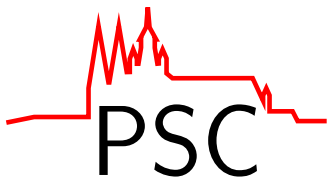
On-line Searching in IUPAC Nucleotide Sequences

Jan Holub
(joint work with Petr Procházka)

The Prague Stringology Club
Faculty of Information Technology
Czech Technical University in Prague



LSD & LAW 2019
February 7, 2019

Outline

1. Motivation
2. Basic Concepts
3. *BADPM* data structures
4. *BADPM* pattern preprocessing
5. *BADPM* searching
6. *BADPM* complexities
7. Experiments

Motivation

- DNA sequencing the population of many individuals.
- 1000 Genomes Projects, UK10K project.
- Pan-genomics: a consensus sequences is a way of representing the sequenced population.
- Consensus sequence can be expressed as so-called degenerate string.
- Need for fast on-line algorithms searching for different patterns in the consensus sequence.

Basic Concepts: IUPAC alphabet

IUPAC symbol	Subset	Bit coding
A	$\{A\}$	$\langle 0001 \rangle$
C	$\{C\}$	$\langle 0010 \rangle$
G	$\{G\}$	$\langle 0100 \rangle$
T	$\{T\}$	$\langle 1000 \rangle$
R	$\{A, G\}$	$\langle 0101 \rangle$
Y	$\{C, T\}$	$\langle 1010 \rangle$
S	$\{C, G\}$	$\langle 0110 \rangle$
W	$\{A, T\}$	$\langle 1001 \rangle$
K	$\{G, T\}$	$\langle 1100 \rangle$
M	$\{A, C\}$	$\langle 0011 \rangle$
B	$\{C, G, T\}$	$\langle 1110 \rangle$
D	$\{A, G, T\}$	$\langle 1101 \rangle$
H	$\{A, C, T\}$	$\langle 1011 \rangle$
V	$\{A, C, G\}$	$\langle 0111 \rangle$
N	$\{A, C, G, T\}$	$\langle 1111 \rangle$

Basic Concepts: DNA Consensus Sequence






homo sapiens:	T	C	T	A	G	C	A	C	T	T	A	C	T	C	T	A	T	G	C	C	T	G	C
pan paniscus:	T	C	T	A	G	C	A	C	T	T	A	C	T	C	T	A	T	G	C	C	T	G	C
chlorocebus sabaeus:	T	C	C	A	G	C	A	C	T	T	A	C	T	C	T	G	T	G	C	C	C	G	C
macaca fascicularis:	T	C	C	A	G	C	A	C	T	T	A	C	T	C	T	G	T	G	C	C	C	A	C
macaca mulatta:	T	C	C	A	G	C	A	C	T	T	A	C	T	C	T	G	T	G	C	C	C	A	C
papio anubis:	T	C	C	A	G	C	A	C	T	T	A	C	T	C	T	G	T	G	C	C	C	G	C
callithrix jacchus:	T	C	C	A	G	C	G	C	T	T	A	C	T	C	T	A	T	A	C	C	T	A	A
CONSENSUS:	T	C	Y	A	G	C	R	C	T	T	A	C	T	C	T	R	T	R	C	C	Y	R	M

Figure 1: Consensus sequence over IUPAC alphabet for different species (chromosome 7: $55\,187\,593 - 55\,187\,615$).

Basic Concepts: Degenerate Pattern Matching

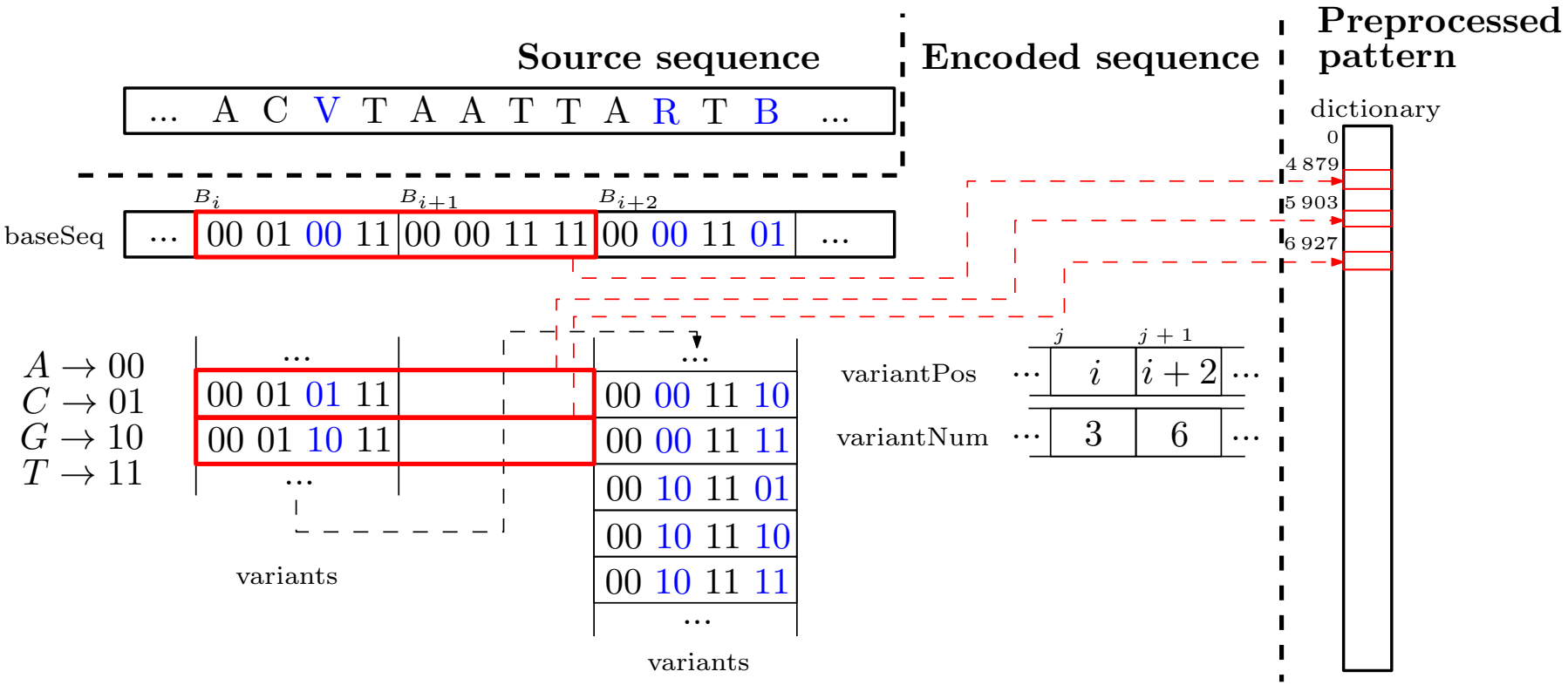
Problem

Given a degenerate text $\mathcal{T}$ and a degenerate pattern $\mathcal{P}$. The problem is to find all the occurrences of $\mathcal{P}$ in $\mathcal{T}$, i.e., to find all i such that for all j in $[1, m]$, $\mathcal{T}_{i+j-1} \cap \mathcal{P}_j \neq \emptyset$.

