N-gram-based Classification and Hierarchical Clustering of Genome Sequences

Andrija Tomović
Friedrich Miescher Institute for Biomedical Research
Maulbeerstrasse 66, CH-4058 Basel, Switzerland

Predrag Janičić
Faculty of Mathematics, University of Belgrade,
Studentski trg 16,11000 Belgrade, Serbia and Montenegro

Vlado Kešelj
Faculty of Computer Science, Dalhousie University,
Halifax, NS, Canada, B3H 1W5

Abstract

In this paper we address the problem of automated classification of isolates, i.e., the problem of determining the family of genomes to which a given genome belongs. Additionally, we address the problem of automated hierarchical clustering of isolates according only to their statistical substring properties. For both of these problems we present algorithms based on nucleotide n-grams. Results obtained experimentally are very positive and suggest that the proposed techniques can be successfully used in a variety of related problems.

1 Introduction

The number and sizes of genome databases grow rapidly over the last years. Huge amount of information requires new ways for processing them and using them in efficient ways. One of the most important problems is clustering of genomes and classification of genomes, i.e., determining a group to which it belongs. For example, distinguishing virus subspecies, strains and isolates is important in vaccine development, diagnostics and other fields of biological and medical research and practice.

The genetic information of every organism is written in the universal language of DNA sequences, and the DNA sequence of any given organism can be obtained by the standard biochemical techniques. Using these sequences, it is now possible to catalogue and characterize any set of living organisms. From such comparisons we can estimate the place of each organism in the family tree of living species—the "tree of life."

In this paper we address the following two problems:

- given several families of genomes and a genome, determine the family to which it most likely belongs;
- define a procedure for clustering of genomes (according only to their statistical substring properties); such procedure should be effective and should not require any expert knowledge.

This work follows some of the ideas from [13]. This paper includes results on using a character n-gram technique for the problem of authorship attribution, i.e., the problem of identifying the author of an anonymous text, or text whose authorship is in doubt. We address the problem of genome sequences classification and extend the approach and ideas reported in [13].

The results obtained following the proposed technique are very positive and encouraging. We believe that the technique can find many applications, both in academic research and in medicine and industry.

Overview of the paper In Section 2 we give some background information and basic notions. In Section 3 we introduce the notion of dissimilarity measures and present several dissimilarity functions. In Section 4 we report on our experimental results that led us to good dissimilarity functions. In Section 5 we discuss how the proposed technique can be used for genome sequences classification and in Section 6 we discuss how the proposed technique can be used for genome clustering and we present some experimental results. In Section 7 we briefly discuss the related work. In Section 8 we present some plans for future work and in Section 9 we draw final conclusions.

2 Background

2.1 N-grams

Definition 1 *Given a sequence of tokens $S = (s_1, s_2, ..., s_{N+(n-1)})$ over the token alphabet A with N and n positive integers, an n-gram of the sequence is an n-long subsequence of consecutive tokens. The i^{th} n-gram of S is the sequence* $(s_i, s_{i+1}, ..., s_{i+n-1})$ [18].

Note that there are N such n-grams in S. There are $(|A|)^n$ different n-grams over the alphabet A ($|A|$ is the size of A).

For $n \leq 5$ Latin names are commonly used for n-grams (e.g., trigrams) and for $n > 5$ numeric prefix are common (e.g., 6-grams).[1]

N-grams have been successfully used for a long time in a wide variety of problems and domains, including: text compression (1953) [20], spelling error detection and correction (1962) [2, 21], optical character recognition (1967) [1],

[1]Since "gram" is a Greek word, some authors prefer using names *monogram, digram, trigram, tetragram, ...* instead of *unigram, bigram, trigram, quadrigram, ...*.

information retrieval (1973) [8], language identification (1991) [17], automatic text categorization (1994) [5], music representation (1999) [9], computational immunology (2000) [15], analysis of whole-genome protein sequences (2002) [11], authorship attribution (2003) [13].

In many domains, techniques based on using n-grams gave very good results. For instance, in natural language processing, n-grams can be used to distinguish between documents in different languages in multi-lingual collections and to gage topical similarity between documents in the same language [5, 7], but also in some other problems. In this field, n-grams show some of its good features:

- robustness: relatively insensitive to spelling variations/errors;
- completeness: token alphabet known in advance;
- domain independence: language and topic independent;
- efficiency: one pass processing;
- simplicity: no linguistic knowledge is required.

On the other hand, the problem which can appear in using n-grams is *exponential explosion*. If A is the Latin alphabet with the space delimiter, then $|A| = 27$. If one distinguishes between upper and lower case letters, and also places significance in numerical digits, then $|A| = 63$. It is clear that many of algorithms with n-grams are computationally too expensive even for $n = 5$ or $n = 6$.

2.2 Definitions of Relevant Biological Terms

Definition 2 (Genome) *Genome is the complete genetic material of an organism. Its size is generally given as its total number of base pairs.[12]*

Definition 3 (Base pair) *Base pair are two nitrogenous bases (adenine and thymine or guanine and cytosine) held together by weak bonds. Two strands of DNA are held together in the shape of a double helix by the bonds between base pairs.[19]*

Definition 4 (Base sequence) *Base sequence is the order of nucleotide bases in a DNA molecule.[19]*

Definition 5 (Nucleotide) *Nucleotide is a subunit of DNA or RNA consisting of a nitrogenous base (adenine, guanine, thymine, or cytosine in DNA; adenine, guanine, uracil, or cytosine in RNA), a phosphate molecule, and a sugar molecule (deoxyribose in DNA and ribose in RNA). Thousands of nucleotides are linked to form a DNA or RNA molecule.[19]*

Definition 6 (Isolate) *Isolate is a genome instance.*

As it can be seen from above definitions, genome is a general term, but isolate is genom from concrete specific organism.

Genome (isolate) can be represented as a base sequence. It can be seen as word on alphabet $\{A, G, C, T, N, R, W, Y, M, K, S, H, B, V, D\}$ where the dominant letters are $\{A, G, C, T\}$. This set of dominant letters represents standard nucleotide codes [2]. A represents Adenosine, T - Thymidine, C - Cytosine, G - Guanosine. They are dominant because DNA sequence is made of 4 nucleotide which they represent. Other letters represent ambiguous nucleotide codes: $N \in \{A, G, C, T\}$, $R \in \{G, A\}$, $W \in \{A, T\}$, $Y \in \{C, T\}$, $M \in \{A, C\}$, $K \in \{G, T\}$, $S \in \{G, C\}$, $H \in \{A, C, T\}$, $B \in \{C, G, T\}$, $V \in \{A, C, G\}$, $D \in \{A, G, T\}$. [4] These letters appears in DNA sequence, because of genetic variability.

Clustering is the process of grouping data objects together on the basis of the features they have in common. It is a standard data mining task in which items are grouped in clusters of objects with the objective of maximizing the intra-cluster similarity and the inter-cluster dissimilarity between objects. *Hierarchical clustering* is the clustering in which the clusters do not simply make a partition of the set of objects, but they organized into a tree hierarchy, so that any child cluster is a subset of the parent cluster and the sibling clusters are disjoint. When applied to genomes, hierarchical clustering produces a biological taxonomy, which helps us to make sense of the enormous diversity of living organisms. In any organism, there are many different kinds of features to choose from, and in principle all of them can be used in classification. For example, one could use external anatomy, internal anatomy, chromosomes, molecules, genome etc. [16]

Ideally, classification should be based on *homology;* that is, shared characteristics that have been inherited from a common ancestor. The more recent ancestor is shared between two species,

- the more homologies they share, and
- the more similar these homologies are.

