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

Andrija Tomović
Friedrich Miescher Institute for Biomedical Research
Part of Novartis Research Foundation
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 unsupervised hierarchical clustering of isolates according only to their statistical substring properties. For both of these problems we present novel algorithms based on nucleotide n-grams, with no required preprocessing steps such as sequence alignment. Results obtained experimentally are very positive and suggest that the proposed techniques can be successfully used in a variety of related problems. The reported experiments demonstrate better performance than some of the state-of-the art methods. We report on a new distance measure between n-gram profiles, which shows superior performance compared to many other measures, including commonly used Euclidean distance.

1 Introduction

The number and sizes of genome databases have grown rapidly over the last few years. A huge amount of information requires new ways for processing them and using them in efficient ways. Two of the most important problems are classification and clustering of genomes, i.e., automatically determining a group to which a previously unseen genome sequence belongs and grouping the genome sequences into a tree structure according to their similarity. 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 code 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. Using such comparisons we can estimate the place of each organism in the family tree of living species—the "tree of life." Phylogenetic inference is the process of developing hypothesis about evolutionary relatedness of organisms based on their observable characteristics. There are several techniques for constructing phylogenetic trees used in bioinformatics, including techniques on neighbor joining (e.g., [1, 2, 3]), maximum parsimony (e.g., [3, 4, 5]), maximum likelihood estimation (e.g., [6, 7, 8]), and others. Most of them are, directly or indirectly, based on multiple sequence alignment. Multiple alignment of complete large genomes can be very expensive and, in addition, it is practically impossible to align some highly plastic genomes to each other, since they can significantly differ in size, gene number and gene order. Therefore, there is a need for classification and clustering techniques that do not rely on sequence alignments. It is worth pointing that a technique for classification and clustering (like the one presented in this paper) may not be based upon biological models, but can still give very good potential for handling these problems. A recent method presented in [9] does not require multiple alignment and, in that sense, is related to the methods we propose in this paper, which also rely on n-gram analysis rather than on sequence alignments.[1]

In this paper we address the following two problems:

classification: given several families of genomes and a genome, determine the family to which it most likely belongs;

clustering: define a procedure for genome clustering using exclusively statistical substring properties of their nucleotide bases; such procedure should be effective, unsupervised, and should not require any expert knowledge for using it, which implies that it can be fully automated.

This work follows some of the ideas from [10], which 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 sequence classification using a similar method and extend the approach and ideas reported in [10].

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.

[1] The research presented in [9] was partly done over the same time as our research. We will discuss the method from [9] (its similarities and differences with respect to our work) also in Section 7.

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 sequence 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 compare algorithms proposed in this paper with other perviously published methods. In Section 9 we present some plans for future work and in Section 10 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* $\mathcal{A}$*, where* N *and* n *are positive integers, an n-gram of the sequence* S *is any n-long subsequence of consecutive tokens. The* i^{th} *n-gram of* S *is the sequence* $(s_i, s_{i+1}, ..., s_{i+n-1})$ [11].

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

For example, if $\mathcal{A}$ is English alphabet, and l string on alphabet $\mathcal{A}$, l = "life_is_a_miracle" then 1-grams are: l,i,f,_,i,s,a,m,r,c,e; 2-grams are: li,if, fe, e_, _i is, s_, _a, ...; 3-grams are: lif, ife, fe_, e_i, ... ; 4-grams are: life, ife_, fe_i, ... and so on.

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).[2]

N-grams have been successfully used for a long time in a wide variety of problems and domains, including: text compression (1953) [12], spelling error detection and correction (1962) [13, 14], optical character recognition (1967) [15], information retrieval (1973) [16], language identification (1991) [17], automatic text categorization (1994) [18], music representation (1999) [19], computational immunology (2000) [20], analysis of whole-genome protein sequences (2002) [21], authorship attribution (2003) [10], protein classification (1993) [22], (2005) [23] and phylogenetic tree reconstruction (2004)[9].

In many domains, techniques based on n-grams gave very good results. For instance, in natural language processing, n-grams can be used to distinguish between documents written in different languages in multi-lingual collections and to gage topical similarity between documents in the same language [18, 24], but also in some other problems. In this field, n-grams show some of its good features:

- robustness: relatively insensitive to spelling variations/errors;

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

- completeness: token alphabet known in advance;
- domain independence: language and topic independent;
- efficiency: one pass processing; and
- 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 $n = 6$ (e.g., $63^5 \approx 10^9$, and with larger n the n-gram cardinality grows exponentially).

2.2 Definitions of Relevant Biological Terms

Definition 2 (Genome) *A* genome *is the complete genetic material of an organism. Its size is generally given as its total number of base pairs [25].*

Definition 3 (Base pair) *A* base pair *consists of 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 [26].*

Definition 4 (Base sequence) *Base sequence is the order of nucleotide bases in a DNA molecule [26].*

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 [26].*

Definition 6 (Amino acid) *Any of a class of 20 molecules that are combined to form proteins in living things. The sequence of amino acids in a protein and hence protein function are determined by the genetic code [26].*

Definition 7 (Isolate) *Isolate is a genome or a peptide instance.*

To clarify the above definition, genome is a general term, while isolate is genome from concrete specific organism.

