Information Models for Prediction

Alexandre Ribeiro
Student 108122, DETI
Aveiro University, Portugal
alexandrepribeiro@ua.pt

Maria Sardinha
Student 108756, DETI
Aveiro University, Portugal
mariasardinha@ua.pt

Miguel Pinto
Student 107449, DETI
Aveiro University, Portugal
miguel.silva48@ua.pt

***Abstract*—This report presents the implementation and evaluation of the MetaClass program, developed for the Algorithmic Theory of Information course [1]. The primary objective is to identify genomic similarities between a given metagenomic sample and a reference database of known organisms, by using the Normalized Relative Compression (NRC) metric. The methodology involves training a Finite-Context Model (FCM) exclusively on the metagenomic sample and then estimating the relative compressibility of each reference sequence using the frozen model. This allows for the inference of biological similarity between sequences, potentially uncovering evidence of contamination or novel life forms. The program was implemented in C++, with auxiliary Python scripts used for result analysis and visualization. The results were evaluated based on both accuracy in ranking the closest organisms and the computational performance across varying parameters, such as context order k and smoothing factor α.**

***Index Terms*—Finite-Context Model, Markov Model, Normalized Relative Compression, Metagenomics, Information Theory, DNA Sequence Analysis, Probabilistic Modeling, Compression-based Similarity.**

I. Introduction

In the area of Exobiology, the identification and classification of organisms based on their genetic sequences is particularly demanding when dealing with metagenomes — collections of genetic material recovered directly from environmental samples potentially containing multiple unknown organisms. This challenge becomes even more significant when considering samples that might originate from extraterrestrial environments, such as those collected from the European Space Station.

Building upon our previous work implementing Finite-Context Models (FCMs) for text analysis and prediction [2], this project extends the application of information-theoretic approaches to the domain of genomic sequence analysis. In our prior work we developed a C++ implementation of FCMs that demonstrated effective probabilistic modeling of textual data by utilizing context-based symbol distribution estimation. These models successfully captured statistical properties and patterns within text, allowing both analysis and generation of content with recognizable stylistic characteristics.

The current project transitions from natural language to the language of genetics — DNA sequences — while using similar principles of information theory as our previous work. The primary objective is to develop a metagenome classification system (MetaClass) that utilizes Normalized Relative Compression (NRC) as a similarity metric to identify which known organisms from a reference database share the highest similarity with genomes found in a metagenome sample of potentially extraterrestrial origin.

This report documents our methodological approach, implementation details, experimental results, and analysis of findings. We explore how parameters such as context size (k) and smoothing factor ($alpha$) affect the performance of our model, and evaluate the overall effectiveness of using compression-based metrics for metagenome classification in the context of exobiology research.

II. Background and Related Work

A. Theoretical Foundation

Our approach is grounded in fundamental principles from information theory (Shannon [5]) and algorithmic complexity (Kolmogorov [6]). Claude Shannon's information theory provides the mathematical framework for quantifying information content in sequences (which are biological in this study), while Kolmogorov complexity theory establishes the theoretical relationship between compression and similarity. Specifically, the Normalized Information Distance (NID [7]) offers a theoretically ideal measure of similarity between any two objects, which can be approximated through compression algorithms.

B. DNA Sequence Characteristics

Some of DNA sequences' properties make them particularly suitable for compression-based analysis, mainly their limited alphabet - consisting of only four nucleotides (A, C, G, T) - which creates a constrained symbolic space compared to natural language. This reduced alphabet simplifies probability estimation in statistical models while still preserving the complex patterns that define genetic information.

Additionally, genomic sequences contain various repetitive elements at different scales, from short tandem repeats [8] to larger structural patterns, namely:

- Coding regions with specific codon usage patterns
- Regulatory elements with conserved motifs
- Structural repeats and palindromic sequences
- Species-specific genomic signatures

These patterns create statistical regularities that compression algorithms can effectively capture. Besides, related organisms typically share similar genomic patterns due to evolutionary conservation, making compression-based similarity metrics particularly effective for taxonomic classification.

C. From Text FCMs to DNA Analysis

In our previous work [2], we implemented Finite-Context Models for natural language analysis and generation. The transition from text to DNA analysis requires some adaptations:

- **Alphabet Restriction:** While our previous implementation handled arbitrary Unicode characters, the current implementation is optimized for the four-letter DNA alphabet (A, C, G, T), ignoring case sensitivity.
- **Testing Conditions Optimization:** Text analysis typically benefits from context lengths between 3-5 characters, as established in our prior experiments. For DNA sequences, optimal context lengths are expected to be longer (around 8-12 nucleotides) to capture biologically meaningful patterns such as codons and binding motifs. The smoothing parameter also required recalibration to balance between over-smoothing and under-smoothing.

D. Normalized Relative Compression

Normalized Relative Compression (NRC) is a measure of similarity between sequences based on the concept that similar sequences compress better together than dissimilar ones.

The NRC between two sequences x and y is defined as:

$$\text{NRC}(x \parallel y) = \frac{C(x \parallel y)}{|x| \log_2(A)}$$

Where:

- $C(x \parallel y)$ is the number of bits needed to losslessly compress x given exclusively a model trained with y.
- $|x|$ is the size of the sequence x.
- $\log_2(A)$ is the log of the alphabet size of sequence x.

This metric is particularly appropriate for DNA sequence comparison for several reasons:

- **Alignment-Independence:** Unlike traditional methods like BLAST that require sequence alignment, NRC can detect similarity even when sequences have undergone significant rearrangements, insertions, or deletions.
- **Computational Efficiency:** For metagenome analysis involving multiple organisms, pairwise alignment approaches scale poorly, while compression-based methods can efficiently process large datasets.
- **Sensitivity to Higher-Order Patterns:** NRC can detect subtle patterns shared between sequences that may not be evident through direct base-by-base comparison, making it suitable for detecting distant evolutionary relationships.

Lower NRC values indicate greater similarity between sequences. This measure approximates the Normalized Information Distance, which is based on Kolmogorov complexity and provides a universal distance metric for comparing any two objects [9].