BADPM: Basic Properties

- Byte-Aligned Degenerate Pattern Matching (*BADPM*).
- Sublinear average time complexity in searching over consensus DNA sequences.
- Extremely fast for long patterns because of long shifts.
- Simple pattern preprocessing: tabulating all pattern factors.
- Processing at the byte level (omitting most of the bitwise operations).
- Easy cooperating with n-gram inverted index.

BADPM: Data Structures



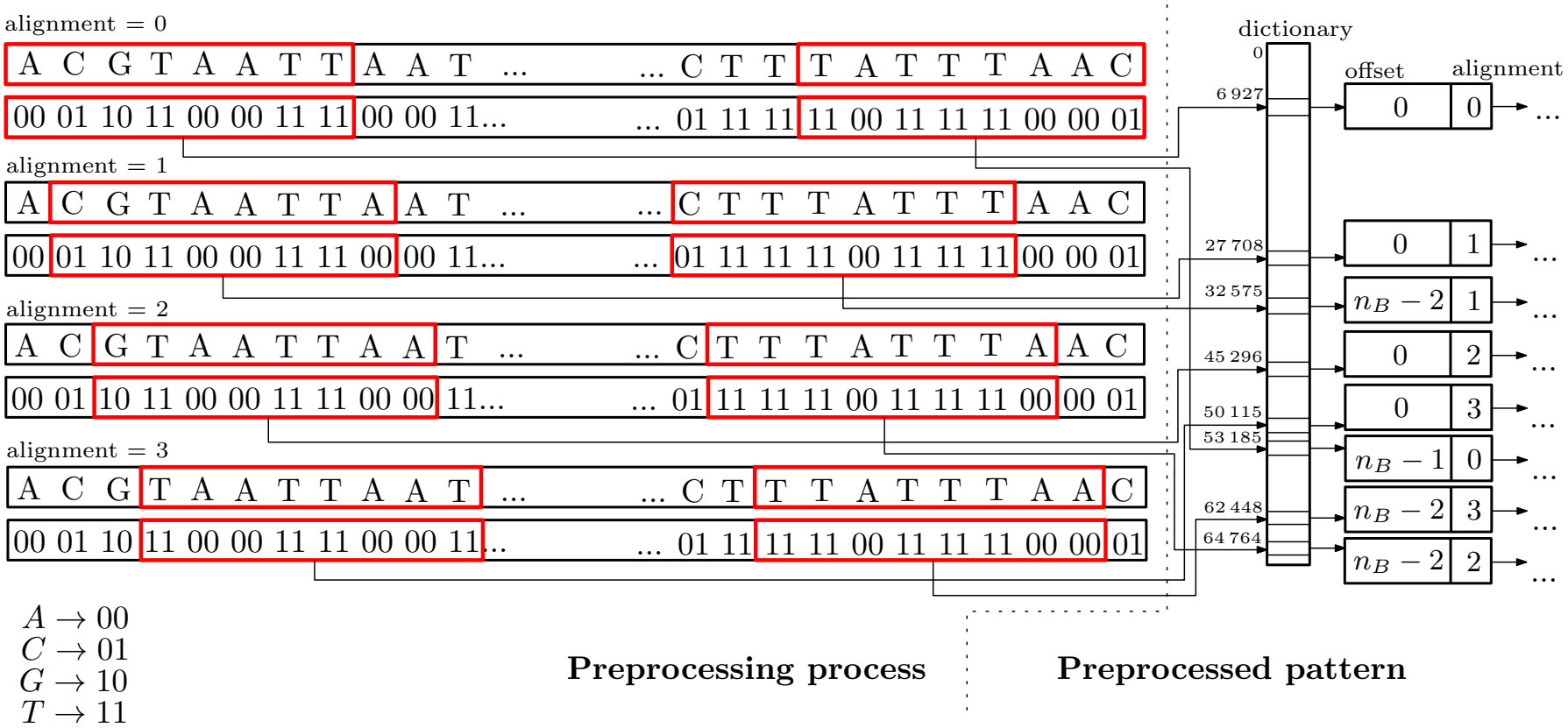
BADPM: Data structures (2)

- Consensus sequence divided into:
  - Base sequence. Consisting of only solid symbols.
  - Variants. Encoded variants (given by the degenerate symbols) in terms of a whole byte.
- Base sequence and variants encoded using bytes substituting 4-grams of symbols/bases.
- Auxiliary array *variantPos* storing positions of "degenerate bytes" in base sequence.
- Auxiliary array *variantNum* storing number of "byte variants" for a given byte.

BADPM: Data structures (3)

- Dictionary of all possible two-byte values ($256^2 = 65\,536$ values).
- Dictionary entries point to lists of occurrences (of a two-byte values) in the encoded pattern P_C.
- List elements:
  - Byte offset in terms of the encoded pattern P_C.
  - Alignment to the encoded pattern P_C.