A genome (isolate) can be represented as a base sequence. It can be seen as a word on the 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[3]. A represents Adenosine, T — Thymidine, C —

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

Cytosine, and G — Guanosine. They are dominant because DNA sequence is made of 4 nucleotide which they represent. Other letters represent ambiguous nucleotide codes: $N \in \{A, G, C, T\}$, $R \in \{G, A\}$, $W \in \{A, T\}$, $Y \in \{C, T\}$, $M \in \{A, C\}$, $K \in \{G, T\}$, $S \in \{G, C\}$, $H \in \{A, C, T\}$, $B \in \{C, G, T\}$, $V \in \{A, C, G\}$, $D \in \{A, G, T\}$ [27]. The ambiguous-code letters appear in DNA sequences due to genetic variability.

2.3 The problems of *Classification* and *Clustering*

The problems of *classification* and *clustering* are two standard data mining tasks [28]. Given a set of objects, which is partitioned into a finite set of classes, *classification* is the task of automatically determining the class of an unseen object, based typically on a model trained on a set of objects with known class memberships. *Clustering* is the process of grouping data objects together on the basis of the features they have in common. The objects are grouped into clusters 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 are organized into a tree hierarchy, so that any child cluster is a subset of the parent cluster and the sibling clusters are disjoint. Classification and clustering are two typical examples of *supervised* and *unsupervised* data mining. Classification is *supervised* in that it typically requires labelled training data to train a classifier. Clustering is *unsupervised* since it is performed on raw input data with no prior knowledge, or supervision over method. Unsupervised learning is one of the main strengths of our hierarchical clustering methodology, and its high performance becomes even more significant when compared to some supervised methods. 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. For example, one could use external anatomy, internal anatomy, chromosomes, molecules, genome etc. [29] Automatic generation of a phylogenetic tree from a set of genomes can be regarded as a special case of hierarchical clustering.

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 similar they 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)[30]. Genome analysis gives powerful way to determine evolutionary relationships. A genome contains a large number of characters, which for related isolates provide a lot of information useful in exploring homology. This specification is in a digital form (a string of letters, i.e., a word on a given alphabet) and can be easily stored on a computer and compared to 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., 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. In the following text, by *similarity of sequences* we denote a measure of similarity of two n-gram distributions.

In [31], some pioneer methods for authorship attribution problem[4] 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 $\{1, 2, \ldots, 26\}$, i.e., all letters of English alphabet; M and N are two texts written in 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 "the standard English." The technique is based on the following idea/expectation: the smaller d(M,N), the more likely is that author of the text N is the same as the author of the text M. As the bigram frequencies of "the standard English" are obviously language-dependent parameters, another dissimilarity measure is given:

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

Following the ideas from [31], the method is adopted and further developed in [10] on the task of authorship attribution. Namely, the above dissimilarity functions (given in equations (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 [31], because all of them were reasonably frequent and the

[4] 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.

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 [10], led to the following dissimilarity measure (which we will denote by d_1 within this paper):

$$d_1(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(\frac{2 \cdot (f_1(n) - f_2(n))}{f_1(n) + f_2(n)} \right)^2 \tag{3}$$

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 [10] 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 [10], 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 d_1 measure.

$$d_2(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{2|f_1(n) - f_2(n)|}{f_1(n) + f_2(n)} \tag{4}$$

$$d_3(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \left(\frac{f_1(n) - f_2(n)}{\sqrt{f_1(n) f_2(n)} + 1} \right)^2 \tag{5}$$

$$d_4(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} \frac{|f_1(n) - f_2(n)|}{\sqrt{f_1(n) \cdot f_2(n)} + 1} \tag{6}$$

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:

$$d_5(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \left(\frac{(f_1(n) - f_2(n))(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)} \right)^2 \tag{7}$$

$$d_6(\mathcal{P}_1, \mathcal{P}_2) = \sum_{\substack{n \in profile \\ f_1(n) f_2(n) \neq 0}} \frac{|f_1(n) - f_2(n)|(f_1(n) + f_2(n))}{2 f_1(n) f_2(n)} \tag{8}$$

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

$$d_7(\mathcal{P}_1,\mathcal{P}_2) = \sum_{\substack{n\in profile\\ f_1(n)f_2(n)\neq 0}} \left(\frac{f_1(n)-f_2(n)}{\sqrt{f_1(n)f_2(n)}}\right)^2 \tag{9}$$

$$d_8(\mathcal{P}_1,\mathcal{P}_2) = \sum_{\substack{n\in profile\\ f_1(n)f_2(n)\neq 0}} \frac{|f_1(n)-f_2(n)|}{\sqrt{f_1(n)f_2(n)}} \tag{10}$$

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

$$d_9(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n\in profile} \left(A|f_1(n)-f_2(n)| + B|f_1(n)^2 - f_2(n)^2|\right) \tag{11}$$

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

$$d_{10}(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n\in profile} \left(A|f_1(n)-f_2(n)| + B|f_1(n)^2 - f_2(n)^2|\right) \tag{12}$$

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

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

$$d_{11}(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n\in profile} \left(\frac{\sqrt{2}(f_1(n)-f_2(n))}{\sqrt{f_1(n)^2+f_2(n)^2}}\right)^2 \tag{13}$$

$$d_{12}(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n\in profile} \frac{\sqrt{2}|f_1(n)-f_2(n)|}{\sqrt{f_1(n)^2+f_2(n)^2}} \tag{14}$$

