universidade
de aveiro

# Copy Model

David Raposo | 93395
Gonçalo Machado | 98359
Catarina Marques | 81382
Bruno Nunes | 80614

**Grupo 1**

*Teoria Algorítmica da Informação, 2023/2024*

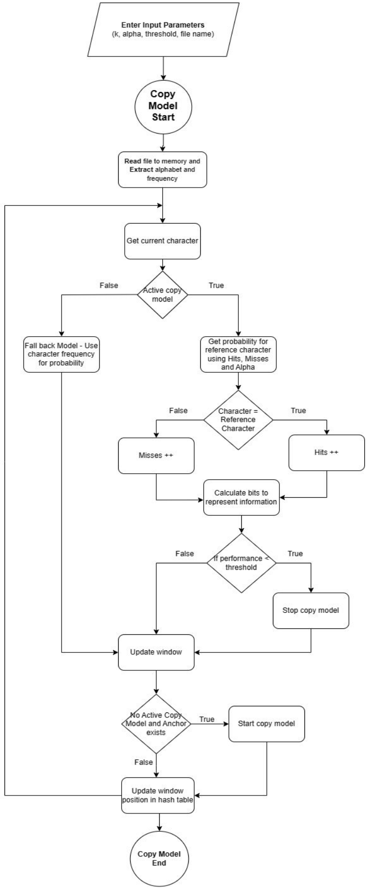
# Copy Model

The copy model is a data compression algorithm that predicts future symbols based on previously seen symbols.

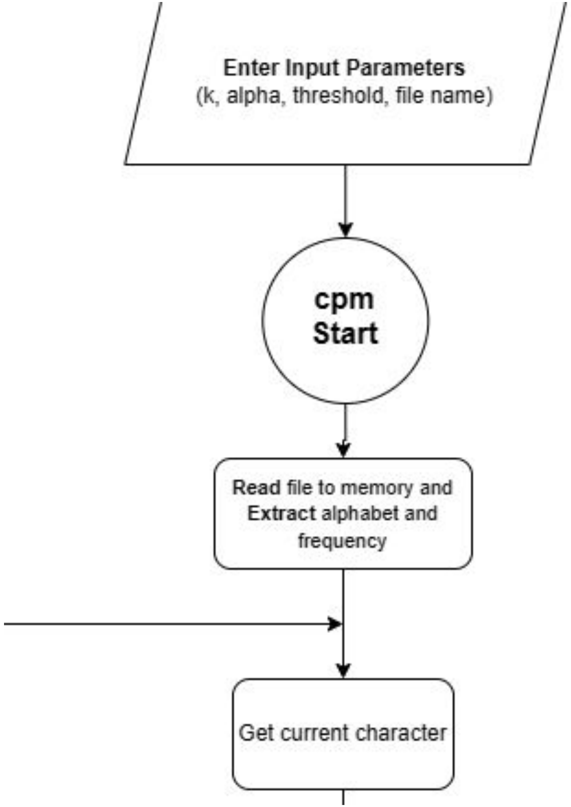
Input Parameters:

- **k** - Size of an anchor
- **alpha** - Smoothing factor
- **threshold** - Used to check if an active copy model should be stopped
- **filename** - Name of file to be “compressed”

# Methodology (Flowchart)


Enter Input Parameters
(k, alpha, threshold, file name)
Copy Model Start
Read file to memory and Extract alphabet and frequency
Get current character
Active copy model
False
True
Fall back Model - Use character frequency for probability
Get probability for reference character using Hits, Misses and Alpha
Character = Reference Character
False
True
Misses ++
Hits ++
Calculate bits to represent information
If performance < threshold
False
True
Stop copy model
Update window
No Active Copy Model and Anchor exists
True
Start copy model
False
Update window position in hash table
Copy Model End

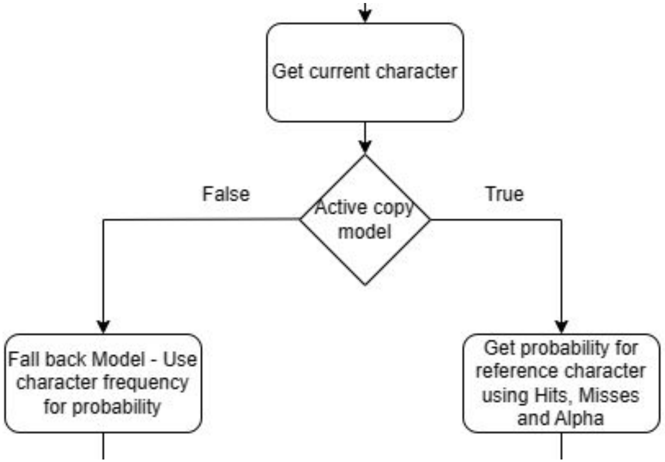
# Methodology


Enter Input Parameters
(k, alpha, threshold, file name)
cpm
Start
Read file to memory and Extract alphabet and frequency
Get current character