However, since the birth of molecular biology, homologies can now also be studied at the level of proteins and DNA (DNA-DNA Hybridization, Chromosome Painting, Comparing DNA Sequences).[14] Genome analysis gives powerful way to determine evolutionary relationships. The complete DNA sequence (genome) of an organism defines the species with a big precision. This specification is in a digital form (a string of letters, word on a given alphabet) and can be easy stored in computer and compared with genomes of other living things.

3 Dissimilarity Functions

Dissimilarity measure d is a function on two sets of sequences $\mathcal{P}_1$ and $\mathcal{P}_2$ (defining specific *profiles*) and it should reflect the dissimilarity between these two, i.e.,

[2] U is also standard nucleotide code and represents Uridine which is replacement of T in RNA

it should meet the following conditions:

- $d(\mathcal{P},\mathcal{P}) = 0$;
- $d(\mathcal{P}_1,\mathcal{P}_2) = d(\mathcal{P}_2,\mathcal{P}_1)$;
- the value $d(\mathcal{P}_1,\mathcal{P}_2)$ should be *small* if $\mathcal{P}_1$ and $\mathcal{P}_2$ are *similar*.
- the value $d(\mathcal{P}_1,\mathcal{P}_2)$ should be *large* if $\mathcal{P}_1$ and $\mathcal{P}_2$ are *not similar*.

The last two conditions are informal as the notion of *similarity* is not strictly defined. By *similar sequences* we will usually mean sequences with similar distributions of n-grams.

In [3], some pioneer methods for authorship attribution problem[3] and dissimilarity measures were discussed. In that book, in the chapter about the use of computers for language processing, a range of problems from some early ideas about language modelling to cryptography, language evolution and authorship attribution, are discussed and tackled using character-level n-grams. Specifically, for authorship attribution problem (i.e., *author identification problem* as called in the book), the bigram letter statistic was used. Two texts are compared for the same authorship, using the dissimilarity formula:

$$d(M,N) = \sum_{I,J}[M(I,J) - E(I,J)] \cdot [N(I,J) - E(I,J)], \tag{1}$$

where I and J are indices over the range $\{1,2,\dots,26\}$, i.e., all letters of English alphabet, $M(I,J)$ and $N(I,J)$ are normalized character bigram frequencies for one and the other author and $E(I,J)$ is the same normalized frequency for "standard English". As the bigram frequencies of "standard English" are obviously language-dependent parameters, another dissimilarity measure is given:

$$d(M,N) = \sum_{I,J}[M(I,J) - N(I,J)]^2. \tag{2}$$

In [13], ideas from [3] are followed and adapted for larger n-grams (and also used for the author attribution problem). Namely, the above dissimilarity functions (functions (1) and (2)) give equal weight to frequency differences of all n-grams included in a profile. This may be justified for bigrams that were used in [3], because all of them were reasonably frequent and the sparse data problem is not an issue. However, with larger n-grams the frequency varies more and more, so if we used this absolute difference measure the more frequent n-grams would be emphasized more because the absolute differences in their frequencies are larger. In order to "normalize"these differences, they are divided by the average frequency for a given n-gram. This, in [13], led to the following dissimilarity measure (which we will denote by d_1 within this paper):

$$d_1(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n \in profile} \left(\frac{2 \cdot (f_1(n) - f_2(n))}{f_1(n) + f_2(n)} \right)^2$$

[3]Authorship attribution problem is as follows: given texts written by authors A_1, A_2, ... A_n, and one additional piece of text, guess who of the given authors wrote that piece of text.

where $f_1(n)$ and $f_2(n)$ are frequencies of an n-gram n in the author profile ($\mathcal{P}_1$) and the document profile ($\mathcal{P}_2$).

A document profile in [13] is the set of L most frequent n-grams in a set of documents, with their attached relative frequencies. The value of parameter L ranges from 20 to 5000. We define genome profiles in the analogous way.

In this paper, we introduce several new dissimilarity measures. Some of them are based on similar considerations as the above one from [13], while we explore some additional variations. In the function d_1 frequency differences are divided by the "average" (arithmetic mean value — $(f_1(n) + f_2(n))/2$) frequency for a given n-gram. In some of the functions we introduce in this paper, we divide frequency differences not by arithmetic mean value, but by geometric mean value for a given n-gram ($\sqrt{f_1(n) \cdot f_2(n)}$), or harmonic mean value $(2/(1/f_1(n)+1/f_2(n)))$ or quadratic mean value $\sqrt{(f_1(n)^2 + f_2(n)^2)/2}$. Also, elements in the sums may be squared, or we may sum the absolute values of differences, in the fashion of the L_1 measure.

$$
\begin{aligned}
d_2(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{n \in \mathit{profile}} \frac{2|f_1(n) - f_2(n)|}{f_1(n) + f_2(n)} \\
d_3(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{n \in \mathit{profile}} \left(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)} + 1} \right)^2 \\
d_4(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{n \in \mathit{profile}} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) \cdot f_2(n)} + 1}
\end{aligned}
$$

An additive constant 1 is used in the numerator of the function d_4 since $f_1(n)$ or $f_2(n)$ can be zero. This function (d_4) will be in focus of our attention in the rest of the paper.

The following two functions are based on the harmonic mean:

$$
\begin{aligned}
d_5(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{\substack{n \in \mathit{profile} \\ f_1(n) f_2(n) \neq 0}} \left(\frac{(f_1(n) - f_2(n))(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)} \right)^2 \\
d_6(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{\substack{n \in \mathit{profile} \\ f_1(n) f_2(n) \neq 0}} \frac{|f_1(n) - f_2(n)|(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)}
\end{aligned}
$$

The following functions are based on the geometric mean value without the use of the additive constant:

$$
\begin{aligned}
d_7(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{\substack{n \in \mathit{profile} \\ f_1(n) f_2(n) \neq 0}} \left(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)}} \right)^2 \\
d_8(\mathcal{P}_1, \mathcal{P}_2) &= \sum_{\substack{n \in \mathit{profile} \\ f_1(n) f_2(n) \neq 0}} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) f_2(n)}}
\end{aligned}
$$

In order to explore the affect of square differences, the following two functions are constructed as weighted linear combinations of linear and square differences:

$$d_9(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(A|f_1(n) - f_2(n)| + |f_1(n)^2 - f_2(n)^2| \right)$$

for $A(\mathcal{P}_1, \mathcal{P}_2) = 100$ and $B = 1$

$$d_{10}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(A|f_1(n) - f_2(n)| + B|f_1(n)^2 - f_2(n)^2| \right)$$

for $A(\mathcal{P}_1, \mathcal{P}_2) = 1000$ and $B = 0.1$.

The following two functions are based on the quadratic mean value:

$$d_{11}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(\frac{\sqrt{2}(f_1(n) - f_2(n))}{\sqrt{f_1(n)^2 + f_2(n)^2}} \right)^2$$

$$d_{12}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{\sqrt{2}|f_1(n) - f_2(n)|}{\sqrt{f_1(n)^2 + f_2(n)^2}}$$

Using the following function we explore the affect of the additive constant on the geometrical mean based function:

$$d_{13}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)} + 10} \right)^2$$

Although following ideas and considerations from [3] and [13], the above functions are only heuristic measures. Their quality is to be tested and ensured by experiments that follow.

