Algorithms on Sequences

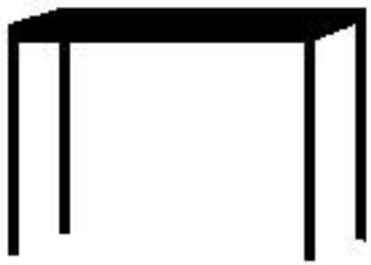
Robert Mercaș

King's College London

March 2016

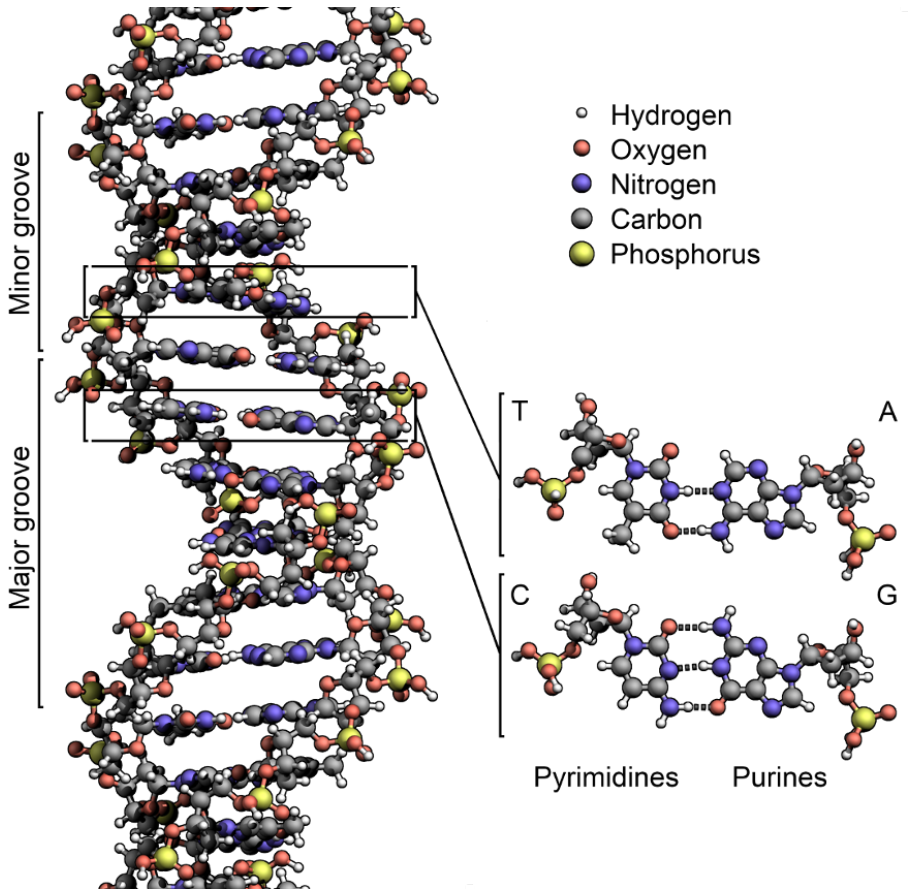
Sequences

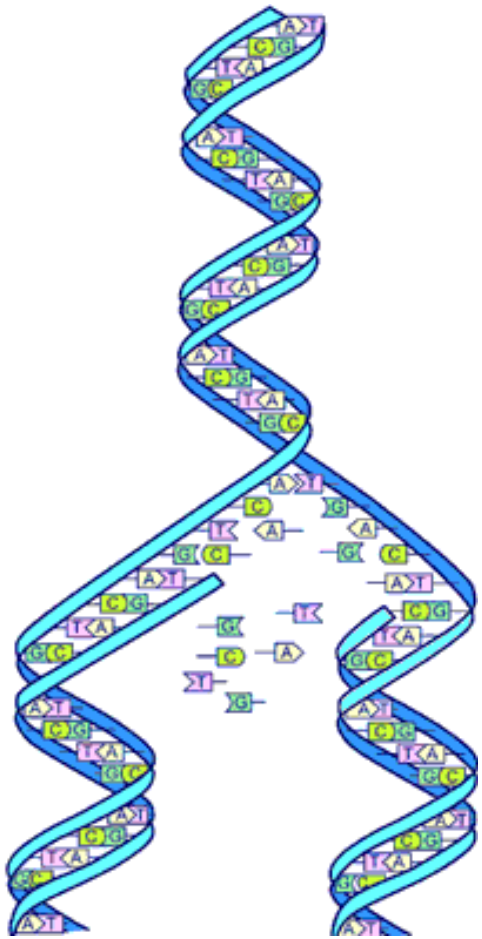
Table



DNA structure

Images from wikipedia and wiki-commons





ACGTAGGTAAAAGTACC

Vibrio Cholerae: 2,586bp from 1,108,260bp: Look for ATG

CTTGATCATACAATGAGGTCACTATGTTCGAGCTCTTCAAACCGGCTG
CGCATACGCAGCGGCTGCCATCCGATAAGGTGGACAGCGTCTATTCAC
GCCTTCGTTGGCAACTTTTCATCGGTATTTTTTGTTGGCTATGCAGGCT
ACTATTTGGTTCGTAAGAACTTTAGCTTGGCAATGCCTTACCTGATTG
AACAAGGCTTTAGTCGTGGCGATCTGGGTGTGGCTCTCGGTGCGGTTT
CAATCGCGTATGGTCTGTCTAAATTTTTGATGGGGAACGTCTCTGACC
GTTCTAACCCGCGCTACTTTCTGAGTGCAGGTCTACTCCTTTCGGCAC
TAGTGATGTTCTGCTTCGGCTTTATGCCATGGGCAACGGGCAGCATTAC
TGCGATGTTTATTCTGCTGTTCTTAAACGGCTGGTTCCAAGGCATGGG
TTGGCCTGCTTGTGGCCGTACTATGGTGCACTGGTGGTCACGCAAAGA
GCGTGGTGAGATTGTTTCGGTCTGGAACGTCGCTCACAACGTCGGTGG
TGGTTTGATTGGCCCCATTTTCCTGCTCGGCCTATGGATGTTTAACGAT
GATTGGCGCACGGCCTTCTATGTCCCCGCTTTCTTTGCGGTGCTGGTT
GCCGTATTTACTTGGCTAGTCATGCGCGATACTCCTCAATCTTGTGGTT
TACCACCGATTGAAGAGTACAAAAACGACTATCCCGATGATTACGATAAG
TCGCATGAAAATGAGATGACTGCGAAAGAGATCTTCTTTAAGTATGTCTT
CTTCATCATACAATCAGCTCACTATGTTCGAGCTCTTCAAACCGGCTGCGCATACGC

Tools

- Searching
- Comparison
- Indexing

Notations

- Alphabet: set of (ordered) distinct symbols - $\Sigma = A, C, T, G, \ldots$
- Word: adjoined symbols of the alphabet - $w = ACAACTGACAA$.
  - length: number of symbols - $|w| = 11$.
  - empty word: word of length 0 - ε.
  - factor: every subword - $ACAA$, $CAACTG$, TGA, etc.
  - prefix: any starting factor - ε, A, AC, ACA, $ACAA$, etc.
  - suffix: any ending factor - $ACAACTGACAA$, ε, A, AA, $GACAA$, etc.
  - border: any prefix that is a suffix (excluding the word) - ε, A, $ACAA$.