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

$$d_{13}(\mathcal{P}_1,\mathcal{P}_2) = \sum_{n\in profile} \left(\frac{f_1(n)-f_2(n)}{\sqrt{f_1(n)f_2(n)}+10}\right)^2 \tag{15}$$

Although following ideas and considerations from [31] and [10], 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 [28]:

Euclidean distance:

$$d_{14}(\mathcal{P}_1,\mathcal{P}_2) = \sqrt{\sum_{n\in profile} (f_1(n)-f_2(n))^2} \tag{16}$$

Manhattan distance:

$$d_{15}(\mathcal{P}_1, \mathcal{P}_2) = \sum_{n \in profile} |f_1(n) - f_2(n)| \tag{17}$$

$$d_{16}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{2 \sum_{n \in profile} f_1(n) f_2(n)}{\sum_{n \in profile} f_1(n)^2 + \sum_{n \in profile} f_2(n)^2} \tag{18}$$

$$d_{17}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n \in profile} f_1(n) f_2(n)}{\sum_{n \in profile} f_1(n)^2 + \sum_{n \in profile} f_2(n)^2 - \sum_{n \in profile} f_1(n) f_2(n)} \tag{19}$$

$$d_{18}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n \in profile} f_1(n) f_2(n)}{\sqrt{(\sum_{n \in profile} f_1(n)^2)(\sum_{n \in profile} f_2(n)^2)}} \tag{20}$$

$$d_{19}(\mathcal{P}_1, \mathcal{P}_2) = 1 - \frac{\sum_{n \in profile} f_1(n) f_2(n)}{\min((\sum_{n \in profile} f_1(n)^2)(\sum_{n \in profile} f_2(n)^2))} \tag{21}$$

4 Evaluation of dissimilarity functions

Following the successful approach of the authorship attribution method presented in [10], we explore the use of the same or a similar technique to the analogous problem of classifying genome sequences. Given several groups of genome sequence and a genome sequence, the task is to determine a group to which the sequenced most likely belongs. The method can be described in the following way: 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 $k = 1$ [28].

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.[5] Scientists have produced SHIV, simian-human immunodeficiency virus, by putting the outer envelope of HIV onto an SIV core.[6]

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

[6] 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.

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 than SIV). The corpus is arbitrarily chosen to demonstrate our method. The method is not specially adapted for HIV/SHIV corpora, but it can be used on other genome collections as well. The method uses complete n-gram profiles (compared to filtered n-grams; e.g., [22]), and the distance function is a general function, not adapted for any specific domain.

Corpus 1 *The corpus is made out of three groups 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 ;*
- *a group of 8 isolates of SHIV.*

For all experiments presented, we used an originally developed software, but also software package Ngrams written by Vlado Kešelj.[7]

4.1 Preliminary Experiments

In order to test whether the technique proposed in [10] 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 conjecture *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 [10]). The results are shown in Figure 1[8]. 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.

[7] Ngrams package is available at http://www.cs.dal.ca/~vlado/srcperl/Ngrams/.
[8] All experimental data can be obtained on request from the first author.

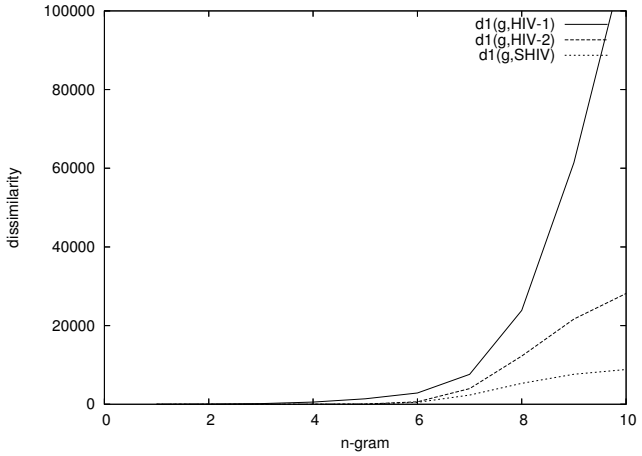

Figure 1: Results for Experiment 1 performed by using dissimilarity function d_1

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 produced positive results. They are shown in Figure 2 (left). As required, for each n ($n = 1, 2, \ldots, 10$), the value $d(\{g\}, \text{HIV-1} \setminus \{g\})$ is the smallest one.

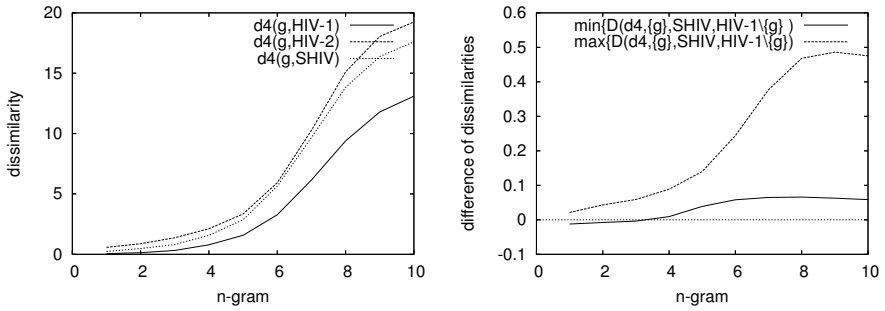

Figure 2: Results for Experiment 1 by using dissimilarity function d_4 (left) and the same results in terms of ratio of dissimilarities (right)

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 conditions will be replaced by the equivalent conditions, expressed in terms of the function $\mathcal{Q}$. The function $\mathcal{Q}$, *ratio of dissimilarities*,[9] over the

[9]Since all investigated dissimilarity functions are of additive type, it is sensible to use differences of dissimilarities (rather than ratios of dissimilarities) as a measure of their quality.

dissimilarity function d and corpora $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$, in the following way:

$$\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) = \frac{d(\mathcal{P}_1, \mathcal{P}_2)}{d(\mathcal{P}_1, \mathcal{P}_3)} \,. \tag{22}$$

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.