E. Compression-Based Classification in Metagenomics

Metagenome classification presents unique challenges compared to single-genome analysis due to the mixture of genetic material from multiple organisms, often with varying abundance levels. Compression-based classification offers several advantages in this context:

- **Reference Model Building:** Instead of building models for each reference genome, our approach inverts the process by training a model on the metagenome sample and evaluating how well it compresses the reference sequences. This approach is more efficient when analyzing a single metagenomic sample against a large reference database.
- **Classification Confidence:** The ranking of NRC scores provides a natural measure of confidence in taxonomic assignments. Organisms that are closely related tend to cluster with similar NRC values, whereas distinct taxa show greater compression distances.
- **Handling Mixed Samples:** When dealing with metagenomic samples containing multiple organisms, compression-based methods can identify the dominant organisms without requiring prior separation of sequences.
- **Computational Considerations:** Our implementation focuses on optimizing the classification process through efficient data structures and targeted model training, making it suitable for the analysis of large metagenomic datasets.

In the following sections, we describe the implementation of our classifier and evaluate its performance on metagenomic samples potentially containing novel or extraterrestrial genetic material.

III. Methodology

A. Overview

To achieve our goal of developing an efficient DNA sequence classifier using information-theoretic principles, we created a methodology that builds upon our previous work with Finite-Context Models that will be described in the current Section. We will be going over the program's components, which metrics are collected, visualizations created, as well as how tests are performed and sequences were generated for the synthetic tests.

B. System Architecture

The implementation consists of the following major components:

- **FCMModel:** Core implementation of the Finite-Context Model.
  - Maintains frequency tables tracking context-symbol occurrences.
  - Implements Laplace smoothing for probability estimation with parameter α.
  - Supports model serialization and de-serialization in both binary (BSON) and JSON formats.
  - Provides methods for entropy calculation and sequence prediction (used only in our previous project).
- **MetaClass:** Command-line application for sequence classification.
  - Processes command-line parameters.
  - Manages workflow for training models and calculating metrics.

  - Ranks and reports results based on similarity scores.
- **Testing Framework:**
  - Text based interface that allows parameter testing with both k and α values.
  - Generate JSON or CSV reports of classification performance.
  - Allow model performance evaluation across parameter combinations.
- **Visualization Framework:**
  - Generated in Python and based on obtained results.
  - Allow a better perception of the comprehensive tests done with various parameters.
- **Utility Files:**
  - Handle file loading (DNA sequence preprocessing and reference database loading).
  - Assist other modules by providing reusable code (e.g. to save results, handle user input, and calculate metrics).

C. Metrics Collected

The developed system calculates several performance metrics to evaluate sequence similarity and model performance:

- **NRC (Normalized Relative Compression):** Primary similarity metric, with lower values indicating higher similarity.
- **KLD (Kullback-Leibler Divergence):** Measures the difference between probability distributions.
- **Compression Bits:** Theoretical minimum number of bits needed to encode a sequence.
- **Time:** Reports processing time for each parameter combination in milliseconds.
- **Ranking:** Orders references by similarity to the query sequence.
- **Classification Metrics:** Including accuracy, precision, recall, F1-score, and ROC AUC for synthetic data evaluation.

1) Compression Bits Estimation: The compression algorithm calculates the theoretical minimum number of bits needed to encode a sequence using the trained model. This is based on Shannon's information theory [5], where each symbol's information content is calculated as $-\log_2(\text{probability})$.

For a DNA sequence, the theoretical maximum information content per symbol is 2 bits (since $\log_2(4) = 2$ for the four nucleotides A, C, G, T). However, the actual information content depends on the context-specific probabilities from our FCM model.

Algorithm 1 iterates through the sequence, calculating the information content for each symbol given its preceding context. The total bit count represents the theoretical minimum number of bits required to encode the sequence using the model.

2) NRC Metric Implementation: The Normalized Relative Compression (NRC) metric is a similarity measure between sequences based on compression efficiency. It normalizes the bits required to compress a sequence using a model against the theoretical maximum entropy for DNA sequences.

Algorithm 1 Bits Calculation Algorithm

```
function CalculateBits(sequence)
    totalBits ← 0
    k ← model.getK()                          ▷ Get context size
    if length(sequence) ≤ k then
        return 2 × length(sequence)           ▷ Default 2 bits per nucleotide
    end if
    for i = 0 to length(sequence) − k − 1 do
        context ← sequence[i : i + k]         ▷ Get k-length context
        nextSymbol ← sequence[i + k]          ▷ Symbol to predict
        probability ← model.getProbability(context, nextSymbol)   ▷ Get probability
        totalBits ← totalBits + (−log2(probability))   ▷ Shannon information
    end for
    return totalBits
end function
```

Algorithm 2 NRC Calculation Algorithm

```
function CalculateNRC(sequence)
    if length(sequence) ≤ model.getK() then
        return 1.0                            ▷ Handle short sequences
    end if
    bits ← CalculateBits(sequence)
    theoreticalMaxBits ← 2 × length(sequence)   ▷ 2 bits per nucleotide
    return bits/theoreticalMaxBits            ▷ Normalized compression ratio
end function
```

The NRC value ranges from near 0 (excellent compression, high similarity) to values potentially exceeding 1 (poor compression, low similarity). When comparing a metagenomic sample to reference genomes, lower NRC values indicate that the model trained on the reference genome compresses the sample efficiently, suggesting genetic similarity.

3) KLD Metric Implementation: The Kullback-Leibler Divergence (KLD) measures how one probability distribution diverges from another. In our context, it quantifies the difference between the empirical distribution of k+1-grams in the sequence and the distribution predicted by our model.

The KLD value is always non-negative and equals zero only when the distributions are identical. Higher KLD values indicate greater dissimilarity between the sequence's empirical distribution and the model's predicted distribution, suggesting less genetic relatedness.

4) Other Classification Metrics: When evaluating the system's performance on synthetic datasets with known ground truth, we apply standard classification metrics:

- **Accuracy:** The proportion of correctly classified samples.

