Politecnico di Torino

Politecnico di Torino

Master's Degree in Computer Engineering

A.a. 2025/2026

Graduation Session 07/2026

Statistical Modeling of Genomic Sequences

Supervisors:

Prof. Renato Ferrero

PhD Candidate Chiara Panico

Candidate:

Pegah Yarahmadi

Abstract

Traditional genomic analysis and identification of patterns and regularities within genomic sequences is a cornerstone of bioinformatics. It frequently relies on sequence alignment that is often computationally intensive and requires precise positional anchoring. This thesis explores alternative, alignment-free methodologies to identify structural patterns and functional regions within Deoxyribonucleic Acid (DNA) sequences.

K-mer compositions are extracted from each sequence and are converted into a probability distribution using a shared vocabulary. Several types of alignment-free methods are compared, such as vector-based, divergence-based, and entropy-weighted approaches, to see how well they perform. The main contribution is a weighted computation of Kullback–Leibler (KL) Divergence method. In this approach, each k-mer gets a weight based on its statistical importance in the dataset, instead of treating all k-mers the same. Different weighting strategies are tested, from moderate information-based methods to those that focus on rare k-mers or use sample-frequency weighting.

The methods are first applied to a controlled set of ASH1L-derived human sequences, divided into functional and pathological groups. Overall, the separation between groups is limited, which makes sense since the sequences are very similar and come from the same genomic region. Still, the weighted methods, especially the less aggressive ones, perform a bit better than the more aggressive weighting strategies. This suggests that giving more weight to informative k-mers helps, as long as very rare and possibly noisy patterns do not take over the comparison.

A second test is done on a larger set of complete Brucella genomes, where the results are more clear. The weighted KL-based methods separate the species much better than the unweighted and baseline approaches, with the information-based and moderately frequency-weighted methods working most reliably. Overall, the two rounds of experiments show that weighting improves alignment-free genomic comparison when there is a real compositional signal in the data, although the amount of improvement still depends on the dataset, how the groups are defined, and how strongly the weighting is applied.

Keywords: Bioinformatics, Statistical modelling, Genome analysis, Kullback–Leibler Divergence

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Renato Ferrero and Dr. Chiara Panico, for their invaluable guidance, insightful feedback, and constant encouragement throughout the development of this thesis. Their expertise and dedication have been a continuous source of inspiration, shaping both the technical and methodological aspects of my work.

I would like to thank my family, especially my big sister, whose love, patience, and support have carried me through every stage of this journey, and my friends, whose encouragement and presence made the difficult moments easier.

I dedicate this work to Nika, Sarina, Hamidreza, Abolfazl, and all the brave, beautiful, innocent people of my country, Iran, who lost their lives for freedom on January 8,9 2026.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Context and Motivation

The rapid growth of genomic data has created new opportunities for understanding biological variation through computational and statistical methods [1, 2]. A genome is seen as a long sequence made up of four letters: A, C, G, and T [3]. Beyond its biological function, it also has a statistical structure, with repeated patterns, irregularities, and areas of varying complexity that can be studied using mathematics [3, 4]. Because of this, genomic sequences are well suited for methods from information theory and probability. This is also one reason why alignment-free analysis has become more popular as genomic data has rapidly increased [3, 5].

Comparing DNA sequences is a key part of this work, whether it is for detecting mutations, classifying samples, or understanding relationships between individuals or conditions [6]. Traditionally, researchers use sequence alignment, but this approach becomes expensive when working with large datasets. Another option is to represent each sequence by its k-mer frequencies and compare these statistical profiles without any alignment [7, 8]. With this method, entropy can describe the sequence's internal complexity, while divergence or distance measures show how two distributions differ [9].

This thesis explores how well these information-theoretic tools work on real genomic data. It focuses on using k-mer distributions as the main representation and tests several comparison methods: cosine similarity, Jensen–Shannon divergence, k-mer Weighted Inner Product (KWIP), and a group of weighted KL divergence variants. The goal is to find out if a weighted form of KL divergence, adapted to genomic k-mer data, can match or improve on current alignment-free methods. [10, 11].

1.2 Research Objectives

To achieve the overall goal of the thesis, the following research objectives are defined:

1. **Presenting genomic sequences using k-mer distributions**
   Convert DNA sequences into numerical representations based on the frequency of short subsequences, called k-mers, so that statistical and information-theoretic methods can be applied.

2. **Implementing alignment-free methods for genomic sequence comparison**
   Examine representative categories of alignment-free approaches, including vector-based, divergence-based, and entropy-weighted methods. Apply each one under consistent experimental conditions so that their behavior can be compared using the same genomic datasets and evaluation procedure.

3. **Developing KL-divergence-based measures**
   Develop a weighted version of the KL divergence in which k-mers are weighted according to their statistical importance in the dataset.

4. **Analyzing the properties and limitations of the proposed methods**
   Studying whether the proposed measure satisfies desirable properties such as non-negativity, symmetry, and the triangle inequality, and discuss whether it can be considered a formal metric.

5. **To evaluate the ability of each measure to separate genomic samples**
   Using Silhouette Scores, supported by distance matrices and visual analysis, to assess how well each method reflects the predefined genomic groups.

6. **Comparing performance across different genomic datasets**
   Evaluating the methods on both ASH1L-derived dataset and the complete Brucella genome dataset in order to determine how dataset size, sequence similarity, and class structure affect method performance.

1.3 Method Overview

This thesis uses an alignment-free approach to compare genomic sequences. Each sequence is divided into overlapping k-mers. Their frequencies are then counted and used to build a numerical representation of the sequence.

Several categories of comparison methods are applied to these representations. Some methods treat the k-mer profiles as vectors, while others treat them as probability distributions. The main focus is on KL-based dissimilarity measures and on testing whether different weighting functions can improve the comparison.

The analysis is repeated for different k-mer lengths and smoothing settings. For each method, a pairwise dissimilarity matrix is generated. The results are evaluated using overall and per-sample Silhouette Scores, together with visual inspection of the distance structure.

The methods are tested on two datasets. The first contains controlled ASH1L-derived human genomic sequences. The second contains complete Brucella genomes. This makes it possible to examine how the methods behave on datasets with different sizes, sequence characteristics, and class structures.

1.4 Contributions

This thesis develops and evaluates a weighted KL-based approach for comparing genomic sequences through their k-mer distributions. Different weighting functions are tested to control the influence of common, rare, and sample-specific k-mers.

Another contribution is the comparison of several alignment-free method categories under the same evaluation procedure. The methods are tested on both a controlled ASH1L-derived dataset and a larger Brucella genome dataset.

The work also includes a memory-efficient implementation for large genomic data. This made it possible to process the Brucella dataset using chunked and disk-based computation without changing the underlying mathematical results.

1.5 Thesis Structure

This thesis is organized into six chapters.

Chapter 1 introduces the topic, the research objectives, the main contributions, and the overall approach.

Chapter 2 provides the biological and computational background needed to understand genomic sequences, file formats, variant data, and k-mer representations.

Chapter 3 reviews the main categories of alignment-free sequence comparison, with particular attention to divergence-based, vector-based, and entropy-weighted methods.

Chapter 4 describes the datasets, preprocessing steps, weighted KL formulation, comparison methods, and evaluation procedure.

Chapter 5 presents and discusses the results obtained on the ASH1L-derived and Brucella datasets.

Chapter 6 summarizes the main findings, discusses the limitations of the study, and outlines possible directions for future work.

Chapter 2

Biological and Computational Background

2.1 Genomic Sequences and DNA Representation

Genomic sequences encode the biological information necessary for the development, function, and reproduction of living organisms. In computational biology, DNA is typically represented as a symbolic sequence over the alphabet consisting of four nucleotide bases: adenine, cytosine, guanine, and thymine. These bases are denoted by the letters A, C, G, and T, respectively. Therefore, a DNA sequence can be modeled as a string over the alphabet

$$\Sigma = \{A, C, G, T\}.$$

For example, a short DNA sequence may be written as

$$S = ACGTACGTTAGC.$$

This symbolic representation allows genomic sequences to be studied using mathematical, statistical, and computational methods. Although DNA has a biological meaning, from a computational point of view it can also be considered as a long discrete sequence. This makes it possible to analyze regularities, repetitions, variations, and statistical patterns inside the sequence.

A genome is the complete set of genetic material of an organism. In humans, the genome is organized into chromosomes. Each chromosome contains long DNA

molecules, and specific regions of these molecules may correspond to genes or regulatory elements. Since different individuals have highly similar genomes but also contain variations, comparing genomic sequences is an important problem in bioinformatics. These variations may include single nucleotide changes, insertions, deletions, or more complex structural differences.

In many genomic studies, individual sequences are compared with a reference genome. A reference genome is a representative sequence used as a coordinate system for describing genomic regions and variants. For human genomic analysis, $GRCh38$ is one of the commonly used reference genome assemblies. By comparing individual samples with a reference genome, it is possible to identify variants and study how different samples differ from each other.

In this thesis, genomic sequences are treated as symbolic sequences that can be represented statistically. Instead of comparing sequences only position by position, the focus is on extracting frequency-based features from the sequences and using these features to compute similarity or distance between samples.

2.2 Reference Genomes and Genetic Variants

A reference genome is a representative sequence used as a common coordinate system for genomic analysis. Variants in each sample are described by comparing its sequence with this reference.

A genetic variant refers to any deviation from the reference sequence. For example, if the reference genome contains an adenine (A) at a specific position and a sample has a cytosine (C) at the same position, this is a variant. The reference gene is the nucleotide present in the reference genome, whereas the alternative sequence represents the nucleotide observed in the sample.

Variants can be described using the reference allele and the alternative allele. The reference allele is the nucleotide or sequence present in the reference genome, while the alternative allele represents the observed difference in the sample. In diploid organisms such as humans, each individual has two copies of most chromosomes, and therefore genotype information is also important. The genotype describes which alleles are present in a given sample.

Understanding variants is important because they can influence biological features, disease susceptibility, or functional properties of genomic regions. In this thesis, variants are relevant because sample-specific sequences may be derived from variant information and compared using statistical measures.

2.3 Genomic File Formats

Genomic data are commonly stored in specialized file formats. These formats are designed to represent sequences, variants, annotations, and other biological information in a structured way. In this work, the most relevant formats are FASTA and Variant Call Format (VCF).

2.3.1 FASTA Format

The FASTA format is one of the most widely used formats for storing biological sequences. It can be used for nucleotide sequences such as DNA and RNA, as well as protein sequences. A FASTA file is composed of one or more sequence records. Each record begins with a header line starting with the character `>`, followed by one or more lines containing the sequence.

An example of a FASTA record is:

```
>sample1
ACGTACGTACGTACGT
```

The header line usually contains the name or identifier of the sequence. The following lines contain the sequence itself. In the case of DNA, the sequence is typically composed of the characters A, C, G, and T, although additional symbols may sometimes appear. For example, the character N is often used to represent an unknown or unspecified nucleotide.

FASTA files are particularly useful for sequence-based analysis because they provide the raw sequence information needed for methods such as K-mer counting. In this thesis, FASTA files are used as the input representation for extracting K-mer frequency distributions from genomic sequences.

2.3.2 VCF Format

The Variant Call Format, commonly known as VCF, is used to store genetic variants. A VCF file describes differences between one or more samples and a reference genome. Each variant record usually contains information such as chromosome name, genomic position, reference allele, alternative allele, quality values, annotations, and genotype information.

A simplified VCF record may contain the following fields:

```
CHROM   POS    ID   REF   ALT   QUAL   FILTER   INFO   FORMAT   SAMPLE
1       1000   .    A     C     .      PASS     .      GT       0/1
```

In this example, the variant is located on chromosome 1 at position 1000. The reference allele is A, and the alternative allele is C. The genotype value `0/1` indicates that the sample is heterozygous, meaning that one allele corresponds to the reference and the other corresponds to the alternative.

In this thesis, VCF files were used during preprocessing to select ASH1L variants and generate sample-specific consensus sequences. The resulting FASTA files were then used for the K-mer-based comparison.

2.3.3 Compressed and Indexed Genomic Files

Genomic files are typically compressed since they can be very large. For example, FASTA files may be compressed as `.fa.gz`, and VCF files may be compressed as `.vcf.gz`. Compression reduces storage requirements and can make data transfer more efficient.

Index files are also commonly used to allow fast access to specific genomic regions. For example, FASTA files can be indexed using `.fai` files, while compressed VCF files can be indexed using `.tbi` or `.csi` files. These index files make it possible to retrieve only a specific chromosome or genomic interval without reading the entire file.

2.4 Tools for Genomic Data Processing

Several computational tools are commonly used to process genomic data before statistical analysis. These tools are not the main theoretical focus of the thesis, but they are important for preparing the data correctly.

2.4.1 BCFtools

`BCFtools` is a command-line tool used for manipulating and analyzing VCF and BCF files. It provides functions for viewing, filtering, indexing, querying, and processing variant data. For example, `BCFtools` can be used to inspect variant records, extract specific samples, filter variants according to quality criteria, and analyze genotype information.

In this thesis, `BCFtools` is relevant during the preprocessing stage. It can be used to inspect variant files, extract variant information, and prepare sample-specific data for subsequent analysis. Since the final comparison is based on genomic sequences or K-mer distributions, `BCFtools` acts mainly as a preprocessing tool rather than as a similarity measure itself.

2.4.2 FASTA Indexing Tools

FASTA indexing tools, such as those provided by `SAMtools`, are used to create index files for FASTA sequences. Indexing allows efficient access to specific regions of a genome. This is useful when only a particular gene, chromosome, or genomic interval is needed for analysis.

For example, if a study focuses on a specific gene region, indexing allows that region to be extracted without loading the entire genome into memory. This is important when working with large reference genomes such as the human genome.

2.4.3 K-mer Counting Tools

Since this thesis is based on K-mer frequency distributions, K-mer counting is a central preprocessing step. A K-mer counting tool takes a sequence as input and counts the occurrences of all subsequences of length k. These counts can then be converted into frequency vectors or probability distributions.

Different tools and programming libraries can be used for K-mer counting. In this work, K-mer frequencies may be computed using Python scripts or specialized tools depending on the experimental setup. The resulting K-mer counts provide the numerical representation used by the similarity and divergence measures.

2.5 K-mer Representation of Genomic Sequences

A K-mer is a substring of length k extracted from a biological sequence. For a DNA sequence, a K-mer is therefore a short word composed of the alphabet {A, C, G, T}. The value of k determines the length of the extracted patterns.

For example, consider the sequence

S=ACGTAC.

If k =3, the 3-mers are obtained by sliding a window of length 3 across the sequence:

ACG,CGT,GTA,TAC.

For a sequence of length L, the total number of overlapping K-mers is

$$L - k + 1.$$

The K-mer representation transforms a genomic sequence into a vector of counts. Each possible K-mer corresponds to one feature, and the value of that feature is the number of times the K-mer appears in the sequence. Since DNA has four possible nucleotides, the total number of possible K-mers is

$$4^k.$$

For small values of k, the number of possible K-mers is limited. However, as k increases, the feature space grows exponentially. For example, for k=3, there are $4^3 = 64$ possible K-mers, while for k=15, there are 4^{15} possible K-mers.

After counting K-mers, the count vector can be normalized to obtain a frequency distribution. If $c_i(x)$ is the count of K-mer x in sequence i, then the normalized frequency can be written as

$$P_i(x) = \frac{c_i(x)}{\sum_y c_i(y)}.$$

Here, $P_i(x)$ represents the empirical probability of observing K-mer x in sequence i. This representation allows each genomic sequence to be treated as a probability distribution over K-mers.

The choice of k is important. Small values of k capture short and general patterns, but they may not be specific enough to distinguish between similar sequences. Large values of k capture longer and more specific patterns, but they also create sparse distributions because many possible K-mers may not appear in a given sequence. Therefore, different values of k, such as $k = 15$, $k = 21$, and $k = 31$, can be tested to study how the K-mer length affects the comparison between genomic samples.

Chapter 3

Information-Theoretic Context

3.1 Overview of Sequence Comparison

Among alignment-free methods, K-mer-based approaches are particularly common. A K-mer is a contiguous subsequence of length (k) extracted from a longer biological sequence. By counting the occurrence of all K-mer in a sequence, each sample can be transformed into a numerical representation that can be compared without requiring explicit alignment. This makes K-mer methods useful for genome comparison, metagenomic analysis, classification, population-genetic analysis, and reference-free sequence comparison [12, 13, 14, 15].

Recent studies show that K-mer methods are not limited to one type of analysis. For example, Roberts et al. discuss the role of K-mer in connecting reference-free pangenomic approaches with population-genetic analyses [12]. Bussi et al. use large-scale K-mer set comparison to study informational properties of genomes and taxonomic structure [13]. Other studies apply K-mer-based methods to metagenomic datasets, environmental genomic data, or alternative sequence representations such as strobemers [14, 15, 16]. Therefore, K-mer similarity approaches should not be viewed as a single method, but rather as a family of related methods that differ in how K-mer information is represented and how similarity or dissimilarity is computed.

3.2 Literature Review and Taxonomy of Similarity Approaches

The literature on K-mer-based sequence comparison can be organized into several methodological families. This classification is useful because different studies use different representations, similarity measures, and evaluation datasets. In this thesis, the taxonomy is organized into four main categories: divergence-based methods, kernel-based, vector-based, and other alignment-free approaches. A further distinction is made according to the type of dataset used for evaluation, since methods validated on whole genomes, metagenomes, simulated reads, or gene-level datasets may not have the same behavior.

3.3 Divergence-based Methods

Divergence-based methods represent each genomic sequence as a probability distribution over its observed K-mers. The difference between two sequences is then measured by comparing their probability distributions rather than their positions in the original sequence.

Common examples include KL divergence, Jensen–Shannon divergence, and weighted variants of these measures. These approaches are closely related to the main focus of this thesis, where genomic sequences are converted into normalized K-mer distributions and compared using both standard and weighted divergence formulations.