  - period: length of repeating sequence - $11, 10, 7$.

Note that $|w| = p(w) + |b(w)|$

Complexity

for $\ell = 0..3$
    $count_i = 0;$
    $count_j = 0;$
    $count_{const} = 0;$
    for $i = 0..n$
        $count_i = count_i + 1$
        for $j = 17..mn$
            $count_j = count_j + 1$
        for $k = 1..5$
            $count_{const} = count_{const} + 1$

Number of operations: $3+(n+1)(1+m+5) = n \cdot m + 6n + m + 9 \in \mathcal{O}(n \cdot m)$
$3n \cdot m + 18n + 3m + 27 \in \mathcal{O}(n \cdot m)$ $3n^2 - 3n - 6 \in \mathcal{O}(n^2)$

Searching

Problem

Look for the pattern x in the text y.

Questions

- How fast can we find all occurrences of the pattern in the text?
- How much extra space do we need?

Elements

- Text $y = \texttt{ACAGACAATACAAACAACAAGACA}$ and $|y| = n = 24$
- Pattern $x = \texttt{ACAA}$ and $|x| = m = 4$

Sliding window

```
ACAGACAATACAAACAACAAGACA
ACAA
 ACAA
  ACAA
   ACAA
    ACAA
```

Naïve algorithm:

- running time $\mathcal{O}(m \cdot n)$
- extra space $\mathcal{O}(1)$
- comparisons $< m \cdot n$

Bad character shift

Σ	A	C	G	T
$DA[]$	1	2	4	4

```
ACAGACAATACAAACAACAAGACA
ACAA
    ACAA
```

Naïve algorithm:

- running time $\mathcal{O}(m \cdot n)$
- extra space $\mathcal{O}(m)$
- comparisons $< m \cdot n$

Morris–Pratt

i	0	1	2	3	4
$x[i]$	A	C	A	A	
$MP[]$	-1	0	0	1	1

```
ACAAACAACACAAACAACAAGACA
ACAA
    ACAA
     ACAA
```

MP algorithm:

- running time $\mathcal{O}(n)$
- extra space $\mathcal{O}(m)$
- comparisons $< 2n$
- delay m

Boyer–Moore

i	0	1	2	3
$x[i]$	A	C	A	A
$D[]$	3	3	1	2

```
ACAAACAACACAAACAACAAGACA
ACAA
   ACAA
    ACAA
```

KMP algorithm:

- running time $\mathcal{O}(n)$
- extra space $\mathcal{O}(1)$
- comparisons $\leq 1.5n$

Sequence comparison

Problem

Compare the sequences x and y.