Algorithm 3 KLD Calculation Algorithm

function CALCULATEKLD(sequence)
  $k \leftarrow$ model.getK()
  if length$(sequence) \leq k$ **then**
    return 0.0 ▷ Not enough data to calculate KLD
  end if
  Initialize $empiricalCounts$ ▷ Store counts of symbols after each context
  Initialize $contextTotals$ ▷ Store total occurrences of each context
  for $i = 0$ **to** length$(sequence) - k - 1$ **do**
    $context \leftarrow$ sequence$[i : i + k]$
    $nextSymbol \leftarrow$ sequence$[i + k]$
    $empiricalCounts[context][nextSymbol] \leftarrow empiricalCounts[context][nextSymbol] + 1$
    $contextTotals[context] \leftarrow contextTotals[context] + 1$
  end for
  $kld \leftarrow 0.0$
  for each $context$ in $empiricalCounts$ **do**
    for each $symbol$ in $empiricalCounts[context]$ **do**
      $count \leftarrow empiricalCounts[context][symbol]$
      $total \leftarrow contextTotals[context]$
      $empiricalProb \leftarrow count/total$
      $modelProb \leftarrow$ model.getProbability$(context, symbol)$
      $kld \leftarrow kld + empiricalProb \times \log_2(empiricalProb/modelProb)$
    end for
  end for
  return kld
end function

- **Precision:** The ratio of true positives to all positive predictions (true positives + false positives).
- **Recall (Sensitivity):** The ratio of true positives to all actual positives (true positives + false negatives).
- **F1-Score:** The harmonic mean of precision and recall, balancing both metrics.
- **ROC AUC:** Area under the Receiver Operating Characteristic curve, measuring the trade-off between true positive rate and false positive rate at various threshold settings.
- **Confusion Matrix:** A table showing true positives, false positives, true negatives, and false negatives to evaluate classification performance.

By having these additional metrics we can further evaluate the system's classification capabilities, especially when working with datasets where the correct taxonomic assignments are known in advance. They help assess both the accuracy of individual classifications and the overall discriminative power of our compression-based approach.

5) Synthetic Data Generation: To evaluate our classification approach, we developed a synthetic data generation pipeline that creates metagenomic samples with known ground truth. This allows quantitative assessment of classification accuracy.

The synthetic data consists of:

- Reference sequences with known taxonomic identifiers
- Metagenomic samples created by combining fragments from reference sequences with controlled proportions
- Ground truth files mapping each position to its source organism

Synthetic samples were designed to include various challenges common in metagenomic analysis:

- Sequence variations (mutations at 1-5% of positions)
- Sequence insertions and deletions (up to 5%)
- Regions of high similarity between distinct organisms
- Contamination with "unknown" sequences not present in the reference database

More specifically, we created a script that takes advantage of a tool called GTO [10] to generate these sequences, and we allow the user to customize the parameters. We produced 100 samples (80 of which random, 10 used in the meta file and other 10 mutations of the previous ones but not in the meta file).

These controlled samples allowed calculation of standard classification metrics including precision, recall, F1-score, and ROC AUC, providing objective measures of classification performance.

D. Testing Framework

The testing framework in our project provides a comprehensive infrastructure for evaluating DNA sequence classification performance using Normalized Relative Compression.

1) Framework Architecture: The testing codebase was refactored into several utility modules:

- `test_utils`: Core testing functions including parameter generation, chunk analysis, and cross-comparison
- `io_utils`: File operations, data reading/writing, and result storage
- `interface_utils`: User interaction and input validation
- `dna_compressor`: DNA-specific compression metrics calculation

2) Testing Capabilities: Our framework offers several testing approaches:

- **Interactive Testing:** The `tests.cpp` file implements an interactive menu for parameter selection, allowing researchers to easily explore different configurations.
- **Parameter Grid Search:** The testing framework supports systematic grid search over various combinations of context size (k) and smoothing parameter (α), enabling researchers to identify optimal parameter combinations.
- **Configuration-Driven Testing:** Tests can be executed using JSON configuration templates, allowing for reproducible experiments without manual intervention. The `config_template.json` provides a standardized format for test configuration.

- **Advanced Analysis:**
  - *Symbol Information Analysis*: Examines the probability distribution of nucleotides in different contexts
  - *Chunk Analysis*: Divides sequences into overlapping chunks to identify regions of varying similarity
  - *Cross-Comparison*: Compares top-matching organisms against each other to reveal phylogenetic relationships
  - *Synthetic Data Evaluation*: Tests classification accuracy against datasets with known ground truth

3) Result Management: Results are systematically organized and preserved:

- **Timestamped Results:** Each test run generates a uniquely timestamped directory (e.g., `results/20250414_101430/`) for result preservation.
- **Latest Results:** A symbolic `results/latest/` directory always points to the most recent test results for convenient access.
- **Multiple Formats:** Results can be saved in both JSON and CSV formats, with test metadata included.
- **Hierarchical Storage:** Subdirectories organize different analysis outputs:
  - `/symbol_info/`: Stores context-specific probability distributions
  - `/chunk_analysis.json`: Contains position-specific matching results
  - `/cross_comparison.json`: Stores organism-to-organism similarity metrics

The framework also has parallel processing capability to accelerate testing by distributing computational load across multiple threads. The ideal number is detected automatically based on the amount of logical CPU cores but can be adjusted as needed.

E. Visualization Tools

The visualization tools provides comprehensive analysis of NRC (Normalized Relative Compression) results through multiple visualization approaches, enabling both parameter optimization and biological insights. The system processes JSON experiment results to generate standardized visualizations, with outputs organized by timestamp for easy reference.

