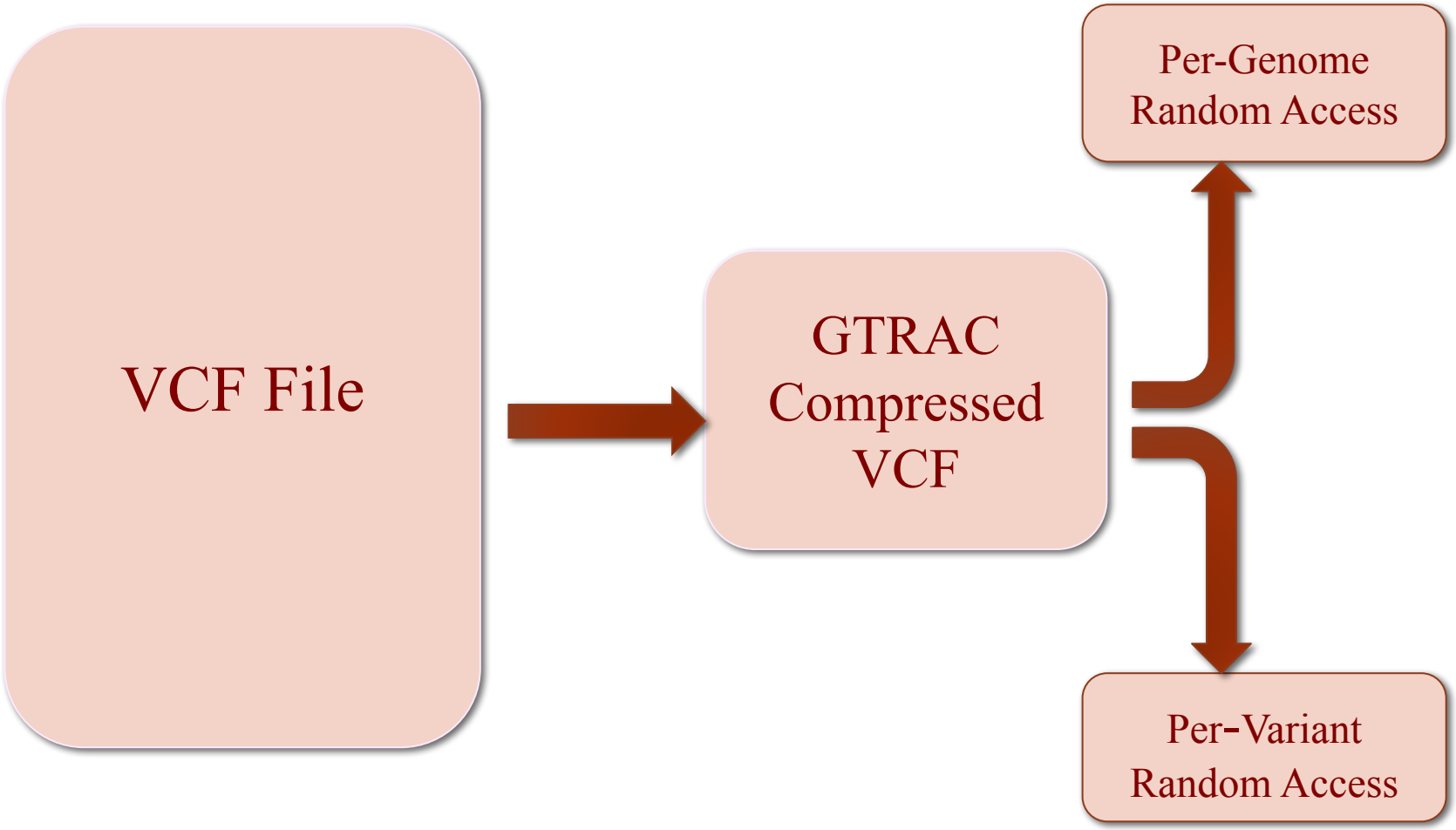
Per-Genome Random Access

#CHROM	POS	REF	ALT	SAMP1	SAMP2	SAMP3
20	14370	G	A	1|0	1|1	0|0
20	17340	T	A	0|1	1|0	1|1
20	112500	A	.	0|1	0|1	0|0
20	112800	A	G	0|1	0|1	0|1
20	201215	T	.	0|1	1|0	1|1
20	251513	T	G	1|1	0|0	0|0

Single Genotype extraction

#CHROM	POS	REF	ALT	SAMP1	SAMP2	SAMP3
20	14370	G	A	1|0	1|1	0|0
20	17340	T	A	0|1	1|0	1|1
20	112500	A	.	0|1	0|1	0|0
20	112800	A	G	0|1	0|1	0|1
20	201215	T	.	0|1	1|0	1|1
20	251513	T	G	1|1	0|0	0|0

GTRAC workflow



Results

H.SAPIENS DATASETS

H.Sapiens Results

- Experiments with the 1000 Genome Project [1] dataset containing 1092 genome samples (2184 haplotypes)

- For comparison, we also show results with:
  - **7zip** : 7zip generic compression algorithm
  - **TGC** : Thousand Genome Compressor [2]

- We compare:
  - Compression
  - Per-Variant Random Access
  - Per-Genome Random Access