Questions

- How to measure the similarity?
- How fast can we compare them?
- How much extra space do we need?

Elements

- Sequence $y = \texttt{ACAGACAATACAAACAACAAGACA}$ and $|y| = n = 24$
- Sequence $x = \texttt{ACTGACATACAATACACAAGAACT}$ and $|x| = m = 24$

Hamming distance

Compare the symbols at each position (number of substitutions).

ACAGACAATACAAACAACAAGACA
ACTGACATACAATACACAAGAACT

Naïve algorithm (best):

- running time $\mathcal{O}(n)$
- extra space $\mathcal{O}(1)$
- comparisons n
- score: number of different symbols (11).

Levenshtein distance

Number of deletions, insertions, substitutions to make the sequences equal.

ACAGACAATACAATACAACAAGAACTA
ACATGACAATACAATACAACAAGAACT

Naïve algorithm (best):

- running time $\mathcal{O}(n^2)$
- extra space $\mathcal{O}(n)$
- comparisons n^2
- distance: number of operations (6).

Needleman–Wunsch

We apply the Levenshtein distance to globally align two sequence.

```
ACAGACAATACAAACAACAAGACA
ACTGACATACAATACACAAGAACT
```

Increase the distance between the sequences by $\alpha 1$ whenever we perform an insertion, deletion or substitution.

A	B
C	D

$$\mathtt{D} = \min(\mathit{cost\_op} + \mathtt{A}, \mathit{cost\_op} + \mathtt{B}, \mathit{cost\_op} + \mathtt{C})$$

Needleman–Wunsch

	ε	A	C	T	G	A	C	A	T	A	C	A	A	T	A	C	A	C	A	A	G	A	A	C	T
ε	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	3	2	1	1	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
G	4	3	2	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19	20
A	5	4	3	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19
C	6	5	4	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18
A	7	6	5	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	8	7	6	6	5	4	3	2	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T	9	8	7	6	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15	16
A	10	9	8	7	7	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15
C	11	10	9	8	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13	14
A	12	11	10	9	9	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13
A	13	12	11	10	10	9	8	7	6	5	4	3	2	3	4	5	6	6	6	7	8	9	10	11	12
A	14	13	12	11	11	10	9	8	7	6	5	4	3	3	3	4	5	6	6	6	7	8	9	10	11
C	15	14	13	12	12	11	10	9	8	7	6	5	4	4	4	3	4	5	6	7	7	8	9	9	10
A	16	15	14	13	13	12	11	10	9	8	7	6	5	5	4	4	3	4	5	6	7	7	8	9	10
A	17	16	15	14	14	13	12	11	10	9	8	7	6	6	5	5	4	4	4	5	6	7	7	8	9
C	18	17	16	15	15	14	13	12	11	10	9	8	7	7	6	5	5	4	5	5	6	7	8	7	8
A	19	18	17	16	16	15	14	13	12	11	10	9	8	8	7	6	5	5	4	5	6	6	7	8	8
A	20	19	18	17	17	16	15	14	13	12	11	10	9	9	8	7	6	6	5	4	5	6	6	7	8
G	21	20	19	18	17	17	16	15	14	13	12	11	10	10	9	8	7	7	6	5	4	5	6	7	8
A	22	21	20	19	18	17	17	16	15	14	13	12	11	11	10	9	8	8	7	6	5	4	5	6	7
C	23	22	21	20	19	18	17	17	16	15	14	13	12	12	11	10	9	8	8	7	6	5	5	5	6
A	24	23	22	21	20	19	18	17	17	16	15	14	13	13	12	11	10	9	8	8	7	6	5	6	6

