Miguel Alcón Doganoc
Advanced Data Structures
June 21, 2019

Exact String Matching Algorithms

Introduction

This document is related with the first work proposed at the Bioinformatic lectures. The purpose of this work is to check if the theory explained in class, about the search algorithms for only one pattern, is still true. As it is said in the statement, we have to compare the performance of 4 algorithms: *Brute Force* (BF), *Horspool* (H), *Backward Nondeterministic Dawg Matching* (BNDM) and *Backward Oracle Matching* (BOM). The comparison is made over a DNA alphabet file of at least 2GB, with random patterns of length $k > 0$.

Experiment

First, I downloaded the codes of the algorithms from [1] and the 2.25GB string from [2]. Then, I had to modify each of the codes to let them read an external file and accept several parameters. These parameters are the DNA alphabet file and n pairs composed by the size of a pattern and the pattern itself. As an example of how to call one of the algorithms:

$$\$ \text{ ./algorithm file\_path size}_1 \text{ pattern}_1 \text{ size}_2 \text{ pattern}_2 \text{ ... size}_n \text{ pattern}_n$$

Moreover, I created a *Python* script to generate a random pattern of length k (input). Finally, I created a *Bash* script that is in charge of generating a string with all the parameters, and then to call each of the algorithms with it. I executed it with $n = 100$, with the size of the patterns from 10 to 1000 with steps of 10, but once $k > 10$ it does not find anything. So I executed it again with $n = 100$, but with the size of the patterns from 1 to 20.

Results

As you can see in figure 1 and 2, the best algorithm when searching small patterns ($k \leq 5$) is H, and BNDM is the best when searching big ones ($k > 5$). In figure 3 we can check that all the algorithms find the same number of patterns. Because the difference between the found patterns with $k = 1$ and $k = 15$ is huge (429877019 v.s. 1), in the figure you cannot see clear the number of patterns found. I put them in table 1. With this, we can conclude that the theory explained in class is true.

References

[1] Thierry Lecroq Christian Charras. Exact string matching algorithms. http://www-igm.univ-mlv.fr/~lecroq/string/index.html, 1997.

[2] National Center for Biotechnology Information. Vicugna pacos. ftp://ftp.ncbi.nih.gov/genomes/Vicugna_pacos/CHR_Un/vpa_ref_Vicugna_pacos-2.0.1_chrUn.fa.gz, 2015.