The main advantage of this category is its probabilistic interpretation. It allows differences in K-mer composition to be expressed in terms of information loss or distributional change. However, some divergence measures are sensitive to zero probabilities and sparse data. This is especially important for large K-mer values, where many features may be absent from individual samples.

3.3.1 Probability Distributions and Smoothing

After K-mer counting, each sequence can be represented as a probability distribution. This is done by normalizing the K-mer counts so that the sum of all probabilities equals one:

$$\sum_{x} P(x) = 1.$$

Here, $P(x)$ is the relative frequency of K-mer x in the sequence. This form is needed because information-theoretic measures compare probability distributions rather than raw counts.

A difficulty appears when a K-mer is present in one sequence but absent from another. One distribution then assigns it a positive probability, while the other assigns zero. This creates a problem for KL divergence because division by zero makes the logarithmic term undefined. To avoid this, a small positive value is added to every K-mer count before normalization:

$$P_i(x) = \frac{c_i(x) + \alpha}{\sum_y (c_i(y) + \alpha)},$$

The parameter α controls the amount of smoothing. When $\alpha > 0$, all K-mers receive a non-zero probability, which makes the divergence calculation more stable. The effect of different smoothing values is examined later in the experimental analysis.

3.3.2 Entropy and Information-Theoretic Measures

Information theory provides tools for describing uncertainty and differences between probability distributions. Since genomic sequences can be represented through their K-mer frequencies, these tools can be used to study their statistical structure. One of the main quantities is Shannon entropy:

$$H(P) = -\sum_x P(x) \log P(x),$$

where $P(x)$ is the probability of K-mer x. Entropy describes how evenly the K-mers are distributed within a sequence. A sequence with many K-mers occurring at similar frequencies has higher entropy, while a sequence dominated by a small number of repeated patterns has lower entropy.

Entropy describes one sequence at a time. To compare two sequences, divergence or similarity measures are needed. These measures show how different their K-mer distributions are. For this reason, information-theoretic methods are useful in alignment-free genomic analysis. They make it possible to compare sequences through their statistical composition without matching them position by position.

3.3.3 Kullback–Leibler Divergence

The KL divergence is an information-theoretic measure used to compare two probability distributions. For two distributions P and Q, it is defined as:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}.$$

In this thesis, P and Q represent the K-mer probability distributions of two genomic sequences. If they are identical, the KL divergence is zero. The divergence

measures how different these distributions are and is useful for genomic comparison because it focuses on changes in the relative frequencies of K-mers as shown in the figure 3.1. A K-mer that is common in one sample but rare in another can make a larger contribution to the final value. This allows the measure to capture differences in sequence composition without aligning the sequences position by position.

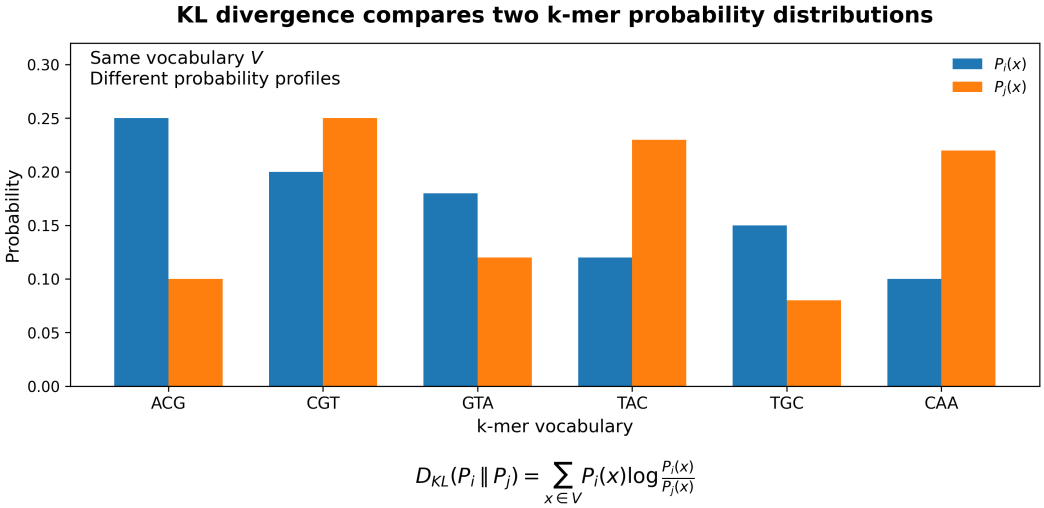

Figure 3.1: Conceptual intuition of KL divergence comparing two distinct K-mer probability profiles, $P_i(x)$ and $P_j(x)$

The standard KL formulation is used as the mathematical basis of the proposed approach. It is then adapted to the characteristics of genomic K-mer data, which are often sparse and high-dimensional. Smoothing is used to handle zero probabilities, while weighting functions are introduced to control the influence of common and rare K-mers.

3.3.4 Properties and limitations of KL divergence

KL divergence is useful for comparing probability distributions, but it also has some limitations. The first is that it is not symmetric:

$$D_{\mathrm{KL}}(P \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P).$$

In other words, comparing P with Q does not always give the same result as comparing Q with P. Since the analyses in this thesis require one value for each pair of samples, the two directions are averaged:

$$D_{\mathrm{KL}}^{\mathrm{sym}}(P_i, P_j) = \frac{1}{2}\left[D_{\mathrm{KL}}(P_i \parallel P_j) + D_{\mathrm{KL}}(P_j \parallel P_i)\right].$$

KL divergence is also not a true distance metric because it does not always satisfy the triangle inequality. For this reason, it is better described as a divergence or dissimilarity measure. Another issue occurs when a K-mer is present in one sample but absent from another. If one probability is positive and the other is zero, the logarithmic term becomes undefined. This is common in sparse K-mer distributions, especially for larger values of K. Smoothing is therefore needed before the divergence is calculated.

It can also give a large contribution to rare K-mers. This may be useful when those K-mers carry meaningful biological information, but it can also increase the effect of noise or sample-specific patterns. These limitations explain why the proposed approach uses symmetrization, smoothing, and weighting.

3.3.5 Jensen–Shannon Divergence

Jensen–Shannon divergence is another information-theoretic measure for comparing two probability distributions. It is closely related to KL divergence, but it is symmetric and more stable when zero probabilities are present. For two distributions P and Q, their midpoint distribution is defined as

$$M = \frac{1}{2}(P + Q).$$

The Jensen–Shannon divergence is then

$$D_{\mathrm{JS}}(P, Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M).$$

Because both distributions are compared with the same midpoint, the result is symmetric:

$$D_{\mathrm{JS}}(P, Q) = D_{\mathrm{JS}}(Q, P).$$

This makes Jensen–Shannon divergence easier to use in pairwise sequence comparison. It is also bounded when logarithm base 2 is used, and the square root of the divergence defines a proper distance:

$$d_{\mathrm{JS}}(P, Q) = \sqrt{D_{\mathrm{JS}}(P, Q)}.$$

In this thesis, Jensen–Shannon distance is applied to the K-mer probability distributions of the genomic samples. It provides a standard divergence-based baseline that can be compared with the KL-based formulations. Alignment-free sequence comparison methods commonly use word or K-mer counts to represent biological sequences numerically, avoiding the need for explicit sequence alignment [17]. These approaches have been widely used in applications such as genome comparison, protein family classification, and next-generation sequencing analysis [6, 7].

3.4 Kernel-based Methods

A third family consists of kernel-based and weighted inner-product methods. These approaches compare sequences by computing inner products or kernel functions over K-mer profiles. Instead of treating all K-mers equally, some methods assign different weights to different K-mers according to their informativeness.

The KWIP method is an important example of this category. It computes a K-mer weighted inner product and uses entropy-based weighting to reduce the influence of highly common K-mers while emphasizing more informative ones [18]. This idea is relevant to the present thesis because the proposed KL-based analysis also investigates the effect of weighting K-mers. Although KWIP is not a divergence-based measure, it is methodologically important because it shows that weighting can improve the biological relevance of K-mer similarity estimation.

3.4.1 KWIP and Entropy-Weighted K-mer Similarity

KWIP is an alignment-free method that compares genomic samples through their K-mer counts. Its main idea is that not all K-mers are equally useful. Very common or very rare K-mers may provide little information, while those shared by an intermediate number of samples can be more helpful for distinguishing them [18]. For each K-mer x, KWIP calculates its presence frequency across the dataset:

$$F(x) = \frac{\text{number of samples containing } x}{\text{total number of samples}}.$$

This value is converted into an entropy weight:

$$H(x) = -\left[F(x)\log_2 F(x) + (1 - F(x))\log_2(1 - F(x))\right].$$

The weight is highest when a K-mer appears in about half of the samples and approaches zero when it appears in almost none or almost all of them. The weighted similarity between samples i and j is then computed as

$$K_{ij} = \sum_x c_i(x)c_j(x)H(x),$$

where $c_i(x)$ and $c_j(x)$ are the counts of K-mer x in the two samples.

The original KWIP method uses compact count sketches to handle large sequencing datasets efficiently [18]. In this thesis, exact K-mer count vectors are used because the datasets are small enough to avoid sketch approximation. The entropy-weighted inner product is then converted into a pairwise distance matrix and evaluated in the same way as the other methods. This method is used in the

thesis because it provides an established example of weighted K-mer comparison. Unlike the KL-based approach, it uses a weighted inner product rather than a divergence between probability distributions.

3.5 Vector-based Methods

Vector-based methods represent each genomic sequence as a numerical vector built from its K-mer counts, frequencies, or presence values. Once the samples are placed in the same feature space, they can be compared using measures such as cosine similarity, Euclidean distance.

This representation is widely used in alignment-free sequence analysis because it is simple, flexible, and suitable for different types of genomic data. Previous studies have shown that K-mer vectors can capture useful genomic and taxonomic patterns and can support analyses such as similarity matrices, clustering, and visualization [12, 13].

In this thesis, cosine similarity is used as the main vector-based baseline. It compares the direction of two K-mer vectors rather than their absolute size, making it useful when the relative composition of the sequences is more important than the total number of observed K-mers.

3.5.1 Cosine Similarity

Cosine similarity is a vector-based measure commonly used to quantify the similarity between two objects represented in a shared feature space. It originated from the vector space model, where objects are represented as vectors and their similarity is evaluated by the angle between them rather than by their absolute magnitude.

The classical vector space model was introduced by Salton, Wong, and Yang [19] for automatic indexing and information retrieval, where documents were represented as weighted term vectors and compared geometrically in a multidimensional space. In the context of biological sequence analysis, the same idea can be adapted by representing each sequence as a vector of K-mer counts or K-mer frequencies.

For two sequences represented by K-mer vectors A and B, cosine similarity is defined as:

$$\text{cosine}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{3.1}$$

where A_i and B_i are the frequencies or counts of the i-th K-mer in the two sequences. If the two vectors point in the same direction, the cosine similarity

approaches 1, indicating a highly similar K-mer composition. If the vectors are orthogonal, the value approaches 0, indicating little overlap in their K-mer patterns. Since K-mer count and frequency vectors contain non-negative values, cosine similarity usually ranges between 0 and 1 in this application.

Cosine similarity is particularly suitable as an alignment-free baseline because it does not require explicit sequence alignment. Instead, it compares the composition of sequences through their K-mer profiles. Alignment-free methods have become important in bioinformatics because they provide alternatives to alignment-based approaches, especially when handling large datasets or when exact positional alignment is not the main objective [20]. These approaches often rely on the distribution of short words or K-mers and have been applied in genome comparison, phylogenetic analysis, classification, and metagenomic analysis [21].

In K-mer-based sequence comparison, each sample is transformed into a high-dimensional vector whose dimensions correspond to the observed K-mers in the dataset. The comparison is then performed in this shared vocabulary space. This representation is conceptually similar to the "bag-of-words" model in text analysis, where the order of features is not directly modeled, but the composition of features is used to compare objects. In genomic applications, the "words" are nucleotide subsequences of length K, and their counts or frequencies summarize the local composition of each sequence.

One advantage of cosine similarity is that it is symmetric. This makes it computationally simple and useful as a reference method.

$$\text{cosine}(A, B) = \text{cosine}(B, A) \tag{3.2}$$

To use cosine similarity in clustering or silhouette score analysis, the similarity value is usually converted into a distance:

$$d_{\text{cosine}}(A, B) = 1 - \text{cosine}(A, B) \tag{3.3}$$

In this form, smaller values indicate more similar samples, while larger values indicate more dissimilar samples. This conversion allows cosine similarity to be compared with other distance- or divergence-based approaches such as Jensen–Shannon divergence and weighted KL-based measures.

Recent bioinformatics work has also applied K-mer-based cosine similarity to biological structure comparison. Lasher and Hendrix [22] introduced `bpRNA-CosMoS`, a method for RNA secondary structure comparison based on K-mer count vectors derived from RNA structure arrays. Their method computes cosine similarity between structural K-mer vectors and further introduces a length-weighted correction to reduce unexpectedly high similarity scores between RNAs of very different

lengths. They also discuss a fuzzy-counting extension, where similar K-mers can contribute pseudo-counts to reduce the effect of small structural variations.

Overall, cosine similarity provides a simple, symmetric, and computationally efficient baseline for alignment-free genomic sequence comparison. It is useful for evaluating whether samples share a similar K-mer composition.

3.6 Other Alignment-free Approaches

Alignment-free sequence comparison also includes methods that do not rely on full k-mer frequency vectors. Some approaches compare only the presence or absence of k-mers, while others use compact sketches or alternative sequence words to reduce memory and computational cost.

Set-based methods often use Jaccard similarity to compare the overlap between k-mer sets. For large genomic datasets, sketching methods provide a faster approximation. Mash uses MinHash sketches to estimate genomic distance, while CMash extends this idea by estimating Jaccard and containment values across multiple k-mer lengths [23, 24]. The CMash study is particularly relevant to this thesis because its associated *Brucella* dataset is used in the final evaluation.

Other methods change the structure of the sequence words themselves. Spaced words and strobemers, for example, are designed to remain informative when substitutions or sequencing errors break ordinary contiguous k-mer matches [16]. Compression-based approaches form another group and estimate similarity from how efficiently sequences can be compressed separately or together. These methods are not implemented in this thesis, but they help place the selected approaches within the wider field of alignment-free genomic comparison.

3.7 Datasets Used for Evaluation in Previous Studies

The performance of a k-mer-based method often depends on the type of dataset used for evaluation. Previous studies have tested alignment-free approaches on complete genomes, simulated reads, metagenomic samples, fragmented assemblies, and long-read sequencing data.

These datasets differ in size, sequence quality, biological origin, and level of variation. As a result, a method that works well on whole genomes may behave differently on short reads or gene-level sequences.

In this thesis, the selected methods are evaluated on two datasets. The first is a controlled set of ASH1L-derived human sequences. The second is a collection of complete *Brucella* genomes obtained from the dataset associated with the CMash study [24]. Using both datasets makes it possible to examine how the methods behave under different sequence scales and class structures.

3.8 Evaluation with Silhouette Score

The Silhouette Score was used to measure how well the genomic samples matched their predefined groups. It provides a numerical summary of both within-group similarity and between-group separation. For each sample i, let $a(i)$ be its average dissimilarity from the other samples in the same group. Let $b(i)$ be its lowest average dissimilarity from samples in another group. The silhouette value is then defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

The value of $s(i)$ ranges from -1 to 1. A value close to 1 means that the sample fits well within its own group. A value close to 0 means that it lies near a group boundary. A negative value means that the sample is, on average, closer to another group than to its assigned group. The overall Silhouette Score is the mean of the individual sample values:

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i),$$

where n is the total number of samples.

In this thesis, the score was calculated from the precomputed pairwise dissimilarity matrix produced by each method. Both the overall score and the per-sample values were examined. The overall score was used to compare methods, while the individual values helped explain which samples were clearly grouped and which ones remained ambiguous.

The Silhouette Score was interpreted together with the distance matrices and visual results. This was especially important for small or unbalanced datasets, where a single global value may hide the behavior of individual samples.

3.9 Comparative Classification of Reviewed Studies

Table 3.1 summarizes the main studies discussed in this chapter. The comparison focuses on the method category, the main idea of each approach, and the dataset used for evaluation. The table shows that alignment-free methods have been tested on a wide range of data, including complete genomes, sequencing reads, metagenomic samples, simulated data, and fragmented assemblies. In this thesis, several representative method categories are evaluated under the same procedure on both ASH1L-derived sequences and complete *Brucella* genomes.

Table 3.1: Classification of reviewed k-mer similarity studies

Study	**Methodological category**	**Main approach**	**Evaluation dataset**
Roberts et al. (2025) [12]	Vector-based / reference-free k-mer approaches	Comparison of k-mer profiles using measures such as cosine, Jaccard, and Bray–Curtis.	Population-genetic simulations, sequencing-depth experiments, and compression settings.
Bussi et al. (2021) [13]	Set-based alignment-free approach	Genome comparison using k-mer Jaccard similarity, similarity matrices, heatmaps, and hierarchical clustering.	Complete genomes from KEGG and NCBI.
Liu and Koslicki (2022) [24]	Sketch-based alignment-free approach	Efficient estimation of k-mer Jaccard and containment indices across multiple k values.	Microbial reference genomes Brucella
Ponsero et al. (2023) [14]	Metagenomic k-mer-based approach	Comparison of de novo comparative metagenomic tools based on k-mer representations.	Simulated short-read metagenomes and real fecal metagenomic datasets.
Van Etten et al. (2023) [15]	Phylogenetic alignment-free approach	k-mer-based distance scoring for phylogenetic classification.	Complete, fragmented, and environmental genomic datasets.
Sahlin (2021) [16]	Alternative k-mer-like representation	Strobemers as an alternative to ordinary contiguous k-mers for sequence similarity detection.	Simulated data and Oxford Nanopore sequencing data.
Murray et al. (2017) [18]	Kernel-based / weighted inner-product method	Entropy-weighted k-mer inner product for genetic similarity estimation.	Sequencing-read datasets for genetic similarity estimation.
Present thesis	Divergence-based, vector-based, and weighted inner-product comparison	KL-based measures, Jensen–Shannon divergence, cosine similarity, and KWIP applied to the same dataset.	Gene-level ASH1L-derived sequence and Brucella dataset.