Needleman–Wunsch

	ε	A	C	T	G	A	C	A	T	A	C	A	A	T	A	C	A	C	A	A	G	A	A	C	T
ε	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	3	2	1	1	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
G	4	3	2	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19	20
A	5	4	3	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19
C	6	5	4	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18
A	7	6	5	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	8	7	6	6	5	4	3	2	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T	9	8	7	6	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15	16
A	10	9	8	7	7	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15
C	11	10	9	8	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13	14
A	12	11	10	9	9	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13
A	13	12	11	10	10	9	8	7	6	5	4	3	2	3	4	5	6	6	6	7	8	9	10	11	12
A	14	13	12	11	11	10	9	8	7	6	5	4	3	3	3	4	5	6	6	6	7	8	9	10	11
C	15	14	13	12	12	11	10	9	8	7	6	5	4	4	4	3	4	5	6	7	7	8	9	9	10
A	16	15	14	13	13	12	11	10	9	8	7	6	5	5	4	4	3	4	5	6	7	7	8	9	10
A	17	16	15	14	14	13	12	11	10	9	8	7	6	6	5	5	4	4	4	5	6	7	7	8	9
C	18	17	16	15	15	14	13	12	11	10	9	8	7	7	6	5	5	4	5	5	6	7	8	7	8
A	19	18	17	16	16	15	14	13	12	11	10	9	8	8	7	6	5	5	4	5	6	6	7	8	8
A	20	19	18	17	17	16	15	14	13	12	11	10	9	9	8	7	6	6	5	4	5	6	6	7	8
G	21	20	19	18	17	17	16	15	14	13	12	11	10	10	9	8	7	7	6	5	4	5	6	7	8
A	22	21	20	19	18	17	17	16	15	14	13	12	11	11	10	9	8	8	7	6	5	4	5	6	7
C	23	22	21	20	19	18	17	17	16	15	14	13	12	12	11	10	9	8	8	7	6	5	5	5	6
A	24	23	22	21	20	19	18	17	17	16	15	14	13	13	12	11	10	9	8	8	7	6	5	6	6

```
ACAGACAATACAA-TACAACAAG-AACA
ACTGAC-AATACAATAC-AACAAGAACT
```

Needleman–Wunsch

	ε	A	C	T	G	A	C	A	T	A	C	A	A	T	A	C	A	C	A	A	G	A	A	C	T
ε	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
A	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
C	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	3	2	1	1	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
G	4	3	2	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19	20
A	5	4	3	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	16	17	18	19
C	6	5	4	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18
A	7	6	5	5	4	3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	8	7	6	6	5	4	3	2	2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T	9	8	7	6	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15	16
A	10	9	8	7	7	6	5	4	3	2	3	3	4	5	5	6	7	8	9	10	11	12	13	14	15
C	11	10	9	8	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13	14
A	12	11	10	9	9	8	7	6	5	4	3	2	3	4	5	6	5	6	7	8	9	10	11	12	13
A	13	12	11	10	10	9	8	7	6	5	4	3	2	3	4	5	6	6	6	7	8	9	10	11	12
A	14	13	12	11	11	10	9	8	7	6	5	4	3	3	3	4	5	6	6	6	7	8	9	10	11
C	15	14	13	12	12	11	10	9	8	7	6	5	4	4	4	3	4	5	6	7	7	8	9	9	10
A	16	15	14	13	13	12	11	10	9	8	7	6	5	5	4	4	3	4	5	6	7	7	8	9	10
A	17	16	15	14	14	13	12	11	10	9	8	7	6	6	5	5	4	4	4	5	6	7	7	8	9
C	18	17	16	15	15	14	13	12	11	10	9	8	7	7	6	5	5	4	5	5	6	7	8	7	8
A	19	18	17	16	16	15	14	13	12	11	10	9	8	8	7	6	5	5	4	5	6	6	7	8	8
A	20	19	18	17	17	16	15	14	13	12	11	10	9	9	8	7	6	6	5	4	5	6	6	7	8
G	21	20	19	18	17	17	16	15	14	13	12	11	10	10	9	8	7	7	6	5	4	5	6	7	8
A	22	21	20	19	18	17	17	16	15	14	13	12	11	11	10	9	8	8	7	6	5	4	5	6	7
C	23	22	21	20	19	18	17	17	16	15	14	13	12	12	11	10	9	8	8	7	6	5	5	5	6
A	24	23	22	21	20	19	18	17	17	16	15	14	13	13	12	11	10	9	8	8	7	6	5	6	6

