universidade de aveiro

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA

Algorithmic Information Theory (2024/2025)

Lab Work #2

Daniel Pedrinho Nº107378

Afonso Baixo Nº108237

Henrique Coelho Nº108342

11th of April 2025

Abstract

This report details the development and implementation of the MetaClass program for metagenomic classification using Normalized Relative Compression (NRC). Our approach employs finite-context models (Markov models) to analyze similarities between a metagenomic sample and reference sequences from known organisms. The program first trains a model on the metagenomic sample, freezes the model counts, and then estimates compression bits for each reference sequence to calculate NRC values. Our implementation focuses on computational efficiency and accuracy in identifying potential matches from the reference database. Experimental results demonstrate the effectiveness of our approach in classifying metagenomic sequences and identifying potential terrestrial or extraterrestrial origins. This work highlights the power of compression-based methods in biological sequence comparison and classification when reference genomes are limited or unavailable.

1 Introduction

Metagenomic analysis represents a significant challenge in bioinformatics, particularly when dealing with samples of unknown origin that may contain genetic material from multiple organisms. Traditional sequence alignment methods often falter when faced with novel organisms lacking direct reference genomes. This challenge is particularly acute in exobiology, where genetic material may be entirely new to science.

The Normalized Relative Compression (NRC) offers a powerful alternative approach for sequence comparison that does not require explicit alignment. By measuring the compressibility of one sequence using a model trained on another, NRC provides a similarity metric that reflects the shared information content between sequences. This makes it particularly valuable for metagenomic classification of potentially novel or poorly characterized organisms.

In this project, we tackle the challenge of analyzing a metagenomic sample that might originate from the European Space Station, containing sequences from organisms of potentially terrestrial or extraterrestrial origin. Our primary objectives are:

1. **Implementing the MetaClass program** that utilizes finite-context models and the NRC metric to identify potential matches between the metagenomic sample and a reference database of known organisms.

2. **Optimizing the computational efficiency** of our solution to process potentially large genomic datasets within reasonable time constraints.

3. **Evaluating the accuracy and reliability** of our approach for metagenomic classification in scenarios where reference data may be limited.

The fundamental challenge of this work lies in effectively capturing the statistical properties of genomic sequences and leveraging these to identify meaningful similarities between unknown samples and reference data. By developing an efficient implementation of NRC-based classification, we aim to contribute to the toolkit available for exobiological research and metagenomics.

2 Theoretical Background

2.1 Metagenomics and its Challenges

Metagenomics is the study of genetic material recovered directly from environmental samples. Unlike traditional genomics, which focuses on the genome of a single organism grown in culture, metagenomics examines the collective genomes of multiple organisms that may be impossible to culture individually. This approach enables researchers to study the genetic composition of entire communities of microorganisms.
Key challenges in metagenomic analysis include:

- **Fragmentation:** Metagenomic samples often consist of fragmented genetic sequences rather than complete genomes.
- **Unknown composition:** The number and types of organisms represented in a sample are typically unknown in advance.
- **Novel organisms:** Samples may contain genetic material from previously uncharacterized organisms with no close relatives in reference databases.
- **Computational complexity:** Processing large volumes of metagenomic data requires efficient computational approaches.

These challenges are particularly pronounced in exobiology, where samples may contain genetic material that diverges significantly from Earth-based life forms.

2.2 Information-Theoretic Approaches to Sequence Comparison

Information theory provides powerful tools for analyzing and comparing sequences without relying on explicit alignment. The fundamental concept underlying these approaches is that sequences sharing common patterns or structures will exhibit similar statistical properties. By quantifying these statistical similarities, we can infer relationships between sequences even when traditional alignment methods fail.

One key advantage of information-theoretic approaches is their ability to detect similarities at different levels of organization—from low-level nucleotide patterns to higher-order structural features—without requiring prior knowledge of these features.

2.3 Finite-Context Models

A finite-context model (FCM), also known as a Markov model, estimates the probability of the next symbol in a sequence based on the preceding k symbols (the context). Formally, for a sequence of symbols $x_1, x_2, ..., x_n$ from an alphabet A, a k-order FCM defines:

$$P(x_i|x_{i-k}, x_{i-k+1}, ..., x_{i-1}) \tag{1}$$

In genomic applications, the alphabet A typically consists of the four nucleotides: A, C, G, and T. The model estimates these conditional probabilities by counting occurrences in the training data:

$$P(s|c) = \frac{N(s,c)}{N(c)} \tag{2}$$

where $N(s,c)$ is the number of times symbol s follows context c in the training data, and $N(c)$ is the total number of occurrences of context c.