We also use several functions, based on measures for similarity/dissimilarity between patterns from [10]:

Euclidean distance:

$$d_{14}(\mathcal{P}_1, \mathcal{P}_2) = \sqrt{\sum_{n \in profile} (f_1(n) - f_2(n))^2}$$

Manhattan distance:

$$d_{15}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} |f_1(n) - f_2(n)|$$

$$d_{16}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{2 \sum_{n \in profile} f_1(n) f_2(n)}{\sum_{n \in profile} f_1(n)^2 + \sum_{n \in profile} f_2(n)^2}$$

$$d_{17}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n \in profile} f_1(n) f_2(n)}{\sum_{n \in profile} f_1(n)^2 + \sum_{n \in profile} f_2(n)^2 - \sum_{n \in profile} f_1(n) f_2(n)}$$

$$d_{18}(\mathcal{P}_1,\mathcal{P}_2) = 1 - \frac{\sum_{n\in profile} f_1(n)f_2(n)}{\sqrt{(\sum_{n\in profile} f_1(n)^2)(\sum_{n\in profile} f_2(n)^2)}}$$

$$d_{19}(\mathcal{P}_1,\mathcal{P}_2) = 1 - \frac{\sum_{n\in profile} f_1(n)f_2(n)}{\min((\sum_{n\in profile} f_1(n)^2)(\sum_{n\in profile} f_2(n)^2))}$$

4 Looking for Good Dissimilarity Functions

Encouraged by the success rate of the system for authorship attribution presented in [13], we try to use the same or a similar technique to the analogous problem of classifying genome sequences. Namely, we want to build a system that, given several groups of genome sequence and a genome sequence, determine a group to which it most likely belongs. The basic idea is simple: for the given set of families $\mathcal{P}_i$, $i = 1, 2, \ldots, k$ and the given genome sequence g, compute the dissimilarity measures $\mathcal{D}(\{g\}, \mathcal{P}_i)$, $i = 1, 2, \ldots, k$. If the value $\mathcal{D}(\{g\}, \mathcal{P}_s)$ is the smallest one, then the guess is that g belongs to the family $\mathcal{P}_s$. Thus, the algorithm for classifying genome sequences is trivial and its quality completely relies on the appropriateness of the dissimilarity measure used. This is essentially the well-known k Nearest Neighbours (kNN) classification method, with $k = 1$ [10].

In the following experiments we used isolates with complete genome sequences of HIV-1 and HIV-2 virus. HIV — Human immunodeficiency virus is categorized in the family of viruses known as retroviruses. Within this family of viruses, HIV is further classified in the genus lentiviruses. HIV-1 and HIV-2 are the two species of human immunodeficiency viruses. They differ in the nature of some of the accessory genes.[4] Scientists have produced SHIV, simian-human immunodeficiency virus, by putting the outer envelope of HIV onto an SIV core.[5] SIV is also a lentivirus, but this virus infects only monkeys. In the following experiments we use also isolates with complete genome of SHIV virus to make classification more demanding (instead of SIV, because SHIV closer related to HIV then SIV). In any case, we just decided to take this corpora to demonstrate our method. Our method is not specially adapted for HIV/SHIV corpora, it can be applied to any other corpora as well, it is not based on corpora selection.

Corpus 1 *The corpus is made out of three group of isolates with complete genomes (available from http://www.ncbi.nlm.nih.gov/, as in October 2004):*

- *a group of 445 isolates of HIV-1;*
- *a group of 18 isolates of HIV-2 ;*

[4] http://biology.fullerton.edu/courses/biol_302/Web/Browser/index.html Understanding Human Immunodeficiency Virus

[5] http://www.niaid.nih.gov/daids/vaccine/advoslide/sld001.htm NATIONAL AIDS VACCINE ADVOCATES FORUM Vaccine Basic Science Mary A. Allen, R.N, M.S. November 8, 1997.

- *a group of 8 isolates of SHIV.*

For all experiments presented, we used an originally developed software, but also softver package Ngrams written by Vlado Keselj.[6]

4.1 Preliminary Confirmation of Expectations

In order to test whether the technique proposed in [13] can be used for genome sequences classification, we performed the following experiment (using Corpus 1).

Experiment 1 *Take one (random) genome sequence (isolate)g from HIV-1 and compute the values:*

$$d(g, \textit{HIV-1} \setminus \{g\}), \quad d(g, \textit{HIV-2}), \quad d(g, \textit{SHIV})$$

for different n-gram lengths ($n = 1, 2, \ldots, 10$).

The plausible outcome *is that $d(g, \textit{HIV-1} \setminus \{g\})$ is the smallest value for each n ($n = 1, 2, \ldots, 10$).*

We performed the above experiment using the dissimilarity function d_1 (from [13]). The results are shown in Figure 1.[7] Despite the very high success rate in the author attribution problem, this function and this experiment did not meet our expectations. Namely, as can be seen from Figure 1, $d(g, \text{HIV-1} \setminus \{g\})$ is not smallest among $d(g, \text{HIV-1} \setminus \{g\})$, $d(g, \text{HIV-2})$, $d(g, \text{SHIV})$ (moreover, for most n (n=1,...,10) $d(g, \text{HIV-1} \setminus \{g\})$ is the largest value. Hence, this dissimilarity function cannot be successfully used for genome sequences classification.

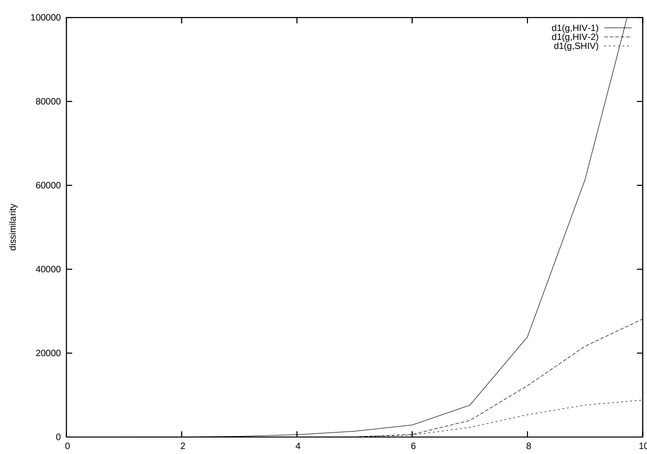

Figure 1: Results for Experiment 1 performed by using dissimilarity function d_1

[6]Ngrams package is available at http://www.cs.dal.ca/˜vlado/srcperl/Ngrams/.

[7]All experimental data can be obtained on request from the first author.

In addition to the attempt with the function d_1, we performed Experiment 1 using the dissimilarity function d_4 (and the same random genome sequence as with the function d_1). Unlike d_1, the function d_4 produces really encouraging results. They are shown in Figure 2. As required, for each n ($n = 1, 2, \ldots, 10$), the value $d(g, \text{HIV-1} \setminus \{g\})$ is the smallest one.

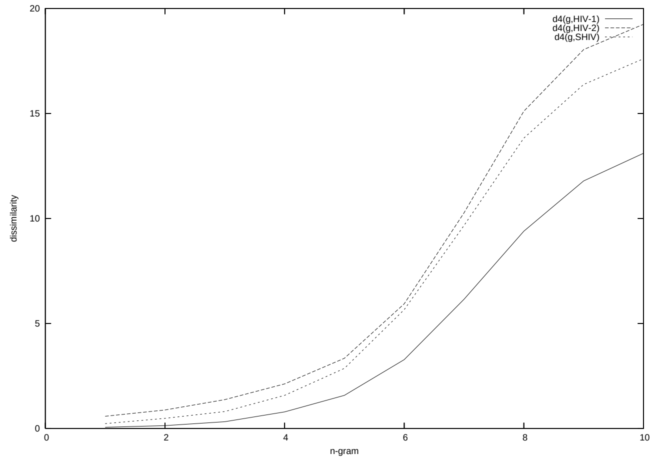

Figure 2: Results for Experiment 1 by using dissimilarity function d_4

4.2 Establishing Preliminary Findings

The outcome of Experiment 1 using the dissimilarity function d_4 is encouraging, but might be misleading if the random genome selected (from HIV-1) within experiment has some specific properties. Therefore, we want to verify that this is not the case. More precisely, we want to check that $d_4(g, \text{HIV-1} \setminus \{g\})$ is the smallest among the values $d_4(g, \text{HIV-1} \setminus \{g\})$, $d_4(g, \text{HIV-2})$, $d_4(g, \text{SHIV})$ for all (or *almost all*) genomes g from HIV-1.