ACAGACAATACAATACAACAAGAACA

ACTGACAATACAATACAACAAGAACT

Edit distance (with affine gap penalty) – score: 0101520253036

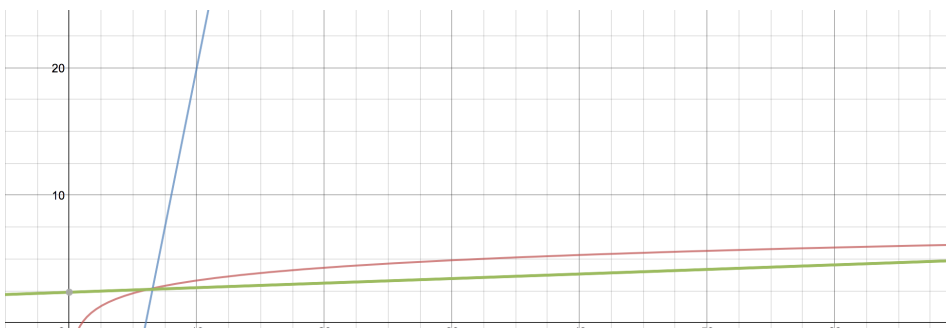
Score the operations, indels with penalty $\alpha 5$, extensions with penalty $\beta 1$, and substitutions with penalty $\gamma 10$, to equal the sequences (usually $\beta \ll \alpha < \gamma$).

```
ACAGACAATACAATACAACAAGACACT
ACATGACAATACAATACAACAAGACACT
```

$$g(k) = \alpha + \beta(k-1)$$

Edit distance algorithm:

- running time $\mathcal{O}(n^2)$
- extra space $\mathcal{O}(n)$
- comparisons n^2
- distance: sum of costs (36).



Smith–Waterman

We adjust the edit distance to locally align two sequence.

```
AAGACAATACAAACAACAAGGG
GGGAATACAAACAACTCTGTTT
```

Give a positive score of $\alpha 2$ to a match, and negative scores $\beta -1$ to indels and $\gamma -2$ substitutions.

A	B
C	D

$$\mathtt{D} = \max(0, \mathit{cost\_op} + \mathtt{A}, \mathit{cost\_op} + \mathtt{B}, \mathit{cost\_op} + \mathtt{C})$$

Smith–Waterman

	ε	A	A	G	A	C	A	A	T	A	C	A	A	A	C	A	A	C	A	A	G	G	G
ε	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	4
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	6
A	0	2	2	1	4	3	2	2	1	2	1	2	2	2	1	2	2	1	2	2	1	3	5
A	0	2	4	3	3	2	5	4	3	3	2	3	4	4	3	3	4	3	3	4	3	2	4
T	0	1	3	2	2	1	4	3	6	5	4	3	3	3	2	2	3	2	2	3	2	1	3
A	0	2	3	2	4	3	3	6	5	8	7	6	5	5	4	4	4	3	4	4	3	2	2
C	0	1	2	1	3	6	5	5	4	7	10	9	8	7	7	6	5	6	5	4	3	2	1
A	0	2	3	2	3	5	8	7	6	6	9	12	11	10	9	9	8	7	8	7	6	5	4
A	0	2	4	3	4	4	7	10	9	8	8	11	14	13	12	11	11	10	9	10	9	8	7
A	0	2	4	3	5	4	6	9	8	11	10	10	13	16	15	14	13	12	12	11	10	9	8
C	0	1	3	2	4	7	6	8	7	10	13	12	12	15	18	17	16	15	14	13	12	11	10
A	0	2	3	2	4	6	9	8	7	9	12	15	14	14	17	20	19	18	17	16	15	14	13
A	0	2	4	3	4	5	8	11	10	9	11	14	17	16	16	19	22	21	20	19	18	17	16
C	0	1	3	2	3	6	7	10	9	8	11	13	16	15	18	18	21	24	23	22	21	20	19
T	0	0	2	1	2	5	6	9	12	11	10	12	15	14	17	17	20	23	22	21	20	19	18
C	0	0	1	0	1	4	5	8	11	10	13	12	14	13	16	16	19	22	21	20	19	18	17
T	0	0	0	0	0	3	4	7	10	9	12	11	13	12	15	15	18	21	20	19	18	17	16
G	0	0	0	2	1	2	3	6	9	8	11	10	12	11	14	14	17	20	19	18	21	20	19
T	0	0	0	1	0	1	2	5	8	7	10	9	11	10	13	13	16	19	18	17	20	19	18
T	0	0	0	0	0	0	1	4	7	6	9	8	10	9	12	12	15	18	17	16	19	18	17
T	0	0	0	0	0	0	0	3	6	5	8	7	9	8	11	11	14	17	16	15	18	17	16