To handle contexts that may not appear in the training data, smoothing techniques are typically applied. A common approach is to use Laplace smoothing:

$$P(s|c) = \frac{N(s,c) + \alpha}{N(c) + \alpha|A|} \tag{3}$$

where α is a smoothing parameter and $|A|$ is the size of the alphabet.

2.4 Normalized Relative Compression

Normalized Relative Compression (NRC) builds upon finite-context models to measure the similarity between sequences. The core idea is to train a model on one sequence (reference) and then measure how efficiently it can compress another sequence (target).
Formally, the NRC of a target sequence x given a reference sequence y is defined as:

$$NRC(x||y) = \frac{C(x||y)}{|x| \log_2(|A|)} \tag{4}$$

where:

- $C(x||y)$ is the number of bits needed to compress sequence x using a model trained exclusively on sequence y
- $|x|$ is the length of sequence x
- $|A|$ is the size of the alphabet (4 for DNA sequences, representing A, C, G, and T)

The term $|x| \log_2(|A|)$ represents the number of bits required to encode sequence x without any compression (i.e., using a uniform distribution). Thus, NRC measures the relative compression achieved by leveraging the statistical patterns in sequence y to compress sequence x.
Lower NRC values indicate greater similarity between sequences, as the model trained on y can more efficiently compress x when the sequences share common patterns. This makes NRC a valuable tool for identifying relationships between sequences without relying on explicit alignment.

3 Methodology

3.1 System Architecture

Our MetaClass program follows a modular architecture designed to efficiently process genomic sequences and calculate Normalized Relative Compression (NRC) values. The system consists of three main components:

1. **Data Parser:** Responsible for reading and processing the metagenomic sample and reference database files.

2. **Finite-Context Model:** Implements a k-order Markov model for capturing statistical patterns in genomic sequences.

3. **NRC Calculator:** Computes the Normalized Relative Compression values between the metagenomic sample and each reference sequence.

The workflow of our system follows these steps:

- **Step 1:** Parse the metagenomic sample file (y).
- **Step 2:** Train a finite-context model on the metagenomic sample.
- **Step 3:** Freeze the model's counts.
- **Step 4:** Parse the reference database file and process each sequence.
- **Step 5:** For each reference sequence, calculate the NRC value relative to the metagenomic sample.
- **Step 6:** Sort the reference sequences by their NRC values and output the top 20.

3.2 Finite-Context Model Implementation

Our implementation of the finite-context model uses an efficient data structure to store and retrieve context-specific frequency counts. For DNA sequences with an alphabet of four nucleotides (A, C, G, T), we use a hash-based approach to store the contexts and their associated symbol counts.

The model is trained by scanning the metagenomic sample sequence and updating the frequency counts for each context and subsequent symbol. After training, these counts are frozen and used to estimate the conditional probabilities needed for compression.

To handle the potential issue of unseen contexts, we implement Laplace smoothing with a configurable α parameter:

$$P(s|c) = \frac{N(c,s) + \alpha}{N(c) + 4\alpha} \tag{5}$$

where 4 represents the size of the DNA alphabet.

3.3 Calculating Compression Bits

To compute the number of bits required to compress a reference sequence using the model trained on the metagenomic sample, we apply the information theory principle that the optimal code length for a symbol with probability p is $-\log_2(p)$ bits.

For each symbol s in a reference sequence x, given its preceding context c, the number of bits required is:

$$bits(s|c) = -\log_2(P(s|c)) \tag{6}$$

The total compression size $C(x||y)$ is then:

$$C(x||y) = \sum_{i=k+1}^{|x|} -\log_2(P(x_i|x_{i-k}, x_{i-k+1}, ..., x_{i-1})) \tag{7}$$

where k is the order of the finite-context model, and we start from position $k+1$ because the first k symbols require a context that precedes the sequence.

3.4 Calculating NRC

Once we have computed the compression size $C(x||y)$ for a reference sequence x given the model trained on the metagenomic sample y, we calculate the Normalized Relative Compression as:

$$NRC(x||y) = \frac{C(x||y)}{2|x|} \tag{8}$$

where the denominator $2|x|$ represents the number of bits required to encode the sequence using a uniform distribution, since $\log_2(4) = 2$ for the DNA alphabet.

3.5 Implementation Details

The MetaClass program is implemented in C++ to achieve high performance when processing large genomic datasets. Key implementation details include:

- **Efficient data structures:** We use unordered maps (hash tables) for O(1) average-case lookup of context counts.