[1] http://www.1000genomes.org/: *A global reference for human genetic variantion*
[2] *Genome compression: A novel approach for large collections*
2013, Deorowicz et.al

H.Sapiens Results

- Input dataset file size (VCF) : **170 GB**

	TGC	7zip	GTRAC
Compressed dataset size	**422 MB**	706 MB	1.1 GB
Per Variant decompression time	7 sec	9 sec	**17 ms**
Per Variant decompression memory	1 GB	1 GB	**90 MB**
Per Genome decompression time	7 sec	9 sec	**1 sec**
Single Variant decompression time	7 sec	9 sec	**3 ms**

H.Sapiens Results

- Input dataset file size (VCF) : **170 GB**

	TGC	7zip	GTRAC
Compressed dataset size	**422 MB**	706 MB	1.1 GB
Per Variant decompression time	7 sec	9 sec	**17 ms**
Per Variant decompression memory	1 GB	1 GB	**90 MB**
Per Genome decompression time	7 sec	9 sec	**1 sec**
Single Variant decompression time	7 sec	9 sec	**3 ms**

H.Sapiens Results

- Input dataset file size (VCF) : **170 GB**

	TGC	7zip	GTRAC
Compressed dataset size	**422 MB**	706 MB	1.1 GB
Per Variant decompression time	7 sec	9 sec	**17 ms**
Per Variant decompression memory	1 GB	1 GB	**90 MB**
Per Genome decompression time	7 sec	9 sec	**1 sec**
Single Variant decompression time	7 sec	9 sec	**3 ms**

H.Sapiens Results

- Input dataset file size (VCF) : **170 GB**

	TGC	7zip	GTRAC
Compressed dataset size	**422 MB**	706 MB	1.1 GB
Per Variant decompression time	7 sec	9 sec	**17 ms**
Per Variant decompression memory	1 GB	1 GB	**90 MB**
Per Genome decompression time	7 sec	9 sec	**1 sec**
Single Genotype extraction time	7 sec	9 sec	**3 ms**

H.Sapiens Results

- Input dataset file size (VCF) : **170 GB**

	TGC	7zip	GTRAC
Compressed dataset size	**422 MB**	706 MB	1.1 GB
Per Variant decompression time	7 sec	9 sec	**17 ms**
Per Variant decompression memory	1 GB	1 GB	**90 MB**
Per Genome decompression time	7 sec	9 sec	**1 sec**
Single Variant decompression time	7 sec	9 sec	**3 ms**