In order to simplify further presentation (and to consider only two values), the above condition will be replaced by the following equivalent conditions: $d_4(g, \text{HIV-2}) - d_4(g, \text{HIV-1} \setminus \{g\}) > 0$, $d_4(g, \text{SHIV}) - d_4(g, \text{HIV-1} \setminus \{g\}) > 0$. We introduce the function $\mathcal{D}$, *difference of dissimilarities*, in the following way:

$$\mathcal{D}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) = d(\mathcal{P}_1, \mathcal{P}_3) - d(\mathcal{P}_1, \mathcal{P}_2) \ .$$

Therefore, now we can state the above conditions in the following way:

$$\mathcal{D}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\}) > 0$$

$$\mathcal{D}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\}) > 0$$

These conditions were met for the genome g used in the described experiment and the results of the experiment in this form are presented in Figure 3.

Now, let us describe the next experiment in terms of function $\mathcal{D}$.

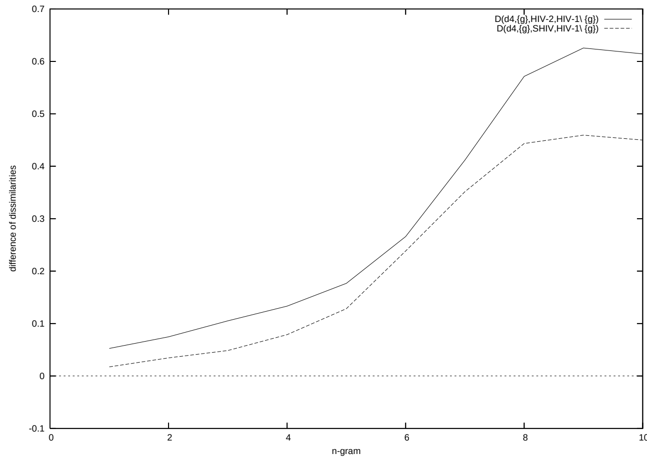

Figure 3: Results for Experiment 1 by using dissimilarity function d_4 and presented in terms of function $\mathcal{D}$

Experiment 2 *For all genome sequences g from HIV-1 compute the values:*

$$\mathcal{D}(d, \{g\}, \textit{HIV-2}, \textit{HIV-1} \setminus \{g\})$$

$$\mathcal{D}(d, \{g\}, \textit{SHIV}, \textit{HIV-1} \setminus \{g\})$$

for different n-gram lengths ($n = 1, 2, \ldots, 10$*).*

The plausible outcome *is that* $\mathcal{D}(d, \{g\}, \textit{HIV-2}, \textit{HIV-1} \setminus \{g\}) > 0$ *and* $\mathcal{D}(d, \{g\}, \textit{SHIV}, \textit{HIV-1} \setminus \{g\}) > 0$ *hold for all (or* almost all*) genomes g from HIV-1 and for all n-gram lengths (n=1,...10).*

The results of Experiment 2 for dissimilarity function d_4 are summarized in Figures 4 and 5. Figure 4 shows minimal and maximal values over all genomes g from HIV-1 for $\mathcal{D}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$. Figure 5 shows minimal and maximal values over all genomes g from HIV-1 for $\mathcal{D}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\})$. We can see that minimal values for HIV-2 are greater than 0 for all n-grams, $\text{n} = 1, \ldots, 10$ and that minimal values for HIV-2 are greater than 0 for n-grams such that $\text{n} > 3$. Although the plausible outcome of Experiment 2 is not reached for all values n (for SHIV) and minimal values for HIV-2 are in some cases close to 0, maximal values suggest that in most cases the values $\mathcal{D}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$ $\mathcal{D}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\})$ are safely above 0. In particular, we can note that the difference of dissimilarities is uniformly above 0 for n larger than 3. For small values of n, the n-gram profiles are small and "information poor" so low performance in such cases is not unexpected.

4.3 Comparing Dissimilarity Functions

The results of Experiment 2 suggest that function $\mathcal{D}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3)$ can serve as a good measure of quality for a dissimilarity function d. Of course, there

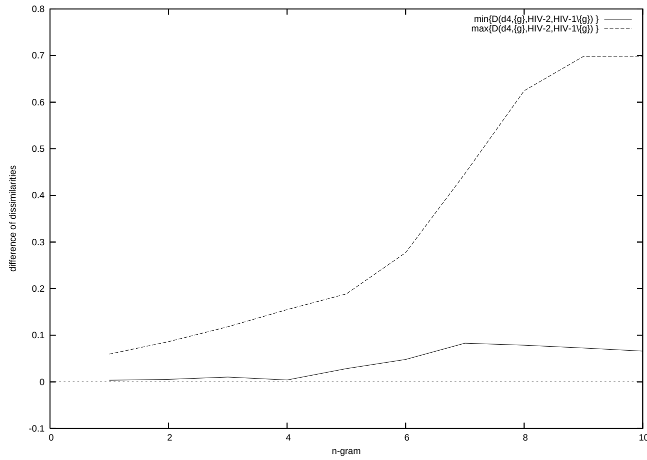

Figure 4: Minimal and maximal values over all genomes g from HIV-1 for $\mathcal{D}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$.

are many candidates for dissimilarity function d used for classifying genome sequences. In this subsection we report on experiments aimed at comparing different candidates.

Since all investigated dissimilarity functions are of additive type, it is sensible to use function $\mathcal{D}$ as a measure of their quality (since it based on subtraction between values of a dissimilarity function). However, for different dissimilarity functions their values (and hence values of function $\mathcal{D}$) can vary even for several orders of magnitude (especially for larger n). Thus, for comparing different dissimilarity functions, we introduce the function $\mathcal{Q}$, *ratio of dissimilarities*, in the following way:

$$\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) = \frac{d(\mathcal{P}_1, \mathcal{P}_2)}{d(\mathcal{P}_1, \mathcal{P}_3)} .$$

If $d(\mathcal{P}_1, \mathcal{P}_2) = 0$ and $d(\mathcal{P}_1, \mathcal{P}_3) = 0$, we define $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3)$ to be 1. If $d(\mathcal{P}_1, \mathcal{P}_3) = 0$ and $d(\mathcal{P}_1, \mathcal{P}_2) \neq 0$, we define $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3)$ to be ∞, where $\infty > r$, for any real number r.

Conditions $\mathcal{D}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) > 0$ and $\mathcal{D}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) > 0$ are equivalent to $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) > 1$ and $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) > 1$. These conditions were met for the genome g used in the above described experiment with function d_4 and the results of the experiment in this form are presented in Figure 6.

Minimal and maximal values for $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$ (for n-grams n=1,...10) are shown in Figure 7. Minimal and maximal values for $\mathcal{Q}(d_4, \{g\}$, $\text{HIV-2}, \text{SHIV} \setminus \{g\})$. These results are transformed results presented in in Figures 4 and 5. Although the minimal values for SHIV are not always greater than 1 and although minimal values for HIV-2 are in some cases close to 1, maximal values suggest that in most cases the values $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$ and $\mathcal{Q}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\})$ are safely above 1.