BADPM: Pattern Preprocessing



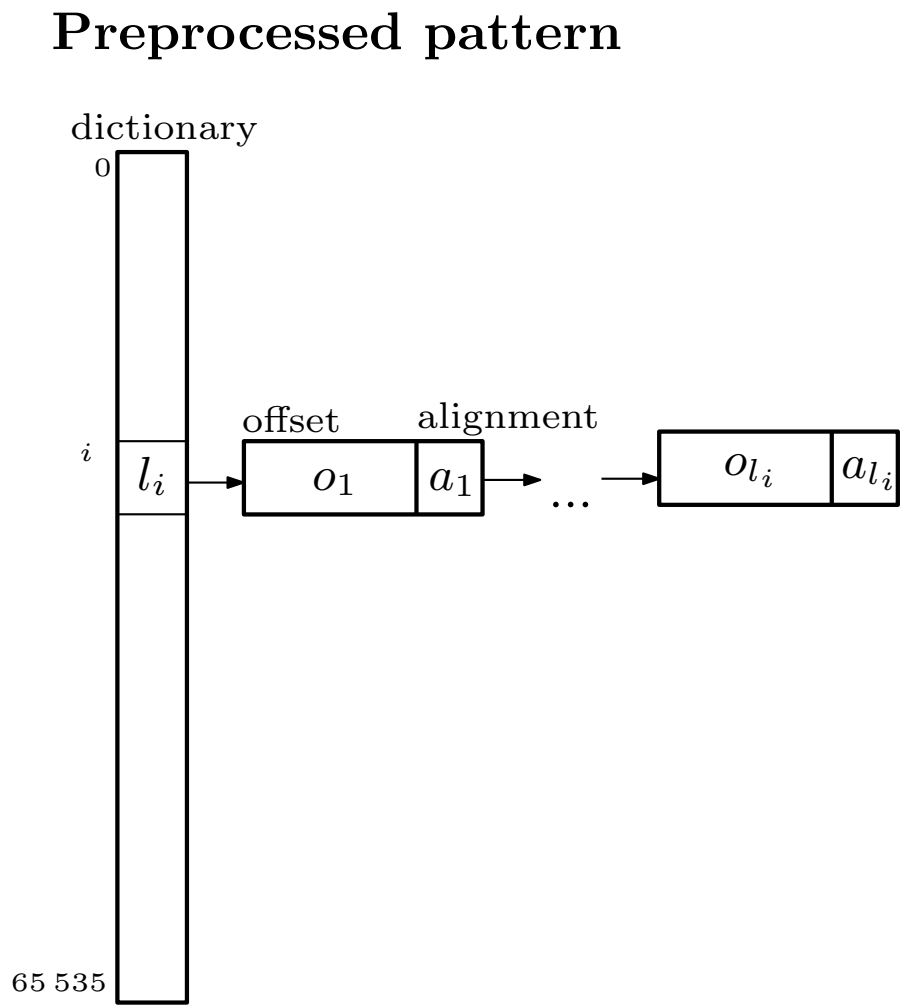
BADPM: Pattern Preprocessing (2)

For different alignments $a \in \{0, 1, 2, 3\}$:

1. Scan all relevant double-byte values.
2. Store byte offset (in terms of the encoded pattern P_E) and alignment a to the corresponding list (a dictionary entry corresponding to the double-byte value).