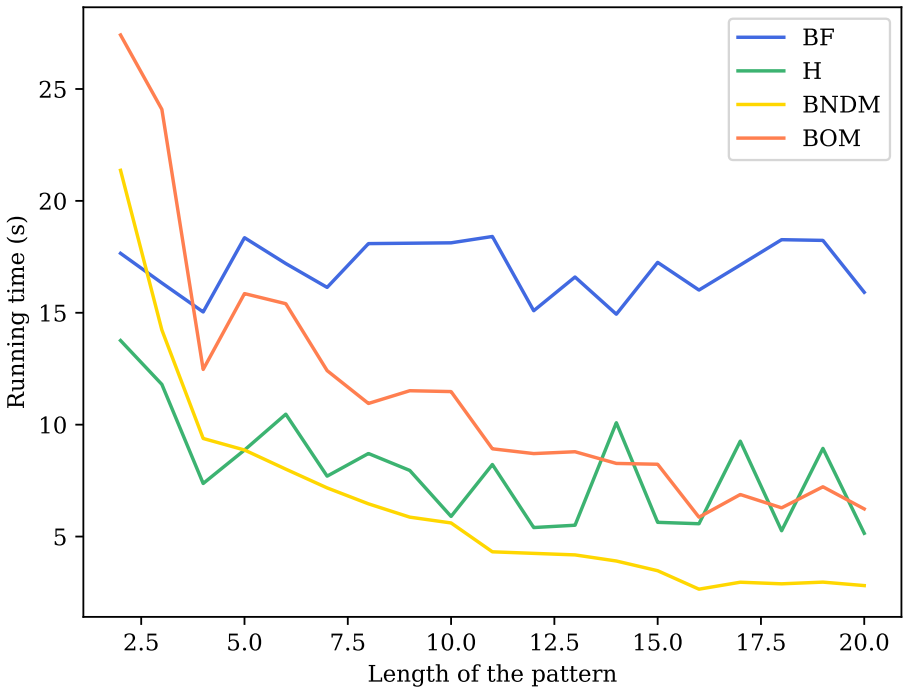

Figure 1: Running time of each algorithm

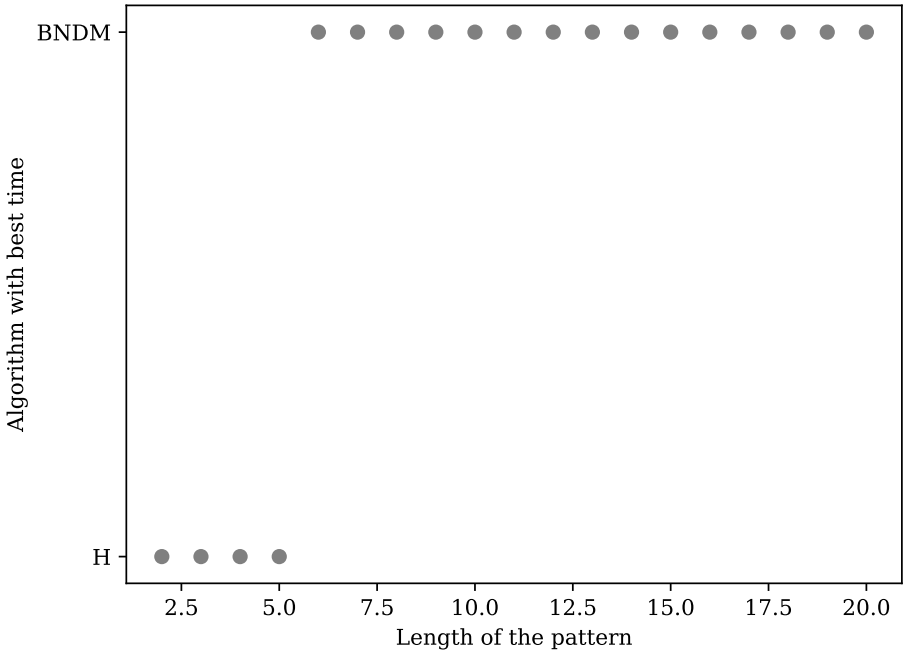

Figure 2: Best algorithms

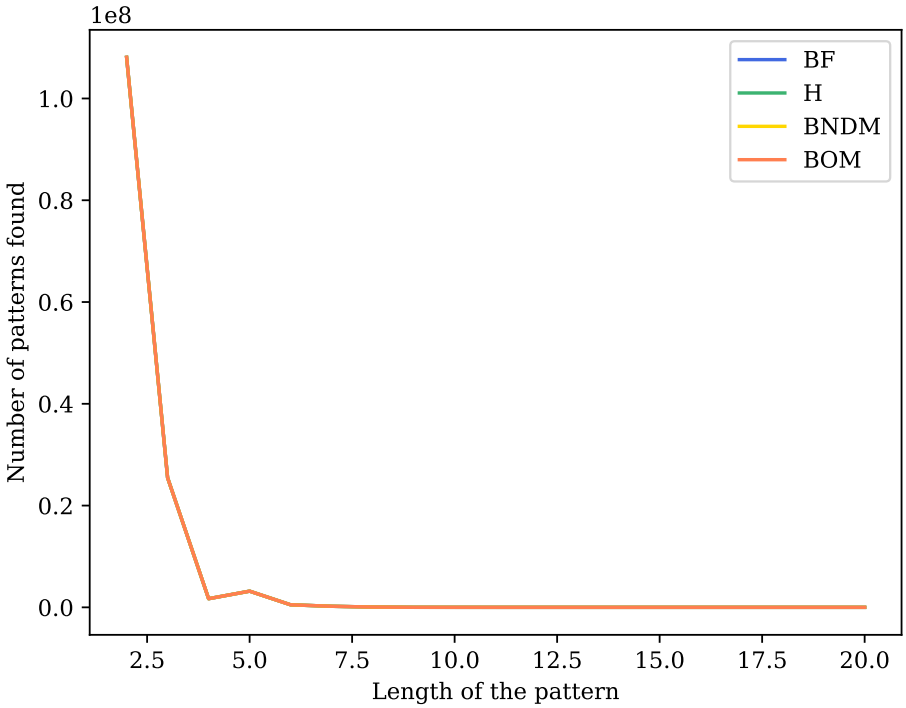

Figure 3: Number of patterns found

Length of the pattern	Number of patterns found
1	429877019
2	108081606
3	25462401
4	1697740
5	3180369
6	499409
7	211306
8	58644
9	9120
10	3056
11	100
12	263
13	35
14	3
15	1
16	0
17	0
18	0
19	0
20	0

Table 1: Number of patterns found depending on the length of the pattern