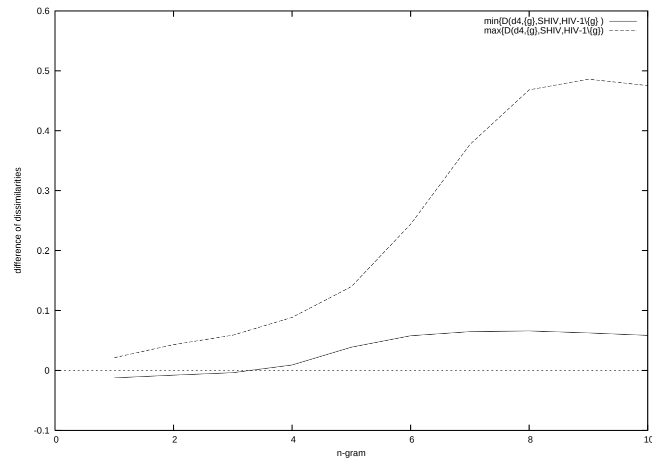

Figure 5: Minimal and maximal values over all genomes g from HIV-1 for $\mathcal{D}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\})$.

Experiment 3 *For all genome sequences g from HIV-1 compute the minimums of values:*

$$\mathcal{Q}(d, \{g\}, \mathcal{P}, \textit{HIV-1} \setminus \{g\})$$

for different n-gram lengths ($n = 1, 2, \ldots, 10$). Do it for different dissimilarity functions d and for $\mathcal{P}$=HIV-2 and $\mathcal{P}$=SHIV. The outcome *is comparison between several dissimilarity functions d. The greater are the above minimal values, the better the function is.*

The results of the Experiment 3 are shown in Figure 9. The minimums are shown only for the functions that gave best results: d_4, d_9, d_{10}, d_{16}, d_{17}, d_{18}, and d_{19}. As it can be seen from Figure 9, for all these functions, for $n \geq 4$, minimal values for $\mathcal{Q}(d, \{g\}, \mathcal{P}, \text{HIV-1} \setminus \{g\})$ are greater than 1. We find these results to be significant and encouraging. One of their consequences is: if we use any of these dissimilarity functions for classifying genome isolates (using the Corpus 1), each HIV-1 isolate will be correctly classified into the group HIV-1. The isolates are correctly classified when n-gram profiles of length 4 or higher up to 10 are used.

Having made a selection of the best candidates for dissimilarity functions, in the next experiment, we will use them for the genome classification problem.

5 Genome Sequence Classification: Experimental Results

Experiment 4 *Take two thirds genome sequences (isolates) from HIV-1 as a corpus $\mathcal{P}_{\textit{HIV-1}}$. Take two thirds genome sequences (isolates) from HIV-2 as a*

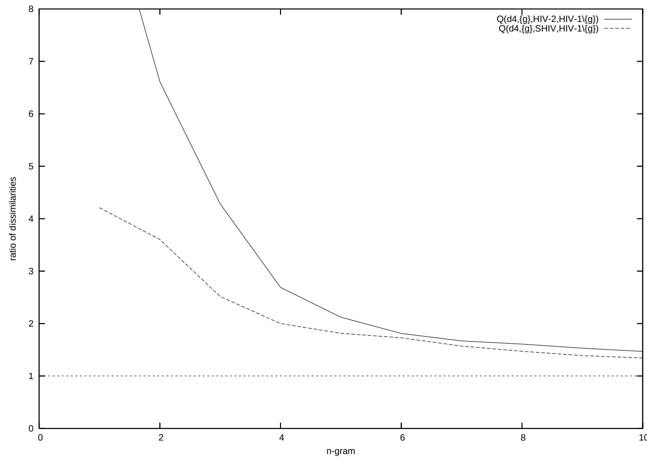

Figure 6: Results for Experiment 1 by using dissimilarity function d_4 presented via ratio of dissimilarities

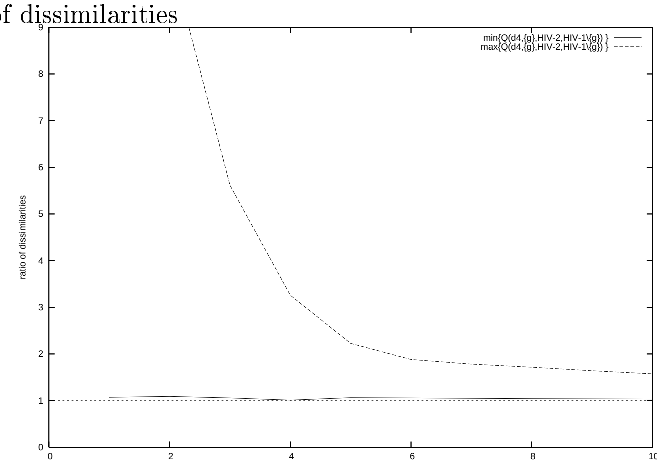

Figure 7: Minimal and maximal values for $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$

corpus $\mathcal{P}_{HIV\text{-}2}$. Take two thirds genome sequences (isolates) from SHIV as a corpus $\mathcal{P}_{SHIV}$.

Take a genome sequence (isolate) from HIV-1 $\setminus\mathcal{P}_{HIV\text{-}1}$ or from HIV-2 $\setminus\mathcal{P}_{HIV\text{-}2}$, or from SHIV $\setminus\mathcal{P}_{SHIV}$. Classify the test genome sequences using the kNN method, that is: compute the values $d(\{g\}, \mathcal{P}_{HIV\text{-}1})$, $d(\{g\}, \mathcal{P}_{HIV\text{-}2})$ and $d(\{g\}, \mathcal{P}_{SHIV})$; and classify the sequence g into one of the three classes according to the rules:

- *g belongs to HIV-1, if $d(\{g\}, \mathcal{P}_{HIV\text{-}1})$ is the smallest value*
- *g belongs to HIV-2, if $d(\{g\}, \mathcal{P}_{HIV\text{-}2})$ is the smallest value*
- *g belongs to SHIV, if $d(\{g\}, \mathcal{P}_{SHIV})$ is the smallest value.*

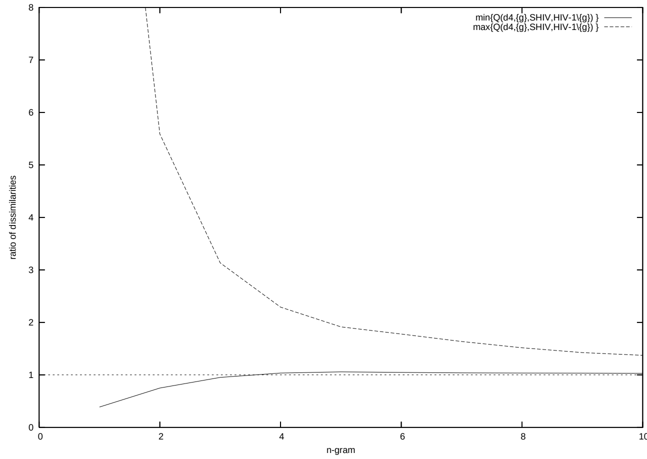

Figure 8: Minimal and maximal values for $\mathcal{Q}(d_4, \{g\}, \mathrm{SHIV}, \mathrm{HIV\text{-}1} \setminus \{g\})$

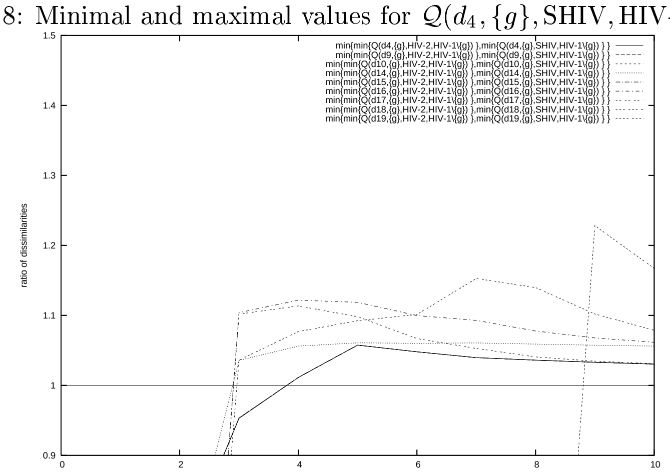

Figure 9: Results for Experiment 3

The guess is correct *if g indeed belongs to the returned set of genome sequences and* wrong *otherwise.*

The plausible outcome *is: for all genome sequences from HIV-1* $\setminus \mathcal{P}_{HIV\text{-}1}$*, HIV-2* $\setminus \mathcal{P}_{HIV\text{-}2}$ *or from SHIV* $\setminus \mathcal{P}_{SHIV}$*, all (or almost all) guesses given by a dissimilarity function d is correct.*

We performed Experiment 4 for all functions given in Section 3. Table 1 shows the results for the functions selected as good candidates for dissimilarity functions in §4.3, while Table 2 shows the results for the remaining functions.