BADPM: Pattern Preprocessing Space

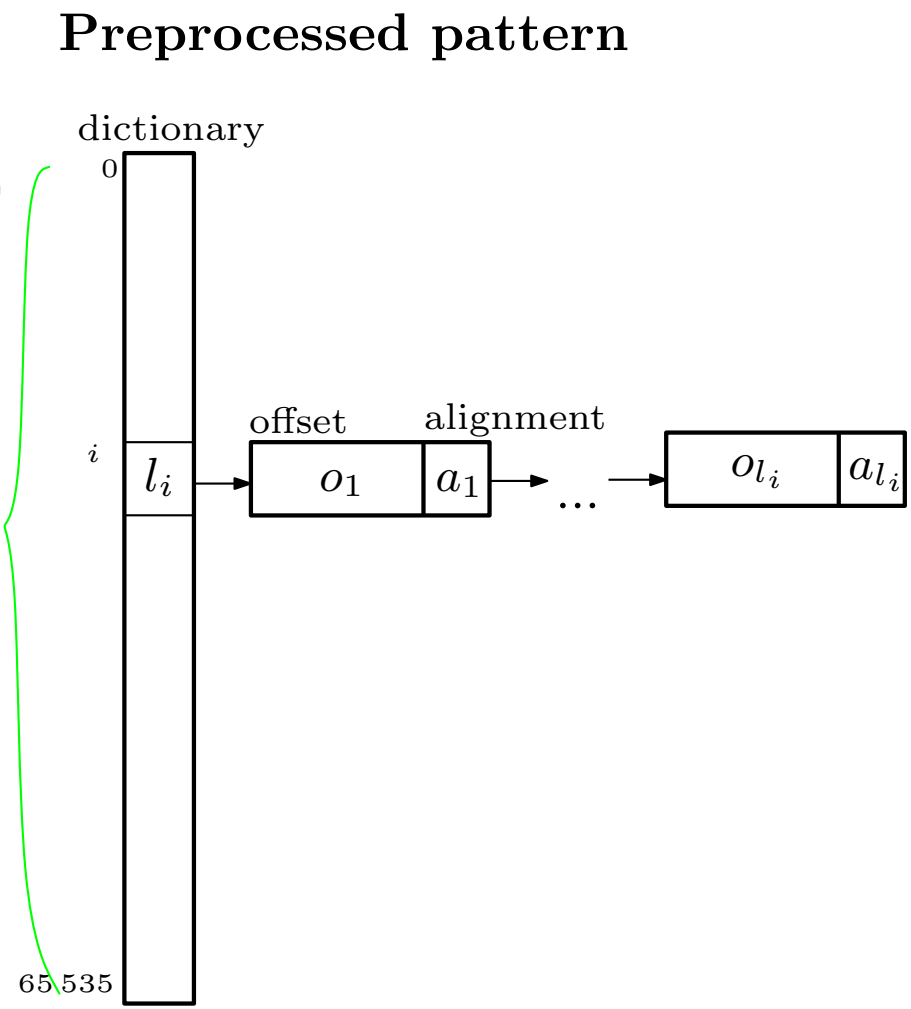
$$\mathcal{O}(m\alpha^2 \log m)$$



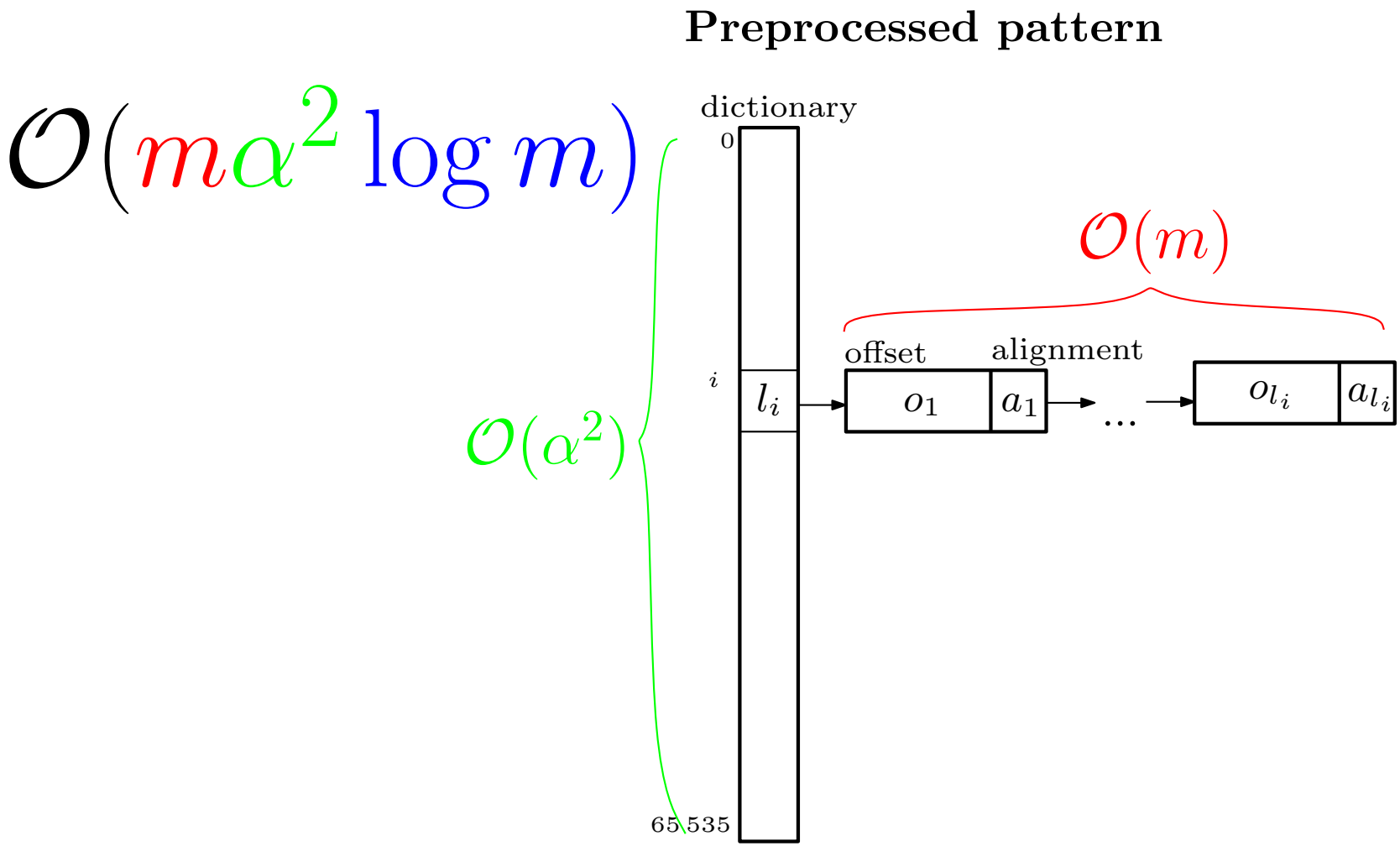
BADPM: Pattern Preprocessing Space

$$\mathcal{O}(m\alpha^2 \log m)$$

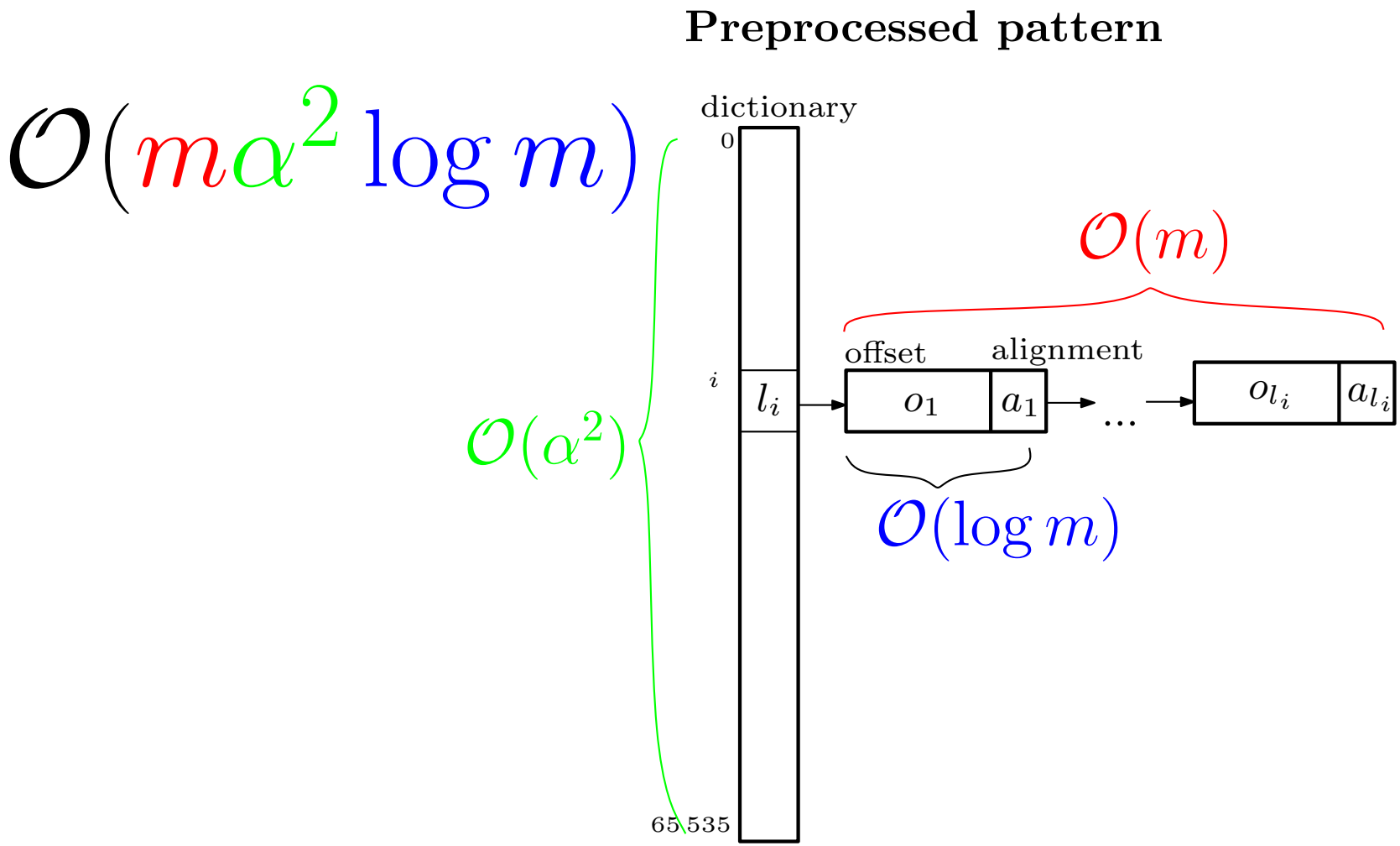
$$\mathcal{O}(\alpha^2)$$



