Music Identification Using Normalized Compression Distance and Frequency Signatures

Group 7 : Afonso Baixo - Daniel Pedrinho - Henrique Coelho
Universidade de Aveiro - DETI

Introduction

Music identification systems are essential tools for recognizing audio segments and matching them to known tracks in a database. This project explores the use of Normalized Compression Distance (NCD) as a metric for comparing frequency signatures extracted from audio files. By leveraging various compression algorithms, the system aims to provide robust identification even under noisy conditions. The project also evaluates the performance of different compressors with various noise levels, analysing the impact of different noise levels.

Theoretical Foundation

The Normalized Information Distance (NID) is defined as

$$\mathrm{NID}(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}},$$

where K(x) represents the Kolmogorov complexity of string x. Since Kolmogorov complexity is non-computable, we approximate it using practical compression algorithms, leading to the Normalized Compression Distance:

$$\mathrm{NCD}(x,y) = \frac{C(x,y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

The key insight is that compression algorithms can serve as practical approximations to Kolmogorov complexity. When applied to audio signatures, this distance metric captures the algorithmic similarity between musical segments. Values near zero indicate high similarity, while values approaching one suggest dissimilarity. This theoretical foundation provides a universal framework for comparing musical content regardless of genre, instrumentation, or recording conditions.

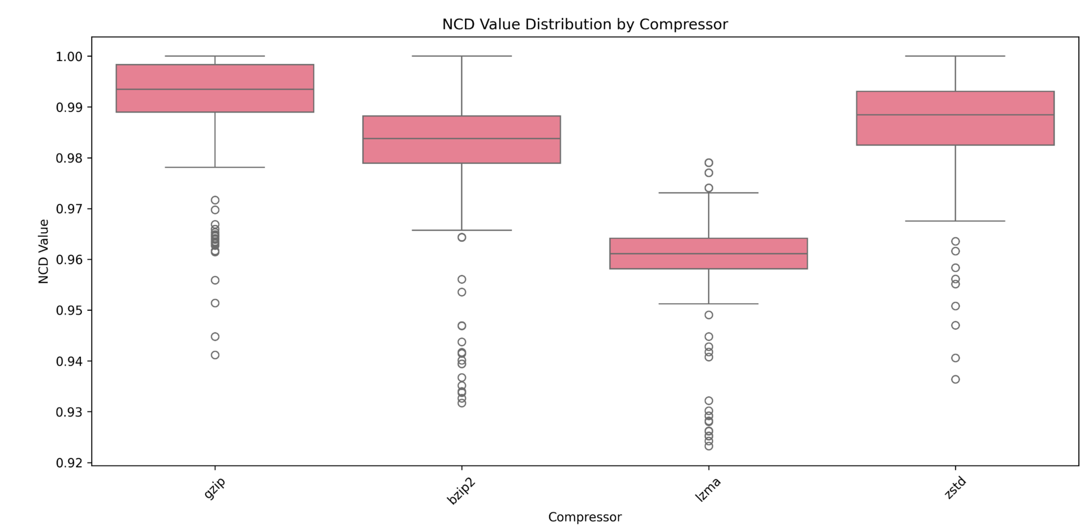
Methodology

The system architecture consists of three main components working in concert to achieve robust music identification. The Audio Processor handles format conversion, segment extraction, and noise addition using SoX for audio manipulation and librosa for Python-based processing. Raw audio files are converted to standardized formats (stereo, 44.1kHz) before signature generation to ensure consistency across different input formats. The signature generation process transforms audio into frequency-domain representations using the GetMaxFreqs tool, which segments audio into overlapping windows and extracts the most significant frequency components. These signatures capture the essential spectral characteristics of musical content while reducing the data to a form suitable for compression-based analysis. The NCD Calculator implements multiple compression algorithms (gzip, bzip2, lzma, zstd) to compute distances between query signatures and database entries, with the system selecting the best match based on the lowest NCD value.