As we can see, almost all functions given in Table 1 gave excellent performances. Almost each of them, for n ≥ 5 gave (maximal) success rate 99.6%. It is interesting to note that none of the functions reached 100% success rate for

n-gram	d_4	d_9	d_{10}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}
1	97,0	97,0	97,0	97,4	97,0	21,3	56,2	20,4	86,4
2	98,7	98,7	98,7	98,3	98,7	91,9	96,6	91,4	84,3
3	99,1	99,1	99,1	99,1	99,1	98,3	99,1	98,7	80,0
4	99,1	99,1	99,1	99,6	99,1	99,6	99,6	98,7	49,8
5	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	38,7
6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	94,0
7	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,1
8	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6
9	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6
10	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6	99,6

Table 1: Results for Experiment 4 for functions d_4, d_9, d_{10}, d_{14}, d_{15}, d_{16}, d_{17}, d_{18}, d_{19}

n-gram	d_1	d_2	d_3	d_5	d_6	d_7	d_8	d_{11}	d_{12}	d_{13}
1	3,0	5,1	4,7	57,0	63,4	57,0	63,4	3,0	5,1	16,2
2	5,1	5,1	5,5	63,0	63,4	63,0	63,4	5,1	5,1	26,0
3	5,1	5,1	9,4	62,6	63,4	63,0	63,4	5,1	5,1	9,8
4	5,1	5,1	11,1	65,1	65,1	65,1	67,7	5,1	5,1	11,1
5	5,5	5,5	10,6	66,4	71,9	67,2	81,3	5,5	5,5	11,1
6	4,7	4,3	10,6	12,3	60,4	56,2	84,3	4,7	4,3	10,6
7	1,7	1,7	10,6	1,7	1,7	1,7	1,7	1,7	1,7	10,6
8	1,7	1,7	10,2	1,7	1,7	1,7	1,7	1,7	1,7	10,2
9	1,7	1,7	10,2	1,7	1,7	1,7	1,7	1,7	1,7	10,2
10	1,7	1,7	10,6	1,7	1,7	1,7	1,7	1,7	1,7	10,2

Table 2: Results for Experiment 4 for functions d_1, d_2, d_3, d_5, d_6, d_7, d_8, d_{11}, d_{12}, d_{13}

any n. In almost all cases for which the success rate 99.6% was reached, the very same isolate was wrongly classified: the isolate AF465242.1. Simion-Human immunodeficiency virus isolate 1B3 was guessed to belong to HIV-1. It would be worthwhile to analyze this anomaly using some deeper biological knowledge.

Table 2 shows results for the remaining dissimilarity functions. All of them, including d_1, from [13] gave very poor results.

Notice, from the given tables, that bigger n does not necessarily mean better success rate. Namely, sometimes smaller n-grams can carry information that is outwith reach for larger n-grams. To obtain a higher level of confidence, one can perform multiple tests (for several values for n) while classifying a genome sequence.

An interesting observation is that the classification accuracy for functions d_5, d_6, d_7, and d_8 in Table 2 is relatively good for n $\in \{1,\dots,6\}$ and then it suddenly drops to 1.7. All of these functions use only n-grams common to two profiles, that is only such n-grams n for which $f_1(n)f_2(n) \neq 0$. As the n-grams grow longer, they become more sparse and unique for a particular profile. Thus, the number n-grams used in summation becomes so small that it become impossible to successfully detect the genome class.

As our final choice for dissimilarity function, we used d_4 for the rest of the experiments reported in this paper.

6 Hierarchical Clustering Problem

With positive results in genome sequence classification (Section 5), now we address a related, but more complex problem: hierarchical clustering of genome sequences. Our goal is to define an algorithm that can provide fully unsupervised hierarchical clustering of genome sequences. This clustering method would be based on pure statistical n-gram information, without using any additional domain knowledge, and it would rely on dissimilarity functions described in the previous text.

We introduce two clustering methods. Both, as a result, give a classification tree, usually called *genome tree.* [8] A genome tree as an unordered binary tree with genome sequences attached to its leafs. Each leaf has a genome sequence attached to it. Each node of genome tree that is not leaf, we annotate with a numerical value that characterize dissimilarity between successor nodes in left and right subtree, and hence, can be used in determining whether these two subtrees belong to the same output group or not.

Clustering Method 1 *At the beginning, the genome tree is empty. The set of input genome sequences is given as an array.*

The genome tree $\mathcal{T}$ is being built in an incremental manner in the following way (let us denote the current genome sequence by g):

- *if $\mathcal{T}$ is empty, then the root of $\mathcal{T}$ is constructed and, g is attached to it;*
- *if the root of $\mathcal{T}$ is, in the same time, leaf l, then two its successors are constructed; l is attached to the left one (and not to the root anymore) and g is attached to the right one;*
- *if the root has to subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$, then let*

$$M = \max_{g_1 \in \mathcal{T}_1, g_2 \in \mathcal{T}_2} d(g_1, g_2)$$

$$M_1 = \max_{g_1 \in \mathcal{T}_1} d(g_1, g)$$

$$M_2 = \max_{g_2 \in \mathcal{T}_2} d(g_2, g)$$

[8] E.g. http://hc.ims.u-tokyo.ac.jp/JSBi/journal/GIW03/GIW03P005/GIW03P005.html

- *if $M_1 > M$ and $M_2 > M$, then g will establish a new group: a new node is constructed with two successors. The old tree $\mathcal{T}$ is attached to the left one, while g is attached to the right one. The constructed tree is now the new tree $\mathcal{T}$.*
- *otherwise, if $M_1 < M_2$, then g will be inserted to $\mathcal{T}_1$ (recursively, using this same algorithm) and if $M_1 \geq M_2$, then g will be inserted to $\mathcal{T}_2$ (recursively, using this same algorithm).*

When the building of the tree $\mathcal{T}$ is finished, we can look for genome groups.

For different orderings of isolates processed, one can get different genome trees and different genome groups.

Within the above algorithm, we can always, for each node and its subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ keep up-to-date the value $M = \max_{g_1 \in \mathcal{T}_1, g_2 \in \mathcal{T}_2} d(g_1, g_2)$. Note that this value M for one node is always less then these values for any of its successors. Thus, for any given threshold value V, we get one genome clustering: all genomes that have one predecessor with $M < V$ belong to the same group. In this way, clustering can be fine tuned via the threshold value V. Note that, an appropriate threshold value can depend on the ordering of isolates being processed.