3.10 Summary

This chapter reviewed the main families of alignment-free sequence comparison. The methods were grouped into divergence-based, vector-based, weighted inner-product, and other alignment-free approaches. Each category uses k-mer information in a different way. Divergence-based methods compare probability distributions, vector-based methods compare numerical profiles, and weighted methods give more importance to selected k-mers. Other approaches use sketches, set overlap, alternative sequence words, or compression.

The reviewed studies also show that method performance depends strongly on the type of dataset. For this reason, the methods in this thesis are evaluated on both ASH1L-derived sequences and complete *Brucella* genomes. The literature supports the use of weighted k-mer comparison, but it also shows that rare features should be handled carefully. This motivates the weighted KL-based formulations introduced in the next chapter.

Chapter 4

Methodology

4.1 Dataset and Preprocessing Pipeline

Two genomic datasets were used in this thesis. The first was a controlled gene-level dataset built from the human ASH1L region. The second contained complete Brucella genomes and was used to test the methods on larger, naturally occurring sequences.

The ASH1L dataset was created so that all samples came from the same genomic region and differed only in the variants applied to the reference sequence. The ASH1L region was extracted from chromosome 1 of the GRCh38 human reference genome using data from Ensembl release 115. Variants located in this region were annotated with the Ensembl Variant Effect Predictor and then divided into two groups according to their predicted consequences: functional variants and predicted pathological variants.

Five samples were generated from each group. For every sample, 50 variants were selected and stored in a sample-specific VCF file. The selected variants were then applied to the ASH1L reference sequence using `bcftools consensus`. This produced ten FASTA files: five functional sequences and five predicted pathological sequences. Since all samples were derived from the same reference region, this dataset provided a controlled setting for the first comparison experiments.

The second dataset was obtained from the resources associated with the CMash paper [24]. It contains 29 complete *Brucella* genome assemblies from several species and strains. Unlike the ASH1L dataset, the Brucella dataset contains complete genomes and has an unbalanced class structure. This limitation was considered during the interpretation of the Silhouette Scores, since singleton classes cannot provide meaningful within-class distances.

For both datasets, the FASTA sequences were cleaned and converted to uppercase. Header lines were removed, and only K-mers containing the standard nucleotide symbols A, C, G, and T were retained. The resulting sequences were then used to construct K-mer count vectors and probability distributions for the comparison methods described in the following sections. The complete preprocessing workflow included:

1. Collecting the reference sequences and variant data;
2. Extracting and annotating the ASH1L variants;
3. Creating the functional and predicted pathological variant groups;
4. Generating sample-specific VCF and FASTA files;
5. Downloading and organizing the Brucella genome assemblies;
6. Assigning class labels from the accompanying CSV file;
7. Cleaning the FASTA sequences;
8. Preparing the sequences for k-mer counting and statistical comparison.

The ASH1L dataset was used in the first stage of the experiments. A second dataset containing complete Brucella genomes was later used to evaluate the methods on larger and naturally occurring genomic sequences.

4.1.1 Genomic Region

ASH1L, also known as ASH1 like histone lysine methyltransferase, is a protein-coding gene located on human chromosome 1. In Ensembl, ASH1L has the gene identifier ENSG00000116539 and is located on the GRCh38 assembly from chr1:155,335,268 to chr1:155,563,162. The Ensembl annotation describes ASH1L as a histone lysine methyltransferase gene and reports multiple transcripts for this gene, making it a suitable region for studying the effect of different variant consequences on generated sequences.

Therefore, the ASH1L reference sequence used in this work was a regional FASTA sequence of length 227,895 base pairs. ASH1L was selected because it is a gene with a relatively large genomic span and multiple annotated transcripts, making it suitable for studying sequence variation and comparing generated individual sequences. Ensembl reports ASH1L as a histone lysine methyltransferase gene located on chromosome 1 and associated with multiple transcripts and phenotypic annotations. [25]

The original reference sequence and variation data were obtained from Ensembl release 115. Ensembl provides downloadable genome sequence data in FASTA format and variant datasets through its FTP resources. The VCF metadata used in this work indicated that the source data came from Ensembl version 115 and that the reference was based on the Ensembl Homo sapiens DNA FASTA resources. The original chromosome-level VCF file is `homo_sapiens-chr1.vcf.gz` that only the ASH1L genomic interval is extracted from this file. This produced an ASH1L-specific regional VCF file. The extracted ASH1L region was then annotated and filtered to create the final functional and pathological variant pools.

4.1.2 VCF Format

The variant information was stored in VCF files used to represent genomic variants such as single-nucleotide variants, insertions, deletions, and other sequence changes. A VCF file contains metadata lines, a header line, and variant records. Each variant record includes fields such as chromosome, position, variant ID, reference allele, alternative allele, quality information, filters, and additional annotation information. The standard VCF fields include `CHROM`, `POS`, `ID`, `REF`, `ALT`, `QUAL`, `FILTER`, and `INFO`; optional genotype information may also be stored in `FORMAT` and `sample` columns.

In this study, the most important VCF field was the `INFO/CSQ` field. This field was produced by **Ensembl Variant Effect Predictor**, or Variant Effect Predictor (VEP), and contains the predicted consequence of each variant. The `VCF` metadata defined the CSQ field as:

```
Allele | Consequence | IMPACT | SYMBOL | Gene | Feature_type
| Feature | BIOTYPE | EXON | INTRON | HGVSc | HGVSp | ...
```

The two most important subfields used in this work are Consequence, IMPACT. The Consequence field describes the predicted biological effect of the variant, for example `missense_variant`, `intron_variant`, `stop_gained`, or `splice_acceptor_variant`. The IMPACT field gives the predicted severity category, such as HIGH, MODERATE, LOW, or MODIFIER.

4.1.3 Software Tools

The main software tool used for VCF processing and sequence generation was BCFtools. BCFtools is a widely used command-line toolkit for manipulating VCF and BCF files. It supports filtering, querying, sorting, indexing, merging, and

consensus-sequence generation. The official BCFtools project is maintained by the SAMtools/HTSlib group, and the source code is available through the official GitHub repository [26]. Several BCFtools commands were used:

`bcftools view`
Used to extract the ASH1L genomic interval, filter pathological variants, and manage genotype metadata derived from Ensembl release 115.

`bcftools query`
was used to extract specific fields from the VCF files, such as position, reference allele, alternative allele, consequence annotation, and impact.

`bcftools sort`
was used to sort VCF records by coordinate..

`bcftools index`
was used to index compressed VCF files, which is required for efficient coordinate-based access.

`bcftools norm`
was used to check whether the REF alleles in the VCF matched the reference FASTA sequence.

`bcftools consensus`
was used to apply variants from each sample-specific VCF to the ASH1L reference FASTA sequence and generate the final FASTA sequences. According to the BCFtools manual, bcftools consensus creates a consensus sequence by applying VCF variants to a reference FASTA sequence.

The FASTA reference file was indexed using, `samtools faidx`. This creates an .fai index file and allows tools such as BCFtools to access the reference sequence efficiently.

4.1.4 Variant Annotation

After extracting the ASH1L genomic region, the variants were annotated in order to determine their possible biological effects. This annotation step was necessary because the original VCF records mainly describe the genomic position and allelic change of each variant, but they do not directly provide enough information to decide whether a variant is functionally relevant or potentially damaging. Therefore, variant-effect annotation was performed before dividing the variants into functional and pathological groups.

Variant annotation is carried out using Ensembl VEP predicts the consequences of genomic variants by comparing each variant with known gene, transcript, and protein annotations. For each variant, VEP reports whether the change falls in an exon, intron, untranslated region, splice site, upstream/downstream region, or coding sequence, and whether it may affect the encoded protein.

The output of VEP is stored in the INFO/CSQ field of the VCF file. This field contains consequence annotations for each variant. The metadata defined the CSQ field as a structured annotation with the following format:

```
Allele | Consequence | IMPACT | SYMBOL | Gene | Feature_type
| Feature | BIOTYPE | EXON | INTRON | HGVSc | HGVSp |
cDNA_position | CDS_position | Protein_position | Amino_acids
| Codons | Existing_variation | DISTANCE | STRAND | FLAGS |
SYMBOL_SOURCE | HGNC_ID | CANONICAL | ENSP | HGVS_OFFSET
```

Among these subfields, the most important ones for this thesis are, Consequence IMPACT, SYMBOL, Gene, Feature, HGVSc, HGVSp, and CANONICAL.

These annotations were then used to classify variants into two groups. Variants with gene-related or protein-related consequences, such as intronic, UTR, upstream/downstream, synonymous, missense, and in-frame variants, were included in the functional-useful group. Variants with more disruptive predicted consequences, such as frameshift, stop-gained, stop-lost, start-lost, splice donor, splice acceptor, and splice-region variants, were included in the pathological-useful group.

It is important to note that the term “pathological” in this thesis refers to variants selected according to predicted consequence severity, not necessarily to variants confirmed as pathogenic in ClinVar. Some variants may not have a ClinVar pathogenic label but can still be considered predicted damaging candidates based on their VEP consequence. Therefore, the classification used in this study is based primarily on the CSQ consequence and IMPACT fields rather than only on clinical database flags.

In summary, the annotation step transformed the raw ASH1L VCF into a biologically interpretable variant dataset. This made it possible to select variants according to their predicted functional or pathological effect and to generate controlled ASH1L-derived FASTA sequences for downstream sequence-similarity analysis.

4.1.5 Functional and Pathological Variant Filtering

After annotation, the ASH1L variants were divided into two groups: **functional-useful variants** and **pathological-useful variants**. The purpose of this division was to create two biologically meaningful groups of sequences for later comparison using k-mer-based and information-theoretic methods. The functional-useful group contained variants with gene-related or protein-related effects. This group included variants such as:

- missense_variant
- synonymous_variant
- inframe_deletion
- inframe_insertion
- protein_altering_variant
- intron_variant
- 3_prime_UTR_variant
- 5_prime_UTR_variant
- upstream_gene_variant
- downstream_gene_variant

These variants are not necessarily damaging, but they are located in or near the gene and may affect the gene sequence, transcript sequence, protein sequence, or regulatory context. Therefore, they were considered useful for constructing functional sequence variation.

The pathological-useful group contained variants with predicted damaging or severe consequences. This group included:

- frameshift_variant
- stop_gained
- stop_lost
- start_lost
- splice_acceptor_variant
- splice_donor_variant

- splice_region_variant

These variants are more likely to disrupt protein coding, translation, or splicing. Therefore, they were used as predicted pathological or damaging candidate variants. In this thesis, the term "pathological" refers to variants selected by predicted consequence severity, not necessarily to variants confirmed as pathogenic in ClinVar.

The resulting filtered files are, `ASH1L_functional.vcf.gz` with 85,934 variants and `ASH1L_pathological.vcf.gz` with 348 variants. The pathological VCF was much smaller than the functional VCF. Therefore, the number of variants per generated sample was chosen based on the smaller pathological pool. To keep the dataset balanced, five functional samples and five pathological samples were generated, each with exactly 50 selected VCF records.

4.1.6 Creation of Sample-Specific VCF Files

The final dataset contained ten sample-specific VCF files. Five were generated from the functional VCF and five from the pathological VCF. The selected records were randomly shuffled and then divided into non-overlapping groups of 50 records. This ensured that each sample contained exactly 50 selected VCF records, and that variants were not reused within the same class.

4.1.7 Consensus FASTA Generation

The final FASTA sequences were created using `bcftools consensus`. This step applied the selected variants in each sample-specific VCF to the ASH1L reference sequence. Using command `bcftools consensus -s - -f ASH1L_reference.local.fa sample_name.local.vcf.gz > sample_name.fa`, the -f option specifies the reference FASTA file. The -s - option was important in this workflow because the sample-specific VCFs represented artificial sets of selected variants rather than real genotype calls. Using -s - allowed the variants to be applied independently of sample genotype information. Without this option, some FASTA sequences remained identical to the reference because no genotype-based alternate allele was applied.

Each FASTA sequence was created from 50 selected VCF records. During consensus generation, some overlapping variants were skipped by BCFtools, because overlapping variants cannot always be applied simultaneously to the same reference interval. Therefore, although each VCF contained 50 selected records, the number of variants actually applied to the FASTA sequence could be slightly lower in some samples. This was recorded as part of the dataset metadata.

4.2 Weighted KL-Based Measures

The main method examined in this thesis is a weighted KL-based approach for comparing genomic sequences through their k-mer distributions. Standard KL divergence is used as the starting point, while different weighting functions are introduced to control how much each k-mer contributes to the comparison.

The aim is to examine whether common, rare, or sample-specific k-mers should be treated differently. The weighted formulations are not presented as new formal distance metrics. Instead, they are adaptations of KL divergence designed for high-dimensional and sparse genomic data.

For each dataset, the sequences are first converted into k-mer count profiles and then normalized into probability distributions. Smoothing is applied to avoid problems caused by absent k-mers. Pairwise weighted KL values are then computed, symmetrized, and stored in a dissimilarity matrix. The following sections describe the construction of the k-mer distributions, the weighted formulation, and the tested weighting functions.

4.2.1 Construction of K-mer Distributions

Each FASTA sequence was first read as a single continuous string. Header lines were removed, all letters were converted to uppercase, and only K-mers containing A, C, G, and T were kept. For a selected value of K, overlapping K-mers were extracted using a sliding window with a step of one nucleotide. The number of times each K-mer x appeared in sample i was recorded as

$$c_i(x).$$

The total number of valid K-mers in that sample was

$$N_i = \sum_x c_i(x).$$

A common vocabulary was then created by taking the union of all k-mers observed across the samples:

$$V = \bigcup_i V_i,$$

where V_i is the set of K-mers found in sample i. This ensured that all samples were represented in the same feature space. If a K-mer was absent from a sample, its count was set to zero.

The count vectors were normalized to obtain probability distributions:

$$P_i(x) = \frac{c_i(x)}{N_i}.$$

Each genomic sample was therefore represented by a probability distribution over the shared vocabulary V.

The main analysis were carried out using

$$k \in \{15,21,31\}.$$

For the information-content weighting experiment on the ASH1L dataset, an additional value of $k = 41$ was tested to examine whether longer k-mers could capture more specific sequence patterns.

4.2.2 Weighted KL formulation

After constructing the k-mer probability distributions, the samples were compared using a weighted KL-based formulation. For two samples i and j, the directional weighted value was defined as

$$D_w(P_i \parallel P_j) = \sum_{x \in V} w(x)\, P_i(x) \log \frac{P_i(x)}{P_j(x)},$$

where $P_i(x)$ and $P_j(x)$ are the smoothed probabilities of k-mer x, and $w(x)$ controls its contribution to the final value.

When

$$w(x) = 1,$$

the formulation reduces to the standard KL divergence. Other weight functions were used to examine whether common, rare, or sample-specific k-mers should have a larger or smaller influence on the comparison. Since KL divergence is directional, the value from sample i to sample j is not necessarily equal to the value in the opposite direction. A symmetric pairwise dissimilarity was therefore obtained by averaging both directions:

$$D_w^{\mathrm{sym}}(P_i, P_j) = \frac{1}{2}\left[D_w(P_i \parallel P_j) + D_w(P_j \parallel P_i)\right].$$

The resulting values were stored in a symmetric dissimilarity matrix,

$$M_{ij} = D_w^{\mathrm{sym}}(P_i, P_j),$$

with zero values on the diagonal. The weighted formulation is used here as a practical dissimilarity measure for genomic k-mer distributions. It is not treated as a formal distance metric, since it does not necessarily satisfy all metric properties. The weighting functions tested in this thesis are described in the following section.

4.3 Weight Functions

The weighted KL formulation requires a weight $w(x)$ for each k-mer. This weight controls how strongly that k-mer contributes to the final dissimilarity value. The idea behind weighting is that not all k-mers are equally informative. Some are common across most samples and mainly reflect the general sequence background. Others are rarer or appear in only a small group of samples and may carry more useful information for distinguishing them. To study this effect, several weighting strategies were tested. Each one represents a different assumption about which k-mers should receive more importance.

Let $C(x)$ be the total count of k-mer x across all samples. The global probability of x is defined as

$$G(x) = \frac{C(x)}{\sum_{y \in V} C(y)}.$$

For the frequency-based weighting functions, a smoothed global distribution was also used:

$$G^{(\beta)}(x) = \frac{C(x) + \beta}{N_G + \beta|V|},$$

where

$$N_G = \sum_{x \in V} C(x),$$

and β controls the smoothing of the global distribution. Since the vocabulary contains only observed k-mers, $G(x)$ is already positive. The purpose of global smoothing is mainly to reduce extremely small probabilities and prevent rare k-mers from receiving excessively large weights. The following subsections describe the unweighted baseline and the four weighting functions evaluated in this thesis.

4.3.1 Unweighted KL divergence

The first weighting function is the unweighted case:

$$w(x) = 1.$$

Substituting this into the weighted KL formula gives

$$D_w(P_i \parallel P_j) = \sum_{x \in V} P_i(x) \log \frac{P_i(x)}{P_j(x)},$$

which is exactly the standard KL divergence.

This case was included as the baseline. It allows the weighted methods to be compared against the original KL formulation. Without this baseline, it would not be possible to determine whether a weighting function actually improves the separation between samples or simply changes the scale of the divergence values.

The unweighted KL divergence treats all k-mers equally with respect to the weight term. Therefore, the contribution of a k-mer depends only on its probability in the two compared samples. This is useful as a reference, but it does not account for whether a k-mer is globally common or globally rare across the dataset.

4.3.2 Information-content weighting

The second weighting function is based on the global information content of each k-mer:

$$w(x) = -\log G(x).$$

This choice assigns larger weights to rare k-mers and smaller weights to common k-mers. Since $G(x)$ represents the global probability of k-mer x, a small value of $G(x)$ indicates that the k-mer is rare in the dataset. In that case, $-\log G(x)$ becomes larger. On the other hand, if a k-mer is very common, $G(x)$ is larger and the weight becomes smaller.