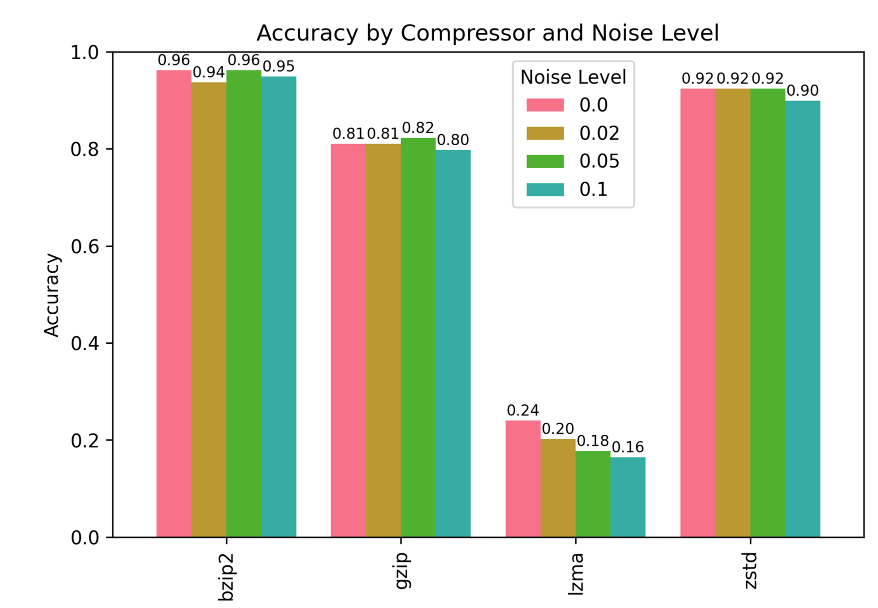
Experimental Setup and Database

The experimental database comprises over 25 diverse musical works spanning multiple genres to ensure comprehensive evaluation. Each complete musical work is processed to generate frequency signatures using configurable parameters including window size (1024), shift (256), down-sampling factor (4), and number of frequency components (4). These parameters were optimized to balance computational efficiency with signature quality.Query generation involves extracting random 10-second segments from database music files, with multiple segments per song to test various portions of each work. The system also generates noisy versions of query segments at different noise levels (0.02, 0.05, 0.1) to evaluate robustness against real-world audio degradation. This comprehensive testing approach ensures that the system can handle both clean and degraded audio input conditions.

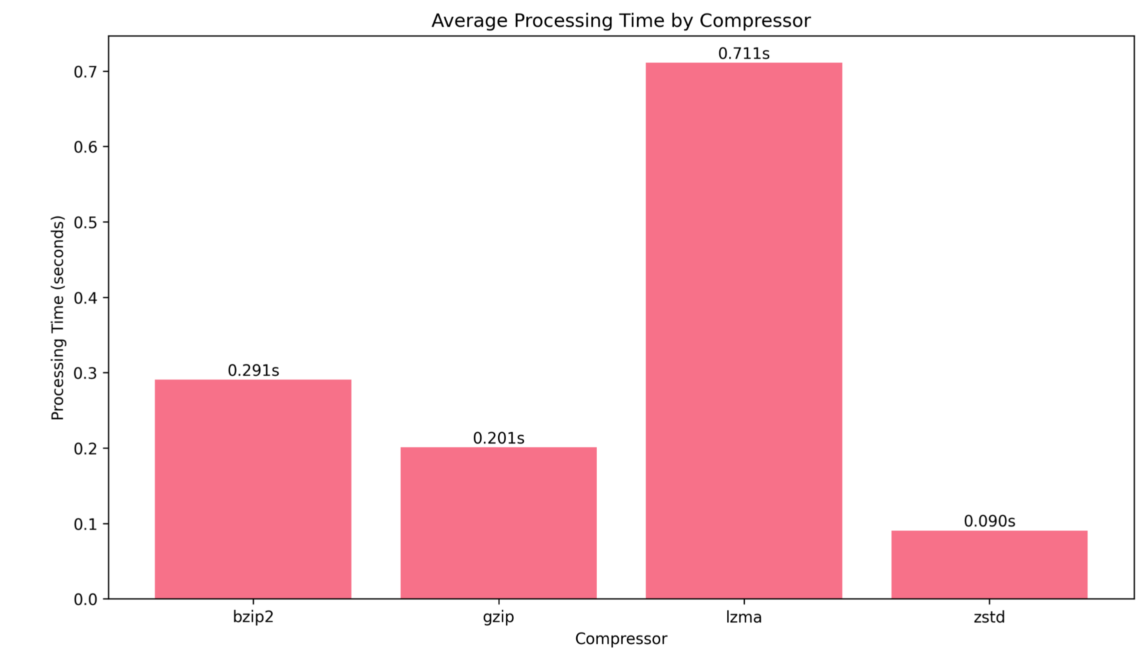
Results



Performance Analysis revealed that NCD values for non-matching pairs clustered near 1.0, while successful matches showed significantly lower values, enabling clear identification. Bzip2 demonstrated the most consistent performance with tightly distributed NCD values.



Bzip2 is remarkably noise-resistant, maintaining (~95%) accuracy across all noise levels, while lzma performs poorly even on clean data (~26%) and degrades further with noise. zstd offers good performance (~91%) with modest noise sensitivity, and gzip provides consistent moderate results (~81%) regardless of noise level, making bzip2 the best choice for noisy data and lzma unsuitable for such applications.



Processing Efficiency showed zstd as the fastest compressor, processing queries in 0.090 seconds, making it ideal for real-time applications. Gzip followed at 0.201 seconds, while bzip2 required 0.291 seconds despite its superior accuracy. Lzma was the slowest at 0.71 seconds.

Noise Robustness testing demonstrated that the system maintained reasonable accuracy under moderate noise levels (0.02, 0.05, 0.1), validating its robustness for real-world scenarios with compromised audio quality, having more variation for the lzma

Conclusion

Bzip2 stands out as the most accurate compressor, achieving over 96% identification success, making it ideal for applications where precision is critical. For scenarios requiring faster processing, zstd offers an excellent compromise, delivering 91% accuracy with processing times under 0.1 seconds. The stark differences in performance across compressors emphasize the importance of choosing the right algorithm for compression-based similarity measures.

The system's robustness was validated through experiments with a database of over 25 music files and 10-second query segments. Noise testing further confirmed its practical applicability, showing that the system gracefully handles moderate noise levels without catastrophic failure, making it suitable for real-world conditions.