Smith–Waterman

	ε	A	A	G	A	C	A	A	T	A	C	A	A	A	C	A	A	C	A	A	G	G	G
ε	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	4
G	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	6
A	0	2	2	1	4	3	2	2	1	2	1	2	2	2	1	2	2	1	2	2	1	3	5
A	0	2	4	3	3	2	5	4	3	3	2	3	4	4	3	3	4	3	3	4	3	2	4
T	0	1	3	2	2	1	4	3	6	5	4	3	3	3	2	2	3	2	2	3	2	1	3
A	0	2	3	2	4	3	3	6	5	8	7	6	5	5	4	4	4	3	4	4	3	2	2
C	0	1	2	1	3	6	5	5	4	7	10	9	8	7	7	6	5	6	5	4	3	2	1
A	0	2	3	2	3	5	8	7	6	6	9	12	11	10	9	9	8	7	8	7	6	5	4
A	0	2	4	3	4	4	7	10	9	8	8	11	14	13	12	11	11	10	9	10	9	8	7
A	0	2	4	3	5	4	6	9	8	11	10	10	13	16	15	14	13	12	12	11	10	9	8
C	0	1	3	2	4	7	6	8	7	10	13	12	12	15	18	17	16	15	14	13	12	11	10
A	0	2	3	2	4	6	9	8	7	9	12	15	14	14	17	20	19	18	17	16	15	14	13
A	0	2	4	3	4	5	8	11	10	9	11	14	17	16	16	19	22	21	20	19	18	17	16
C	0	1	3	2	3	6	7	10	9	8	11	13	16	15	18	18	21	24	23	22	21	20	19
T	0	0	2	1	2	5	6	9	12	11	10	12	15	14	17	17	20	23	22	21	20	19	18
C	0	0	1	0	1	4	5	8	11	10	13	12	14	13	16	16	19	22	21	20	19	18	17
T	0	0	0	0	0	3	4	7	10	9	12	11	13	12	15	15	18	21	20	19	18	17	16
G	0	0	0	2	1	2	3	6	9	8	11	10	12	11	14	14	17	20	19	18	21	20	19
T	0	0	0	1	0	1	2	5	8	7	10	9	11	10	13	13	16	19	18	17	20	19	18
T	0	0	0	0	0	0	1	4	7	6	9	8	10	9	12	12	15	18	17	16	19	18	17
T	0	0	0	0	0	0	0	3	6	5	8	7	9	8	11	11	14	17	16	15	18	17	16

```
GACAATACAAACAAC
GA--ATACAAACAAC
```

Indexing

Problem

Look for different patterns in the text y.

Questions

- How to pre-process the text such that we can find fast occurrences of different patterns in the text?
- How much extra space do we need?

Elements

- Text $y = \texttt{ACAGACAATACAAACAACAAGACA}$ and $|y| = 24$.
- Various patterns.

Naïve algorithm

Search for each of the patterns through the text, one pattern at the time.

Elements

- Text $y = \texttt{ACAGACAATACAAACAACAAGACA}$
- Patterns $\{\texttt{AA}, \texttt{CAA}, \texttt{TAC}, \texttt{AGA}, \texttt{AC}\}$

Naïve algorithm:

- running time $\mathcal{O}(k \cdot n)$
- extra space $\mathcal{O}(k \cdot (\max_{i=0}^{k} m_i)) + occ)$
- comparisons $n \cdot (\sum_{i=0}^{k} m_i))$

Searching a list of strings

Elements

- A list{ACAGACAATACAAACAACAAGACA, CAACAAGACA, AAGACA, CAGACAATACAAACAACAAGACA, GACAATACAAACAACAAGACA, ACA, TACAAACAACAAGACA, AAACAACAAGACA, ATACAAACAACAAGACA}
- Pattern $x =$ ACAGACAA

1. AAACAACAAGACA
2. AAGACA
3. ACA
4. ACAGACAATACAAACAACAAGACA
5. ATACAAACAACAAGACA
6. CAACAAGACA
7. CAGACAATACAAACAACAAGACA
8. GACAATACAAACAACAAGACA
9. TACAAACAACAAGACA

Search algorithm:

- sorting time $\mathcal{O}(\sum_{i=0}^{k} n_i))$
- extra space $\mathcal{O}(k)$
- search time $\mathcal{O}(m + \log k)$
- comparisons $\leq m + \lceil \log(k+1) \rceil$

Suffix array: ACAGACAATACAAACAACAAGACA

ACAGACAATACAAACAACAAGACA	AACAACAAGACA
CAGACAATACAAACAACAAGACA	ACAACAAGACA
AGACAATACAAACAACAAGACA	CAACAAGACA
GACAATACAAACAACAAGACA	AACAAGACA
ACAATACAAACAACAAGACA	ACAAGACA
CAATACAAACAACAAGACA	CAAGACA
AATACAAACAACAAGACA	AAGACA
ATACAAACAACAAGACA	AGACA
TACAAACAACAAGACA	GACA
ACAAACAACAAGACA	ACA
CAAACAACAAGACA	CA
AAACAACAAGACA	A

Suffix array: ACAGACAATACAAACAACAAGACA

```
A
AAACAACAAGACA
AACAAGACA
AACAACAAGACA
AAGACA
AATACAAACAACAAGACA
ACA
ACAAACAACAAGACA
ACAACAAGACA
ACAAGACA
ACAATACAAACAACAAGACA
ACAGACAATACAAACAACAAGACA
AGACA
AGACAATACAAACAACAAGACA
ATACAAACAACAAGACA
CA
CAAACAACAAGACA
CAACAAGACA
CAAGACA
CAATACAAACAACAAGACA
CAGACAATACAAACAACAAGACA
GACA
GACAATACAAACAACAAGACA
TACAAACAACAAGACA
```

