

# **PMT: Pattern Matching Tool**

Wendell Jânio de Oliveira Júnior - wjoj@cin.ufpe.br

## **1. Implementação**

### **1.1 Algoritmos**

Foram implementados os seguintes algoritmos:

bf: Brute-force

ac: Aho-Corasick (1975)

bm: Boyer-Moore (1977)

uk: Ukkonen (1985)

so: Shift-Or (Baeza-Yates–Gonnet, 1992)

wm: Wu-Manber (1992)

### **1.2 Escolha do algoritmo**

Quando um algoritmo não é especificado, a ferramenta decide o algoritmo que será utilizado da seguinte maneira:

- Caso a distância de edição seja maior que 0: Wu-Manber (1992);
- Caso a quantidade de padrões seja maior que 1: Aho-Corasick (1975);
- Caso o padrão seja menor que 4 bytes: Shift-Or (Baeza-Yates–Gonnet, 1992);
- Caso contrario: Boyer-Moore (1977).

### **1.3 Detalhes da implementação**

A ferramenta foi desenvolvida na linguagem de programação C, no padrão ISO/IEC 9899:1999 (C99), buscando manter compatibilidade com a versão anterior do padrão (C90) para facilitar a portabilidade.

#### **1.3.1 Leitura dos arquivos**

Os arquivos são lidos como binários de forma sequencial, sem separação de linhas. Um buffer de 256 KiB é utilizado para otimizar a leitura do arquivo. Na implementação do algoritmo Boyer-Moore (1977) um buffer de 512 KiB é utilizado caso o padrão seja menor do que 256 KiB, caso contrário o tamanho do buffer é o dobro do tamanho do padrão.

#### **1.3.2 Alfabeto**

Todos os algoritmos foram implementados considerando um alfabeto com os caracteres representados com 8 bits (256 caracteres), assim todos os bytes de um arquivo podem ser representados.