1) Core Visualization Categories:

a) Parameter Space Analysis: 3D surface plots and heatmaps display NRC values across k and α parameters, allowing for identification of optimal parameter combinations. Execution time analysis relates computational cost to parameter choices, while rank stability visualizations identify consistently top-performing organisms across parameter variations.

b) Organism Comparison Visualizations: Cross-comparison heatmaps of NRC and KLD values between organisms enable assessment of phylogenetic relationships. Information profile plots show filtered entropy patterns for specific organisms, and outlier analysis identifies phylogenetically distinct organisms based on IQR thresholds.

c) Sequence-based Analysis: Chunk analysis visualizations map best-matching organisms across sequence positions, with color-coded NRC scores identifying sequence regions with strong taxonomic signals. This enables detection of chimeric sequences or regions of horizontal gene transfer.

2) Implementation Details: The Python-based framework incorporates statistical processing techniques such as filtering with Blackman windows and outlier detection. Results are automatically generated in both timestamped and "latest" directories, facilitating experiment tracking and comparison. Summary statistics identify optimal parameters and best-matching organisms, streamlining interpretation of complex NRC data.

F. Experimental Setup

Our experimental evaluations were conducted using a standardized environment to ensure consistency and reproducibility of results. While we explored various configurations during development, all final results presented in this paper were generated using the setup presented in this subsection.

1) Hardware and Software Environment:

- **Operating System:** Ubuntu 24.10
- **Processor:** Intel Core i7-12650H (10 cores, 16 threads)
- **Memory:** 16GB DDR5 4800MHz RAM
- **Compiler:** GCC 13.3.0 with C++17 support
- **Build System:** CMake 3.30.3

2) Parallel Processing Configuration: The testing framework uses multi-threading to accelerate computation (default number automatically detected)

- **Standard Tests:** 16 parallel threads for reference processing
- **Chunk Analysis:** single thread (due to higher memory requirements and issues in development)

3) Test Configuration: All experiments were conducted using a standardized configuration file (`final_config.json`) to ensure reproducibility. Some of the main parameters are described below:

- **Context Size Range:** $k \in [1, 20]$
- **Alpha Value Range:** $\alpha \in [0.01, 1]$ with 10 spaced values within the interval
- **Chunk Size:** 25000 bp with 1000 bp overlap
- **Top Matches:** 20 preserved for further analysis

The complete configuration is available in the project repository for exact replication of our results.

4) Result Processing: Test results were then processed by our visualization tools, generating the graphics presented in this paper. Each visualization preserves the specific test parameters used, allowing for direct correlation between experimental conditions and observed results.

All performance metrics and visualizations presented in subsequent sections reflect this standardized experimental setup.

IV. Experimental Results

This section presents and analyzes the outcomes of the MetaClass system through a series of controlled experiments.

We evaluated the model on its ability to classify DNA sequences based on the Normalized Relative Compression (NRC) metric using various parameter configurations. The analysis includes rankings, performance metrics, execution time, and visual breakdowns by sequence region and organism similarity.

A. Organism Ranking and NRC Scores

The system outputs a ranked list of reference organisms for each input sequence based on similarity, measured by the NRC. Table (I) summarizes the top-10 results for a metagenomic sample, where *Octopus vulgaris* showed the highest similarity. This visualization was done with the best parameters founded (k=17 and α=0.01).

TABLE I
TOP 10 NRC-RANKED ORGANISMS (K = 17, α = 0.01)

Rank	Organism Name	NRC	KLD
1	OR353425.1 Octopus vulgaris mitochondrion, complete genome...	0.023185	719.722054
2	NC 001348.1 Human herpesvirus 3, complete genome...	0.028285	6976.290343
3	NC 005831.2 Human Coronavirus NL63, complete genome...	0.028549	1570.720935
4	Super ISS Si1240...	0.106828	264.934522
5	Super MUL 720...	0.197829	284.873124
6	NC 005831.2— Human Coronavirus NL63, complete genome...	0.426500	23500.206077
7	NC 025220.1— Sweet potato leaf curl virus associated satelli...	0.976808	1432.000000
8	NC 003847.1— Panicum mosaic satellite virus, complete genome...	0.979419	1618.000000
9	NC 024075.1— Cat Que Virus strain VN04-2108 nucle-oprotein ge...	0.982706	1932.000000
10	NC 029618.1 Lake Sarah-associated circular molecule 7 ...	0.984489	2158.000000

Figure 1 complements this ranking with a heatmap visualization of the top organism NRC scores, enabling a rapid visual comparison of similarity across the candidates.

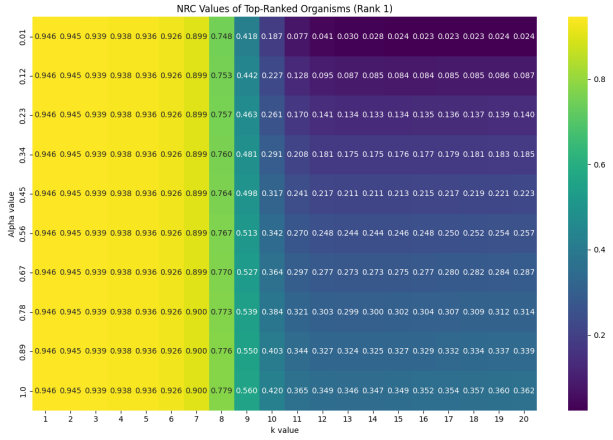

Fig. 1. Heatmap of the Top-20 Organisms (NRC).

B. Parameter Impact Analysis

We performed a grid search across various values of the context length k and the smoothing factor $alpha$. The visualizations in Figures 2 & 3 summarize the results.

According to Figure 2:

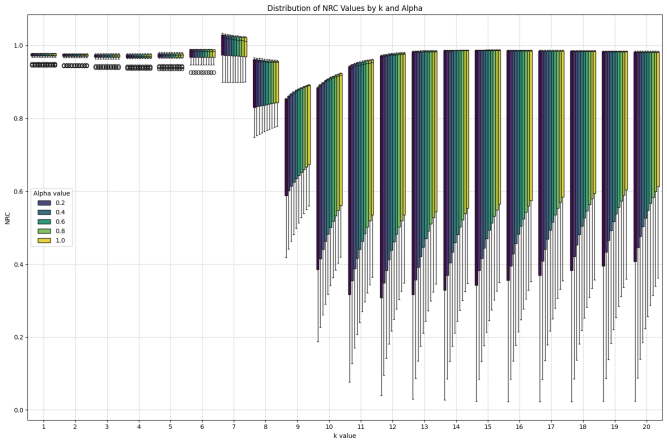

Fig. 2. Boxplot by k and α.