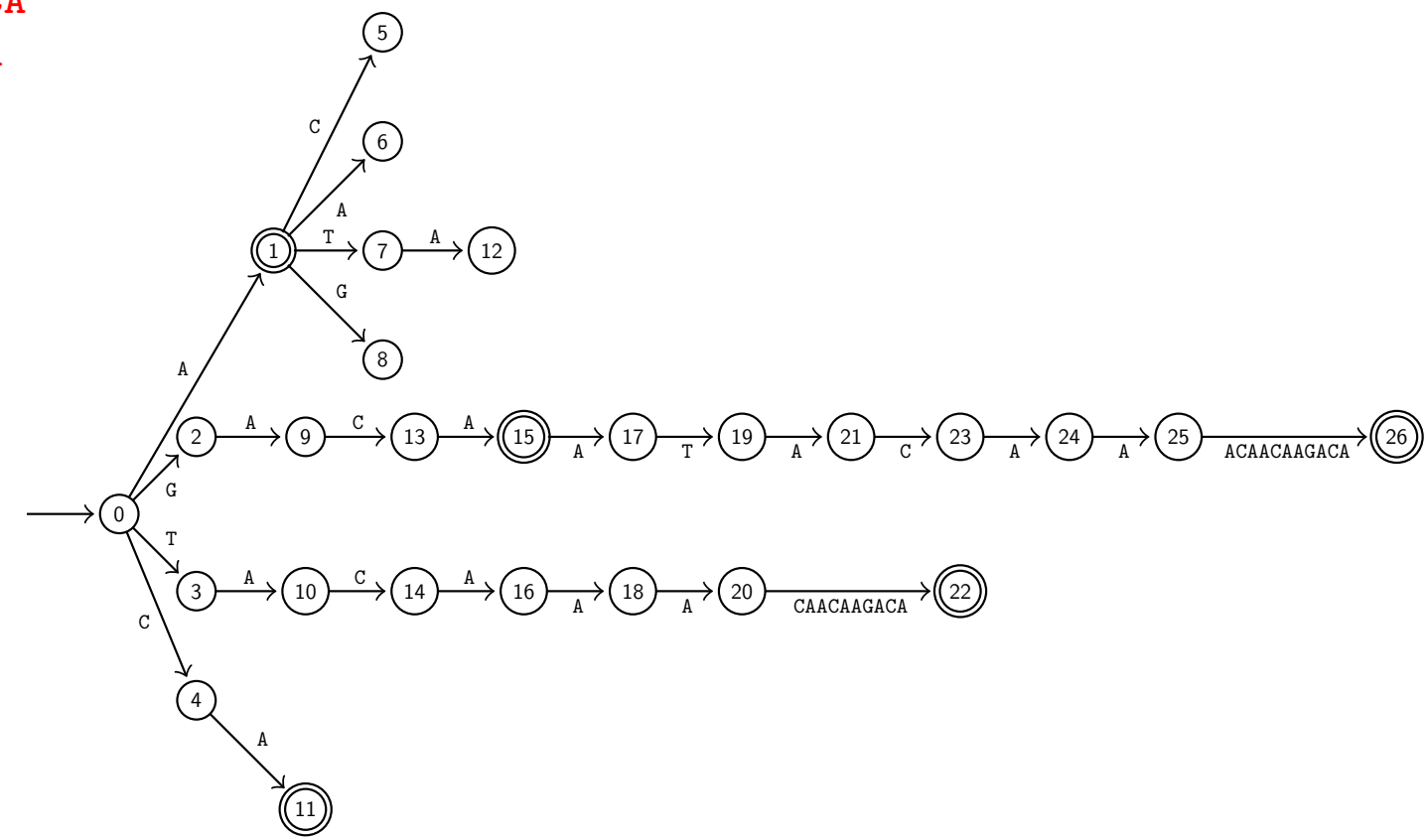
Pattern: ATACAAACAACAAGACA

Suffix algorithm:

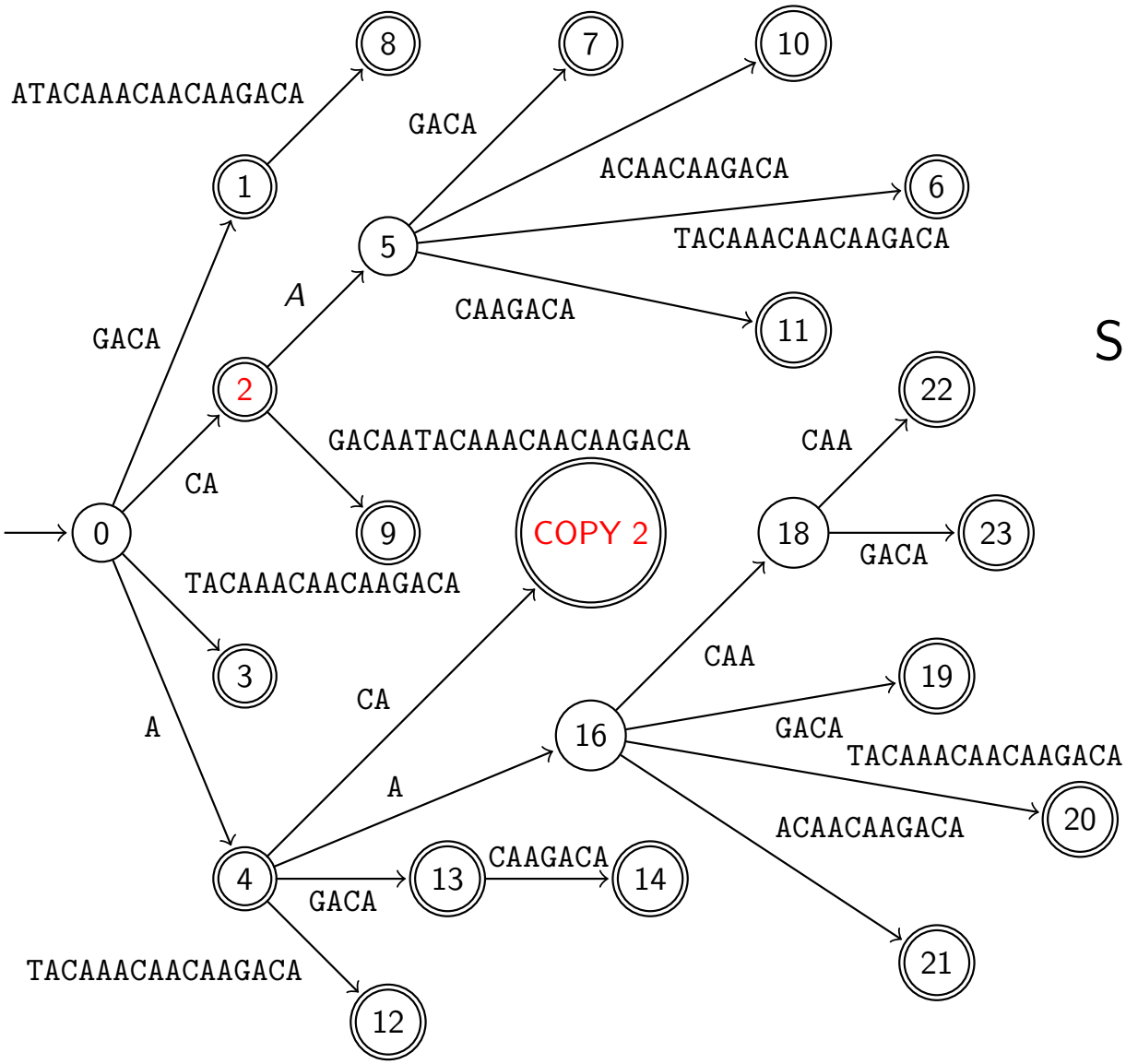
- running time $\mathcal{O}(n)$
- extra space $\mathcal{O}(n)$
- search $\mathcal{O}(m)$

Suffix tree: ACAGACAATACAAACAACAAGACA

ACAGACAATACAAACAACAAGACA
CAGACAATACAAACAACAAGACA
AGACAATACAAACAACAAGACA
GACAATACAAACAACAAGACA
ACAATACAAACAACAAGACA
CAATACAAACAACAAGACA
AATACAAACAACAAGACA
ATACAAACAACAAGACA
TACAAACAACAAGACA
ACAAACAACAAGACA
CAAACAACAAGACA
AAACAACAAGACA
AACAACAAGACA
ACAACAAGACA
CAACAAGACA
AACAAGACA
ACAAGACA
CAAGACA
AAGACA
AGACA
GACA
ACA
CA
A



Suffix tree: ACAGACAATACAAACAACAAGACA



Suffix tree algorithm:

- nodes $\leq 2n$
- extra space $\mathcal{O}(n)$
- running time $\mathcal{O}(n)$
- search $\mathcal{O}(m)$

And then...

Other operations:

- approximate matching
- regular expressions (with reversals)
- various updates (concatenation, diff, deletion, swap)

Usage

- repeats detection (biology, music, forensics)
- auto-prediction
- fraud (plagiat) detection
- phylogeny trees construction