The motivation for this design is that rare k-mers may carry more discriminative information than very common k-mers. Common k-mers may reflect general sequence composition shared by most samples, while rare k-mers may correspond to more specific sequence patterns. Therefore, this weighting function tests the hypothesis that rare k-mers are more useful for distinguishing between genomic samples.

A useful property of this weighting function is that the logarithm grows slowly. This means that rare k-mers are emphasized, but not as aggressively as in inverse-frequency weighting. Therefore, $w(x) = -\log G(x)$ can be interpreted as a moderate rare-k-mer weighting strategy.

The corresponding weighted divergence is

$$D_{-\log G}(P_i \parallel P_j) = \sum_{x \in V} [-\log G(x)] P_i(x) \log \frac{P_i(x)}{P_j(x)}.$$

This formulation is especially relevant for genomic data because the vocabulary can be very large and sparse, particularly for larger values of k. The logarithmic weight provides a way to increase the influence of rare k-mers while reducing the risk that extremely rare k-mers completely dominate the divergence.

4.3.3 Inverse-frequency weighting

The third weighting function uses the inverse of the global k-mer probability:

$$w(x) = \frac{1}{G(x)}.$$

This weighting function gives very large weights to globally rare k-mers and small weights to globally common k-mers. The corresponding weighted divergence is

$$D_{1/G}(P_i \parallel P_j) = \sum_{x \in V} \frac{1}{G(x)} P_i(x) \log \frac{P_i(x)}{P_j(x)}.$$

The motivation for testing this function is to examine a stronger version of rare-k-mer emphasis. If rare k-mers are highly informative for distinguishing functional and pathological samples, then increasing their contribution could improve the separation between groups.

However, this design also has a risk. If $G(x)$ is very small, then $\frac{1}{G(x)}$ can become very large. As a result, a small number of extremely rare k-mers may dominate the divergence value. This can be useful if those rare k-mers correspond to meaningful biological differences, but it can also amplify noise, sequencing artifacts, or sample-specific events.

Therefore, this weighting function was included not because it is guaranteed to be better, but because it tests an important hypothesis: whether strong emphasis on rare k-mers improves genomic sample separation. Its performance must be evaluated empirically using the resulting distance matrices and silhouette scores.

4.3.4 Inverse square-root weighting

The fourth weighting function is a softer version of inverse-frequency weighting:

$$w(x) = \frac{1}{\sqrt{G(x)}}.$$

The corresponding divergence is

$$D_{1/\sqrt{G}}(P_i \parallel P_j) = \sum_{x \in V} \frac{1}{\sqrt{G(x)}} P_i(x) \log \frac{P_i(x)}{P_j(x)}.$$

This function also gives larger weights to rare k-mers, but the square root reduces the strength of the weighting compared with $\frac{1}{G(x)}$. Therefore, it represents

a compromise between the moderate logarithmic weighting and the stronger inverse-frequency weighting.

The motivation for this design is that rare k-mers may be informative, but over-emphasizing them may make the divergence unstable or noisy. The inverse square-root weight allows rare k-mers to contribute more than common k-mers, while limiting the extreme amplification produced by the full inverse-frequency weight.

This weighting function is useful for testing whether a balanced rare-k-mer emphasis provides better separation than either the unweighted KL divergence or the more aggressive inverse-frequency version.

4.3.5 IDF-Weighted KL

A further weighting strategy is inspired by the term frequency–inverse document frequency framework. In this analogy, each genomic sample is treated as a document, and each k-mer is treated as a term. The purpose of this weighting is to reduce the influence of k-mers that appear in almost all samples and increase the influence of k-mers that appear in fewer samples.

Let n be the total number of samples and let $df(x)$ be the number of samples in which k-mer x appears at least once:

$$df(x) = |\{i : c_i(x) > 0\}|.$$

The inverse document frequency of k-mer x is computed as

$$idf(x) = \log\left(\frac{n+1}{df(x)+1}\right) + 1.$$

The TF-IDF-based weight is then defined as

$$w(x) = idf(x).$$

The weighted KL divergence becomes

$$D_{TFIDF}(P_i \parallel P_j) = \sum_{x \in V} idf(x) P_i(x) \log \frac{P_i(x)}{P_j(x)}.$$

The motivation for this design is slightly different from the global-frequency weights. The functions $-\log G(x)$, $\frac{1}{G(x)}$, and $\frac{1}{\sqrt{G(x)}}$ use the total global frequency of each k-mer. In contrast, TF-IDF uses the number of samples in which the k-mer

appears. Therefore, a k-mer that appears in nearly all samples receives a low weight, while a k-mer that appears in only a few samples receives a higher weight.

This is useful because a k-mer may have a low total frequency but still appear across many samples, or it may be concentrated in only a small subset of samples. TF-IDF weighting captures this sample-level distribution and is therefore suitable for evaluating whether group-specific k-mers improve sample separation.

4.3.6 Summary of tested weighting functions

The tested weighting functions are summarized as follows:

- Baseline Unweighted Case:
$$w(x) = 1$$
- Information-Content Weight:
$$w(x) = -\log G(x)$$
- Inverse-Frequency Weight:
$$w(x) = \frac{1}{G(x)}$$
- Inverse Square-Root Weight:
$$w(x) = \frac{1}{\sqrt{G(x)}}$$
- IDF-Weighted KL:
$$w(x) = idf(x)$$

Each function represents a different assumption about the importance of k-mers. The unweighted case provides the baseline. The logarithmic information-content weight moderately emphasizes rare k-mers. The inverse-frequency weight strongly emphasizes rare k-mers. The inverse square-root weight provides an intermediate alternative. The TF-IDF weight emphasizes k-mers that are not widely shared across samples.

The purpose of comparing these functions is to identify which weighting strategy produces the clearest separation between genomic sample groups. Therefore, the weight functions are evaluated empirically using pairwise dissimilarity matrices, and silhouette-based separation scores.

4.3.7 Smoothing Parameters and Sensitivity Testing

Smoothing was applied because k-mer distributions are often sparse. A k-mer may appear in one sample but be absent from another, which creates a problem for KL-based calculations. For each sample, the smoothed probability was defined as

$$P_i^{(\alpha)}(x) = \frac{c_i(x) + \alpha}{N_i + \alpha|V|},$$

where α controls the amount of smoothing. When $\alpha > 0$, every k-mer receives a small positive probability. For the frequency-based weight functions, the global distribution was also smoothed:

$$G^{(\beta)}(x) = \frac{C(x) + \beta}{N_G + \beta|V|}.$$

Here, β reduces the effect of extremely rare k-mers by preventing their global probabilities from becoming too small. The experiments tested different values of k, α, β, and $w(x)$. For each parameter setting, a pairwise dissimilarity matrix was computed and evaluated using overall and per-sample Silhouette Scores. The final comparison considered both the numerical scores and the structure of the resulting distance matrices.

4.4 Evaluation procedure using Silhouette Score

In this thesis, the Silhouette Score was used to quantify the separation between the predefined genomic sample groups after computing the KL-based pairwise dissimilarity matrices. It provides a numerical measure of how well the samples are separated according to their assigned labels. For each tested KL-based formulation, a symmetric dissimilarity matrix was first constructed:

$$M_{ij} = D_w^{sym}(P_i, P_j).$$

This matrix was then used as a precomputed distance matrix for silhouette evaluation. The score was computed for each combination of k-mer length, smoothing parameters, and weight function:

$$k, \quad \alpha, \quad \beta, \quad w(x).$$

The overall Silhouette Score was used to compare the different KL-based variants. In addition, the individual silhouette values of each sample were examined in order to identify which samples were clearly assigned to their group and which samples behaved ambiguously.

Therefore, the evaluation included both the overall Silhouette Score and per-sample silhouette values. The overall score was used for method comparison, while the per-sample values were used to support the interpretation of individual samples in the results section.

4.5 Jensen–Shannon Distance Computation

The complete Jensen–Shannon analysis followed these steps:

1. Read each FASTA file and extract the sequence.
2. Generate overlapping k-mers for each sample.
3. Count the frequency of each k-mer.
4. Construct a global k-mer vocabulary across all samples.
5. Convert each sample's k-mer count vector into a probability distribution.
6. Compute pairwise Jensen–Shannon divergence between all sample pairs.
7. Convert divergence values into Jensen–Shannon distances using the square root.
8. Build a symmetric pairwise distance matrix.
9. Compute overall and per-sample silhouette scores using the precomputed distance matrix.
10. Repeat the analysis for $k = 15$, $k = 21$, and $k = 31$.

4.5.1 Construction of the global K-mer vocabulary

After counting k-mers in each sample, a global k-mer vocabulary was constructed. This vocabulary was defined as the union of all distinct k-mers observed in all samples:

$$V = \bigcup_{i=1}^{N} V_i \tag{4.1}$$

where V_i is the set of k-mers observed in sample i, and N is the number of samples.

Using a shared global vocabulary is necessary because all samples must be represented in the same feature space before pairwise distances can be computed. Therefore, each sample was represented as a vector whose length is equal to the number of distinct k-mers in the entire dataset.

For each sample i, the count vector was defined as:

$$C_i = [c_i(x_1), c_i(x_2), \ldots, c_i(x_m)] \tag{4.2}$$

where $x_1, x_2, \ldots, x_m$ are the k-mers in the global vocabulary, m is the vocabulary size, and $c_i(x)$ is the number of occurrences of k-mer x in sample i. If a k-mer from the global vocabulary did not appear in a specific sample, its count for that sample was set to zero.

4.5.2 Conversion from K-mer counts to probability distributions

Jensen–Shannon divergence compares probability distributions, not raw count vectors. Therefore, each k-mer count vector was normalized into a probability distribution.

For sample i, the probability of k-mer x was computed as:

$$P_i(x) = \frac{c_i(x)}{\sum_{x \in V} c_i(x)} \tag{4.3}$$

where:

$$\sum_{x \in V} P_i(x) = 1 \tag{4.4}$$

Thus, each sample was represented as a k-mer probability distribution over the global vocabulary.

No smoothing was applied in the standard Jensen–Shannon computation. This is because Jensen–Shannon divergence is naturally more stable than KL divergence when zero probabilities occur. If a k-mer appears in one sample but not in another, the midpoint distribution still assigns a non-zero probability to that k-mer, preventing the divergence from becoming infinite.

4.5.3 Pairwise Jensen–Shannon divergence

For every pair of samples i and j, their probability distributions P_i and P_j were compared using Jensen–Shannon divergence.

First, the midpoint distribution was computed:

$$M_{ij}(x) = \frac{1}{2}\left(P_i(x) + P_j(x)\right) \tag{4.5}$$

Then, the Jensen–Shannon divergence was calculated as:

$$D_{JS}(P_i, P_j) = \frac{1}{2} \sum_{x \in V} P_i(x) \log_2 \frac{P_i(x)}{M_{ij}(x)} + \frac{1}{2} \sum_{x \in V} P_j(x) \log_2 \frac{P_j(x)}{M_{ij}(x)} \tag{4.6}$$

Terms with zero probability were handled according to the standard convention:

$$0 \log \frac{0}{q} = 0 \tag{4.7}$$

This means that k-mers absent from a sample do not contribute to the divergence term for that sample.

Logarithm base 2 was used, so the Jensen–Shannon divergence is bounded between 0 and 1:

$$0 \leq D_{JS}(P_i, P_j) \leq 1 \tag{4.8}$$

A value close to 0 indicates that two samples have very similar k-mer probability distributions, while a larger value indicates greater dissimilarity.

4.5.4 Jensen–Shannon distance matrix

Although Jensen–Shannon divergence is symmetric and bounded, the square root of the divergence was used for distance-based analysis:

$$d_{JS}(P_i, P_j) = \sqrt{D_{JS}(P_i, P_j)} \tag{4.9}$$

This quantity is called the Jensen–Shannon distance.

For each value of k, a pairwise distance matrix was constructed:

$$D = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1N} \\ d_{21} & 0 & d_{23} & \cdots & d_{2N} \\ d_{31} & d_{32} & 0 & \cdots & d_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & d_{N3} & \cdots & 0 \end{bmatrix} \tag{4.10}$$

where $d_{ij} = d_{JS}(P_i, P_j)$. The diagonal entries are zero because each sample is identical to itself. The matrix is symmetric because Jensen–Shannon distance satisfies:

$$d_{JS}(P_i, P_j) = d_{JS}(P_j, P_i) \tag{4.11}$$

This distance matrix was used for visualization and quantitative evaluation.

4.6 Cosine Similarity-Based K-mer Comparison

Cosine similarity was used as a vector-based baseline for comparing the genomic samples. Each FASTA sequence was converted into a normalized k-mer frequency vector defined over the shared vocabulary.

For two samples represented by vectors X_i and X_j, cosine similarity was calculated as

$$\text{cosine}(X_i, X_j) = \frac{X_i \cdot X_j}{\|X_i\|\|X_j\|}.$$

Because the vectors contain non-negative frequencies, values close to 1 indicate similar k-mer composition. For clustering and silhouette analysis, similarity was converted into cosine distance:

$$d_{\text{cosine}}(X_i, X_j) = 1 - \text{cosine}(X_i, X_j).$$

The analysis was repeated for the same main k-mer lengths used by the other methods. For each setting, a pairwise distance matrix and overall and per-sample Silhouette Scores were produced. A length-weighted version was also tested to check whether small differences in sequence length affected the results. Since the ASH1L-derived samples had almost the same length, this correction produced little change.

4.7 KWIP Entropy-Weighted

In addition to divergence-based measures, an exact KWIP-style weighted inner product was computed to compare the *ASH1L* consensus sequences. The original KWIP method computes genetic similarity from k-mer counts using an entropy-weighted inner product and then converts the resulting kernel matrix into a distance matrix. In the original implementation, k-mers are stored in probabilistic sketches for computational efficiency on large sequencing-read datasets. In this thesis, because the dataset consisted of a limited number of consensus FASTA sequences, exact k-mer count vectors were used instead of sketches. This allowed the same entropy-weighted similarity principle to be applied without hash collisions or sketch approximation.

The same k values were used as in the KL divergence, Jensen–Shannon divergence, and cosine similarity analysis, allowing direct comparison between methods. K-mers containing ambiguous characters other than A, C, G, and T were excluded. For each value of k, a global vocabulary was created from all distinct k-mers observed in the ten samples. Each sample was then represented as a count vector:

$$C_i = [c_i(x_1), c_i(x_2), \ldots, c_i(x_m)] \tag{4.12}$$

where $c_i(x)$ is the count of k-mer x in sample i, and m is the number of distinct k-mers in the global vocabulary.

For each k-mer, its presence frequency across the sample set was calculated as:

$$F(x) = \frac{1}{N} \sum_{i=1}^{N} I(c_i(x) > 0) \tag{4.13}$$

where N is the number of samples and $I(c_i(x) > 0)$ is an indicator function equal to 1 if k-mer x is present in sample i, and 0 otherwise. This frequency was then converted into a Shannon entropy weight:

$$H(x) = -\left[F(x) \log_2 F(x) + (1 - F(x)) \log_2 (1 - F(x))\right] \tag{4.14}$$

For $F(x) = 0$ or $F(x) = 1$, the entropy weight was explicitly set to zero. Therefore, k-mers present in all samples received no weight, while k-mers present in an intermediate proportion of samples received higher weights. The weighted inner product between each pair of samples was then computed as:

$$K_{ij} = \sum_{x} c_i(x) c_j(x) H(x) \tag{4.15}$$

where K_{ij} is the weighted similarity between samples i and j. This produced a KWIP-style kernel matrix. Following the KWIP formulation, the kernel matrix was normalized using the self-similarity values:

$$K'_{ij} = \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}} \tag{4.16}$$

The normalized kernel matrix was then converted into a distance matrix:

$$D_{ij} = \sqrt{K'_{ii} + K'_{jj} - 2K'_{ij}} \tag{4.17}$$

Since the diagonal values of the normalized kernel are equal to 1, this simplifies to:

$$D_{ij} = \sqrt{2 - 2K'_{ij}} \tag{4.18}$$

This pipeline produced one KWIP-style distance matrix for each k-mer length. The resulting distance matrices were evaluated using the same procedure applied to the other sequence comparison methods. Finally, the separation between functional and pathological samples was quantified using the silhouette score with the precomputed KWIP distance matrix, generating both global and per-sample silhouette profiles.

This method therefore represents an exact adaptation of the weighted inner product component of KWIP to the *ASH1L* consensus FASTA dataset. The original sketching component was not used, because exact k-mer counting was computationally feasible for the present dataset and allowed a cleaner comparison with the other k-mer-based measures used in this thesis.

Chapter 5

Results and Discussion

5.1 Overview of the Experimental Evaluation

This chapter presents the results obtained on the two datasets used in this thesis: the controlled ASH1L-derived sequences and the complete Brucella genomes. The evaluated methods include divergence-based, vector-based, and entropy-weighted approaches, together with the weighted kl-based formulations. For each method, a pairwise dissimilarity matrix was generated and evaluated using overall and per-sample Silhouette Scores.

The ASH1L dataset was used as a controlled gene-level experiment. Since all samples were derived from the same genomic region, the main aim was to test whether small differences in k-mer composition could be detected. The Brucella dataset was used as a larger whole-genome experiment. It allowed the same methods to be tested on naturally occurring genomes with a more complex class structure. The results are discussed not only in terms of the highest score, but also by considering the effect of k-mer length, smoothing, weighting strategy, and dataset structure.

5.2 Standard KL Baseline Scores

The standard KL formulation was first evaluated as a baseline before applying the weighting functions and the results are presented in the following table.

Table 5.1: Sensitivity analysis of the standard KL baseline with respect to k-mer length and smoothing parameter α.

k	α	**Vocab. size**	**Silhouette score**
15	0.001	199251	0.0463
	0.01		0.0464
	0.1		0.0464
	1		0.0468
21	0.001	219315	0.0545
	0.01		0.0545
	0.1		0.0547
	1		0.0555
31	0.001	236913	0.0659
	0.01		0.0660
	0.1		0.0661
	1		**0.0666**

Table 5.1 shows that the Silhouette Score improves as the k-mer length increases. The best result is obtained at $k = 31$ with $\alpha = 1$. In contrast, changing the smoothing parameter produces only small differences within each fixed value of k.