Notice that this clustering method is, in spirit, related to another sort of dissimilarity measures between two corpora $\mathcal{P}_1$ and $\mathcal{P}_2$ (which we do not address in this paper, but may be the subject of our future research):

$$d(\mathcal{P}_1, \mathcal{P}_2) = \max_{g_1 \in \mathcal{P}_1, g_2 \in \mathcal{P}_2} d(g_1, g_2)$$

Clustering Method 2 *The second clustering method is similar to the first method. The only difference is the way in which the values M, M_1 and M_2 are calculated. These values are calculated in the following way*

$$M = d(\mathcal{T}_1, \mathcal{T}_2)$$

$$M_1 = d(\mathcal{T}_1, g)$$

$$M_2 = d(\mathcal{T}_2, g)$$

where by $\mathcal{T}$ we mean the set of all genomes attached to leafs of $\mathcal{T}$.

A tree $\mathcal{T}$ generated using the second method does not necessarily fulfill the condition that the value M for one node is always less then these values for any of its successors.

Experiment 5 *Use the clustering methods 1 and 2 (for particular dissimilarity function d and particular value n) and apply them to the Corpus 1.*

The plausible outcome *is that the groups HIV-1, HIV-2 and SHIV will be detected and separated.*

We performed Experiment 5 for the dissimilarity function d_4 and for n= 10. Results for clustering method 1 are shown in Figure 10. The threshold value 1.735 gives very good clustering with very few incorrectly classified isolates:

isolates of HIV-1 are classified into three groups — of 391 (with one additional SHIV isolate), 22 and 32 elements, isolates of HIV-2 into one group (of 17 elements) and one of them into the group of 7 SHIV isolates. A deeper biological analysis is required for explaining why the HIV-1 isolates are separated into three groups and what makes distinction between them; why one SHIV isolate was classified along with HIV-1 isolates and why one HIV-2 (V27200.1 Human-immunodeficiency virus type 2 EHO) was classified along with SHIV isolates. The node N_3 imposes introducing of two subgroups in the node N_2 (because M in N_3 is greater than the threshold value) and hence distinguishing the node N_4, despite the fact that the value M in N_4 is less than the threshold value. For lower threshold values, one could get more fine-grained clustering.

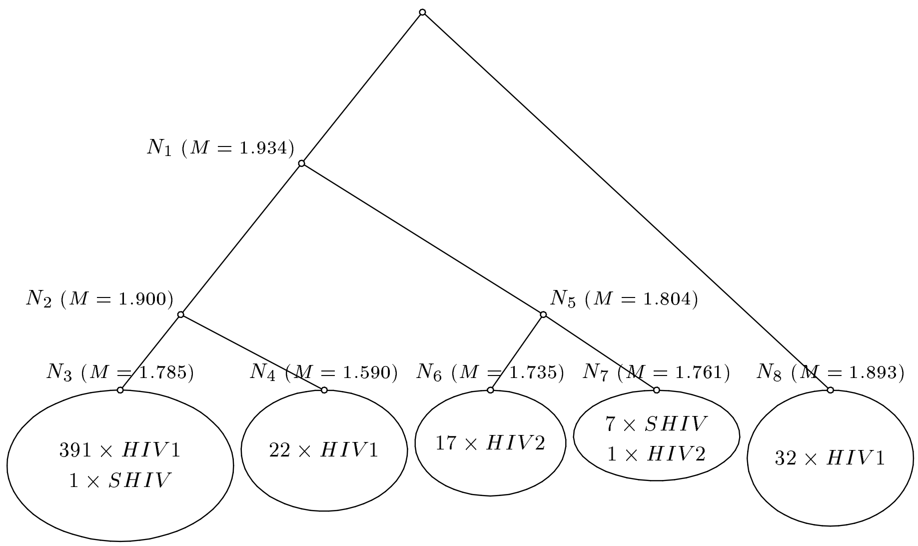

Figure 10: Results for Experiment 5 for threshold value 1.735 and for Method 1, for n=10 and for dissimilarity function d_4

Notice that in the classification problem, we had almost 100% success rate, while in the presented clustering method there were some wrong classification decisions. The main reasons for them are:

- in the clustering problem, HIV-2 and SHIV isolates are processed along with HIV-1 isolates;
- corpora are not the same as in the classification experiments; in the clustering problem, corpora are being built incrementally;
- in the clustering problem, pair-wise metrics is used, and not the one used in the classification problem.

The results from Experiment 5 for the dissimilarity function d_4 and for n= 10 and for clustering method 2 are shown in Figure 11. As already noted, a tree $\mathcal{T}$ generated using the second clustering method does not necessarily fulfill the condition value M for one node is always less then these values for any of its successors. That is why we cannot make fine grained partition based on suitably selected threshold values (which is one of the weaknesses of this method). However, for suitably selected nodes (their values M can still help in that) one can get a tree as one given in Figure 11. It can be noted that the tree produced by clustering method 2 (Figure 11) is better than the tree produce my method 1 (Figure 10) in the sense that it matches better the known class labels of the genomes, even though the number of produced leaf clusters is smaller. This can be expressed more explicitly by the majority class accuracy. Namely, if we label each cluster with the majority class genome, we see that the tree produced by method 1 creates two misclassifications, while the tree produced by method 2 has only one misclassification, giving accuracies of 0.9958 and 0.9979.

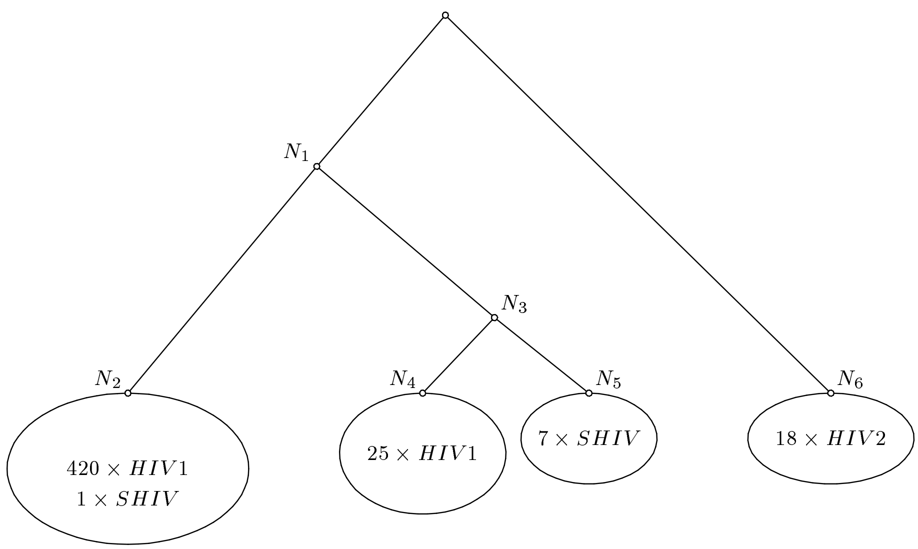

Figure 11: Results for Experiment 5 for Method 2, for n=10 and for dissimilarity function d_4

7 Related Work

This work follows some of the ideas from [13]. That paper reports on using n-grams for authorship attribution, i.e., for identifying the author of an anonymous text, or text whose authorship is in doubt. In that work, there is proposed a novel method for computer-assisted authorship attribution based on character level n-gram author profiles, which is motivated by an almost forgotten, pioneering method in 1976 [3]. We follow ideas from [13], but apply them to another domain and also change the dissimilarity functions used.

The text classification problem is also addressed using n-grams in [6], and, so-called, the out-of-place measure is used as a dissimilarity function. Very good results are reported for application of this technique to the classification of text from the *usenet* newsgroups articles. In the out-of-place measure, the frequencies in two corpora are sorted and for each n-gram the position in the sorted list is determined; then for each n-gram the absolute value of difference of these positions is calculated and then summed for all n-grams. Although the work presented in [6] is similar in spirit to the work presented here, the key difference is the different style of dissimilarity function. In future work, it would be interesting to compare these two styles of dissimilarity functions.