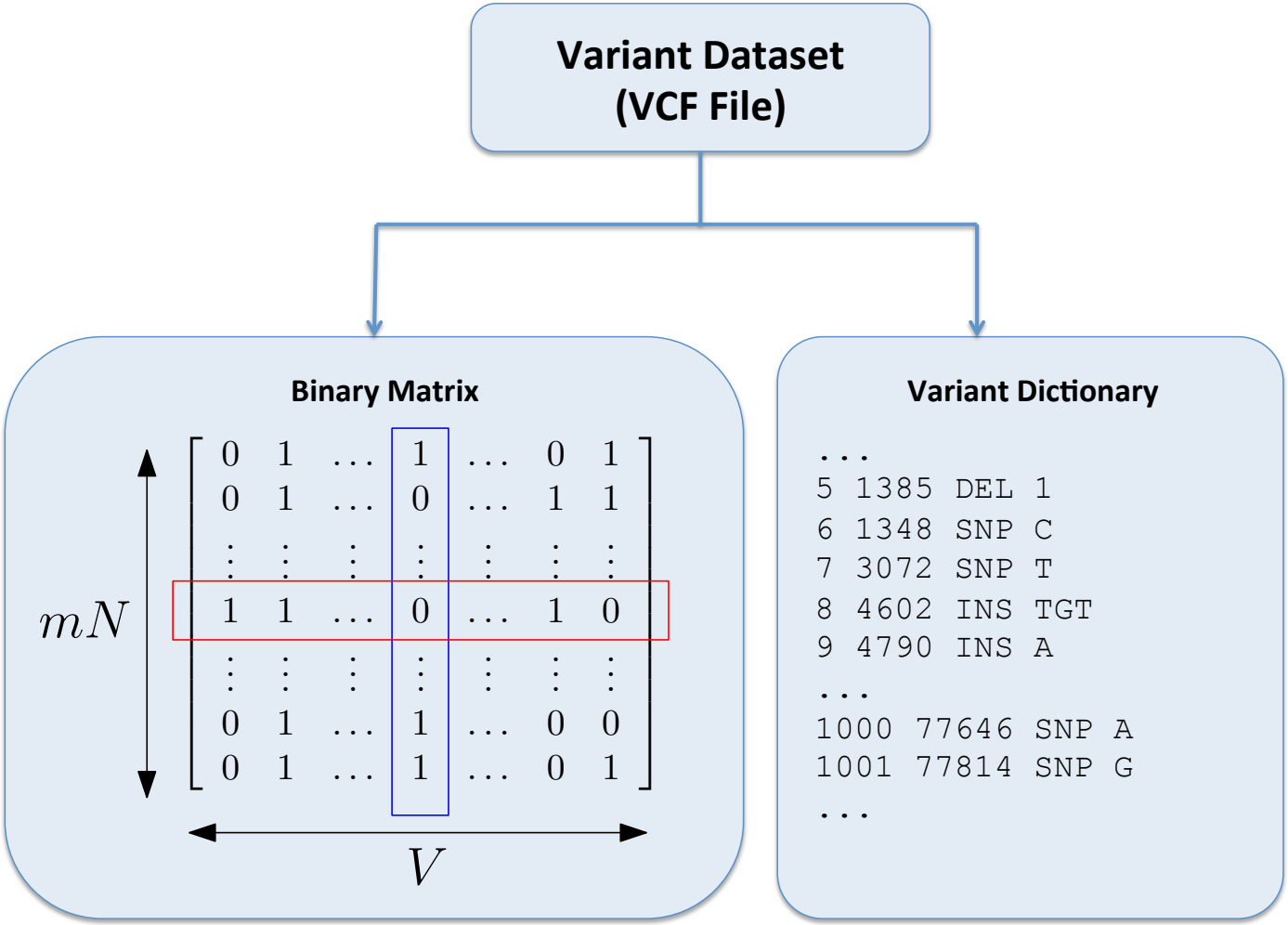
Implementation

- Paper & Implementation available at:https://github.com/kedartatwawadi/GTRAC

Algorithm Details

Compression & Random Access

Preprocessing



Parsing

- Consider the genotype binary matrix **H**, with V = 20, N = 3, and m = 1

$$\mathcal{H} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

K = 2

$$\begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\ \mathcal{H}_K = & \begin{bmatrix} 0 & 3 & 1 & 2 & 2 & 0 & 3 & 1 & 0 & 2 \\ 0 & 3 & 0 & 3 & 2 & 0 & 1 & 1 & 0 & 3 \\ 0 & 3 & 0 & 3 & 2 & 1 & 3 & 1 & 0 & 1 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \end{matrix}$$

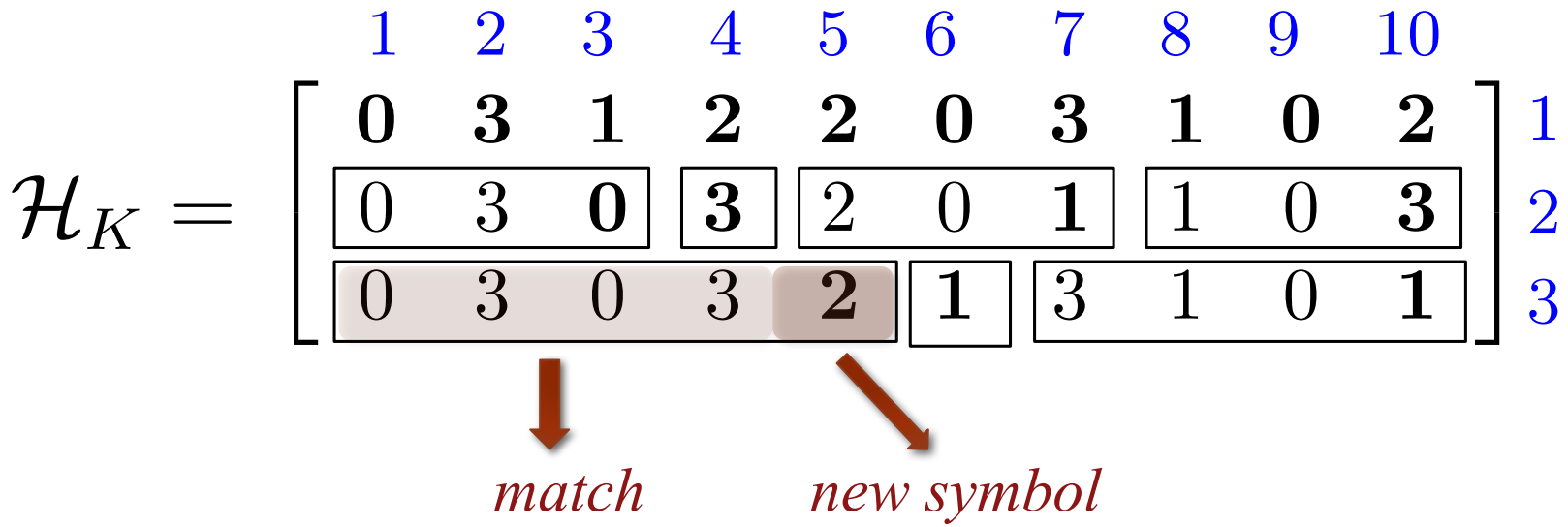
Parsing

- Every row represented in terms of substrings known as **phrases**
- Each phrase represented as **[match, new symbol]**
- Restriction on phrases:
  - Positional aligned
  - match-ending coincides with a phrase-ending

$$\mathcal{H}_K = \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccccccc} \mathbf{0} & \mathbf{3} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{0} & \mathbf{3} & \mathbf{1} & \mathbf{0} & \mathbf{2} \\ \boxed{0} & \boxed{3} & \boxed{\mathbf{0}} & \boxed{\mathbf{3}} & \boxed{2} & \boxed{0} & \boxed{\mathbf{1}} & \boxed{1} & \boxed{0} & \boxed{\mathbf{3}} \\ \boxed{0} & \boxed{3} & \boxed{0} & \boxed{3} & \boxed{\mathbf{2}} & \boxed{\mathbf{1}} & \boxed{3} & \boxed{1} & \boxed{0} & \boxed{\mathbf{1}} \end{array}\right] \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \end{array}$$