Overall, the results suggest that k-mer length has a stronger effect than smoothing. However, the scores remain low, indicating that the standard KL baseline provides only weak separation on the ASH1L dataset.

5.3 Weighted-KL Functions Scores

5.3.1 Information-content weighting Scores

Table 5.2: Sensitivity analysis of the weighted KL method using weight function $w(x) = -\log G(x)$, evaluated across variations in k-mer length, smoothing parameters α, β.

k	**Vocab. size**	α	β	**Silhouette score**
15	199251	0.001	1	0.04405
		0.01	1	0.04406
		0.1	1	0.04411
		1	1	0.04439

Table 5.2: Sensitivity analysis of the weighted KL method using weight function $w(x) = -\log G(x)$, evaluated across variations in k-mer length, smoothing parameters α, β.

k	**Vocab. size**	α	β	**Silhouette score**
21	219315	0.001	1	0.05188
		0.01	1	0.05194
		0.1	1	0.05211
		1	1	0.05273
31	236913	0.001	1	0.06297
		0.01	1	0.06303
		0.1	1	0.06318
		1	1	**0.06356**
15	199251	0.001	0.001	0.04315
			0.01	0.04316
			0.1	0.04327
			1	0.04405
		0.01	0.001	0.04316
			0.01	0.04317
			0.1	0.04329
			1	0.04406
		0.1	0.001	0.04321
			0.01	0.04323
			0.1	0.04334
			1	0.04411
		1	0.001	0.04350
			0.01	0.04351
			0.1	0.04363
			1	0.04439
21	219315	0.001	0.001	0.05086
			0.01	0.05087
			0.1	0.05100
			1	0.05188
		0.01	0.001	0.05092
			0.01	0.05094
			0.1	0.05107

Table 5.2: Sensitivity analysis of the weighted KL method using weight function $w(x) = -\log G(x)$, evaluated across variations in k-mer length, smoothing parameters α, β.

k	**Vocab. size**	α	β	**Silhouette score**
			1	0.05194
		0.1	0.001	0.05109
			0.01	0.05110
			0.1	0.05123
			1	0.05211
		1	0.001	0.05171
			0.01	0.05172
			0.1	0.05185
			1	0.05274
31	236913	0.001	0.001	0.06179
			0.01	0.06181
			0.1	0.06196
			1	0.06298
		0.01	0.001	0.06185
			0.01	0.06187
			0.1	0.06202
			1	0.06304
		0.1	0.001	0.06199
			0.01	0.06201
			0.1	0.06216
			1	0.06318
		1	0.001	0.06238
			0.01	0.06239
			0.1	0.06255
			1	**0.06357**
41	245243	0.001	0.001	0.06880
			0.01	0.06881
			0.1	0.06898
			1.0	0.07009
		0.01	0.001	0.06880
			0.01	0.06881

Table 5.2: Sensitivity analysis of the weighted KL method using weight function $w(x) = -\log G(x)$, evaluated across variations in k-mer length, smoothing parameters α, β.

k	**Vocab. size**	α	β	**Silhouette score**
			0.1	0.06898
			1.0	0.07009
		0.1	0.001	0.06880
			0.01	0.06882
			0.1	0.06898
			1.0	0.07009
		1.0	0.001	0.06880
			0.01	0.06882
			0.1	0.06899
			1.0	**0.07010**

For the first part of the experiment, where β is fixed at 1, the results show a consistent increase in the Silhouette score as the k-mer length becomes larger. At $k = 15$, the scores remain around 0.044, while increasing k to 21 improves the score to approximately 0.052. The best result in this subset is obtained at $k = 31$, where the Silhouette score reaches 0.06356 for $\alpha = 1$. Similar to the unweighted kl baseline, the effect of α is relatively small within each fixed value of k. For example, at $k = 31$, increasing α from 0.001 to 1 changes the score only from 0.06297 to 0.06356. This suggests that the k-mer length has a stronger influence on sample separation than the smoothing parameter.

When the full α-β sensitivity analysis is considered, the same general pattern is observed. For all tested k-mer lengths, larger values of β tend to produce slightly higher Silhouette scores. This indicates that stronger weighting can improve the separation of the samples, although the improvement remains gradual rather than dramatic. The highest result is obtained at $k = 41$, $\alpha = 1.0$, and $\beta = 1.0$, with a Silhouette score of 0.07010. This is higher than the best score obtained for $k = 31$, suggesting that extending the k-mer length beyond 31 can capture more specific sequence information and slightly improve clustering performance.

However, the absolute Silhouette scores are still low and close to zero. Therefore, although the weighted formulation improves the result slightly, especially at larger k and higher β, the overall separation between the sample groups remains weak. This suggests that the method can emphasize informative k-mers to some extent, but the dataset itself may limit how clearly the samples can be separated.

Main observation. In this weighted KL setting, increasing the k-mer length has the strongest effect on the Silhouette score, while changes in α and β produce smaller improvements. The best result is obtained at $k = 41$, $\alpha = 1.0$, and $\beta = 1.0$, with a Silhouette score of 0.07010. Although this shows a slight improvement over shorter k-mer settings, the low absolute score indicates that the separation between sample groups remains limited.

5.3.2 Inverse-frequency weighting Scores

Table 5.3: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/G(x)$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
15	199251	0.001	1	0.01836
		0.01	1	0.01837
		0.1	1	0.01839
		1	1	0.01847
21	219315	0.001	1	0.02265
		0.01	1	0.02267
		0.1	1	0.02270
		1	1	0.02282
31	236913	0.001	1	0.02917
		0.01	1	0.02906
		0.1	1	0.02902
		1	1	0.02900
41	245243	0.001	1	0.03299
		0.01	1	0.03299
		0.1	1	0.03299
		1	1	0.03300
15	199251	0.001	0.001	0.01127
			0.01	0.01135
			0.1	0.01211
			1	0.01836
		0.01	0.001	0.01128
			0.01	0.01136
			0.1	0.01212

Table 5.3: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/G(x)$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
			1	0.01837
		0.1	0.001	0.01130
			0.01	0.01138
			0.1	0.01214
			1	0.01839
		1	0.001	0.01136
			0.01	0.01144
			0.1	0.01221
			1	0.01847
21	219315	0.001	0.001	0.01440
			0.01	0.01449
			0.1	0.01539
			1	0.02265
		0.01	0.001	0.01441
			0.01	0.01450
			0.1	0.01540
			1	0.02267
		0.1	0.001	0.01443
			0.01	0.01452
			0.1	0.01542
			1	0.02270
		1	0.001	0.01451
			0.01	0.01460
			0.1	0.01551
			1	0.02282
31	236913	0.001	0.001	0.01900
			0.01	0.01911
			0.1	0.02021
			1	0.02900
		0.01	0.001	0.01901
			0.01	0.01912

Table 5.3: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/G(x)$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
			0.1	0.02022
			1	0.02902
		0.1	0.001	0.01904
			0.01	0.01915
			0.1	0.02025
			1	0.02906
		1	0.001	0.01911
			0.01	0.01922
			0.1	0.02032
			1	**0.02917**

Table 5.3 presents the sensitivity analysis of the weighted KL formulation using the inverse global-frequency weight function ($w(x) = 1/G(x)$). This weighting strategy gives larger importance to k-mers with lower global frequency and reduces the influence of highly frequent k-mers. The purpose of this experiment is to examine whether emphasizing rare k-mers can improve the separation between the genomic sample groups.

The results show that the Silhouette score increases as the k-mer length becomes larger. For $k = 15$, the best score is 0.01847, while for $k = 21$, it increases to 0.02282. The highest score in this section is obtained at $k = 31$, $\alpha = 1$, and $\beta = 1$, with a Silhouette score of 0.02917. This indicates that longer k-mers provide more useful sequence-specific information than shorter k-mers, even when inverse-frequency weighting is applied.

The effect of the smoothing parameter α is relatively small. For each fixed value of k and β, changing α produces only minor changes in the Silhouette score. This suggests that α mainly supports numerical stability in the kl calculation, rather than strongly affecting the separation between samples.

In contrast, β has a clearer effect on the results. Increasing β generally improves the Silhouette score, showing that stronger application of the inverse-frequency weighting leads to better separation within this formulation. For example, at $k = 31$ and $\alpha = 1$, the score increases from 0.01911 when $\beta = 0.001$ to 0.02917 when $\beta = 1$. However, the overall improvement remains limited.

Main observation. For the weighted KL formulation using $w(x) = 1/G(x)$, the best result is obtained at $k = 31$, $\alpha = 1$, and $\beta = 1$, with a Silhouette score of 0.02917. Although increasing k and β improves the score, the low absolute value indicates weak separation between the sample groups. Compared with the unweighted KL baseline, the inverse-frequency weighting strategy produces lower Silhouette scores, indicating that emphasizing low-probability k-mers is not more effective for these samples.

5.3.3 Inverse square-root weighting Scores

Table 5.4: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/\sqrt{G(x)}$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
15	199251	0.001	1	0.03105
		0.01	1	0.03107
		0.1	1	0.03111
		1	1	0.03126
21	219315	0.001	1	0.03717
		0.01	1	0.03720
		0.1	1	0.03728
		1	1	0.03756
31	236913	0.001	1	0.04605
		0.01	1	0.04608
		0.1	1	0.04616
		1	1	**0.04639**
15	199251	0.001	0.001	0.02536
			0.01	0.02544
			0.1	0.02614
			1	0.03105
		0.01	0.001	0.02538
			0.01	0.02545
			0.1	0.02616
			1	0.03107
		0.1	0.001	0.02543
			0.01	0.02550

Table 5.4: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/\sqrt{G(x)}$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
			0.1	0.02620
			1	0.03111
		1	0.001	0.02559
			0.01	0.02566
			0.1	0.02636
			1	0.03126
21	219315	0.001	0.001	0.03065
			0.01	0.03073
			0.1	0.03154
			1	0.03717
		0.01	0.001	0.03068
			0.01	0.03076
			0.1	0.03157
			1	0.03720
		0.1	0.001	0.03075
			0.01	0.03084
			0.1	0.03165
			1	0.03728
		1	0.001	0.03100
			0.01	0.03109
			0.1	0.03190
			1	0.03756
31	236913	0.001	0.001	0.03833
			0.01	0.03843
			0.1	0.03939
			1	0.04605
		0.01	0.001	0.03836
			0.01	0.03846
			0.1	0.03942
			1	0.04608
		0.1	0.001	0.03844
			0.1	0.03945

Table 5.4: Sensitivity analysis of the weighted KL baseline using weight function $w(x) = 1/\sqrt{G(x)}$, evaluated across variations in k-mer length, smoothing parameters α, and β.

k	**Vocab. size**	α	β	**Silhouette score**
			0.01	0.03853
			0.1	0.03950
			1	0.04616
		1	0.001	0.03863
			0.01	0.03873
			0.1	0.03969
			1	**0.04639**

Table 5.4 presents the results of the weighted KL formulation using $w(x) = 1/\sqrt{G(x)}$. This function increases the contribution of k-mers with lower global probability, but the square-root term makes this effect less aggressive than $w(x) = 1/G(x)$. The results show that the Silhouette score increases with k-mer length: the best score is 0.03126 for $k = 15$, 0.03756 for $k = 21$, and 0.04639 for $k = 31$.

The effect of α is small compared with the effect of k and β. Within each fixed value of k and β, increasing α only slightly changes the Silhouette score. In contrast, increasing β gives a clearer improvement, showing that stronger use of the weighting function increases the score within this setting. The highest result is obtained at $k = 31$, $\alpha = 1$, and $\beta = 1$, with a Silhouette score of 0.04639.

Key takeaway. The moderated inverse-frequency weighting $w(x) = 1/\sqrt{G(x)}$ performs better than the direct inverse weighting $w(x) = 1/G(x)$, increasing the best score from 0.02917 to 0.04639. However, it remains lower than the unweighted kl baseline score of 0.0666, suggesting that the unweighted k-mer probability distribution preserves more useful separation information for the tested samples.

5.3.4 TF-IDF weighting Scores

Table 5.5: Summary of results for TF–IDF KL method with varying k.

k	**Vocabulary size**	**Mean** *df*	**Mean IDF**	**Silhouette Score**
15	199251	9.6868	1.0566	0.0299
21	219315	9.5814	1.0756	0.0359
31	236913	9.4128	1.1059	**0.0447**

Table 5.5 reports the results of the TF–IDF KL method for different k-mer lengths. As k increases from 15 to 31, the vocabulary size increases from 199251 to 236913, while the mean document frequency decreases from 9.6868 to 9.4128. This means that longer k-mers become more sample-specific and appear in slightly fewer samples on average. At the same time, the mean IDF increases from 1.0566 to 1.1059, showing that longer k-mers receive higher inverse-document-frequency weights.

The Silhouette score also increases consistently with k. The score is 0.0299 for $k = 15$, increases to 0.0359 for $k = 21$, and reaches its highest value of 0.0447 for $k = 31$. This suggests that longer k-mers provide more informative patterns for distinguishing the samples when TF–IDF normalization is applied. The increase in IDF also supports this trend, because higher IDF values give more importance to k-mers that are less uniformly shared across all samples.

Key Takeaway. The TF–IDF KL method benefits from increasing the k-mer length, with the best result obtained at $k = 31$ and a Silhouette score of 0.0447. This result is higher than the direct inverse-frequency weighting $w(x) = 1/G(x)$, but slightly lower than the moderated inverse-frequency weighting $w(x) = 1/\sqrt{G(x)}$ and lower than the unweighted KL baseline. Therefore, TF–IDF weighting captures some sample-specific k-mer information, but it does not outperform the best KL-based result in this experiment.

5.4 Jensen–Shannon Divergence Results

Table 5.6: Global silhouette scores and vocabulary sizes for different k-mer configurations using Jensen–Shannon distance.

k	Vocabulary size	Silhouette Score
15	199,251	0.023899
21	219,315	0.028934
31	236,913	0.035676

The results in Table 5.6 show that the Silhouette score increases as the k-mer length becomes larger. The lowest score is obtained at $k = 15$ with a value of 0.023899, while the highest score is obtained at $k = 31$ with a value of 0.035676. This trend suggests that longer k-mers provide slightly more informative sequence representations for this dataset, because they can capture more specific local sequence patterns than shorter k-mers.

However, although $k = 31$ gives the best result among the tested configurations, the absolute Silhouette score remains close to zero. This indicates that the standard Jensen–Shannon distance captures some differences between the samples, but it does not provide strong separation between the functional and pathological groups. Therefore, the improvement from $k = 15$ to $k = 31$ is meaningful as a trend, but limited in magnitude.

Since $k = 31$ gave the highest silhouette score, the corresponding distance matrix was inspected in more detail. The average distances between sample groups are presented in Table 5.7.

Table 5.7: Mean Jensen–Shannon distances by sample pair type for $k = 31$.

Pair type	Mean Jensen–Shannon distance
Functional–functional	0.114672
Pathological–pathological	0.098531
Functional–pathological	0.110465

The pathological samples showed the lowest average within-class distance. This indicates that the pathological samples formed a relatively compact group under Jensen–Shannon distance. This explains why the functional samples had negative silhouette values. They were not closer to their own class than to the opposite class. The per-sample silhouette values for $k = 31$ are outlined in Table 5.8.

Table 5.8: Per-sample silhouette values for $k = 31$ under Jensen–Shannon distance.

Sample	Class	Silhouette Score
sample 01	functional	-0.036919
sample 02	functional	-0.035316
sample 03	functional	-0.035616
sample 04	functional	-0.034215
sample 05	functional	-0.041371
sample 06	pathological	0.104175
sample 07	pathological	0.107264
sample 08	pathological	0.107555
sample 09	pathological	0.111435
sample 10	pathological	0.109767

All functional samples obtained negative silhouette values, while all pathological

samples obtained positive values. This shows that Jensen–Shannon distance mainly captured the internal similarity of the pathological group, but it did not form a compact functional group.

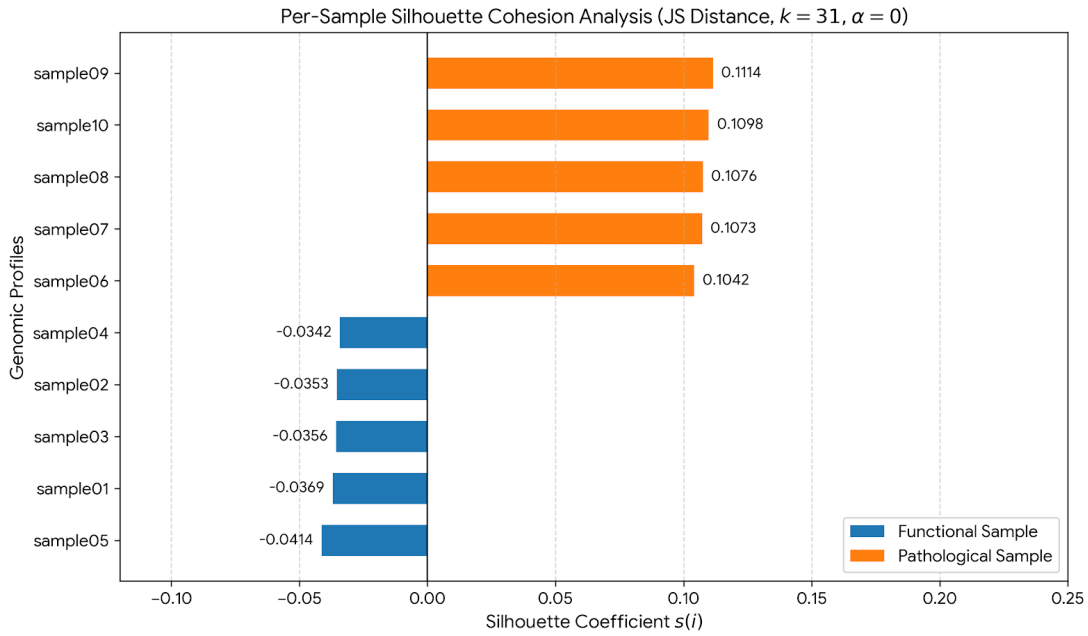

Figure 5.1: Overall Silhouette Score performance as a function of k-mer length (k) using Jensen-Shannon distance metric.