Start Implementation:

- Read the file only once
- Frequencies are used by the fallback model

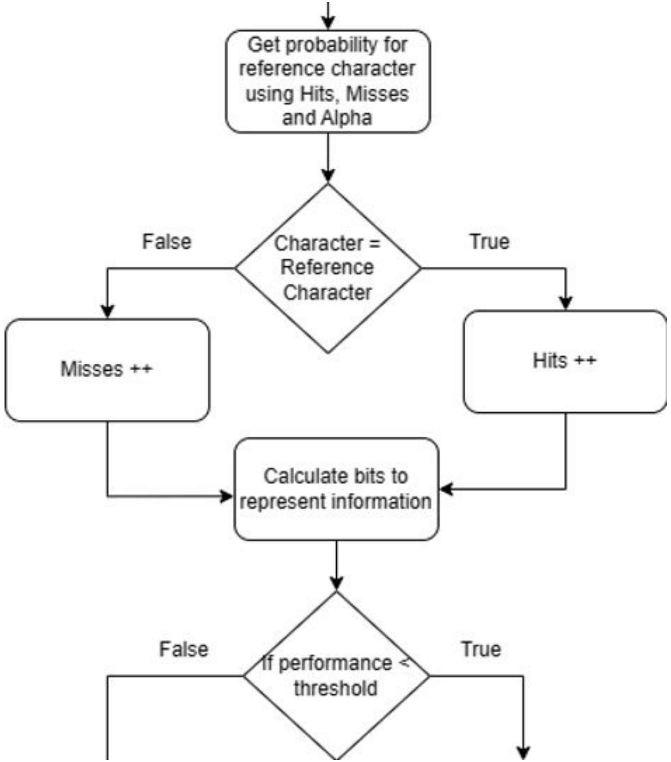
# Methodology


Get current character
Active copy model
False
True
Fall back Model - Use character frequency for probability
Get probability for reference character using Hits, Misses and Alpha

Implementation of Fallback Model:

- Use the relative frequency of characters in the file to calculate the probability of the next character. This reduces the number of bits used compared to using a uniform distribution.

# Methodology

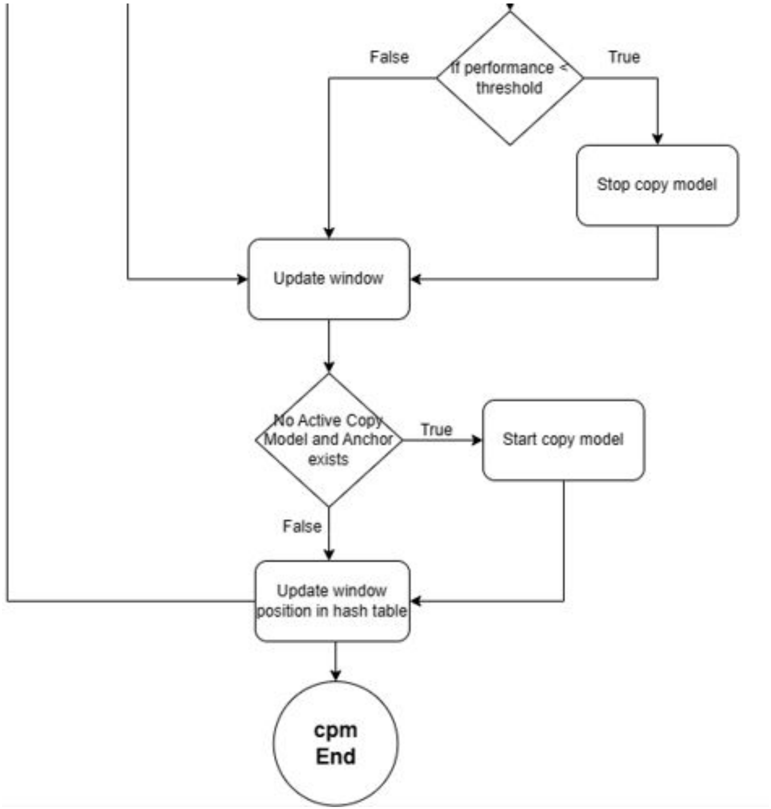

Get probability for reference character using Hits, Misses and Alpha
False
Character = Reference Character
True
Misses ++
Hits ++
Calculate bits to represent information
False
If performance < threshold
True