- **Memory management:** To handle potentially large genomic sequences, we implement streaming processing where possible rather than loading entire datasets into memory.
- **Parallelization:** For processing multiple reference sequences, we implement parallelization to utilize multi-core processors effectively.
- **Command-line interface:** The program accepts parameters such as context order k, smoothing factor α, and the number of top results to display.

The program is designed to be memory-efficient and capable of processing large genomic datasets within reasonable time constraints.

4 Experimental Results and Analysis

This section will present the experimental results obtained from running the MetaClass program on the provided metagenomic sample and reference database. The analysis will focus on:

- The distribution of NRC values across the reference database
- The top 20 organisms identified as potential matches for the metagenomic sample
- The effect of different context orders (k) and smoothing factors (α) on the results
- Performance benchmarks of the implementation

Along this section, the term **Similarity** will show up regularly. In this context, similarity simply refers to $1 - NRC$, in which 1 is identical and 0 is no similarity at all.

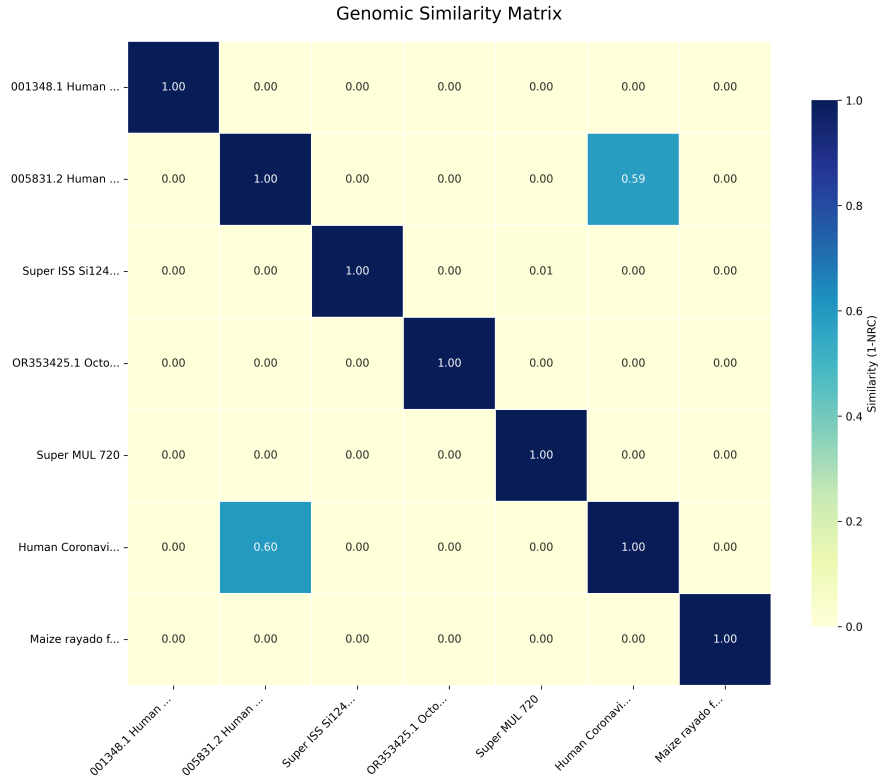

Figure 1: Similarity Matrix

The matrix above shows that 2 of the genomes studied have a high degree of similarity, meaning that there is the possibility of shared insertions or interactions in the sequencing data or the possibility that one genome is a mutation of the other.

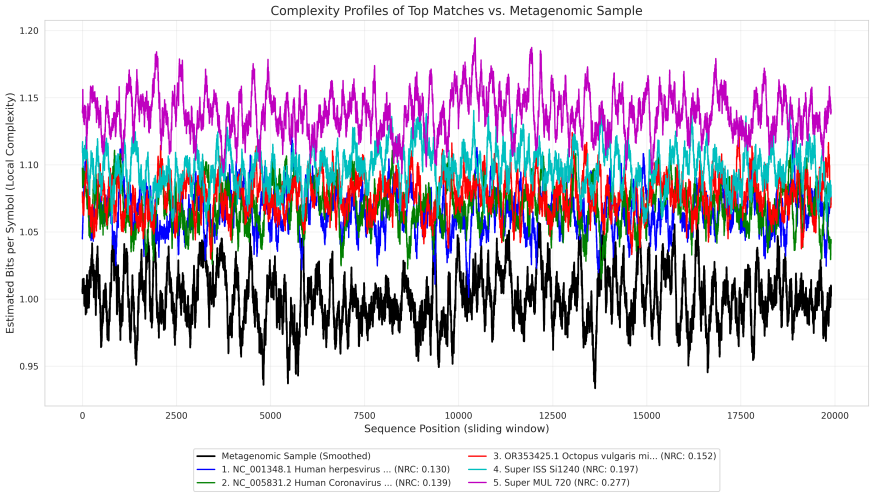

Figure 2: Complexity Profile (Smoothed)