- NRC values are more stable and lower for k around 17 and $\alpha \approx 0.01$.
- Higher values of α tend to over-smooth, reducing the discrimination power.

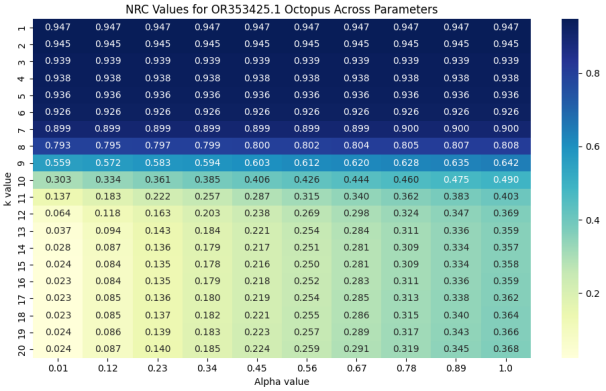

Fig. 3. Parameter Influence on the *Octopus*.

Looking into Figure 3 shows how the Octopus is affected by the different parameter settings.

C. Execution Time Analysis

To assess computational scalability, we evaluated the processing time across multiple configurations.

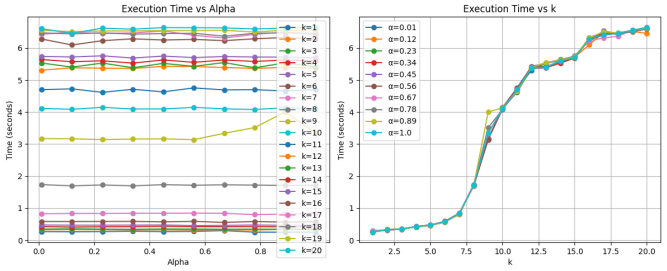

Fig. 4. Execution time for each parameter variation.

The final results shown through Figure 4 reveal that:

- Time grows moderately with context size k.
- The use of multi-threading (number of threads = number of computer cores) keeps the execution feasible even with large reference sets.

D. Rank Stability Across Parameters

We evaluated whether the top-ranking organisms remained consistent across different k and α values.

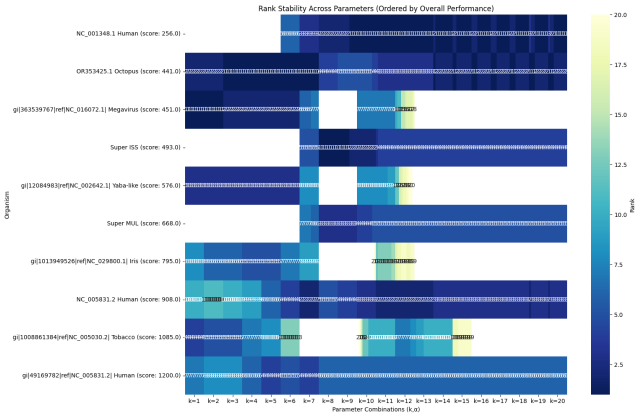

Fig. 5. Rank Stability Across Parameters.

As we can see, through Figure 5 a subset of organisms (3) consistently ranks in the top-10 validating the robustness of the NRC metric.

E. Outlier Detection and Entropy Profiles

To detect outliers or novel sequences, we analyzed the NRC deviation and the symbol entropy profiles, such outcomes are seen in Figures 6 & 7.

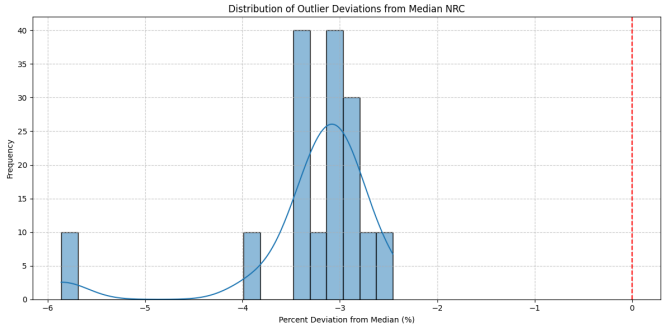

Fig. 6. NRC Outliers - Deviation.

The plot in Figure 8 reveals the entropy variation in the *Octopus vulgaris*.

F. Sequence-Level Chunk Analysis

To detect transferred regions, the metagenomic sequence was divided into overlapping windows for local similarity evaluation and resulted in Figures 9 & 10.

We can see that each chunk shows a low level of similarity to other organisms, with the exception of a few chunks in the

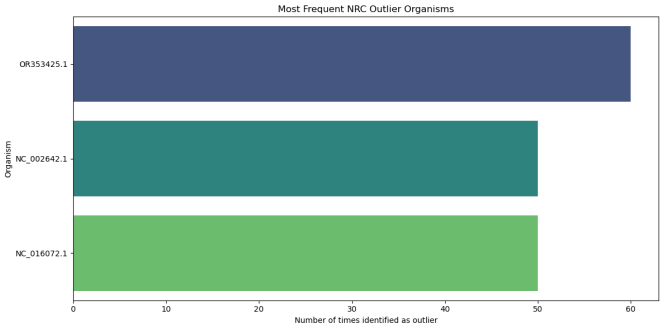

Fig. 7. NRC Outliers - Summary.

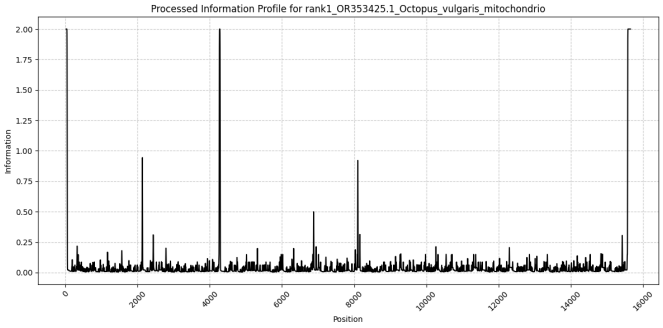

Fig. 8. Information Profile for *Octopus*.