The condition $d(g, \mathcal{P}_1, \mathcal{P}_3) < d(g, \mathcal{P}_1, \mathcal{P}_2)$ is equivalent to $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3) > 1$ (i.e., the conditions $d_4(\{g\}, \text{HIV-1} \setminus \{g\}) < d_4(\{g\}, \text{HIV-2})$ and $d_4(\{g\}, \text{HIV-1} \setminus \{g\}) < d_4(\{g\}, \text{SHIV})$ are equivalent to the conditions $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\}) > 1$ and $\mathcal{Q}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\}) > 1$. As already shown, these conditions were met for the genome g used in the above described Experiment 1 with function d_4; the results of the experiment in terms of function $\mathcal{Q}$ are presented in Figure 2 (right).

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

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

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

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

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

Minimal and maximal values for $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$ and for $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{SHIV} \setminus \{g\})$ (for n-grams n=1,...10) are shown in Figure 3. 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, the results 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 (especially for n-grams such that n > 3)[10]. This suggests that the classification based on function d_4 will work as expected for all of the elements of HIV-1.

4.2 Comparing Dissimilarity Functions

The results of Experiment 2 suggest that function $\mathcal{Q}(d, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3)$ can serve as a good measure of quality for a dissimilarity function d. Of course, there are many candidates for dissimilarity function d used for classifying genome sequences. In this subsection we report on experiments aimed at comparing different candidates.

However, for different dissimilarity functions their values (and hence values of differences) can vary even for several orders of magnitude (especially for larger n). Therefore, for comparing different dissimilarity functions, we will use the function based on ratios of dissimilarities.

[10]For small values of n, the n-gram profiles are small and "information poor" so low performance in such cases is not unexpected.

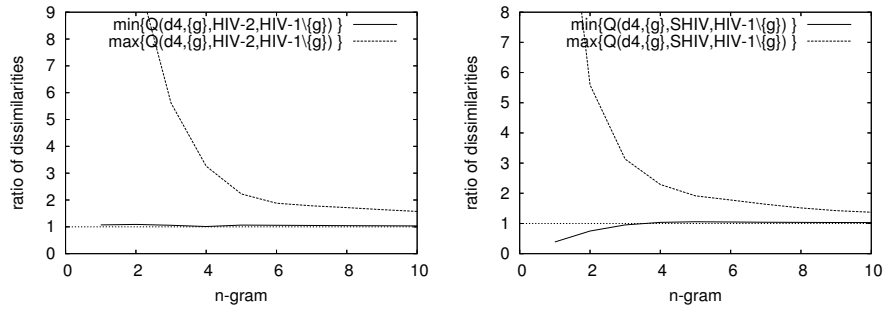

Figure 3: Minimal and maximal values (over $g \in$ HIV-1) for $\mathcal{Q}(d_4, \{g\}, \text{HIV-2}, \text{HIV-1} \setminus \{g\})$ (left) and minimal and maximal values for $\mathcal{Q}(d_4, \{g\}, \text{SHIV}, \text{HIV-1} \setminus \{g\})$ (right)

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 conjecture *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 4. 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 4, 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 *Randomly select*[11] *two thirds of the genome sequences (isolates) from HIV-1 as a new corpus $\mathcal{P}_{\textit{HIV-1}}$, two thirds from HIV-2 as the corpus*

[11]The selection algorithm is as follows: each isolate from the current set can be selected with the same probability (if there are m isolates in the set, then each can be selected with the probability $1/m$). When one element is selected, then it is deleted from the set, and the

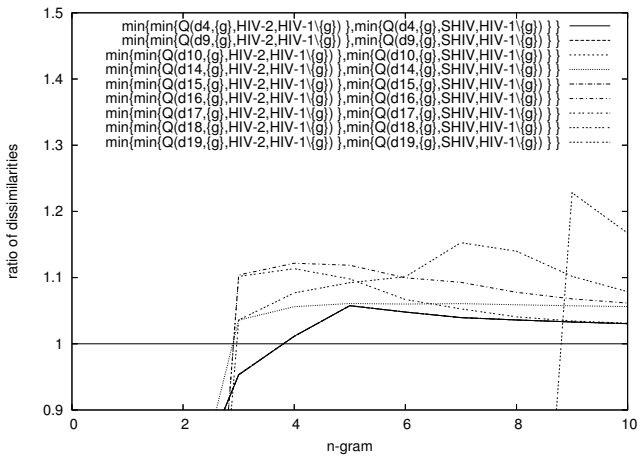

Figure 4: Results for Experiment 3

$\mathcal{P}_{HIV\text{-}2}$, and two thirds from SHIV as the corpus $\mathcal{P}_{SHIV}$.