Key takeaway. The Jensen–Shannon distance performs best at $k = 31$, with a Silhouette score of 0.035676. Increasing the k-mer length improves the score, but the low absolute value shows that the standard Jensen–Shannon distance alone is not sufficient to clearly separate the two sample groups.

5.5 Cosine Similarity Comparison Results

Table 5.9: Global Silhouette Scores obtained using cosine distance.

k	Vocabulary size	Global Silhouette Score
15	199,251	0.0526
21	219,315	0.0603
31	236,913	0.0672

The global silhouette score increased slightly as k increased. The highest value was obtained for $k = 31$, with a score of approximately:

$$S = 0.0672 \tag{5.1}$$

However, this value is still close to zero. Therefore, cosine distance produced only weak separation between the functional and pathological groups. This means that the two classes are not clearly separated in the cosine-based k-mer feature space.

Although the score improves from $k = 15$ to $k = 31$, the improvement is small. This suggests that increasing k-mer length makes the representation slightly more informative, but cosine similarity alone is not sufficient to strongly distinguish the two biological groups.

5.5.1 Per-Sample Silhouette Scores

The per-sample silhouette scores show a clear difference between the two classes. The functional samples have negative silhouette scores for all tested k values, while the pathological samples have positive silhouette scores. Mean values by class are summarized in Table 5.10.

Table 5.10: Mean per-class Silhouette Scores using cosine distance.

k	Functional mean Silhouette	Pathological mean Silhouette
15	-0.0601	0.1652
21	-0.0685	0.1891
31	-0.0794	0.2138

The pathological group consistently shows positive silhouette scores, and the values increase with larger k. This indicates that the pathological samples form a relatively compact group under cosine distance.

In contrast, all functional samples have negative silhouette scores. A negative silhouette score means that a sample is, on average, closer to samples from the opposite class than to samples from its own assigned class. Therefore, the functional samples do not form a coherent cluster using cosine-based k-mer representation.

This pattern is visible in the per-sample silhouette plots for $k = 15$, $k = 21$, and $k = 31$. In all three plots, the first five samples, corresponding to the functional group, are below zero, while the five pathological samples are above zero.

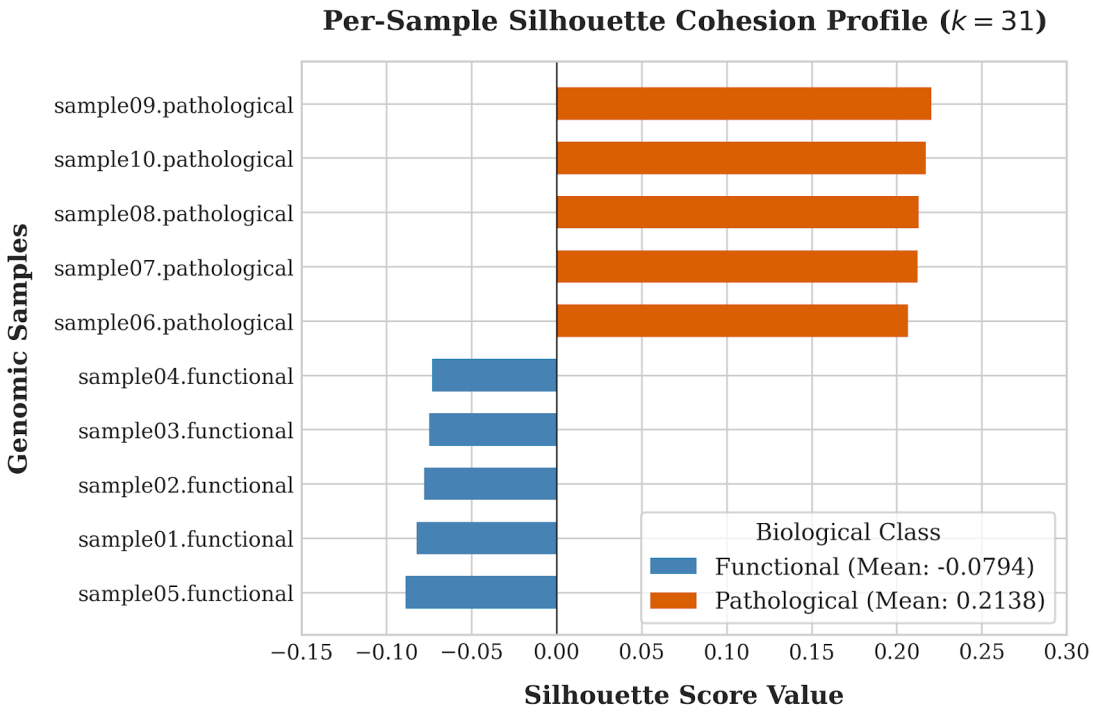

Figure 5.2: Per-sample Silhouette Scores using cosine distance for (a) $k = 15$, (b) $k = 21$, and (c) $k = 31$. The functional samples show negative silhouette scores for all k-mer lengths, whereas the pathological samples show positive scores.

5.5.2 Pairwise Distance Structure

To better understand why the functional samples have negative silhouette scores, the average cosine distances were compared within and between the two classes (Table 5.11).

Table 5.11: Mean pairwise cosine distances within and between classes.

k	Within functional	Within pathological	Between groups
15	0.001642	0.001289	0.001544
21	0.005271	0.003982	0.004910
31	0.012342	0.008932	0.011362

For every tested value of k, the pathological–pathological distance is the smallest. This confirms that the pathological samples are internally more similar to each other.

The critical observation is that the functional–pathological distance is smaller than the functional–functional distance:

$$d_{\text{functional-pathological}} < d_{\text{functional-functional}} \tag{5.2}$$

For example, at $k = 31$:

$$d_{\text{functional-functional}} = 0.012342 \tag{5.3}$$

$$d_{\text{functional-pathological}} = 0.011362 \tag{5.4}$$

This means that, on average, functional samples are closer to pathological samples than to other functional samples. This explains why the functional samples have negative silhouette scores. Therefore, cosine distance identifies a compact pathological group, but it does not identify the functional group as a separate cluster.

5.5.3 Best K-mer Length Selection

Among the tested k-mer lengths, $k = 31$ gave the highest global silhouette score. The detailed per-sample metric breakdowns for this optimal configuration are provided in Table 5.12.

Table 5.12: Per-sample Silhouette Scores for the best cosine result ($k = 31$).

Sample	Class	Silhouette Score
sample01.functional	Functional	-0.0823
sample02.functional	Functional	-0.0778
sample03.functional	Functional	-0.0748
sample04.functional	Functional	-0.0733
sample05.functional	Functional	-0.0889
sample06.pathological	Pathological	0.2067
sample07.pathological	Pathological	0.2123
sample08.pathological	Pathological	0.2128
sample09.pathological	Pathological	0.2203
sample10.pathological	Pathological	0.2171

At $k = 31$, the pathological samples have silhouette scores between approximately 0.207 and 0.220, showing a consistent internal grouping. The highest score belongs to sample09.pathological, with $S = 0.2203$. The functional samples all remain negative, with sample05.functional showing the lowest value ($S = -0.0889$). This

indicates that sample05.functional is the least well-separated functional sample under cosine distance.

5.5.4 Nearest-Neighbor Behavior at K=31

The nearest-neighbor structure also confirms the weak separation of the functional class. At $k = 31$, the nearest sample for every functional sample is systematically sample08.pathological (Table 5.13).

Table 5.13: Nearest opposite-class samples for functional samples at $k = 31$.

Functional sample	Nearest pathological sample	Cosine distance
sample01.functional	sample08.pathological	0.011111
sample02.functional	sample08.pathological	0.011122
sample03.functional	sample08.pathological	0.011063
sample04.functional	sample08.pathological	0.011092
sample05.functional	sample08.pathological	0.011004

This result is important because it shows that the functional samples are not only weakly separated, but are systematically closer to one specific pathological sample than to their own functional group. Therefore, the negative silhouette scores of the functional samples are not random anomalies; they accurately reflect the underlying structure of the cosine distance matrix.

5.5.5 Final Interpretation

Overall, cosine similarity provides a useful baseline for comparing the k-mer composition of the *ASH1L* samples. The best result was obtained at $k = 31$, but the global silhouette score remained low at $S = 0.0672$. This indicates a weak overall separation between the functional and pathological groups.

The per-sample analysis shows that this weak global result is caused by an asymmetric pattern: pathological samples form a relatively compact group, while functional samples do not. In fact, functional samples are closer on average to pathological samples than to other functional samples.

Therefore, cosine similarity captures some structure in the dataset, especially the internal similarity of the pathological group, but it is not sufficient as a strong discriminative measure for separating the two classes. For this reason, cosine similarity should be interpreted as a baseline method rather than the main analytical measure. Its results can be compared against Jensen–Shannon divergence

and the weighted kl-based measures to evaluate whether more distribution-sensitive approaches provide better class separation.

5.6 KWIP Results

Table 5.14: Global KWIP silhouette scores for different k-mer lengths.

k	Vocabulary size	Global silhouette score
15	192,910	0.0385
21	214,589	0.0453
31	234,545	0.0543

The global silhouette score increased as the k-mer length increased, from 0.0385 at $k = 15$ to 0.0543 at $k = 31$. This indicates that longer k-mers provided slightly better separation between the functional and pathological groups. However, all values remained close to zero, meaning that the separation was weak overall. Therefore, the KWIP measure detected a small class-related signal, but did not produce a strong clustering of the two groups.

5.6.1 Entropy-Weighted K-mer Contribution

The entropy-weighting step produced a large number of zero-weight k-mers, especially at smaller k values. Since the vocabulary contains only observed k-mers, zero-weight k-mers mainly correspond to k-mers present in all samples and therefore not useful for distinguishing the two groups. A summary of these features is provided in Table 5.15.

Table 5.15: Summary of entropy-weighted k-mer features.

k	**Vocabulary size**	**Zero-weight k-mers**	**Nonzero-weight k-mers**
15	192,910	182,060	10,850
21	214,589	198,354	16,235
31	234,545	209,686	24,859

The number of informative, nonzero-weight k-mers increased with k-mer length. At $k = 15$, only 10,850 k-mers had nonzero entropy weights, while at $k = 31$, this

number increased to 24,859. This trend explains the improvement in silhouette score at larger k values. Longer k-mers generated a larger set of sample-variable features, which contributed more information to the weighted inner product.

The maximum entropy weight was 1.0 for all three k values, meaning that some k-mers were present in approximately half of the samples. These k-mers receive the highest possible weight in the KWIP formulation because they are potentially the most informative for separating samples.

Because $k = 31$ produced the best global result, the per-sample silhouette scores for this configuration were examined in more detail in Table 5.16.

Table 5.16: Per-sample silhouette scores for kWIP at $k = 31$.

Sample	**Label**	**Silhouette score**
sample01.functional	Functional	0.0265
sample02.functional	Functional	0.0301
sample03.functional	Functional	0.0304
sample04.functional	Functional	0.0322
sample05.functional	Functional	0.0194
sample06.pathological	Pathological	0.0716
sample07.pathological	Pathological	0.0800
sample08.pathological	Pathological	0.0814
sample09.pathological	Pathological	0.0865
sample10.pathological	Pathological	0.0845

All samples had positive silhouette scores, indicating that no sample was more similar to the opposite group than to its assigned group. However, the values were still low, confirming that the separation was not strong.

The pathological samples had consistently higher silhouette scores than the functional samples. The mean silhouette score for the pathological group was 0.0808, whereas the mean silhouette score for the functional group was 0.0277. This shows that the pathological samples formed a more coherent group under the KWIP distance matrix, while the functional samples were closer to the boundary between the two classes.

The weakest sample was sample05.functional = 0.0194. This sample showed the lowest group separation and may be located near the boundary between functional and pathological samples. The clearest pathological samples were

sample09.pathological = 0.0865 and sample10.pathological = 0.0845. These samples showed the strongest consistency with their assigned group according to the KWIP distance.

The KWIP entropy-weighted inner product produced weak but consistently positive separation between the functional and pathological *ASH1L* samples. The best performance was observed at $k = 31$, suggesting that longer k-mers captured more discriminative sequence information than shorter k-mers. This is consistent with the increase in nonzero-weight k-mers at larger k values.

However, the global silhouette scores remained close to zero for all tested k values. Therefore, kWIP should not be interpreted as producing strong class separation in this dataset. Instead, it provides a useful weighted k-mer baseline that detects a weak class-related tendency. Compared with the functional samples, the pathological samples were more clearly grouped, as shown by their higher per-sample silhouette scores.

5.7 Comparison of the Results

To evaluate the performance of the proposed weighted KL formulation, its best result was compared with the main baseline methods used in this study: Jensen–Shannon distance, cosine distance, KWIP, and the unweighted KL baseline. For a fair comparison, the main comparison is based on the shared tested k-mer length $k = 31$, since all methods were evaluated at this value. The comparison is summarized in Table 5.17.

Table 5.17: Comparison of the results of all functions computed on ASH1L dataset, ordered by Silhouette score.

Category	Method	Score	Best Setting
Vector-based	Cosine distance	0.0672	$k : 31$
Divergence-based	standard KL	0.0666	$k : 31$
	$w(x) = -\log G(x)$	0.0635	$k : 31$, $\alpha = 1$, $\beta = 1$
	$w(x) = 1/\sqrt{G(x)}$	0.0463	$k : 31$, $\alpha = 1$, $\beta = 1$
	TF–IDF KL	0.0447	$k : 31$
	$w(x) = 1/G(x)$	0.0291	$k : 15$, $\alpha = 1$, $\beta = 1$
	Jensen–Shannon distance	0.0356	$k : 31$
Kernel-based	KWIP	0.0543	$k : 31$

The comparison shows that the highest score at $k = 31$ is obtained by cosine

distance, with a Silhouette score of 0.0672, followed very closely by the unweighted KL baseline with a score of 0.0666. The proposed weighted kl formulation using $w(x) = -\log G(x)$ obtains a score of 0.06357, which is slightly lower than the unweighted KL baseline but still higher than KWIP, Jensen–Shannon distance, TF–IDF KL, and the inverse-frequency weighting variants. This indicates that the logarithmic weighting scheme preserves much of the useful information captured by the unweighted k-mer probability distributions, while adding a controlled emphasis on globally less frequent k-mers.

Among the tested weighting functions, $w(x) = -\log G(x)$ performs better than the stronger inverse-frequency alternatives. The direct inverse weighting $w(x) = 1/G(x)$ gives the lowest score among the weighted kl variants, with a best score of 0.02917. The moderated inverse weighting $w(x) = 1/\sqrt{G(x)}$ improves this result to 0.04639, but it still remains below the logarithmic weighting result. This suggests that very strong emphasis on low-probability k-mers is not suitable for this dataset, probably because it gives too much influence to rare or highly sample-specific k-mers. In contrast, the logarithmic weight applies a softer adjustment, which appears to provide a better balance between preserving the original k-mer distribution and emphasizing informative k-mers.

Compared with the standard KL method, the weighted KL formulation does not clearly improve the Silhouette score at the shared value $k = 31$. The unweighted KL score is 0.0666, while the best weighted KL score with $w(x) = -\log G(x)$ is 0.06357. Therefore, the weighting function does not provide a direct improvement in global clustering performance for this dataset. However, the result is still close to the unweighted KL baseline and higher than several other baseline methods, showing that the weighted formulation remains competitive.

An additional observation is that when the weighted KL formulation with $w(x) = -\log G(x)$ was extended to $k = 41$, the Silhouette score increased to 0.07010. This value is higher than the best unweighted KL result reported at $k = 31$. However, this improvement should be interpreted carefully, because the comparison is not based on the same k-mer length. The higher score may reflect the effect of using longer k-mers, the weighting function, or a combination of both. Therefore, the result at $k = 41$ suggests that the weighted formulation may benefit from longer and more specific k-mer representations, but it should not be used alone as evidence that the weighting scheme outperforms the unweighted KL baseline.

Final discussion. Overall, the results show that the proposed weighting scheme provides a flexible way to control the contribution of different k-mers in the KL-based comparison. In particular, the logarithmic weighting function $w(x) = -\log G(x)$ performed better than the stronger inverse-frequency weighting

functions, suggesting that moderate weighting is more suitable than overemphasizing very rare k-mers. However, the weighted KL formulation did not clearly outperform the unweighted KL baseline at the shared value $k = 31$. This indicates that, for the present dataset, weighting alone is not sufficient to substantially improve class separation.

The comparison across methods also shows that the separation between functional and pathological samples is limited across several independent approaches, including Jensen–Shannon distance, cosine distance, KWIP, unweighted KL, and the weighted KL variants. Therefore, the limited improvement should not be interpreted only as a failure of the weighting strategy. Instead, it suggests that the dataset itself imposes limitations on separability. In other words, the problem is not only methodological, but also data-driven: the available sequence differences may not be strong or consistent enough to produce clear separation between the two sample groups using alignment-free k-mer distributions alone.

Key takeaway. The weighting scheme allows flexible emphasis on potentially informative k-mers, especially those with lower global frequency, but this flexibility does not automatically lead to better separation. The best weighted formulation remained close to the unweighted KL baseline, while stronger inverse-frequency weighting reduced performance. These results suggest that moderate weighting can preserve useful distributional information, but the separability of the functional and pathological samples is also constrained by the structure of the dataset itself.