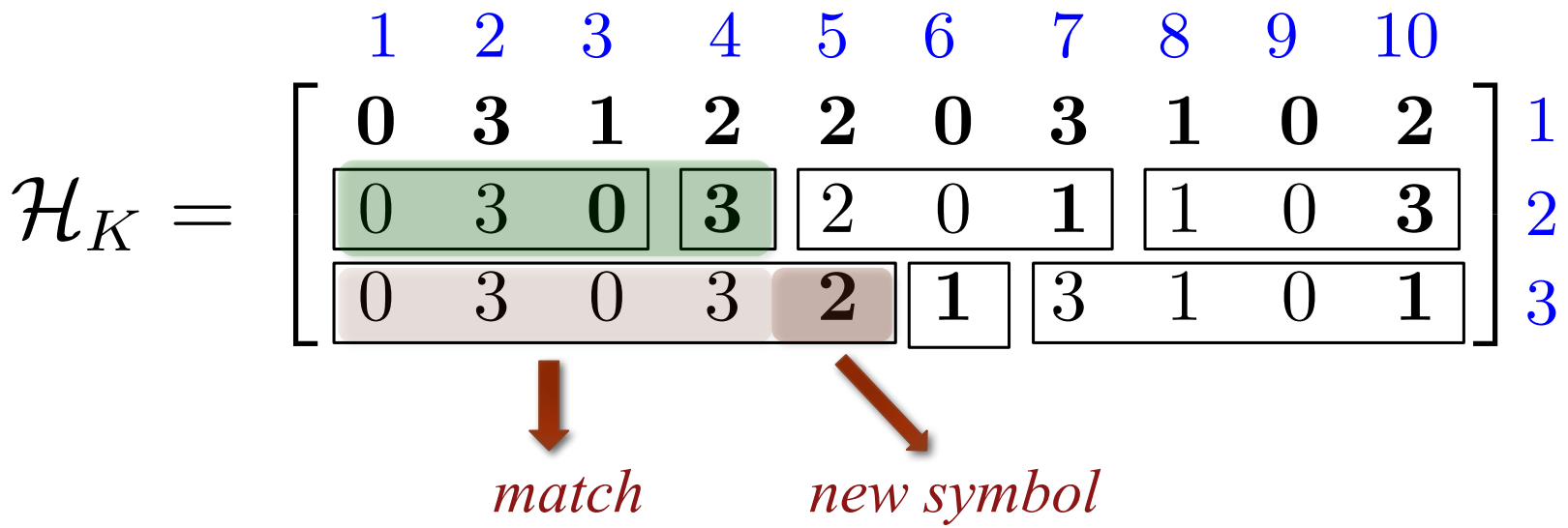
Parsing

- Every row represented in terms of substrings known as **phrases**
- Each phrase represented as **[match, new symbol]**
- Restriction on phrases:
  - Positional aligned
  - match-ending coincides with a phrase-ending in the **source** row



Parsing

- Every row represented in terms of substrings known as **phrases**
- Each phrase represented as **[match, new symbol]**
- Restriction on phrases:
  - Positional aligned
  - match-ending coincides with a phrase-ending in the **source** row