Using sets $\mathcal{P}_{HIV\text{-}1}$, $\mathcal{P}_{HIV\text{-}2}$, and $\mathcal{P}_{SHIV}$ as training data, we use the remaining isolates as the testing data for our kNN classification method. That is, each $g \in (HIV\text{-}1 \cup HIV\text{-}2 \cup SHIV) \setminus (\mathcal{P}_{HIV\text{-}1} \cup \mathcal{P}_{HIV\text{-}2} \cup \mathcal{P}_{SHIV})$ will be classified 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.*

The guess is correct *if g indeed belongs to the returned set of genome sequences and* wrong *otherwise. For each n-gram size and dissimilarity function, we measure the average classification accuracy.*

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 §4.2, 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 $\geq$ 5 gave (maximal) success rate 99.6%. It is interesting to note that none of the functions reached 100% success rate for 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 AF465242.1 (1B3) was guessed to belong to HIV-1.

process continues until required number of isolates is selected. The random number generator provided by the C# (Visual Studio 2003) library is used. Before each experiment, the random generator was initialized using the current absolute time. The same approach was used for other experiments as well.

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} (average accuracy in percentages)

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} (average accuracy in percentages)

Indeed, in [32] authors said that in contrast to other SHIV that are composed of grater than 50% SIV sequences, the sequence AF465242.1 was HIV-1 derived. The isolate AF465242.1 is more similar to HIV-1 genomes, then other SHIV isolates in corpus 1. Our classification method confirmed this fact.

Table 2 shows results for the remaining dissimilarity functions. All of them, including d_1, from [10] 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. It is interesting to note that some functions perform well even with n-grams of size 1. However, our further experiments have shown that larger n-gram sizes provide profiles that produce more reliable and consistently good performance.

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 $\in \{1, \ldots, 6\}$ 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.

On the basis of the above experiments, the following functions gave the best results: d_4, d_9, d_{10}, d_{14} and d_{15}. However, on the basis of some additional experiments (over additional corpora), we decided to use d_4 for the rest of this work. Namely, for instance, on the corpus 2 (see section 8), function d_4 gave better results than the widely used Euclidean distance d_{14} (d_4, d_9, d_{10} and d_{15} had 100% of correct classification guesses, against 60% achieved by d_{14}). Function d_4 is good as functions: d_9, d_{10} and d_{15}. We are sure that we would get very similar results using any other of these functions.

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 objective 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. Since the method does not require any additional domain knowledge, it can be fully automated. Otherwise, the clustering task would require a lot of human expert time and may be subject to human errors. Domain specific methods need to be tuned toward a specific domain, and when the data in a domain changes, they may require readjustments. Some evidence of the soundness of this strategy based on dissimilarity of n-gram profiles was given in a few other publications, as early as 1993 [22], and this work can be regarded as a continuation of this methodology. [22] and other related methods to hierarchical clustering of genome sequences are discussed in section 7.

We introduce two clustering methods. Both, as a result, give a classification tree, usually called *genome tree.*[12] A genome tree as an unordered binary tree with genome sequences attached to its leafs. Each leaf has a genome sequence attached to it. We annotate each node of a genome tree that is not a leaf with a numerical value that characterizes dissimilarity between successor nodes in the left and right subtrees, 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*

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

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 two subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$, then let*

$$M = \max_{g_1 \in \mathcal{T}_1, g_2 \in \mathcal{T}_2} d(g_1, g_2) \tag{23}$$

$$M_1 = \max_{g_1 \in \mathcal{T}_1} d(g_1, g) \tag{24}$$

$$M_2 = \max_{g_2 \in \mathcal{T}_2} d(g_2, g) \tag{25}$$

  - *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 $M_1 \leq M$ and $M_2 \leq M$, then 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.

Note that 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 $M = \max_{g_1 \in \mathcal{T}_1, g_2 \in \mathcal{T}_2} d(g_1, g_2)$. Notice that this value M for one node is always greater 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. More precisely, determining the final resulting groups within method is performed in the following way (for a threshold value V): in one node, its value M is compared with V; if $M < V$, then the whole tree attached to this node makes one group; if $M \geq V$, then successors of the node and their values M_1 and M_2 are examined: if either of them is less then V, then these two nodes define two groups; otherwise (if $M_1 > V$ and $M_2 > V$), the above procedure applies to these successor nodes. 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):

$$(\mathcal{P}_1, \mathcal{P}_2) = \max_{g_1 \in \mathcal{P}_1, g_2 \in \mathcal{P}_2} d(g_1, g_2) \tag{26}$$

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*

$$M = d(\mathcal{T}_1, \mathcal{T}_2) \tag{27}$$

$$M_1 = d(\mathcal{T}_1, g) \tag{28}$$

$$M_2 = d(\mathcal{T}_2, g) \tag{29}$$

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 conjecture *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 5. The threshold value 1.75 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.

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.