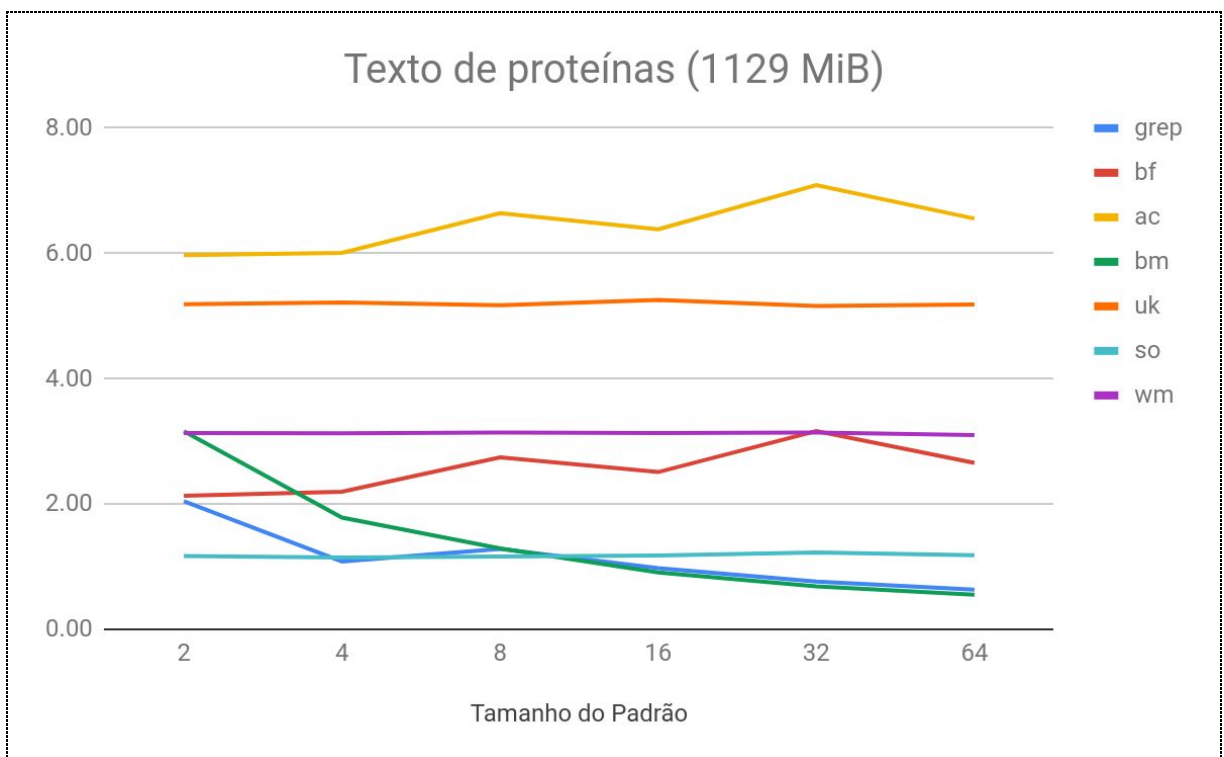
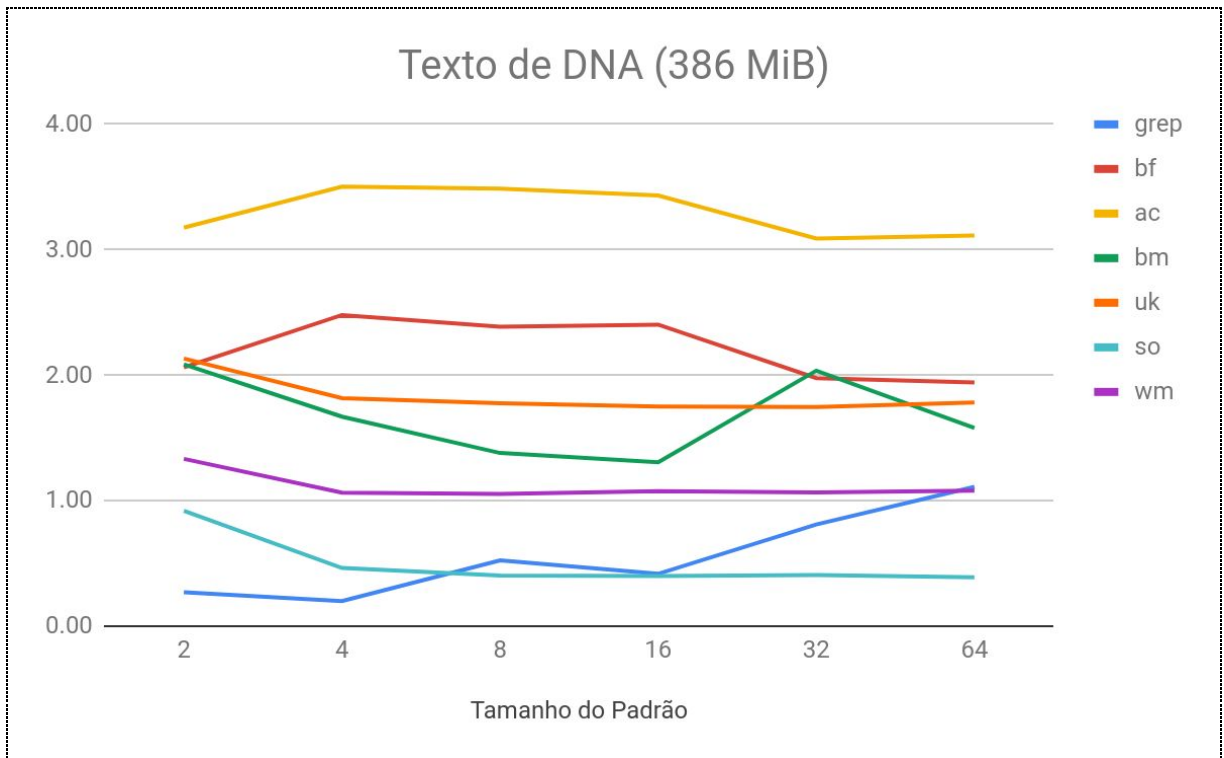
#### **1.3.3 Estruturas de dados**