In [11] n-grams are used for studying languages distribution of members of "vocabulary" (e.g., standard 20 amino acids). The paper reports on the finding that some n-grams occur frequently in some organisms while occur rarely in others. Following this observation, a simple Markovian unigram model from the proteins of Aeropyrum pernix was trained. When training and test set were from the same organism, a perplexity (a variation on cross-entropy) enabled automatically distinguishing between organisms with even the simplest language model. While in [11] distributions of n-grams are considered, in the work presented here we reduce the difference of two genomes to a single number, which serves as a dissimilarity measure.

Concerning the clustering algorithms based on n-grams, we are not aware of such algorithms, and we believe that the algorithms presented here are the first of that kind.

8 Future Work

For our future work, we are planning to further develop techniques presented in this paper: to further investigate and improve the presented dissimilarity functions and the classification and clustering methods. Also, we are planning to apply the technique to other corpora and domains (not only in bioinformatics). We have already performed preliminary experiments on three genus of viruses: Tobamovirus (15 complete genomes), Alphavirus[9] (15) and Sobemovirus (9).[10] We took half of each of them as training corpus and then ran the classification process for the remaining half. The results, for n$= 1, \dots, 10$ were again excellent (they are given in Table 3). These results show that the technique proposed here can be successfully applied also to the cases where we have grouping/classification of different species. Tobamovirus, Alphavirus and Sobemovirus are three groups of viruses which belong to group "ssRNA positive-strand viruses, no DNA stage" There are also other families/genus of viruses which belong to this group like Astroviridae, Baranviridae, Benyvirus

[9] belongs to family Togaviridae

[10] Available from: `http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/ssRNA01.html`.

etc. We could perform our technique to guess in which group(family, subfamily, genus) some species belongs. The number of group (family, subfamily, genus) is not necessary to be three, it can be two, four, five or more. Using out technique it would be, also, possible to cluster many different species in groups which can correspond to their “official” group (class, family, subfamily, genus ...)

Figure 12 shows results of clustering of the viruses using the clustering methods 1 and 2, making only very few wrong classifying decisions (according to the starting, “official” classification). In the first tree, in all nodes that were not distinguished, values M are less than 1.96. However, it is not the case with the second genome tree.

n	1	2	3	4	5	6	7	8	9	10
	66.7%	100%	100%	100%	100%	88,9%	100%	100%	100%	100%

Table 3: Classification results for Tobamovirus, Alphavirus and Sobemovirus

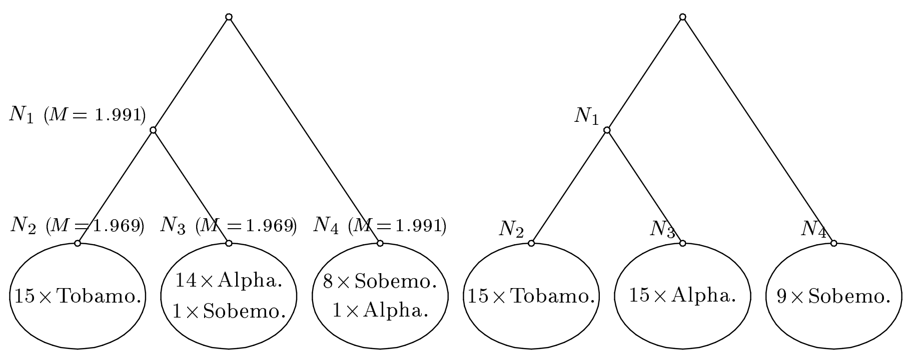

Figure 12: Clustering results for Tobamovirus, Alphavirus and Sobemovirus for threshold value 1.965 (for Methods 1 and 2, for n=10 and for dissimilarity function d_4)

9 Conclusions

In this paper we addressed the problems of automatic isolate classification, and clustering and unsupervised genome tree generation. For both of these problems we use techniques based on n-grams. For the classification problem, we follow some ideas from [13], while we changed the key ingredient of the technique — the dissimilarity function. For the clustering problem we presented two original algorithms and (to the best of our knowledge) the first two of the kind. We tested the techniques on the corpus of 463 HIV isolates and 8 SHIV isolates with complete genomes. Results obtained experimentally are very good: for suitably selected dissimilarity function, accuracy rate for the classification problem was 99.6%. For the clustering problem, both methods gave very good results for suitable selected dissimilarity function and suitable chosen threshold value. The presented experimental results suggest that the proposed techniques can be successfully used.

Our future plans include improving and testing the techniques on other corpora (one such preliminary test is presented in Section 8). We believe that the proposed technique can be used in many practical applications in biological and medical research and practice.

References

[1] E.-N. Adnan, S. Veermachaneni, and G. Nagy. Handwriting recognition using position sensitive letter n-gram matching. In *Proceedings of Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, DocLab, Rensselaer Ploytechnic Institue, Troy, NY 12180, 2003.

[2] R. Angell, G. E. Freund, and P. Willett. Automatic spelling correction using trigram similarity measure. *Information Processing and Management*, 19(4):255–261, 1983.

[3] W. R. Bennett. *Scientific and engineering problem-solving with the computer.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[4] R. Bowen. Molecular Toolkit Help. On-line at (last access Mar 2005): http://arbl.cvmbs.colostate.edu/molkit/help.html.

[5] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the 1994 Symposium On Document Analysis and Information Retrieval*, University of Nevada, Las Vegas, April 1994.

[6] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the 1994 Symposium On Document Analysis and Information Retrieval*, University of Nevada, Las Vegas, April 1994.

[7] M. Damashek. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267:843–848, February 1995.

[8] T. De Heer. Experiments with syntactic traces in information retrieval. *Information Storage Retrieval*, 10:133–144, January 1974.

[9] J. Downie. *Evaluating a Simple Approach to Musical Information Retrieval: Conceiving Melodic N-grmas as Text.* PhD thesis, University of Western Ontario, 1999.

[10] M. H. Dunham. *Data Mining Introduction and Advanced Topics.* Southern Methodist University, Pearson Education Inc., New Jersey, 2003.

[11] M. Ganapathiraju, D. Weisser, R. Rosenfeld, J. Carbonell, R. Reddy, and J. Klein-Seetharaman. Comparative n-gram analysis of whole-genome protein sequences. In *HLT'02: Human Language Technologies Conference*, San Diego, March 2002.

[12] N. C. Institute. Dictionary of Cancer Term. On-line at (last access Mar. 2005): http://www.cancer.gov/dictionary.

[13] V. Kešelj, F. Peng, N. Cercone, and C. Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03*, Dalhousie University, Halifax, Nova Scotia, Canada, August 2003.

[14] J. W. Kimball. Kimball's Bilogy Pages. On-line at (last access Mar. 2005): http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/T/Taxonomy.html.

[15] C. Marceau. Characterizing the behavior of a program using multiple-length n-grams. In *Proceedings of the 2000 workshop on New security paradigms*, pages 101–110, Ballycotton, County Cork, Ireland, 2001.

[16] T. N. H. Museum. Nature Navigator. On-line at (last access Mar. 2005): http://internt.nhm.ac.uk/jdsml/naturenavigator/naturenamed/index.dsml.

[17] J. Schmitt. Trigram-based method of language identification. In *U.S. Patent number:5062143*, October 1991.

[18] D. Tauritz. Application of n-grams. Department of Computer Science, University of Missouri-Rolla.

[19] R. University. A Glossary of Genetics. On-line at (last access Mar. 2005): http://linkage.rockefeller.edu/wli/glossary/genetics.html.

[20] J. Wisniewski. Effective text compression with simultaneous digram and trigram encoding. *Journal of Information Science*, 13:159–164, 1987.

[21] E. M. Zamora, J. J. Pollock, and A. Zamora. The use of trigram analysis for spelling error detection. *Information Processing and Management*, 17(6):305–316, 1981.