BADPM: Pattern Preprocessing Space



BADPM: Pattern Preprocessing Space



BADPM: Pattern Preprocessing Time

$\mathcal{O}(m\alpha^2)$

- Scan $\mathcal{O}(m)$ bytes of the encoded pattern P_E.
- Check $\mathcal{O}(\alpha^2)$ double-byte values at each position (pathological patterns $\ldots NNNNNNNN\ldots$).
- Store offset and alignment for each double-byte value to the corresponding list ($\mathcal{O}(1)$ time).

BADPM Searching

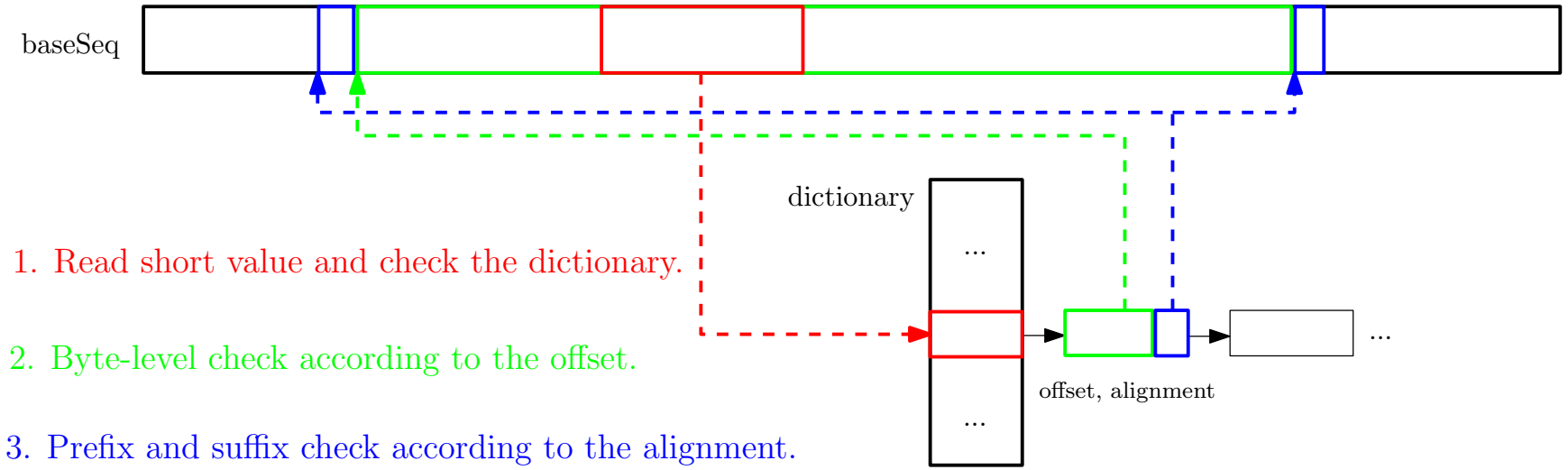
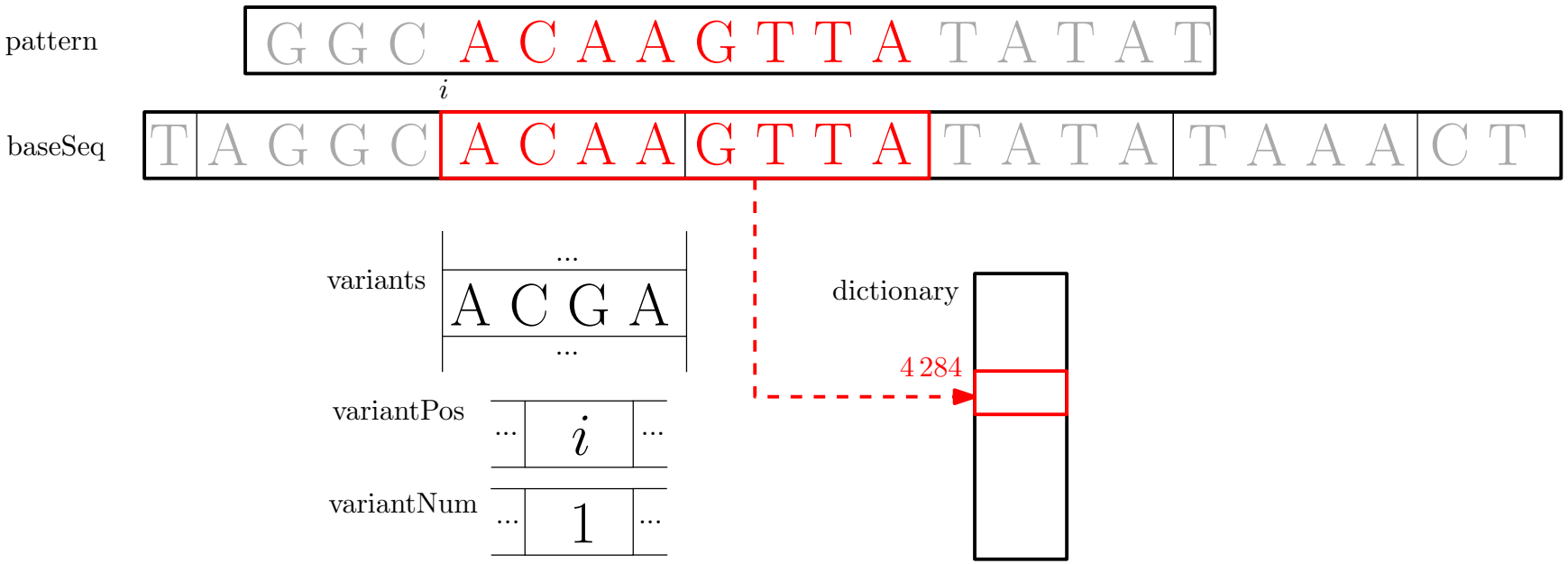
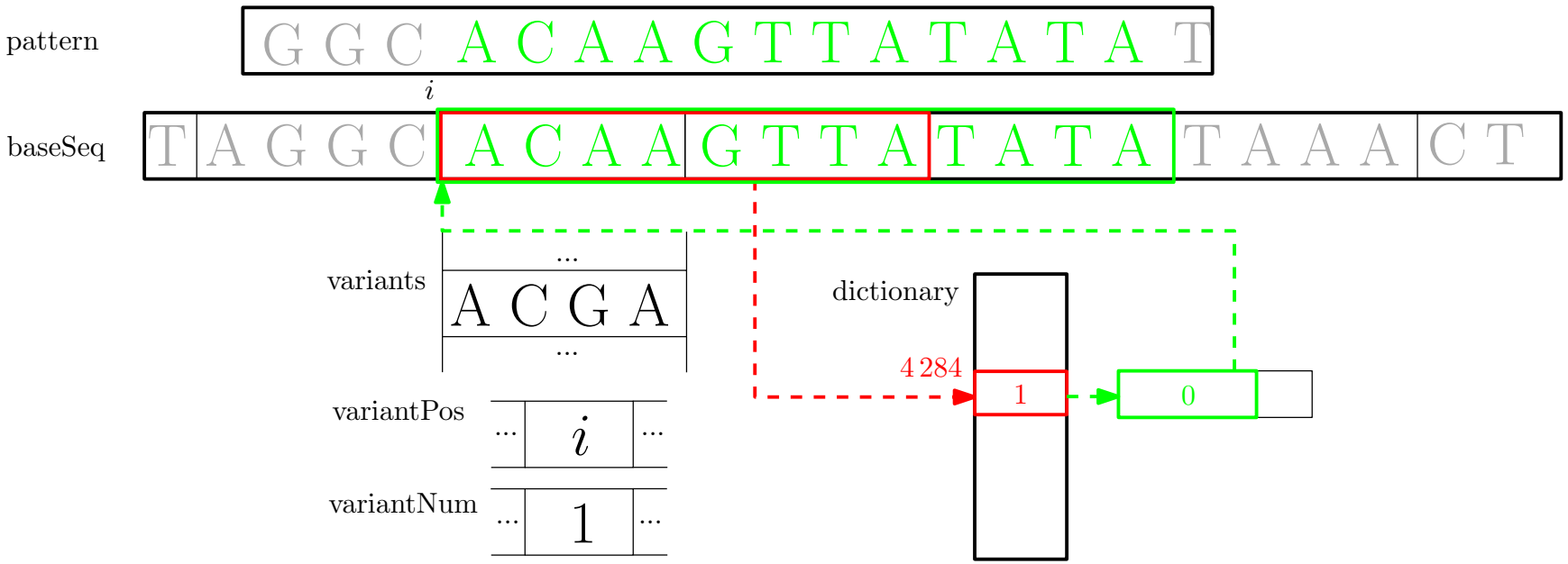