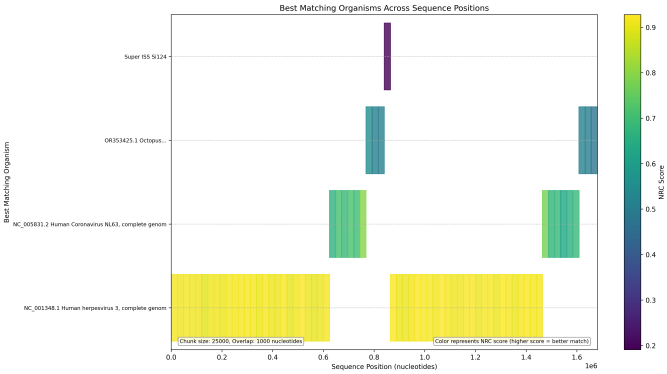

Fig. 9. Chunk-Level Analysis.

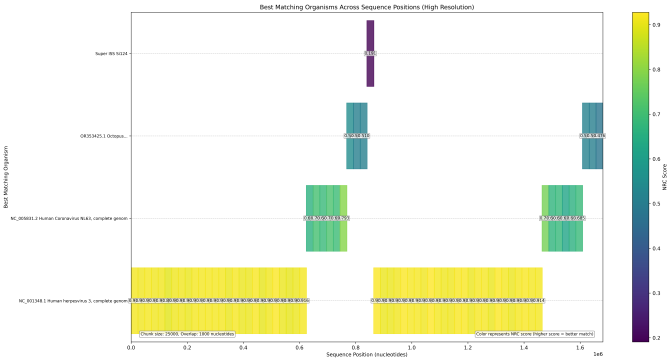

Fig. 10. Chunk-Level Analysis - Hires.

middle and end of the sequence. The presence of different organisms may suggest mixed-origin content.

G. Cross-Organism Similarity Comparison

We compared the top-ranked organisms against one another, using both NRC and KLD to understand the inter-organism relationships. Figure 11 is a resulting NRC heatmap.

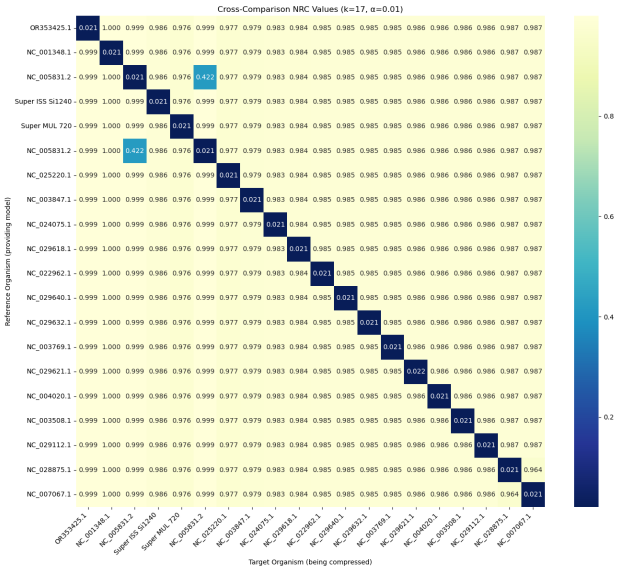

Fig. 11. Cross Comparison - NRC Heatmap.

The NRC revealed similar organisms, while the KLD provided additional context on divergence in information profiles.

H. Human Coronavirus Genomes Comparison

A focused comparative analysis was conducted between two human coronavirus-related genomes, including:

- NC_005831.2 Human Coronavirus NL63, complete genome.
- gi|49169782|ref|NC_005831.2 | Human Coronavirus NL63, complete genome.

The visualizations in Figures 12 & 13 were generated to explore the information complexity within this subset.

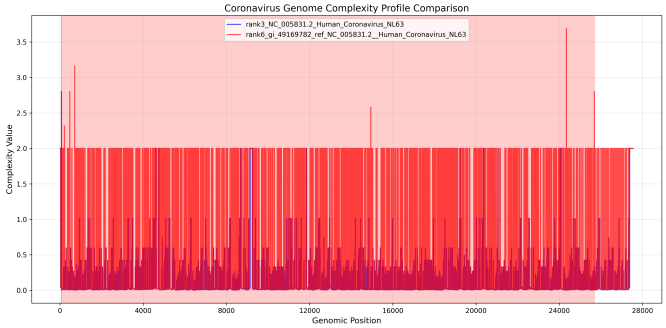

Fig. 12. Human Coronavirus Complexity Profile.

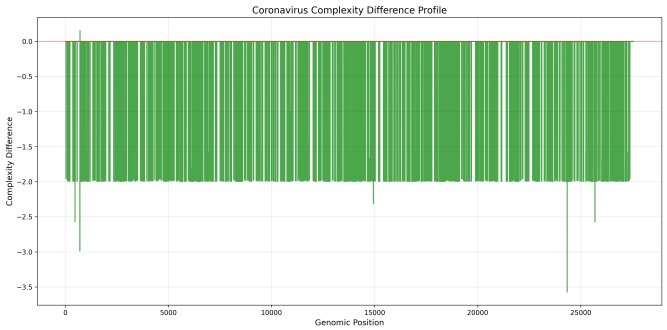

Fig. 13. Human Coronavirus Complexity Difference.

The first plot, Figure 12 shows the entropy variations along the genome, identifying conserved and variable regions. While the second one in Figure 13 highlights the relative compression differences across closely and distant related viruses.

I. Synthetic Data Analysis

To validate the performance of the MetaClass system under controlled conditions, we generated a synthetic dataset with known ground truth. This dataset simulates a metagenomic sample composed of known sequences and their mutated variants, allowing us to evaluate both the classification accuracy and robustness of the NRC metric. Figures 14 & 15 show the output of tests in the synthetic data samples.