5.8 Evaluation on the Brucella Dataset

The final evaluation of the proposed KL divergence methods relies on a benchmark genomic dataset of the genus Brucella. Sourced from the study by Liu et al. [24], this collection includes fully sequenced genomes from several distinct species and strains—namely Brucella abortus, Brucella melitensis, Brucella suis, and Brucella pinnipedialis, alongside some unclassified Brucella sp. samples downloaded from the original authors' associated GitHub repository [27]. Because these genomes are highly homologous, they serve as a strict test case for alignment-free approaches.

There are 29 unique samples in total across all classes in the table. Here is the breakdown of the unique sample counts for each class, along with the specific sample names as evidence:

1. ***Brucella* sp. (21 unique samples)**
   In biological nomenclature and taxonomy, "sp." stands for "species unspecified". When a sample is labeled as `Brucella_sp`, it means the organism has been confidently identified as belonging to the genus *Brucella*, but the specific

species (such as *abortus*, *canis*, or *suis*) is either unknown, unclassified, or was not specified in the source data. It functions as a catch-all category for unspecified *Brucella* species, which is why it doesn't have a more specific name. Since there are more than 10 samples, here are 5 examples:

- `GCA_000157875.1_ASM15787v1_genomic.fa`
- `GCA_000158995.1_ASM15899v1_genomic.fa`
- `GCA_000163135.1_ASM16313v1_genomic.fa`
- `GCA_000370925.1_Bruc_sp_56_94_V1_genomic.fa`
- `GCA_000370945.1_Bruc_sp_63_311_V1_genomic.fa`

2. ***abortus* (2 unique samples)**

   - `GCA_000740135.1_ASM74013v1_genomic.fa`
   - `GCA_000740155.1_ASM74015v1_genomic.fa`

3. **Other Classes (1 unique sample each)**
   These species only have a single representative sample in your dataset; therefore, their silhouette scores default to 0.0.

   - *canis*: `GCA_000691585.1_ASM69158v1_genomic.fa`
   - *suis*: `GCA_000740235.1_ASM74023v1_genomic.fa`
   - *pinnipedialis*: `GCA_000740275.1_ASM74027v1_genomic.fa`
   - *melitensis*: `GCA_001307475.2_ASM130747v2_genomic.fa`
   - *vulpis*: `GCA_900000005.1_BVF60_genomic.fa`
   - *inopinata*: `GCA_900095155.1_BR141012304v1_genomic.fa`

5.9 Dataset Bottleneck

When it came time to scale up my experiments and test methods on the much larger Brucella dataset, I hit a major bottleneck: the system repeatedly crashed with Out-Of-Memory (OOM) errors.The root of the problem was the sheer scale of the vocabulary. As I increased the k-mer size to capture more complex biological patterns (such as $k = 31$), the number of possible unique sequences exploded exponentially.

The original code was designed to load all the raw FASTA files and build massive frequency dictionaries directly in the computer's RAM all at the exact same time, which simply overwhelmed the hardware.To overcome this, I had to completely redesign the code's architecture. First, I shifted to a streaming approach, reading

the genomes and counting the k-mers in small, manageable blocks rather than all at once. Second, instead of holding the giant probability matrices in memory, I saved the intermediate data directly to the hard drive as compact arrays and computed the final distance matrix in small, sequential chunks. This architectural shift allowed me to bypass hardware limitations and successfully process the heavy Brucella dataset without sacrificing the mathematical precision of the divergence model.

5.10 Standard KL Baseline Scores

The main goal here is to determine whether our novel method can accurately capture the subtle sequence regularities that set these strains apart. To test this, the dataset was clustered in an unsupervised manner using different k-mer lengths ($k = 15, 21, 31$) and Dirichlet smoothing parameters ($\alpha = 0.001, 0.01, 0.1, 1$). Global and per-sample silhouette scores were then used to measure the separation quality of the resulting clusters, providing a clear metric for how well the different KL formulations distinguish true biological relationships from background noise.

Based on these scores, the clustering status of each genomic assembly was categorized into three distinct tiers: Clear (highly cohesive and well-separated), Weak/Ambiguous (near zero, on or close to the decision boundary), and Possibly Misplaced (negative scores, indicating potential misclassification or overlapping clusters).

Table 5.18: Sensitivity analysis of the Standard KL method on Brucella dataset

k	α	Overall Score	Clear (*abortus*)	Misclassified (`Brucella_sp`)
15	0.001	−0.333	0.161	−0.864
	0.010	−0.329	0.205	−0.860
	0.100	−0.317	0.301	−0.848
	1.000	−0.292	0.526	−0.812
21	0.001	−0.301	0.637	−0.836
	0.010	−0.297	0.659	−0.833
	0.100	−0.293	0.711	−0.823
	1.000	−0.304	0.817	−0.803
31	0.001	−0.282	0.724	−0.844
	0.010	−0.279	0.741	−0.841
	0.100	−0.279	0.784	−0.830
	1.000	−0.297	0.867	−0.813

Interpretation of the Results

- **Distinct Success with *abortus*:** The *abortus* class (containing 2 samples) is well-separated and responds very positively to hyperparameter tuning. As shown in Figure 5.3, its mean silhouette score increases significantly from 0.210 at $k = 15$ to 0.749 at $k = 31$, indicating strong cluster cohesion.
- **Poor Cohesion for `Brucella_sp`:** The largest group, `Brucella_sp` (21 samples), consistently exhibits negative silhouette scores ranging from −0.45 to −0.49 across all configurations. This results in a status of `possibly_misplaced`, suggesting that these samples are either poorly clustered or need a more granular sub-classification.
- **Singleton Classes:** Six classes (*canis*, *inopinata*, *melitensis*, *pinnipedialis*, *suis*, and *vulpis*) contain exactly one sample each. They consistently yield a silhouette score of 0.0 with a `weak/ambiguous` status, which is typical for singleton clusters where intra-cluster distances cannot be computed.

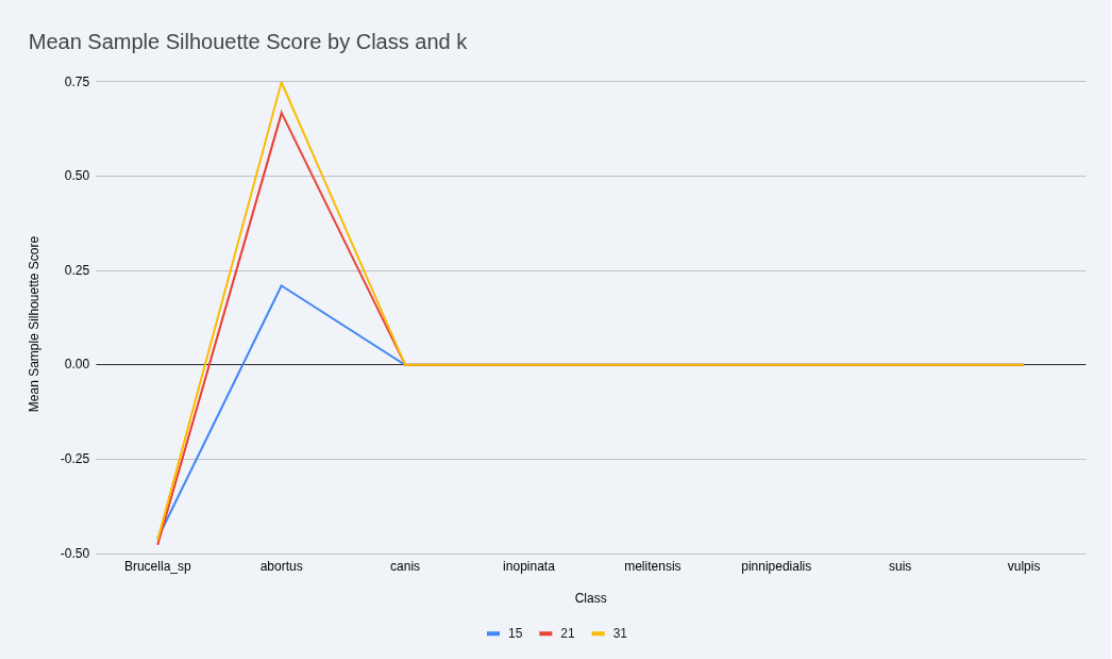

Figure 5.3: Summary of the silhouette score evaluations for the KL divergence method across various hyperparameter configurations (k and alpha for the Brucella dataset

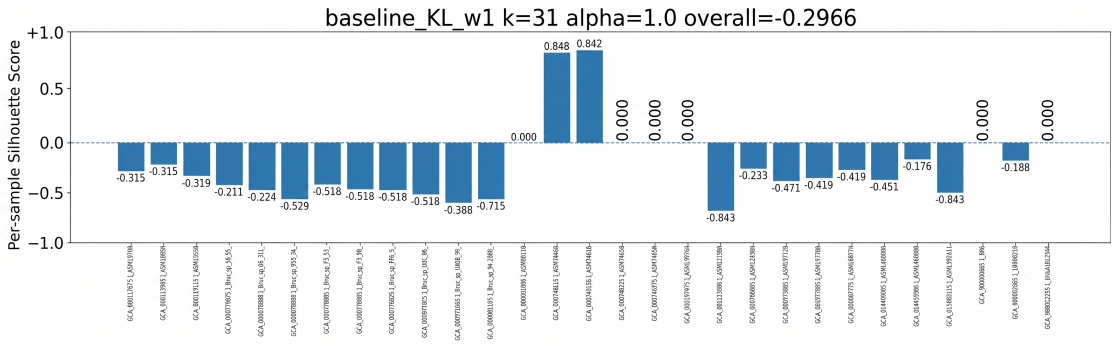

Figure 5.4: Distribution of individual genomic samples by silhouette score ($k = 31, \alpha = 1.0$)

5.11 Weighted-KL Functions Scores

This section presents the empirical evaluation of the novel Context-Aware Weighted KL Divergence approach. The following subsections detail the clustering performance and distance matrix evaluations across various weight functions and hyperparameter configurations.

5.11.1 Information-content weighting Scores

With the memory-mapped pipeline in place, I conducted a hyperparameter grid search across varying k-mer lengths (k), sample smoothing parameters (α), and global smoothing parameters (β). The resulting clustering qualities, measured by the overall Silhouette score, are presented in Table 5.19.

Table 5.19: Scores for the method $-\log G(x)$ across varying k, α, and β parameters.

k	α	β	Overall Score	Clear Samples	Misclassified
15	0.1	1	0.383	0.703, abortus	-0.351, Brucella_sp
	0.1	0.1	0.382	0.605, Brucella_sp	-0.341, Brucella_sp
	1	1	0.377	0.600, Brucella_sp	-0.100, Brucella_sp
	1	0.1	0.376	0.597, Brucella_sp	N/A
21	0.1	1	0.367	0.702, abortus	-0.083, Brucella_sp
	0.1	0.1	0.366	0.678, abortus	-0.297, Brucella_sp
	1	1	0.36	0.569, Brucella_sp	-0.308, Brucella_sp
	1	0.1	0.359	0.566, Brucella_sp	N/A
31	0.1	1	0.342	0.699, abortus	-0.062, Brucella_sp
	0.1	0.1	0.341	0.547, Brucella_sp	-0.240, Brucella_sp
	1	1	0.336	0.543, Brucella_sp	-0.253, Brucella_sp
	1	0.1	0.335	0.539, Brucella_sp	N/A

The overall global silhouette score for non-singleton clusters is 0.3837, indicating a moderate level of cluster separation and cohesion across the dataset. The majority of the computed samples (20 out of 23) are flagged with a clear status, indicating stable and correct cluster assignments.

The `abortus` class demonstrates strong cluster cohesion with a high mean silhouette score of 0.6911 (across 2 samples). In contrast, the larger `Brucella_sp` class has a lower mean score of 0.3544 (across 21 samples), suggesting higher internal diversity or potential sub-clusters.

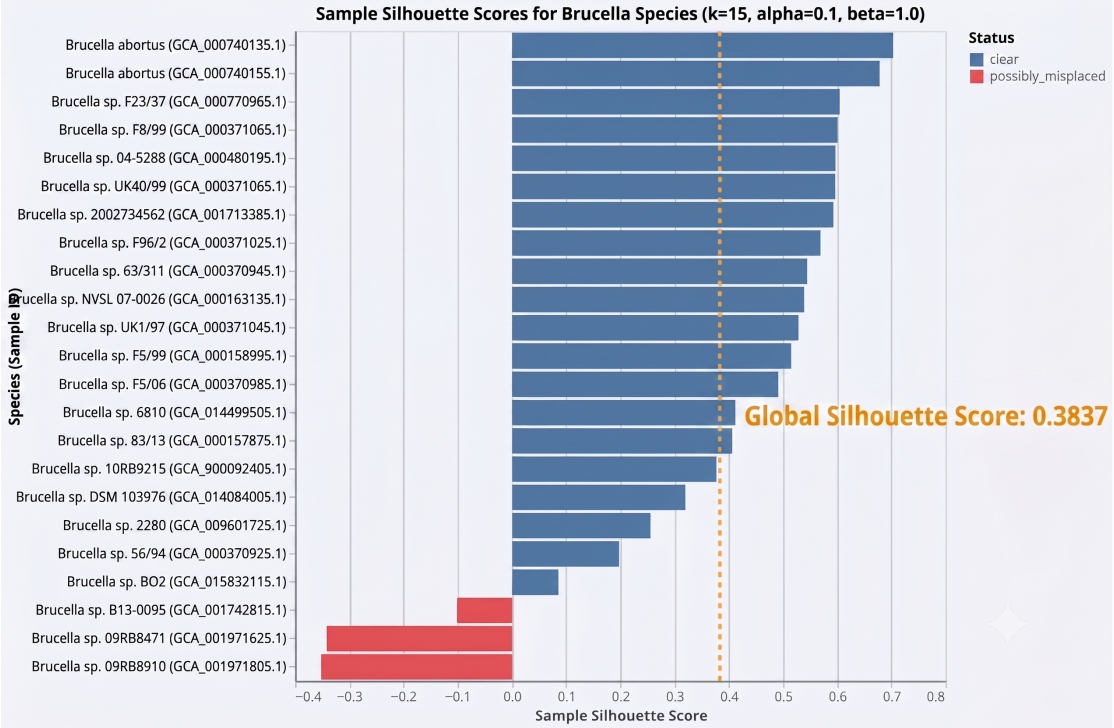

Figure 5.5: Sample-level silhouette analysis for *Brucella* species using the method $-\log G(x)$ with best values of parameters k, α, and β parameters.

5.11.2 Inverse-Frequency Weighting Scores

This section evaluates the performance of the inverse-frequency weighting approach, utilizing the weighting function $w(x) = 1/G(x)$. Table 5.20 details the method's classification effectiveness, presenting the overall scores alongside data on clear and misclassified samples.

Table 5.20: Scores for the method $w(x) = 1/G(x)$, evaluated across variations in k-mer length, smoothing parameter α, and parameter β.

k	α	β	Overall Score	Clear Samples	Misclassified
15	0.1	1	0.288	0.746, abortus	0.01, Brucella_sp
	1	1	0.281	0.730, abortus	-0.05, Brucella_sp
	0.1	0.1	0.216	0.452, Brucella_sp	-0.210, Brucella_sp
	1	0.1	0.209	0.343, Brucella_sp	N/A
21	0.1	1	0.261	0.742, abortus	-0.025, Brucella_sp
	1	1	0.254	0.726, abortus	-0.041, Brucella_sp
	0.1	0.1	0.183	0.408, Brucella_sp	-0.170, Brucella_sp
	1	0.1	0.175	0.266, Brucella_sp	N/A
31	0.1	1	0.224	0.733, abortus	-0.077, Brucella_sp
	1	1	0.216	0.718, abortus	-0.025, Brucella_sp
	0.1	0.1	0.138	0.347, Brucella_sp	-0.122, Brucella_sp
	1	0.1	0.130	0.206, Brucella_sp	N/A

Overall, the performance of the method is heavily influenced by the chosen parameters. The data reveals a clear trend regarding the k-mer length (k), where performance steadily decreases as the length increases; consequently, the shorter length of $k = 15$ yields the highest overall scores. Additionally, lower smoothing, specifically an α of 0.1, consistently outperforms the higher smoothing value of $\alpha = 1$ across all tested configurations.

The parameter β also plays a critical role, as utilizing a higher value ($\beta = 1$) significantly boosts the overall scores compared to a lower value ($\beta = 0.1$). Finally, in terms of errors, the system consistently struggles with *Brucella_sp*, making it the primary misclassified entity across nearly all tested parameter combinations.

5.11.3 Inverse square-root weighting Scores

This section evaluates the performance of the e square-root weighting approach, computing the function $w(x) = 1/\sqrt{G(x)}$. Table 5.21 details the method's classification effectiveness, presenting the overall scores alongside data on clear and misclassified samples.

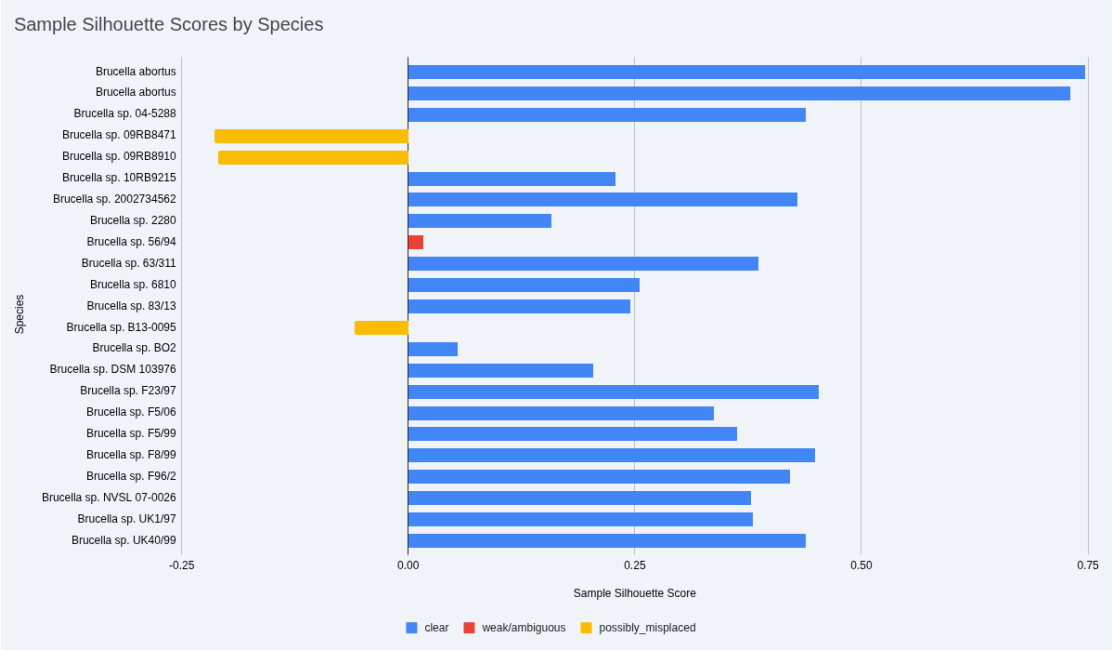

Figure 5.6: Sample-level silhouette analysis for *Brucella* species using the method $w(x) = 1/G(x)$ with best values of parameters k, α, and β parameters.