Figure 2: *BADPM*: Conceptual schema of searching.

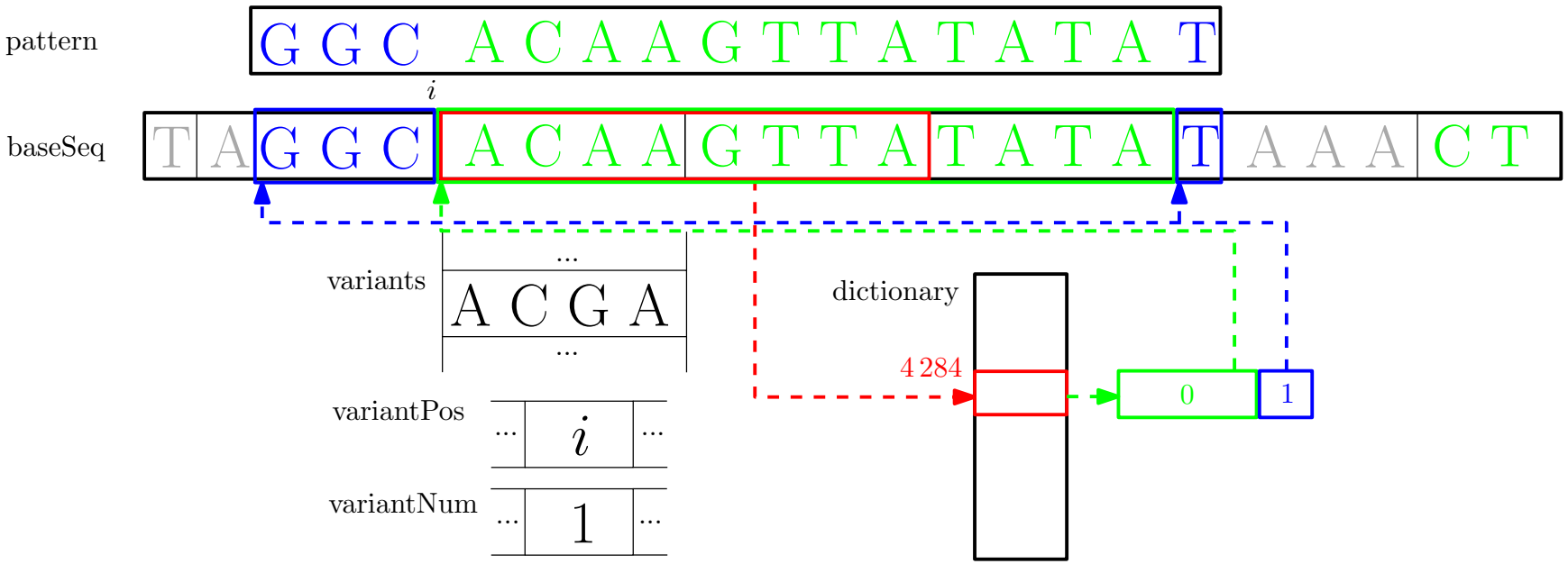
BADPM Searching: Example



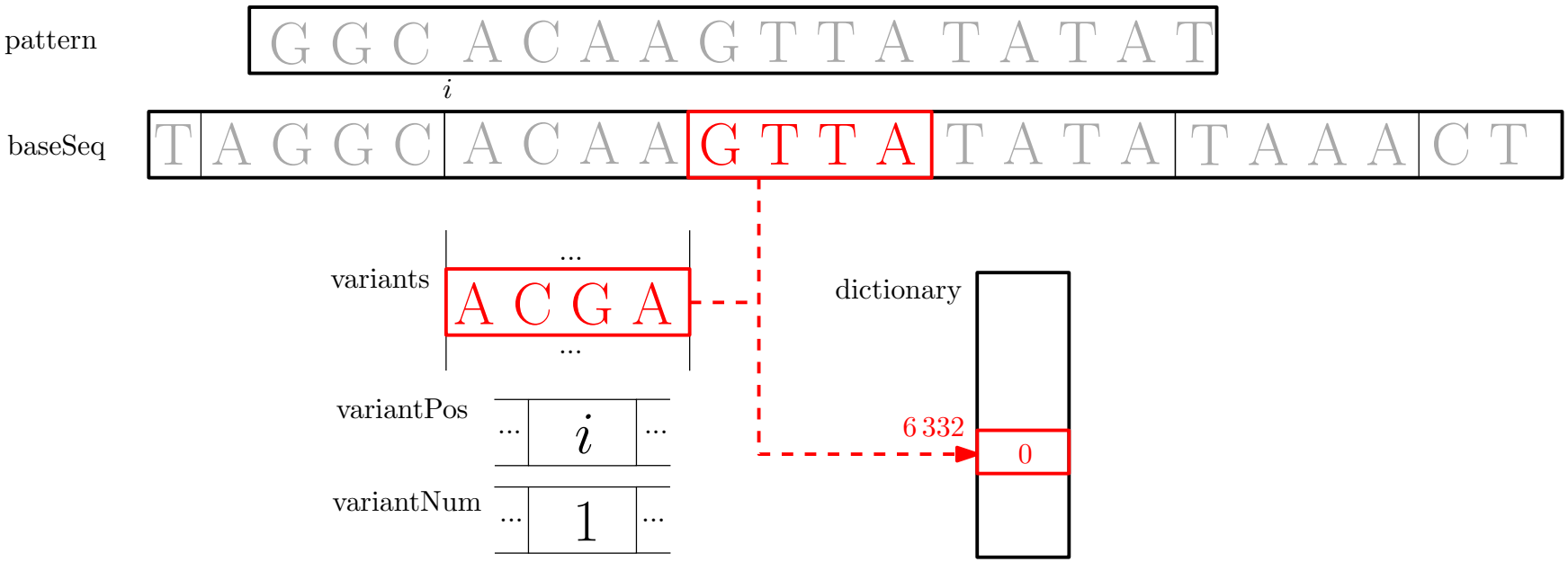
BADPM Searching: Example



BADPM Searching: Example



BADPM Searching: Example



BADPM Searching: Time Complexity

$\mathcal{O}(nm^2\alpha^4)$

- Scan $\mathcal{O}(n)$ bytes of the base sequence.
- Check $\mathcal{O}(\alpha^2)$ double-byte values at each position (pathological sequences $...NNNNNNNN...$).
- Check up to $\mathcal{O}(m)$ offsets for each double-byte value.
- Sequential byte-by-byte comparison with the encoded pattern P_E ($\mathcal{O}(m)$ bytes).
- Considering $\mathcal{O}(\alpha)$ variants for each byte of the sequence and $\mathcal{O}(\alpha)$ variants for each byte of the encoded pattern P_E (pathological sequences and patterns $...NNNNNNNN...$).

Experiments: Locate time

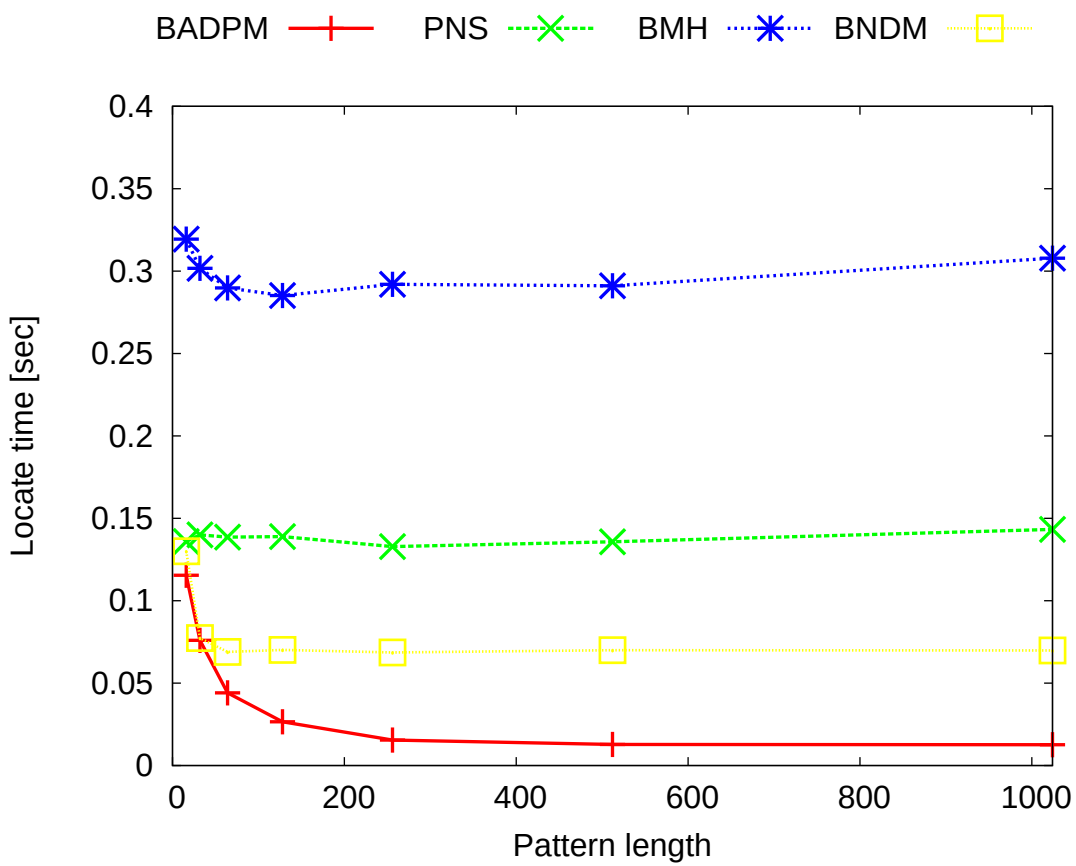


Figure 3: `Human chromosome 7`: Locate time depending on the length of the searched pattern m.