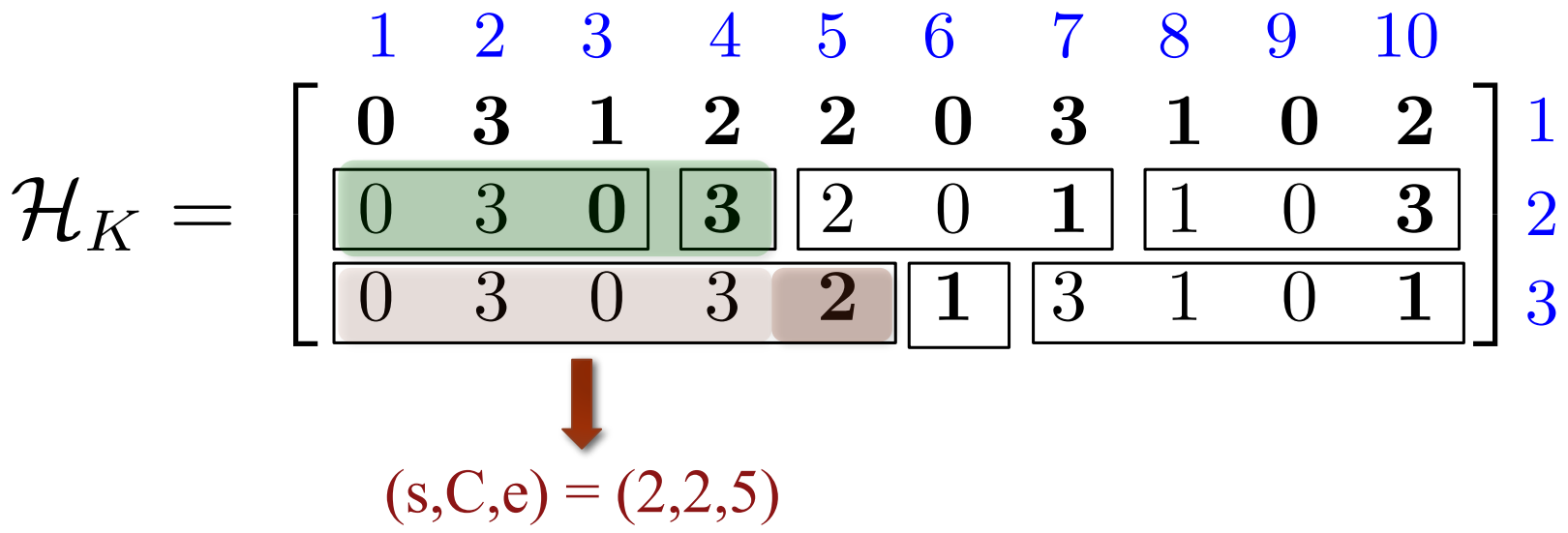
Phrase Parameters

- We represent the phrase with parameters: **(s,C,e)**

  - source row-id **(s)**

  - phrase ending position **(e)**

  - new symbol added at the end **(C)**



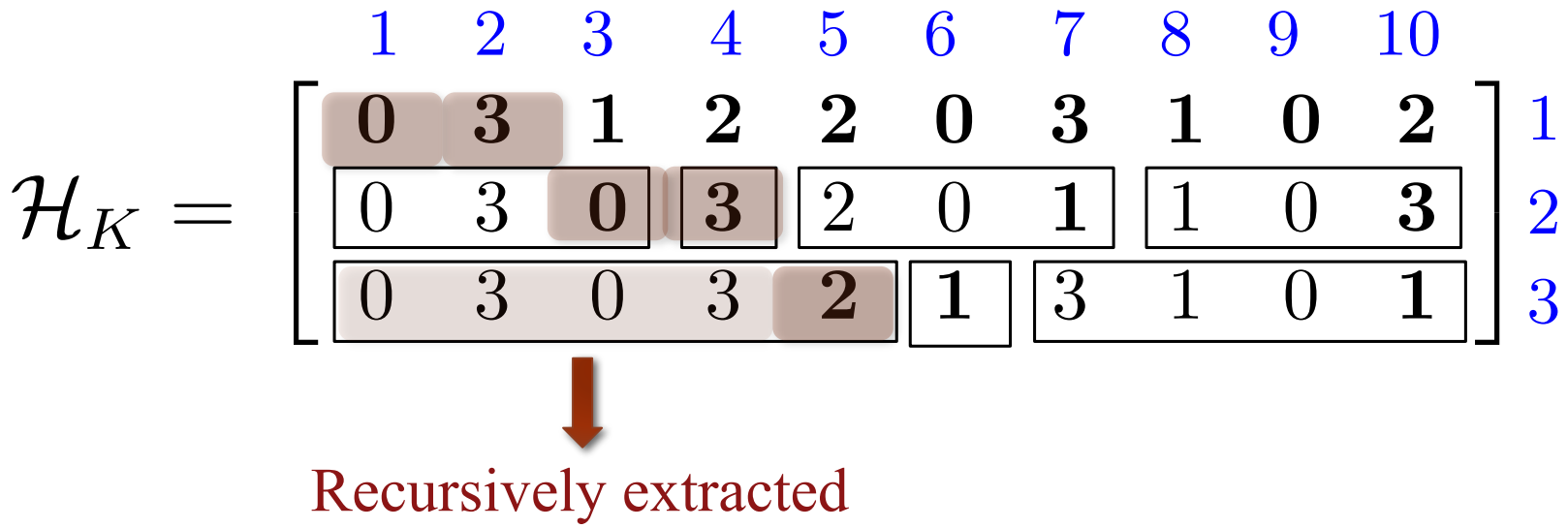
Encoding & Random Access

- Phrase parameters are encoded using
  - succinct bitvectors
  - variable-byte encoding

More details in the paper!