Table 5.21: Scores for the method $w(x) = 1/\sqrt{G(x)}$, evaluated across variations in k-mer length, smoothing parameter α, and parameter β.

k	α	β	Overall Score	Clear Samples	Misclassified
15	0.1	1	0.356	0.721, abortus	0.066, Brucella_sp
	1	1	0.350	0.699, abortus	-0.273, Brucella_sp
	0.1	0.1	0.339	0.557, Brucella_sp	-0.277, Brucella_sp
	1	0.1	0.333	0.528, Brucella_sp	N/A
21	0.1	1	0.336	0.719, abortus	-0.046, Brucella_sp
	1	1	0.329	0.698, abortus	-0.225, Brucella_sp
	0.1	0.1	0.316	0.522, Brucella_sp	-0.232, Brucella_sp
	1	0.1	0.309	0.492, Brucella_sp	N/A
31	0.1	1	0.307	0.715, abortus	-0.024, Brucella_sp
	1	1	0.300	0.6958, abortus	-0.166, Brucella_sp
	0.1	0.1	0.283	0.472, Brucella_sp	-0.176, Brucella_sp
	1	0.1	0.276	0.441, Brucella_sp	N/A

An analysis of the per-sample results for the optimal configuration ($k = 15, \alpha = 0.1, \beta = 1$) reveals that the vast majority of *Brucella* samples achieve positive silhouette scores. This indicates that most samples are clearly classified, with only a small subset falling into the misclassified (negative) range, highlighting the method's overall effectiveness under these specific settings.

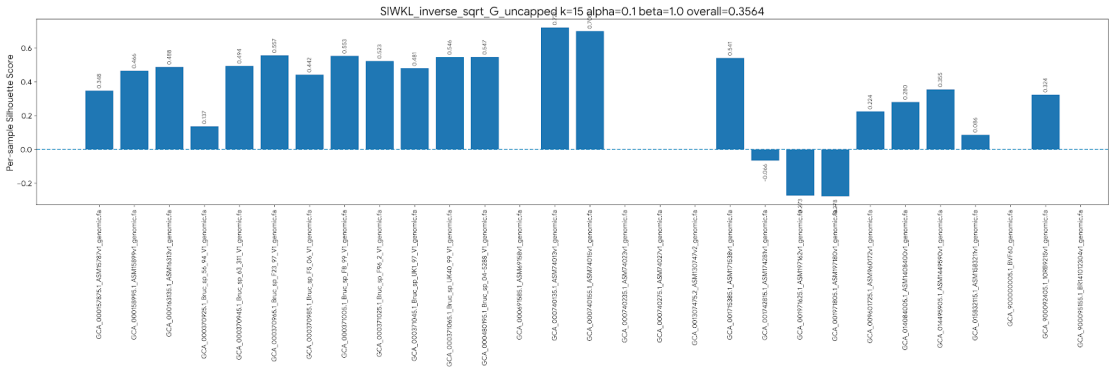

Figure 5.7: Sample-level silhouette analysis for *Brucella* species using the method $w(x) = 1/\sqrt{G(x)}$ with best values of parameters k, α, and β parameters.

5.11.4 TF-IDF Weighting Scores

Table 5.22: Summary of results for TF–IDF KL method with varying k.

k	Overall Score	Clear Samples	Misclassified
15	0.322	0.744, abortus	-0.148, Brucella_sp
21	0.307	0.740, abortus	-0.123, Brucella_sp
31	0.287	0.733, Brucella_sp	-0.091, Brucella_sp

The results show that out of the 29 samples evaluated, 19 are classified with a `clear` status, indicating that they are well-matched to their assigned clusters and distinct from neighboring clusters. There are 3 samples flagged as `possibly_misplaced`. Their negative scores indicate that they are closer on average to a neighboring cluster than to their currently assigned group. There is 1 sample flagged as `weak/ambiguous`, meaning it sits directly on the boundary between two clusters: `Brucella` sp. BO2 (GCA_015832115.1): 0.0030

5.12 Jensen–Shannon Divergence Results

This section explores the classification performance of the JSD method. Table 5.23 summarizes the method's effectiveness, detailing the overall scores alongside the clear and misclassified samples evaluated across varying k-mer lengths k.

Table 5.23: Summary of results for Jensen–Shannon Divergence method with varying k.

k	Overall Score	Clear Samples	Misclassified
15	0.253	0.448, abortus	-0.191, Brucella_sp
21	0.242	0.449, abortus	-0.167, Brucella_sp
31	0.225	0.446, Brucella_sp	-0.133, Brucella_sp

Overall, the Jensen-Shannon Divergence method achieves its highest performance at the shortest k-mer length of $k = 15$ (overall score: 0.253), with effectiveness steadily declining as k increases. While the method achieves its highest confidence scores on *abortus* samples, the per-sample data reveals that the majority of *Brucella_sp* samples are also successfully and clearly classified. However, the system's classification struggles are exclusively isolated to the *Brucella_sp* group, which accounts for all of the misclassified and ambiguous cases. Compared to the previously evaluated inverse-frequency weighting method, the JSD approach exhibits a noticeable reduction in overall classification performance and confidence scores.

5.13 Cosine Similarity Comparison Results

This section presents the classification outcomes derived from applying the Cosine Similarity method.

Table 5.24: Summary of results for Cosine Similarity method with varying k.

k	Overall Score
15	-0.374
21	-0.361
31	-0.342

The results for the Cosine Similarity method stand in stark contrast to previously evaluated approaches, demonstrating significantly poorer performance. The most notable observation is that the overall scores are entirely negative across all tested k-mer lengths.

5.14 KWIP Results

This section evaluates the classification performance of the KWIP method.

Table 5.25: Summary of results for KWIP method with varying k.

k	Overall Score
15	-0.236
21	-0.234
31	-0.230

The results for the KWIP method indicate poor classification performance, with overall scores remaining negative across all evaluated k-mer lengths. Similar to the Cosine Similarity approach, a negative overall score suggests that samples are, on average, more similar to members of neighboring clusters than to those within their assigned groups, reflecting widespread misclassification. While there is a marginal improvement as the k-mer length increases—shifting from -0.236 at $k = 15$ to -0.230 at $k = 31$—the persistently negative values demonstrate that the kWIP method is ineffective for this specific clustering task.

5.15 Comparison of the Methods

To evaluate the performance of the proposed weighted KL formulation, its best result was compared with other methods used in this study. The comparison is summarized in Table 5.26.

Table 5.26: Comparison of the results of all functions computed on Brucella dataset, ordered by Silhouette score.

Category	Method	Score	Best Setting
Divergence-based	standard KL	-0.279	$k : 31$, $\alpha : 1$
	$w(x) = -\log G(x)$	**0.383**	$k : 15$, $\alpha = 0.1$, $\beta = 1$
	$w(x) = 1/\sqrt{G(x)}$	0.356	$k : 15$, $\alpha = 0.1$, $\beta = 1$
	TF–IDF KL	0.322	$k : 15$
	$w(x) = 1/G(x)$	0.288	$k : 15$, $\alpha = 0.1$, $\beta = 1$
	Jensen–Shannon distance	0.253	$k : 15$
Kernel-based	KWIP	-0.230	$k : 31$
Vector-based	Cosine distance	-0.342	$k : 31$

The results demonstrate a clear superiority of the proposed weighted divergence formulations over standard baseline methods. The highest clustering quality is achieved using the divergence-based method with the logarithmic weighting function $w(x) = -\log G(x)$, which yields the top Silhouette score of 0.383 at optimal parameters $k = 15$, $\alpha = 0.1$, and $\beta = 1$. Other weighted approaches, such as $w(x) = 1/\sqrt{G(x)}$ (score 0.356) and the TF-IDF KL (score 0.322), also show robust positive performance, smoothly outperforming the standard Jensen-Shannon distance (score 0.253).

Chapter 6

Conclusion

In this thesis, I set out to improve how I measure sequence similarity and cluster genomic profiles. Traditional baseline methods often struggle to capture the complex biological realities of these sequences, which motivated our exploration into k-mer distributions and advanced entropy modeling. By introducing and testing new weighted KL divergence formulations, I aimed to provide a more nuanced and accurate approach.

Putting these methods to the test, particularly on the Brucella dataset, yielded clear and encouraging results. The proposed weighted divergence formulations significantly improved both the separation and clustering quality of the genomic data. The standout approach was the logarithmic weighting function $w(x) = -\log G(x)$. When parameterized optimally at $k = 15$, $\alpha = 0.1$, and $\beta = 1$, it achieved the highest Silhouette score of 0.383. This smoothly outperformed standard baselines like the unweighted KL, Jensen–Shannon distance, and Cosine distance. Beyond just the clustering scores, reviewing the actual classification outputs showed that the model successfully classified several specific target samples, proving it is a genuinely viable tool for distinguishing closely related sequences.

A major takeaway from this work is the delicate balance required when choosing the k-mer size (k). While setting $k = 15$ gave us optimal clustering resolution, pushing the size up to $k = 31$ caused significant problems. Not only did the clustering quality drop drastically—resulting in negative Silhouette scores across unweighted, kernel-based, and vector-based methods—but it also introduced severe computational roadblocks. Processing these larger k-mers frequently led to out-of-memory errors and unexpected terminations during matrix calculations. Ultimately, this research demonstrates that using weighted KL formulations with moderately sized k-mers offers a powerful, yet computationally realistic, framework for genomic analysis.

6.2 Future Work

While the results with our weighted divergence formulations are highly promising, this research also opens the door for several practical next steps:

- **Solving the Memory Bottleneck:** The most immediate challenge is overcoming the memory crashes I encountered with larger k-mer sizes ($k \geq 31$). Future work should focus on algorithmic optimizations, such as utilizing sparse matrices or distributed computing frameworks, so I can process larger sequence fragments without overloading the system's memory.
- **Testing on Diverse Genomes:** I established that the logarithmic weighting $w(x) = -\log G(x)$ works very well for the Brucella dataset, but I need to see how it handles other types of data. Applying this pipeline to more complex microbiomes or larger eukaryotic genomes will help verify if these specific divergence metrics are universally effective.
- **Smarter Parameter Tuning:** In this study, the parameters α and β were fixed based on empirical testing. A great next step would be introducing machine learning techniques to dynamically tune these parameters on the fly, allowing the model to adapt automatically to the specific entropy profile of whatever dataset is being analyzed.
- **Improving Classification Power:** Even though our model successfully classified some challenging samples, there is always room to drive the misclassification rate down further. Feeding these weighted KL divergence metrics directly into more advanced predictive models, such as deep learning classifiers or ensemble methods, could make sub-species identification even more precise.

The complete source code is maintained in the project repository: `https://github.com/PegahYmd/Statistical-Modeling-of-Genomic-Sequences`

Acronyms

KL Kullback–Leibler

DNA Deoxyribonucleic Acid

KWIP k-mer Weighted Inner Product

VCF Variant Call Format

VEP Variant Effect Predictor

Bibliography

[1] Nicola Mulder and et al. «Genomic Research Data Generation, Analysis and Sharing – Challenges in the African Setting». In: *Data Science Journal* 16 (2017). URL: https://datascience.codata.org/articles/dsj-2017-049.

[2] Bertil Schmidt and Andreas Hildebrandt. «Next-generation sequencing: big data meets high performance computing». In: *Drug Discovery Today* 22.4 (2017). URL: https://www.sciencedirect.com/science/article/abs/pii/S1359644617300582.

[3] Susana Vinga. «Information theory applications for biological sequence analysis». In: *Briefings in Bioinformatics* 15.3 (2014).

[4] Armin O. Schmitt and Hanspeter Herzel. «Estimating the Entropy of DNA Sequences». In: *Journal of Theoretical Biology* 188.3 (1997). URL: https://www.sciencedirect.com/science/article/abs/pii/S0022519397904938.

[5] Pranabananda Chanda, Eliana Costa, Jiali Hu, Sneha Sukumar, Kyle Van Hemert, and Rita Walia. «Information Theory in Computational Biology». In: *Molecular Omics* 16.5 (2020). URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7517167/.

[6] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M. Karlowski. «Alignment-free sequence comparison: benefits, applications, and tools». In: *Genome Biology* 18 (2017). URL: https://link.springer.com/article/10.1186/s13059-017-1319-7.

[7] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, Ying Wang, Gesine Reinert, and Fengzhu Sun. «Alignment-Free Sequence Analysis and Applications». In: *Annual Review of Biomedical Data Science* 1 (2018). DOI: 10.1146/annurev-biodatasci-080917-013431. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC6905628/.

[8] C. Moeckel and et al. «A survey of k-mer methods and applications in bioinformatics». In: *Computational and Structural Biotechnology Journal* 23 (2024). URL: https://www.sciencedirect.com/science/article/pii/S2001037024001703.

[9] J. Lin. «Divergence measures based on the Shannon entropy». In: *IEEE Transactions on Information Theory* 37.1 (1991). URL: https://www.cise.ufl.edu/~anand/sp06/jensen-shannon.pdf.

[10] B. Lasher and et al. «bpRNA-CosMoS: a robust and efficient RNA structural comparison method using k-mers and cosine similarity». In: *Bioinformatics* 41.4 (2025).

[11] S. Kullback and R. A. Leibler. «On Information and Sufficiency». In: *The Annals of Mathematical Statistics* 22.1 (1951). URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full.

[12] M. D. Roberts, O. Davis, E. B. Josephs, and R. J. Williamson. «k-mer-based approaches to bridging pangenomics and population genetics». In: *arXiv preprint arXiv:2409.11683* (2024). URL: https://doi.org/10.48550/arxiv.2409.11683.

[13] Y. Bussi, R. Kapon, and Z. Reich. «Large-scale k-mer-based analysis of the informational properties of genomes, comparative genomics and taxonomy». In: *PLOS ONE* 16.11 (2021). DOI: 10.1371/journal.pone.0258693.

[14] Alise J. Ponsero, Marian Miller, and Bonnie L. Hurwitz. «Comparison of k-mer-based de novo comparative metagenomic tools and approaches». In: *Microbiome Research Reports* 2.1 (2023). DOI: 10.20517/mrr.2023.26.

[15] Julia Van Etten, Timothy G. Stephens, and Debashish Bhattacharya. «A k-mer-Based Approach for Phylogenetic Classification of Taxa in Environmental Genomic Data». In: *Systematic Biology* 72.5 (2023). DOI: 10.1093/sysbio/syad037.

[16] Kristoffer Sahlin. «Effective sequence similarity detection with strobemers». In: *Genome Research* 31.11 (2021). DOI: 10.1101/gr.275648.121.

[17] S. Vinga and J. Almeida. «Alignment-free sequence comparison — a review». In: *Bioinformatics* 19.4 (2003). DOI: 10.1093/bioinformatics/btg005.

[18] Kevin D. Murray, Christfried Webers, Cheng Soon Ong, Justin Borevitz, and Norman Warthmann. «kWIP: The k-mer weighted inner product, a de novo estimator of genetic similarity». In: *PLOS Computational Biology* 13.9 (2017). DOI: 10.1371/journal.pcbi.1005727.

[19] G. Salton, A. Wong, and C. S. Yang. «A vector space model for automatic indexing». In: *Communications of the ACM* 18.11 (1975). DOI: 10.1145/361219.361220.

[20] M. T. Swain and M. Vickers. «Interpreting alignment-free sequence comparison: what makes a score a good score?» In: *NAR Genomics and Bioinformatics* 4.3 (2022). DOI: 10.1093/nargab/lqac062.

[21] N. Pornputtapong, I. Nookaew, and J. Nielsen. «KITSUNE: A tool for identifying empirically optimal k-mer length for alignment-free phylogenomic analysis». In: *Frontiers in Bioengineering and Biotechnology* 8 (2020). DOI: 10.3389/fbioe.2020.556413.

[22] B. Lasher and D. A. Hendrix. «bpRNA-CosMoS: a robust and efficient RNA structural comparison method using k-mer based cosine similarity». In: *Bioinformatics* 41.4 (2025), btaf108. DOI: 10.1093/bioinformatics/btaf108.

[23] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. «Mash: fast genome and metagenome distance estimation using MinHash». In: *Genome Biology* 17.1 (2016). DOI: 10.1186/s13059-016-0997-x.

[24] Shaopeng Liu and David Koslicki. «CMash: Fast, multi-resolution estimation of k-mer-based Jaccard and containment indices». In: *Bioinformatics* 38.Supplement_1 (2022). DOI: 10.1093/bioinformatics/btac237.

[25] Ensembl. *Gene Summary: ASH1L (ENSG00000116539)*. Release 115. URL: https://www.ensembl.org/Homo_sapiens/Gene/Summary?g=ENSG00000116539.

[26] BCFtools Team. *BCFtools: Utilities for variant calling and manipulating VCF and BCF files*. GitHub repository. 2026. URL: https://github.com/samtools/bcftools.

[27] David Koslicki et al. *CMash: Fast, intersection-based similarity of large genomes*. https://github.com/dkoslicki/CMash. 2019.