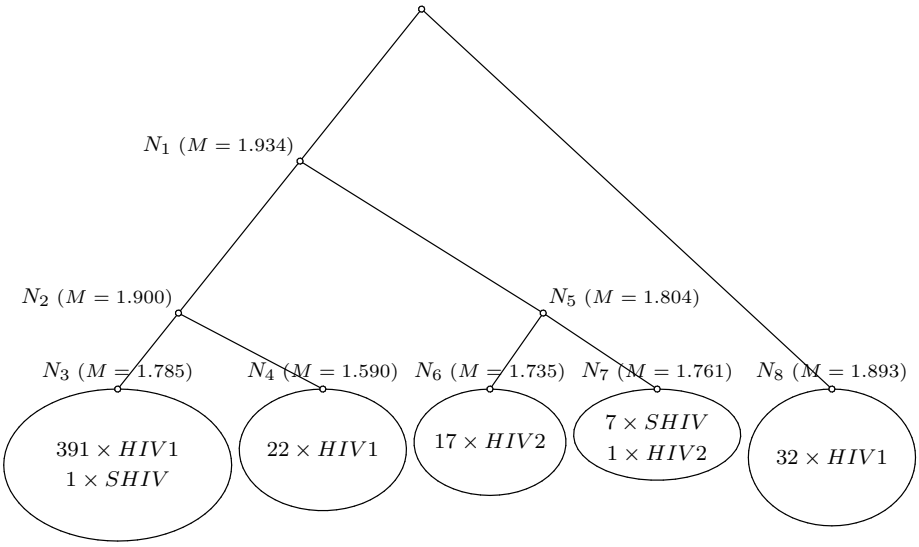

Figure 5: Results for Experiment 5 for threshold value 1.75 and for Method 1, for n=10 and for dissimilarity function d_4

The results from Experiment 5 for the dissimilarity function d_4 and for $n = 10$ and for clustering method 2 are shown in Figure 6. 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 6. The selection of distinguished nodes for easier comparison in this example of clustering is made in the following way:

- initially, each element attached to a leaf makes one group;
- then, if dominating elements in groups attached to two neighboring nodes (sibling-nodes) are equal with respect to known grouping, then then these two groups are joined together into one, bigger group, attached to the parent node of these two nodes.

The above method was used also for producing clustering trees shown in Figures 9, 10, 11 and 12 in the following text. It can be noted that the tree produced by clustering method 2 (Figure 6) is better than the tree produce my method 1 (Figure 5) 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.

It is worth pointing out that the dissimilarity functions explored in this paper (Section 3) can be used in any distance-based algorithm for producing trees, such as NJ [1], UPGMA [33] and so on.

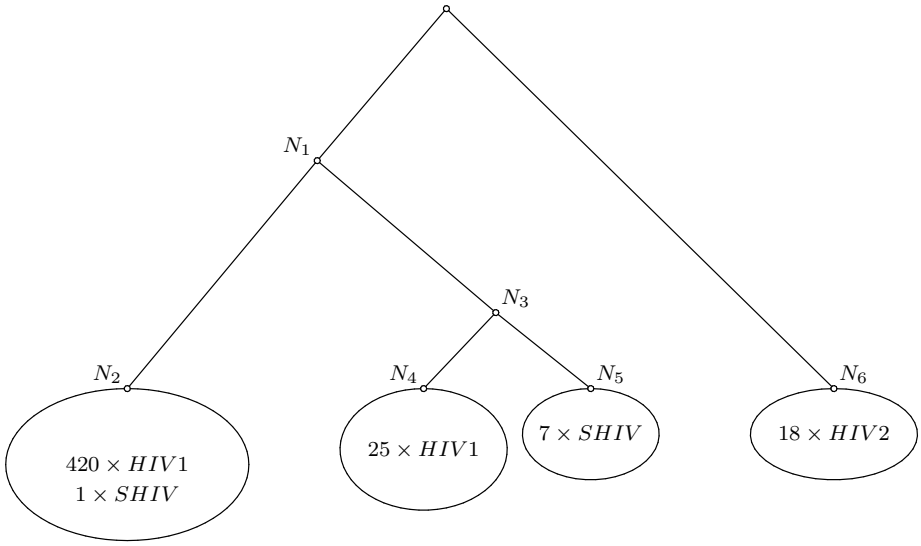

Figure 6: Results for Experiment 5 for Method 2, for n=10 and for dissimilarity function d_4

7 Related Work

This work follows some of the ideas from [10]. 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 a pioneering method in 1976 [31]. We follow ideas from [10], but apply them to another domain and also change the dissimilarity functions used.

The text classification problem is also addressed using n-grams in [34], and 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 [34] 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.

One of the earlier applications of n-gram based methods in the area of bioinformatics is reported in [22] in 1993, where a method for protein classification based on oligopeptide frequency is presented. Oligopeptide frequencies are n-gram frequencies over the alphabet of 20 amino acids. They consider shorter n-grams of up to size $n = 4$, and use the Mahalonobis distance to select a small set of 25 characteristic n-grams. In context of this work, the novelty of our classification approach is in considering the full n-gram profile and thus avoiding the n-gram selection process. The method [22] uses the standard cosine distance function, which seem to work well on a profile of selected n-grams, but not for the full n-gram profiles that we use. We conducted an experiment with this distance function and, indeed, its performance was significantly lower than several other functions presented here, most notably d_4, in the context of full n-gram profiles. More precisely, the use of cosine similarity measure reached the maximal accuracy of only 88% for 6-grams on corpus with Tobamovirus (15 complete genomes), Alphavirus(15) and Sobemovirus(9) (see Section 9), versus 100.0% achieved by measures with d_4 presented in Table 3.

In [21] 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 automatic distinguishing between organisms with even the simplest language model. While in [21] 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.