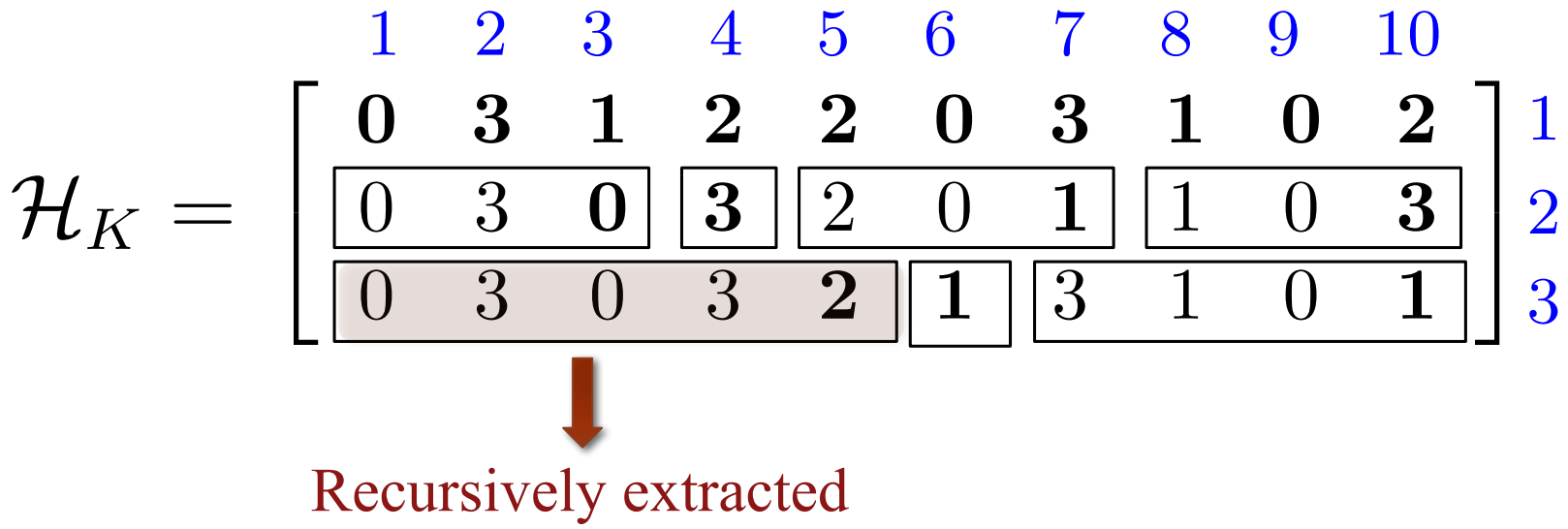
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



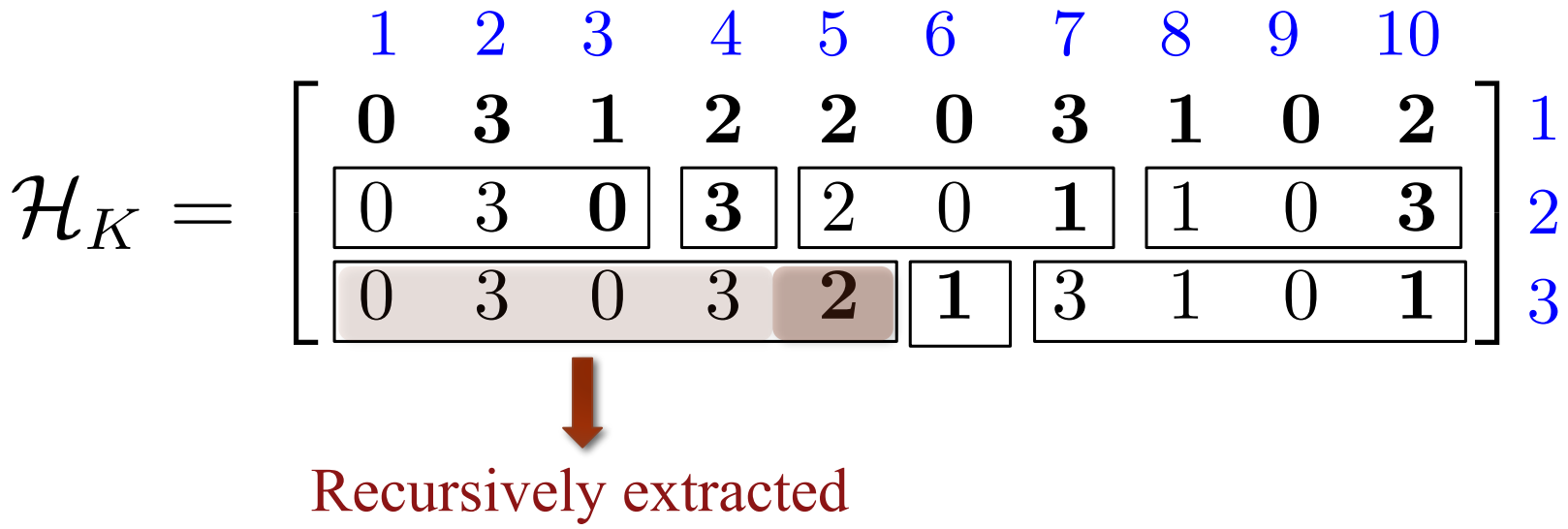
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