Implementation for probability:

- Keep tracks of Hits and Misses and use alpha as a smoothing factor.
- Get probability of hit using:

$$p = \frac{\text{Hits} + \text{alpha}}{\text{Hits} + \text{Misses} + 2 * alpha}$$

# Methodology



False
If performance < threshold
True
Stop copy model
Update window
No Active Copy Model and Anchor exists
True
Start copy model
False
Update window position in hash table
cpm End

Implementation for stopping copy model:

- Keeping track of the last k hits and misses, and comparing the model to the threshold allowed handling the cold start problem (the model stopping very early)

# Results - Copy Model

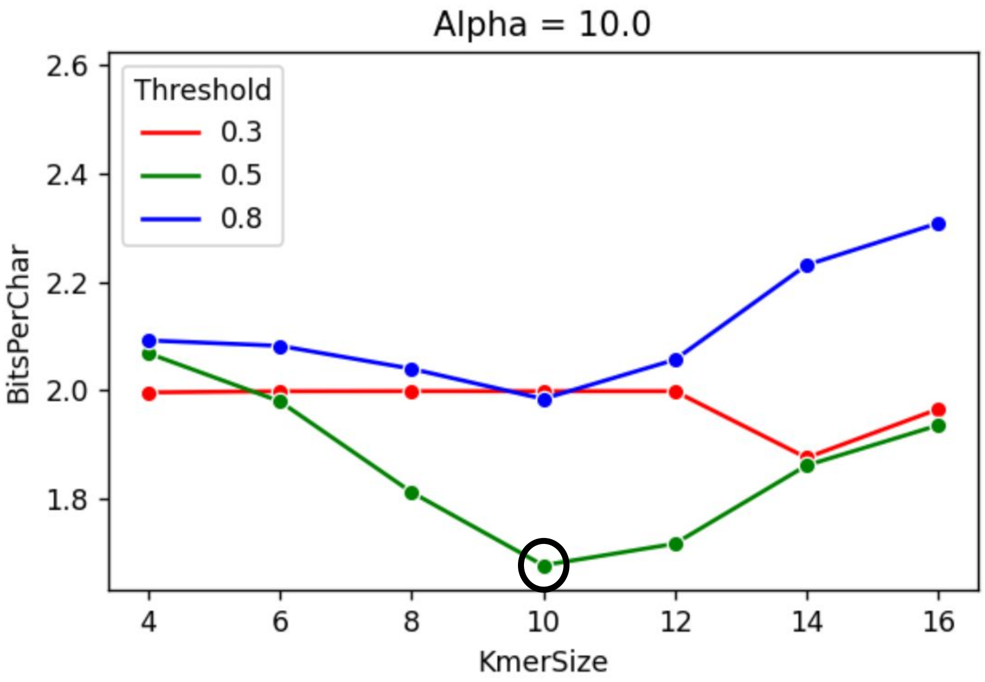
Range of input parameters used:

- k - {4,6,8,10,12,14}
- alpha - {0.1,1,10,100}
- threshold - {0.3,0.5,0.8}

Best parameters found for ‘chry.txt’:

- k - 10
- alpha - 10
- threshold - 0.5

Bits per character : 1,67631



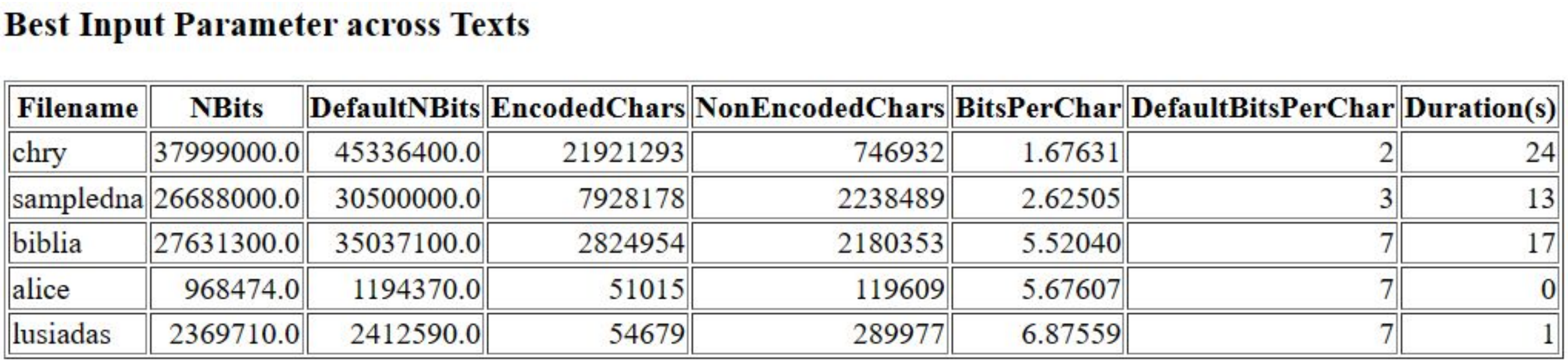
Results for data ‘chry.txt’ with alpha = 10