Experimental Setup (for the synthetic data tests):

- Sample Length: 27312 nucleotides.
- Reference Database: 100 synthetic samples.
- Ground Truth: 20 known positives.
- Parameters: k = 8, α = 0.01.
- Threshold: NRC = 0.5.

Based on the output shown in Figure 14, we can conclude that:

- The classifier correctly identified 17 of 20 true positives.
- All predicted positives were correct (precision = 100%).
- The few false negatives reflect the NRC threshold cutoff, showing the trade-off between recall and filtering confidence.

According to the results from Figure 15:

- All original (non-mutated) sequences appeared in the top-10 with low NRC values (≈0.15).
- Mutated variants still ranked well, but closer to the threshold. This reveals how the sequence divergence affects the compression similarity.
- This supports the NRC's role as a continuous metric of sequence similarity.

V. Discussion

A. Interpretation of Results

Our experimental results demonstrate that the Normalized Relative Compression (NRC) metric effectively captures biological relationships between DNA sequences without requiring sequence alignment. The parameter optimization study

```
==================================================
                EVALUATION RESULTS
==================================================
NRC Threshold: 0.500000

Confusion Matrix:
--------------------------------------------------
                |     Actual     |     Actual     |
                |    Positive    |    Negative    |
--------------------------------------------------
 Predicted      |          17    |           0    |
 Positive       |                |                |
--------------------------------------------------
 Predicted      |           3    |          80    |
 Negative       |                |                |
--------------------------------------------------

Metrics:
--------------------------------------------------
Accuracy:  97.0000%
Precision: 100.0000%
Recall:    85.0000%
F1 Score:  0.9189
ROC AUC:   1.0000
--------------------------------------------------
```

Fig. 14. Synthetic Results - Output.

```
Top 20 matches by NRC:
--------------------------------------------------
Rank |        NRC |     Status | Reference
--------------------------------------------------
   1 |   0.151432 |   TRUE POS | sequence_83_meta
   2 |   0.151577 |   TRUE POS | sequence_86_meta
   3 |   0.152661 |   TRUE POS | sequence_84_meta
   4 |   0.153892 |   TRUE POS | sequence_87_meta
   5 |   0.154849 |   TRUE POS | sequence_81_meta
   6 |   0.160097 |   TRUE POS | sequence_85_meta
   7 |   0.160648 |   TRUE POS | sequence_88_meta
   8 |   0.161236 |   TRUE POS | sequence_82_meta
   9 |   0.162995 |   TRUE POS | sequence_90_meta
  10 |   0.166849 |   TRUE POS | sequence_89_meta
  11 |   0.231097 |   TRUE POS | sequence_98_mut_88
  12 |   0.282013 |   TRUE POS | sequence_91_mut_81
  13 |   0.412662 |   TRUE POS | sequence_97_mut_87
  14 |   0.470191 |   TRUE POS | sequence_94_mut_84
  15 |   0.484995 |   TRUE POS | sequence_96_mut_86
  16 |   0.487677 |   TRUE POS | sequence_99_mut_89
  17 |   0.489892 |   TRUE POS | sequence_93_mut_83
  18 |   0.543829 |  FALSE NEG | sequence_95_mut_85
  19 |   0.548875 |  FALSE NEG | sequence_100_mut_90
  20 |   0.631181 |  FALSE NEG | sequence_92_mut_82
--------------------------------------------------

False negatives (missed sequences that should be detected): 3
```

Fig. 15. Synthetic Metrics - Output.

revealed that context length k values around 17 and smoothing parameters α near 0.01 provide optimal discrimination between organisms, which aligns with biological expectations. This specific k-value range is particularly interesting as it approximates the size of regulatory motifs and other functional elements in many genomes, suggesting that our model captures biologically relevant patterns rather than simply statistical noise.

The consistent performance of certain organisms across parameter variations, particularly *Octopus vulgaris* in our tests, provides strong evidence for the robustness of this approach. When a metagenomic sample consistently shows low NRC scores against a specific reference genome across different parameter settings, it strongly indicates genuine biological similarity rather than algorithmic artifacts.

In chunk analysis, regions showing distinct taxonomic assignments likely represent areas of horizontal gene transfer, chimeric sequences, or contamination. These transitional regions are often overlooked by global alignment approaches but are biologically significant, particularly in metagenomic samples where mixed-origin content is expected. The ability to detect such regions without prior knowledge of sequence boundaries represents a significant advantage of our compression-based approach.

B. Comparison with Traditional Methods

Unlike traditional alignment-based methods such as BLAST, which focus on finding local sequence similarities, our NRC-based approach offers distinct advantages for metagenomic classification:

- **Alignment-Independence:** By avoiding the need for explicit sequence alignment, our approach can detect similarity even when sequences have undergone significant rearrangements or contain insertions and deletions that would confound alignment-based methods.
- **Computational Efficiency:** Our multi-threaded implementation demonstrates efficient scaling with larger reference databases, with execution times growing only moderately with increasing context size. This is particularly important for metagenome analysis, where reference databases can be extremely large.
- **Pattern Recognition:** The FCM-based approach captures higher-order dependencies in nucleotide sequences that simple k-mer counting methods might miss, providing a more nuanced view of genetic relationships.
- **Pattern Recognition:** The FCM-based approach captures higher-order dependencies in nucleotide sequences that simple k-mer counting methods might miss, providing a more nuanced view of genetic relationships.

However, alignment-based methods still offer advantages in specificity and established statistical significance measures that our approach currently lacks. A hybrid approach combining both methodologies might offer the best of both worlds for comprehensive metagenomic analysis.