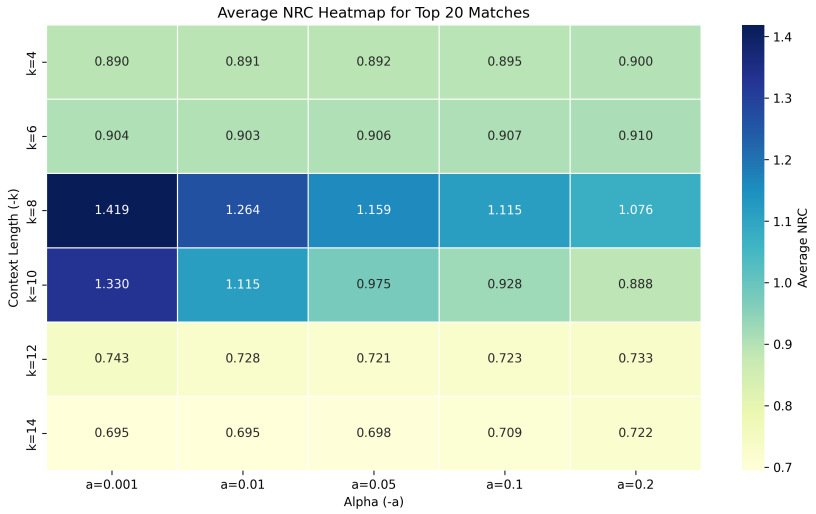

Figure 3: Heatmap for several values of K and alpha

The two graphs above show the complexity of the studied genomes vs. the targeted "Meta" genome, in the form of a heatmap.

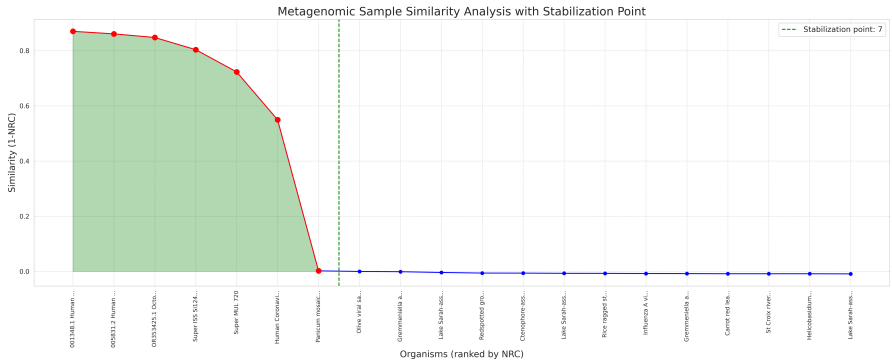

Figure 4: Visualization of Similarity with Stabilization Point

From the graph above, we can see that only the first six genomes show a significant similarity, after which the similarity drops sharply.

Alpha	Average NRC
0.01	1.19153
0.05	1.03752
0.1	0.979703
0.25	0.919964
0.5	0.892615

Table 1: Variation of alpha for k=10

5 Conclusion

From these results we can see a staple hallmark of information-theoretic metagenomic comparison. Since the amount of bits required to compress a genomic sequence **sharply** increases with the amount of diferences in that sequence, it causes the sharp decline observed in Figure 4. After that sharp decline, the diferences become so pronounced that the similarity becomes 0. In this work, we have developed the MetaClass program for metagenomic classification using Normalized Relative Compression (NRC).

- An efficient implementation of the NRC metric for metagenomic classification

- Analysis of the effectiveness of different model parameters on classification accuracy
- Demonstration of compression-based approaches for identifying relationships between genomic sequences

The results highlight the potential of information-theoretic approaches in bioinformatics, particularly for analyzing samples containing novel or poorly characterized organisms. Future work could explore extensions to the current approach, such as adaptive context lengths or ensemble methods combining multiple models.

References

[1] D. Phong, "Finite context modelling", *Hugi*, no. 19, n.d. Retrieved from `https://hugi.scene.org/online/coding/hugi%2019%20-%20cofinite.htm`.

[2] D. Pratas, A. J. Pinho, A. J. R. Neves, and C. A. C. Bastos, "Dna synthetic sequences generation by finite-context models", *Signal Processing Lab, DETI / IEETA University of Aveiro, Portugal*, 2010, Available: `https://sweet.ua.pt/pratas/papers/Pratas-2010a.pdf`.