# Results - Copy Model

Datasets used:

- ‘chry.txt’
- 1 Dna Sample
- 3 books

The best parameters for the chry.txt dataset were used for the other datasets.



### Best Input Parameter across Texts

| Filename | NBits | DefaultNBits | EncodedChars | NonEncodedChars | BitsPerChar | DefaultBitsPerChar | Duration(s) |
|---|---|---|---|---|---|---|---|
| chry | 37999000.0 | 45336400.0 | 21921293 | 746932 | 1.67631 | 2 | 24 |
| sampledna | 26688000.0 | 30500000.0 | 7928178 | 2238489 | 2.62505 | 3 | 13 |
| biblia | 27631300.0 | 35037100.0 | 2824954 | 2180353 | 5.52040 | 7 | 17 |
| alice | 968474.0 | 1194370.0 | 51015 | 119609 | 5.67607 | 7 | 0 |
| lusiadas | 2369710.0 | 2412590.0 | 54679 | 289977 | 6.87559 | 7 | 1 |

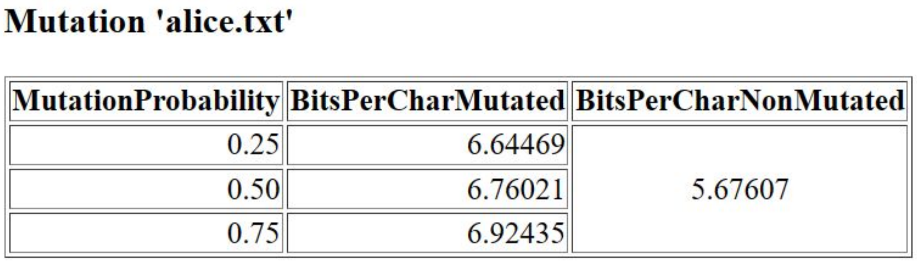
# Results - Mutate

Mutate is a program that changes the contents of a file according to a mutation probability.

Range of probability used : {0.25,0.50,0.75}

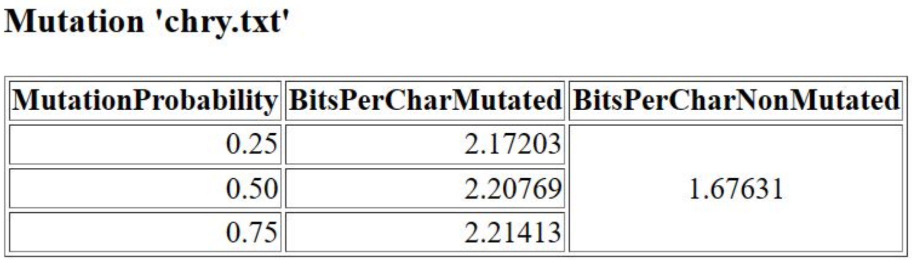
Files mutated:

- chry.txt
- alice.txt



**Mutation 'alice.txt'**

| MutationProbability | BitsPerCharMutated | BitsPerCharNonMutated |
|---|---|---|
| 0.25 | 6.64469 | 5.67607 |
| 0.50 | 6.76021 | |
| 0.75 | 6.92435 | |