C. Limitations and Challenges

Despite promising results, our approach has several limitations that warrant consideration:

- **Parameter Sensitivity:** Though our implementation was robust across a range of values, optimal performance requires careful parameter selection, which may vary depending on the characteristics of the input sequences. Currently this requires empirical testing rather than automated selection.
- **Reference Database Dependency:** Like all classification approaches, the quality and breadth of the reference database significantly impacts performance. Taxonomic groups poorly represented in the database may go undetected or be misclassified.
- **Model Assumptions:** The Markov property underlying FCMs assumes that the probability of a symbol depends only on the preceding k symbols, which may not capture longer-range dependencies in genomic sequences.
- **Computational Constraints:** While efficient for moderate-sized datasets, the memory requirements grow exponentially with context size, limiting practical applications with extremely large k values.
- **Novel Sequence Handling:** Sequences with no close match in the reference database present a challenge, as they will receive uniformly high NRC scores, making it difficult to determine appropriate taxonomic assignment.

VI. Conclusion and Future Work

A. Objectives Achieved

Our project has successfully:

- **Implemented a Metagenome Classifier:** We developed MetaClass, an efficient C++ implementation that utilizes Normalized Relative Compression to identify similarities between metagenomic samples and reference organisms.
- **Created a Comprehensive Testing Framework:** Our modularized testing infrastructure enables systematic parameter optimization and advanced analysis features including chunk analysis, symbol information analysis, and cross-comparison.
- **Demonstrated Effective Parallelization:** The multi-threaded implementation significantly reduces execution time, making comprehensive parameter testing and large database analysis feasible.
- **Validated with Synthetic Data:** Quantitative evaluation using synthetic datasets with known ground truth confirmed the effectiveness of our approach, with high accuracy and AUC values.
- **Developed Visualization Tools:** Our Python-based visualization framework provides intuitive interpretation of complex NRC data, enabling both parameter optimization and biological insight.

B. Key Findings

We have discovered that:

- Context sizes k between 13-17 and smoothing parameters α around 0.01 provide optimal discrimination between organisms for DNA sequence classification using FCM-based compression, but this is highly dependent on sequence size, as our synthetic sequences benefited from k values of around 8-10, keeping the same α.
- Compression-based metrics like NRC can effectively capture biological relationships between DNA sequences without requiring sequence alignment, offering complementary insights to traditional methods.
- The chunk analysis approach successfully identifies regions of varying taxonomic assignment within a single sequence, potentially indicating horizontal gene transfer or mixed-origin content.
- Symbol information analysis reveals distinctive nucleotide distribution patterns that can help identify conserved regions of potential functional importance.
- Cross-comparison between top organisms provides insights into their phylogenetic relationships, with the NRC and KLD metrics offering complementary perspectives on sequence similarity.

C. Future Work

Several promising directions could extend this work:

- **Adaptive Parameter Selection:** Developing algorithms to automatically select optimal context size and smoothing parameter based on sequence characteristics would eliminate the need for manual parameter tuning.
- **Hierarchical Classification:** Implementing a taxonomic hierarchy-aware classification system could improve accuracy by taking into account evolutionary relationships between organisms.
- **Model Extensions:** Exploring variable-order Markov models or incorporating long-range dependencies could improve the detection of complex genomic patterns.
- **Integration with Existing Pipelines:** Creating interfaces to popular bio-informatics workflows would facilitate broader adoption of compression-based metrics in metagenome analysis.
- **Real Metagenomic Applications:** Testing on complex environmental samples with unknown compositions would provide further validation in real-world scenarios relevant to exobiology and environmental science.
- **Confidence Metrics:** Developing statistical measures to quantify classification confidence would enhance the interpretability of results, particularly for novel sequences with no close match in the reference database.

In conclusion, our FCM-based approach to metagenome classification demonstrates the power of information-theoretic methods in bio-informatics. By evaluating sequence similarity based on compression efficiency rather than direct alignment, we gain unique insights into genomic relationships that complement traditional approaches. This may prove particularly valuable in exobiology contexts, where novel genetic sequences might not align well with known terrestrial organisms despite sharing underlying statistical properties.

REFERENCES

[1] T.A.I. - Elearning. Retrieved from https://elearning.ua.pt/course/view.php?id=5431

[2] A. Ribeiro, M. Sardinha, M. Pinto (2025). *TAI_Project1* [Computer software]. GitHub. Retrieved from https://github.com/miguel-silva48/TAI_Project1

[3] T.A.I. 2nd Project Guidelines - Elearning. Retrieved from: https://uapt33090-my.sharepoint.com/:b:/r/personal/an_ua_pt/Documents/DETI/TAI%202024-2025/Pub/Project%20%2302/TAI-W2.pdf?csf=1&web=1&e=QGH9pU

[4] Research Council (US) Task Group on Life Sciences. Space Science in the Twenty-First Century: Imperatives for the Decades 1995 to 2015: Life Sciences. Washington (DC): National Academies Press (US); 1988. 2, Exobiology. Available from: https://www.ncbi.nlm.nih.gov/books/NBK217840/

[5] Shannon, C. E. (1948). *A Mathematical Theory of Communication*. Bell System Technical Journal, 27, 379–423, 623–656.

[6] Kolmogorov, A. N. (1968). *Three approaches to the quantitative definition of information*. In *International Colloquium on Information Theory* (pp. 370–377). Akademiai Kiado.

[7] Ming Li, Xin Chen, Xin Li, Bin Ma and P. M. B. Vitanyi, "The similarity metric," in IEEE Transactions on Information Theory, vol. 50, no. 12, pp. 3250-3264, Dec. 2004, doi: 10.1109/TIT.2004.838101.

[8] Benson G. Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Res. 1999 Jan 15;27(2):573-80. doi: 10.1093/nar/27.2.573. PMID: 9862982; PMCID: PMC148217.

[9] Wikipedia contributors. *Normalized compression distance*. In Wikipedia. Retrieved April 13, 2025, from https://en.wikipedia.org/wiki/Normalized_compression_distance

[10] GTO: A toolkit to unify pipelines in genomic and proteomic research. J. R. Almeida, A. J. Pinho, J. L. Oliveira, O. Fajarda, D. Pratas, SoftwareX, Volume 12, 2020, 100535, doi: https://doi.org/10.1016/j.softx.2020.100535