A relatively similar approach to mitochondrial genome phylogeny is applied in [35]. Unlike our approach, which relies on n-gram profile similarity measure, the results in [35] rely on a distance measure based on estimating Kolmogorov complexity. A known distance-based method for reconstructing phylogenetic trees is the *neighbor-joining* method [1].

Karlin and Burge introduced "genome signature" based on dinucleotide (bigrams) frequency[36]. They used "relative abundance" by taking the ratio $\frac{P(ab)}{P(a)*P(b)}$ where $P(ab)$ is the probability of appearance of the dinucleotide pair 'ab' and $P(a)$ is the probability of appearance of the nucleotide 'a'. This subtraction procedure has been extended to greater n (size of n-grams) in recent work [9, 37], where the authors developed a method for constructing phylogenetic tree, based on n-strings and the neighbor-joining method. The method in [9] uses genetic n-gram frequencies from which random-background frequencies are subtracted. The n-gram profiles "normalized" in this way are compared using the cosine distance function (similar to [22]), and the neighbor-joining method is applied to construct the phylogenetic tree. The novelty and significance of our work, compared to this method is in exploring a range of distance

functions, and in offering two new algorithmic methods for clustering into phylogenetic trees, which are compared to the standard NJ method. We believe that the clustering algorithms presented here are the first of this kind. Our methods produce rooted tree. Some comparison of trees produced by our methods and the method of Qi *et al.* [9] (and some other methods, as well) with the same corpus of genomes is presented in Section 8. In addition to the problem of phylogenetic tree construction, we successfully address the problem of genomes classification using the same dissimilarity functions (Section 5), which represents an additional validation of our approach. Qi et all [9] tested their method on prokaryotic genomes sequences [37] but we, also, preformed experiment with our methods on eukaryotic genome sequences and got very good results (see Section 8).

Cheng *et al.* [23] used n-grams for protein classification using the Decision Tree [38] and the Naive Bayes classifier [39]. Our approach in not based on some of the known machine learning algorithm. Rather, it is designed in the spirit of the dissimilarity functions.

In recent work [40], the authors used singular value decomposition (SVD)-based analysis to generate phylogenetic trees using whole genome protein sequences from a family of single-stranded RNA plant viruses. In this approach authors represented individual protein sequences in a high dimensional space as vector consisting of all possible tripeptide (3-gram) frequency elements, using all possible combinations of the 20 individual amino acids. All proteins vector are then organized into sparse input matrix (A) which is decomposed using SVD to three matrices (U, Σ, V). A measure of relatedness between protein pairs is obtained form the angle between pairs of protein vectors defined within the matrix (V). Using those distances phylogenetic tree is obtained by NJ method. Our approach in not based only on 3-grams, we did not use all possible n-grams (but just those which are appeared in genomes), our dissimilarity functions are different and our methods for clustering is not based on NJ method.

8 Comparison to other methods

In this section we report on comparison results between our methods and some other methods. For all used test corpora our method gave very good results, better than the compared methods.

8.1 Comparison to other methods on HIV corpus

We performed the multiple alignment of the genome sequences from corpus 1 using the CLUSTALX program [2]. The sequences are loaded in the FASTA format and the alignment is performed with the default parameters. The experiment took around 21 computer days on PC 2GHz. The distance matrix is calculated based on the divergence percentage distance function. Using the same program, a tree is calculated using the NJ method, and a rooted tree (Figure 7) is created with the absence of outgroup. The NJ method, as well as parsimony

and the probabilistic models, produces unrooted trees. Finding the root is a secondary task, which can be accomplished by adding an outgroup, or species that is known to be more distantly related to each of the remaining species than they are to each other. In the absence of a convenient outgroup, there are somewhat *ad hoc* strategies, such as picking the midpoint of the longest chain of consecutive edges, which would be expected to identify the root if deviations from molecular clock were not too great [41]. Rooted tree has directions and this directions correspond to evolutionary time. Unrooted tree does not have a root, and therefore do not specify evolutionary relationships in the same way. They do not allow the determination of ancestors and descendants. Method, which we presented in this paper, produce rooted trees. Figure 8 shows the phylogenetic rooted tree of genomes from corpus 1 using the NJ and CVTree method of Qi, Luo, and Hao [9]. Using the software package CVTree we obtained distance matrix for all sequences and then using the NJ method (software neighbor) in the PHYLIP [3] package we calculated the tree. The final phylogenetic rooted tree is obtained using the DRAWGRAM software in the PHYLIP package [3] with the absence of outgroup. It can be seen that our methods gave better results.

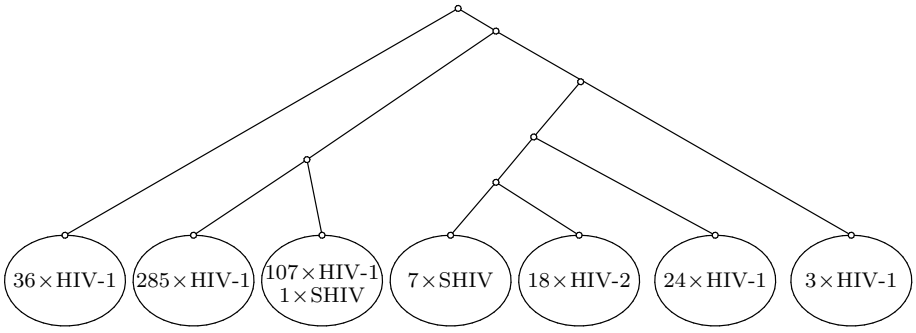

Figure 7: The phylogenetic tree of genomes from corpus 1 obtained by multiple alignment (ClustalX) and NJ method

8.2 Corpus with more distantly related genomes

We also tested our method and compared it with other methods on corpora which contain genome sequences which are more distantly related.