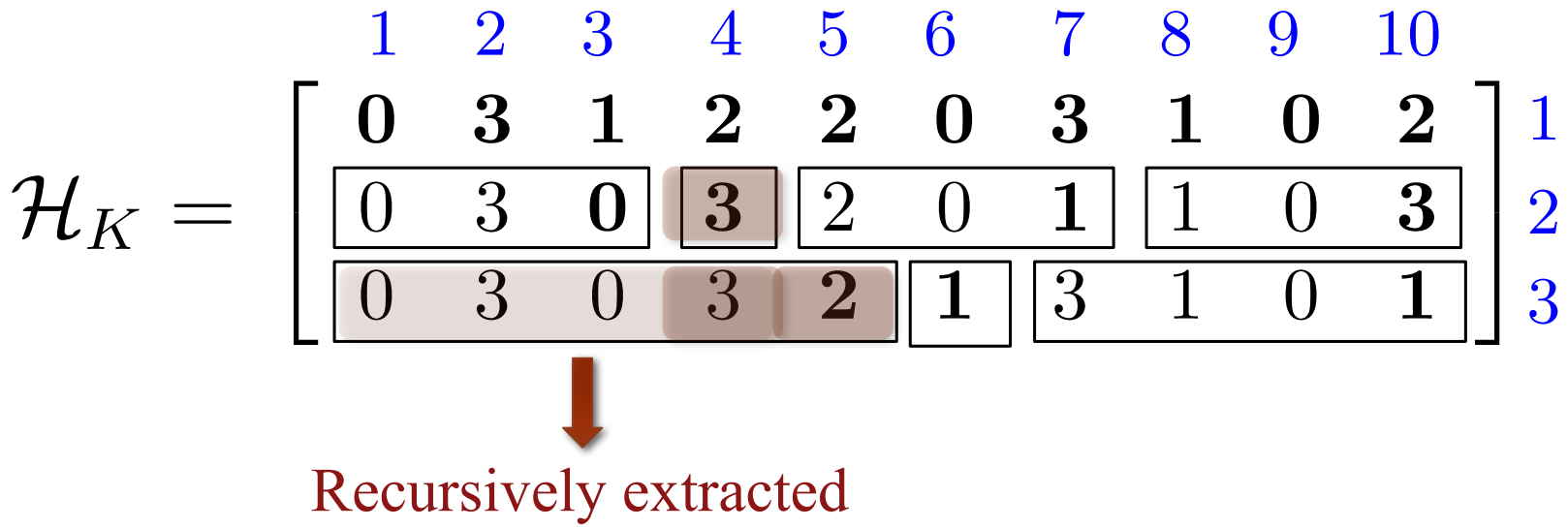
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



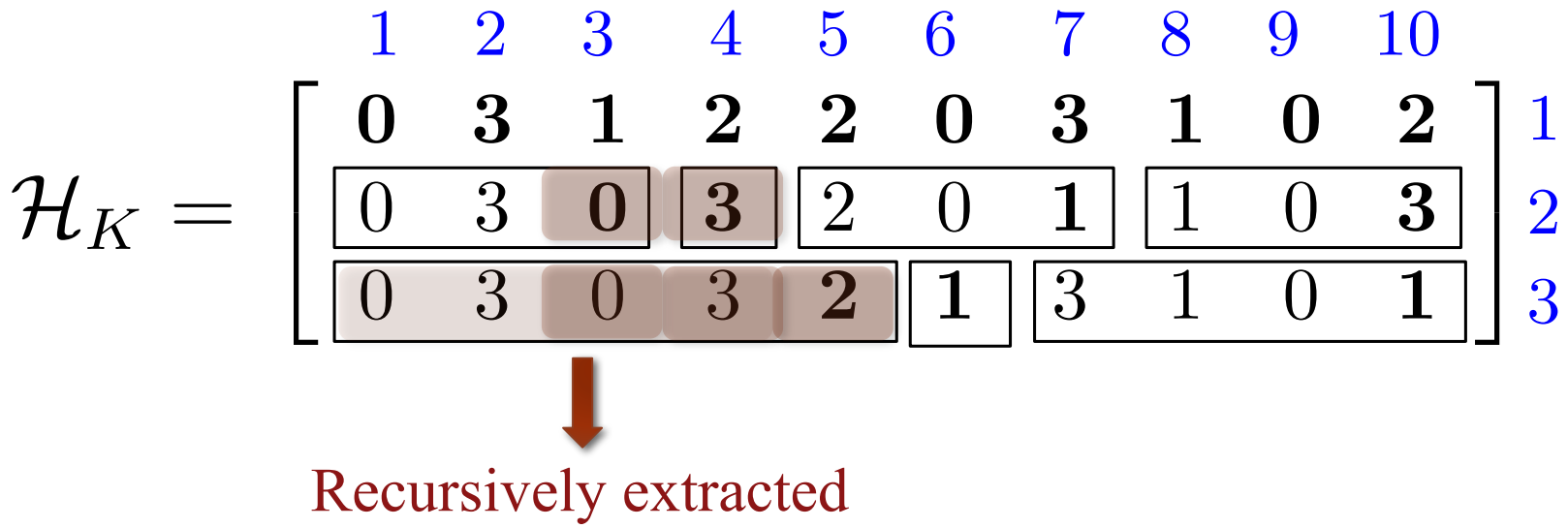
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