Experiments: Locate time for chromosomes

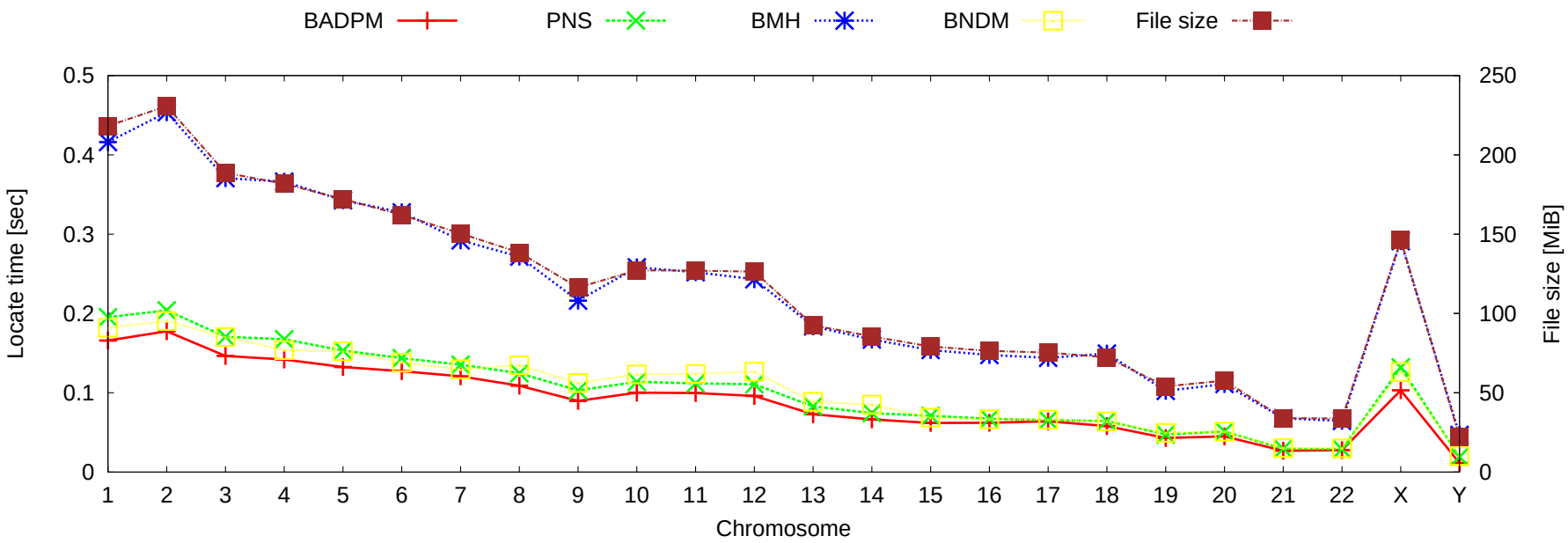


Figure 4: Locate time for different human chromosomes for $m = 16$. The second vertical axis represents the chromosome file size.

Experiments: Inverted index

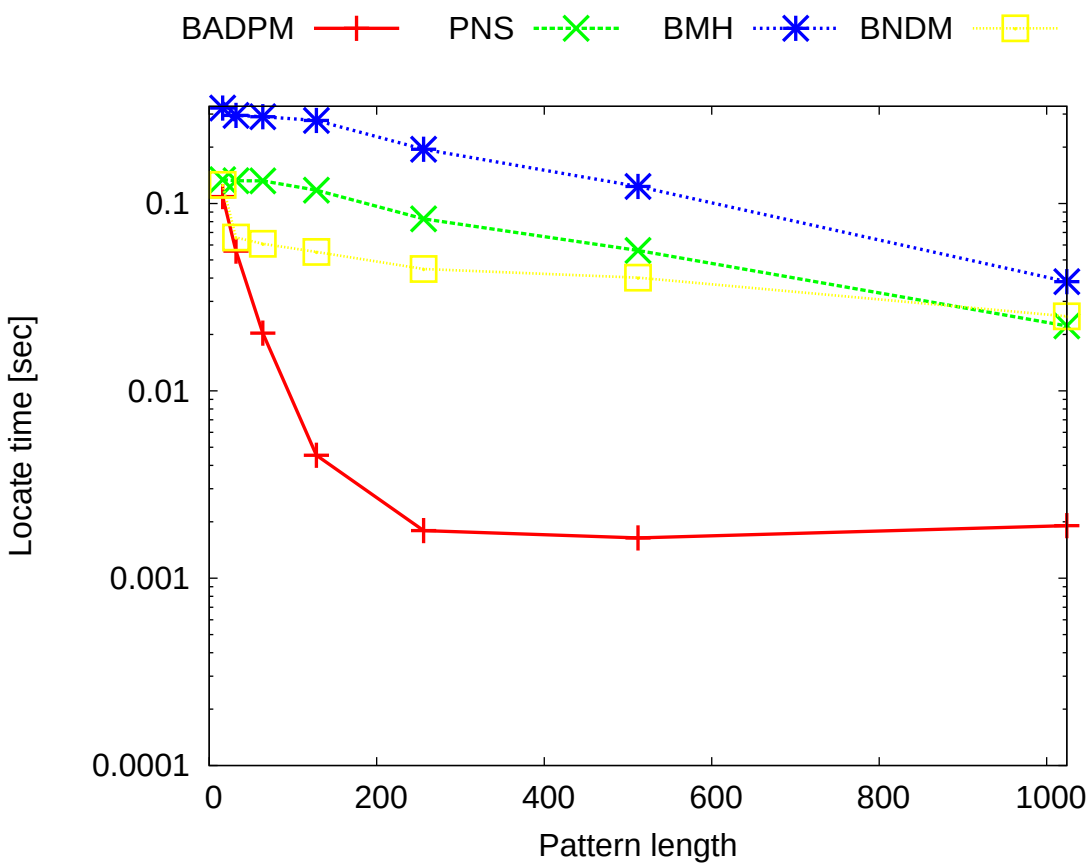


Figure 5: `Human chromosome 7:` Locate time using inverted index, block size $= 102\,400$ bases.

Thank you!

- Any questions?

- Prague Stringology Conference 2019 (August 26–28, 2019)
- postdoc position on succinct data structures in Prague (2019–2022)