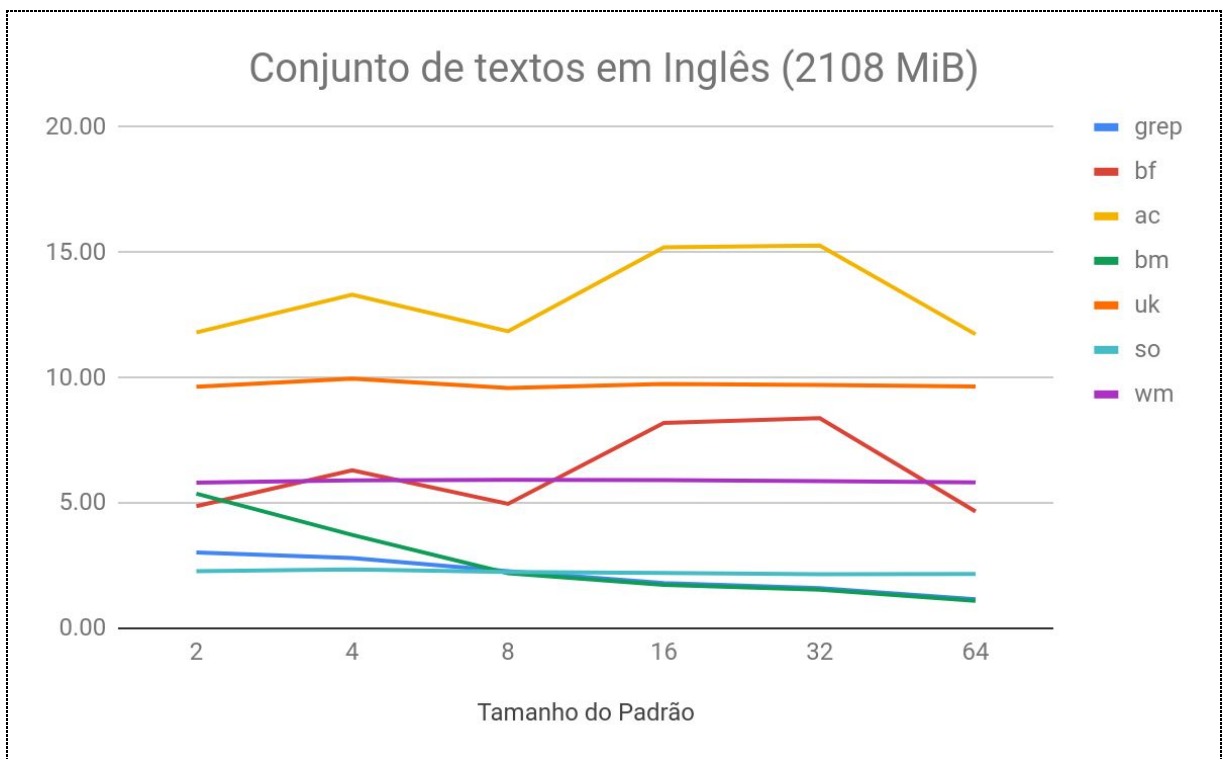
Uma queue (fila) foi utilizada na implementação do algoritmo Aho-Corasick (1975) e do algoritmo Ukkonen (1985), que também se utiliza de uma hash table na criação da máquina de estados. As duas estruturas foram implementadas diretamente na ferramenta.

## 2. Testes

Os teste foram executados em uma máquina com processador Intel(R) Xeon(R) CPU @ 2.30GHz e 7.3 GiB de memória RAM.

### 2.1 Busca exata





## 2.2 Busca aproximada