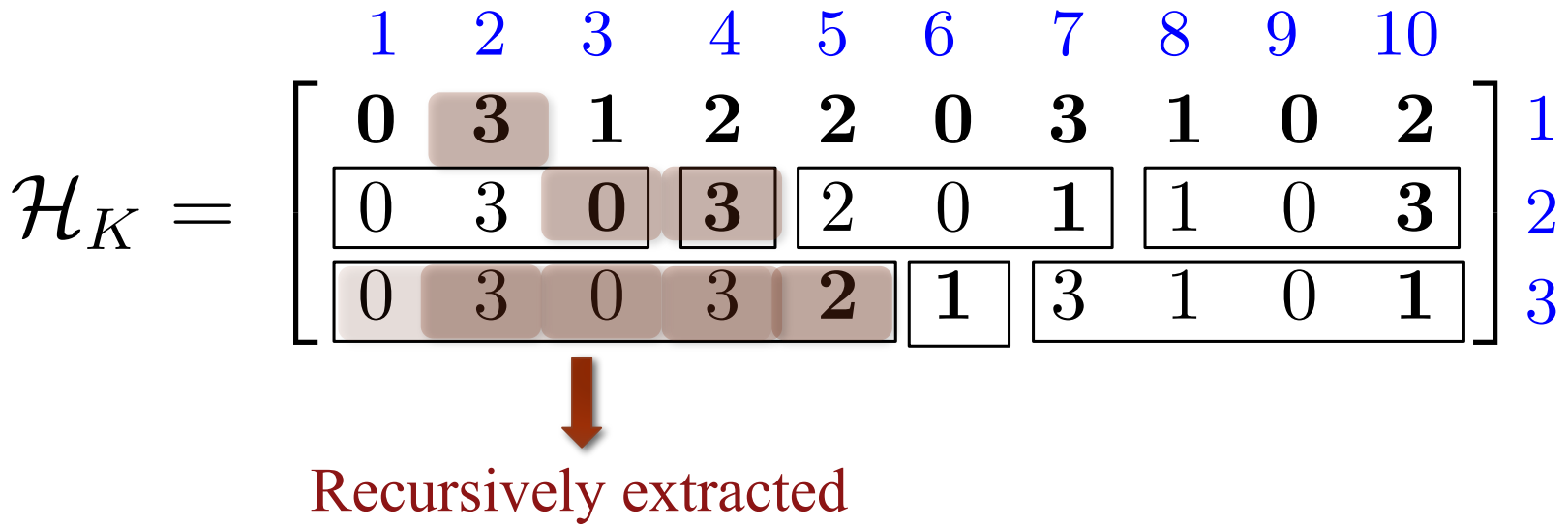
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



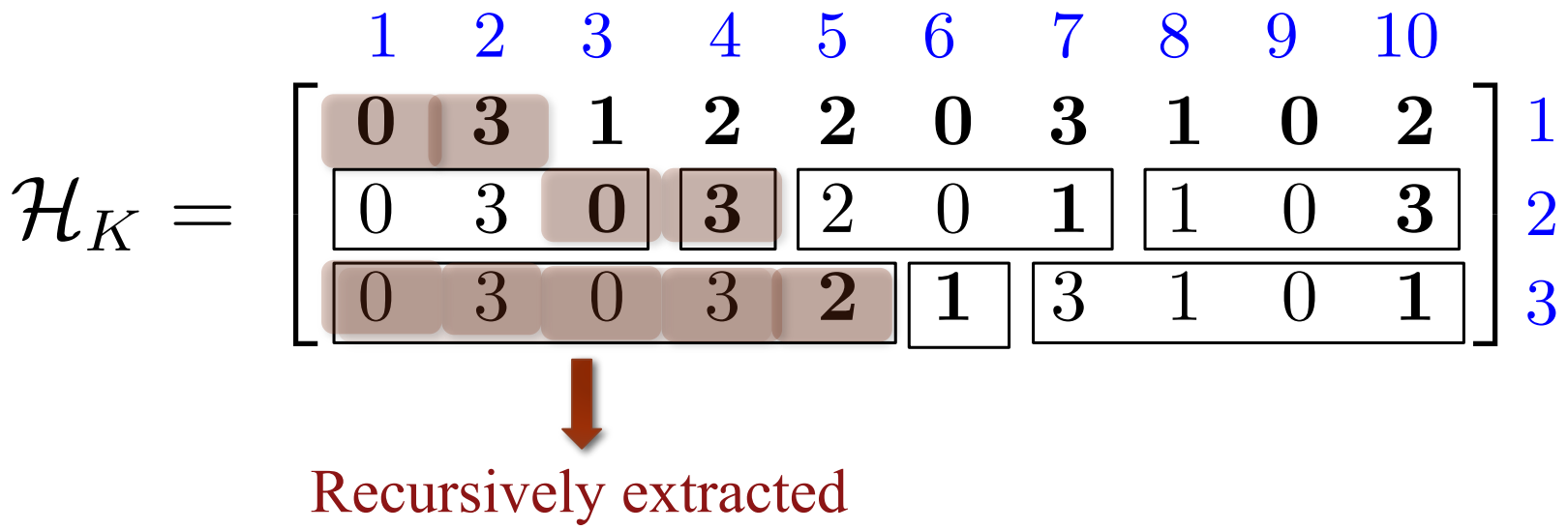
Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



Row-wise (per-genome) extraction

- Recursively extract the last symbol of a phrase
- Linear complexity of extraction (time & memory): **O(L)**
- Supports Parallel Extraction



Summary & Future Work

Summary

- Presented GTRAC, a variant dataset compressor which provides efficient random access (~100x compression, ~millisec decompression)
- GTRAC supports parallel decompression, and thus can be made faster by using higher number of cores.

Further Work

- Supporting dynamic addition and removal of genomes/variants. One step closer to a compressed dynamic variant database
- Supporting fast computations
- Adapting GTRAC algorithm for other genomic datasets

Idoia Ochoa

Mikel Hernaez

Thank You!

QUESTIONS?

Tsachy Weissman

Implementation

- Paper & Implementation available at:https://github.com/kedartatwawadi/GTRAC