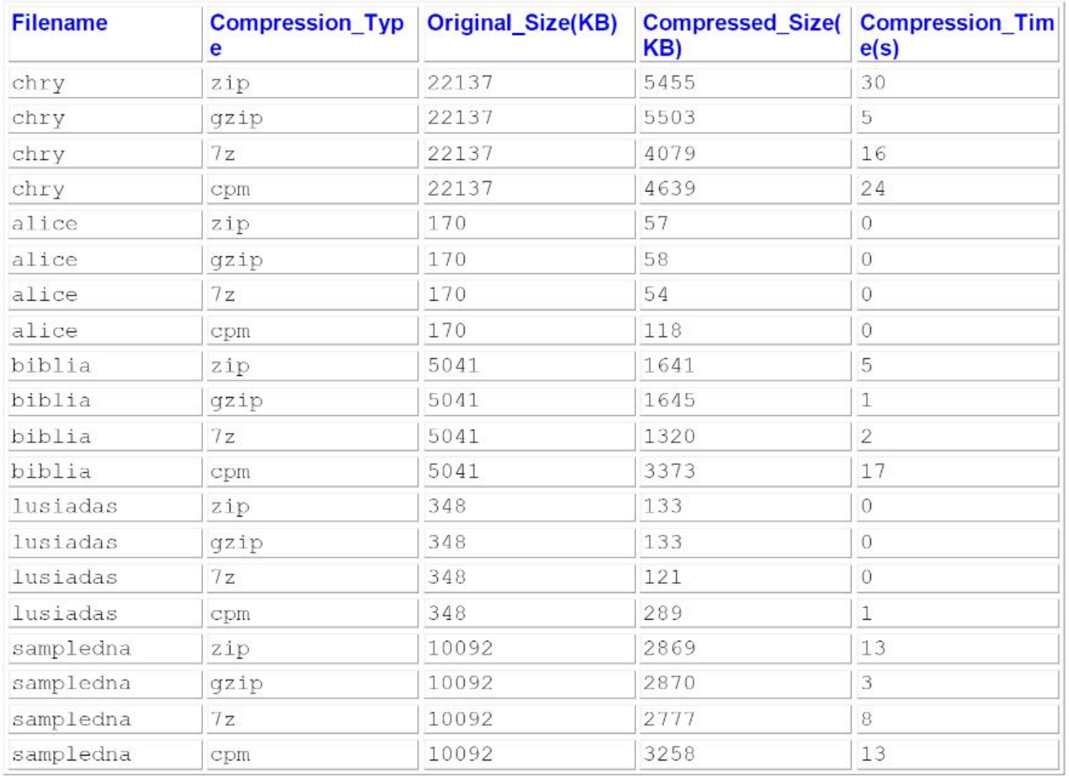
**Mutation 'chry.txt'**

| MutationProbability | BitsPerCharMutated | BitsPerCharNonMutated |
|---|---|---|
| 0.25 | 2.17203 | 1.67631 |
| 0.50 | 2.20769 | |
| 0.75 | 2.21413 | |

# Results - Other compressors

Compressors used:

- 7z
- gzip
- zip



| Filename | Compression_Type | Original_Size(KB) | Compressed_Size(KB) | Compression_Time(s) |
|---|---|---|---|---|
| chry | zip | 22137 | 5455 | 30 |
| chry | gzip | 22137 | 5503 | 5 |
| chry | 7z | 22137 | 4079 | 16 |
| chry | cpm | 22137 | 4639 | 24 |
| alice | zip | 170 | 57 | 0 |
| alice | gzip | 170 | 58 | 0 |
| alice | 7z | 170 | 54 | 0 |
| alice | cpm | 170 | 118 | 0 |
| biblia | zip | 5041 | 1641 | 5 |
| biblia | gzip | 5041 | 1645 | 1 |
| biblia | 7z | 5041 | 1320 | 2 |
| biblia | cpm | 5041 | 3373 | 17 |
| lusiadas | zip | 348 | 133 | 0 |
| lusiadas | gzip | 348 | 133 | 0 |
| lusiadas | 7z | 348 | 121 | 0 |
| lusiadas | cpm | 348 | 289 | 1 |
| sampledna | zip | 10092 | 2869 | 13 |
| sampledna | gzip | 10092 | 2870 | 3 |
| sampledna | 7z | 10092 | 2777 | 8 |
| sampledna | cpm | 10092 | 3258 | 13 |

# Conclusion

- The copy model was able to successfully compress files, compressing the chry.txt file to 20% of its original size.
- According to the results presented, the copy model had a performance as good or better than the other compression algorithms in DNA texts and a worse performance than the other compression algorithms in normal text files.
- Mutated files created a negative impact in the copy model compression, which increases the more mutations the file has.

# Future Work

- Multi-anchor implementation
- Test on files with different size and content
- Leverage concurrency to increase efficiency
- Memory optimizations