Corpus 2 *This corpus is made out of three genus of viruses (randomly selected from the list of all genus of viruses with available genome sequences*[13]

[13]Available from: `http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi`, as in October 2004

- *Tobamovirus — 15 complete genomes;*
- *Alphavirus — 15 complete genomes;*
- *Sobemovirus — 9 complete genomes.*

We took half of available genomes of each of those three genus as training corpus and then ran the classification process for the remaining half. The results, for $n = 1, \ldots, 10$ and function d_4 were again excellent (they are given in Table 3). These results show that the technique proposed here can be successfully applied also to the 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 etc. Our technique can be used to classify specific species into given groups (families, subfamilies, genus).

Figure 9 shows results of clustering of the viruses from corpus 2 using the clustering methods 1 and 2, making only very few wrong classifying decisions. In the first tree, in all nodes that were not distinguished, values M are less than 1.96. The distinguished nodes in the second genome tree are selected according to the method described on page 19.

Figure 10 shows the phylogenetic tree of genomes from corpus 2 obtained using the NJ method (Neighbour Joining). The multiple alignment of the complete genome sequences is performed using the CLUSTALX program [2] with default parameters. NJ method was preformed from CLUSTALX program and a rooted tree is created with the absence of outgroup. Figure 11 shows the phylogenetic rooted tree of Tobamovirus, Alphavirus and Sobemovirus genomes using the NJ from PHYLIP package and CVTree software package of Qi, Luo, and Hao [9]. The final phylogenetic rooted tree is obtained using the DRAWGRAM software in the PHYLIP package [3] with the absence of outgroup. Finally, Figure 12 shows the phylogenetic rooted tree of genomes from corpus 2 using multiple alignment (CLUSTALX) and parsimony method (software DNAPARS form PHYLIP package). Trees in Figure 10, 11 and 12 are drawn with grouped elements in a uniform way. It can be seen that our methods gave better results.

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

Table 3: Classification results for Tobamovirus, Alphavirus and Sobemovirus

8.3 Mixed corpus with large genome sequences

We have, also performed experiments with genome sequences with length more then 10 000 base pairs and not only viral genome sequences but also from higher (eukaryotic) organism. In order to test our methods and compare with other methods we defined new corpus 3 with more distantly related and longer genomes sequences.

Corpus 3 *This corpus is made out of 4 groups of genomes sequences randomly selected from NCBI Genome database*[14]*. This corpus contains:*

- *13 complete genomes of Ebola (prokaryotic-virus) with average length 19 Kb;*
- *10 complete genomes of Acelomata (eukaryotic-fungi) with average length 14 Kb;*
- *4 complete genomes of Mithondrion (eukaryotic-fungi) with average length 18 Kb and*
- *4 complete genomes of Streptomyces (eukaryotic-bacteria) with average length 12 Kb.*

We took half of each of them as training corpus and then ran the classification process for the remaining half. The results, for $n = 1, \ldots, 10$ and function d_4 were again excellent (they are given in Table 4). These results show that the technique proposed here can be successfully applied also to the cases where we have grouping/classification of different species with long genome sequences (longer then 10 Kb) and with more groups/classes (more then 3). The corpus in this experiment contains eukaryotic and prokaryotic genomes sequences and our classification method gave good results. Figure 13 shows results of clustering of genomes from corpus 3 using the clustering methods 1 and 2, making only very few wrong classifying decisions (according to the starting, "official" classification). Results of method 2 are very good, even in higher level of clustering we can see that genomes are divided in three clusters: bacteria (Streptomyces), fungi (Mitohodrion and Acelomata) and virus (Ebola). Proposed classification and clustering methods gave good results in case when we have very closely related genomes sequences (corpus 1), but also with moderate related genomes (corpus 2) and very distantly related genomes (corpus 3. Genomes from corpus 3 are significantly different in size and very large, therefore those genomes cannot be aligned to each other. It is reason why we did not present phylogenetics trees which are produced using multiple alignment. Figure 14 shows the phylogenetic rooted tree of genomes from corpus 3using the NJ from PHYLIP package and CVTree software package of Qi, Luo, and Hao [9]. It can be seen that our methods gave better results.

[14] `http://www.ncbi.nlm.nih.gov/Database/`, as in May, 2005

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

Table 4: Classification results for Ebola, Acelomata, Mitohondrion and Streptomyces

9 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. We will try to explore if (and how) optimal length of n-grams depends on the size of genomes and cardinality of alphabet. Also, we are planning to apply the technique to other corpora and domains (not only in bioinformatics). We have already preformed preliminary experiments on three groups of human diseases genes. From database OMIM [42] we randomly selected tree kind of diseases: Retinoblastoma (disease of eye), Colon cancer (belongs to group of diseases of the digestive system) and DiGeorge syndrome (belongs to the group of diseases of the immune system). We added all available human gene sequences (NCBI database) for those diseases: 7 for Retinoblastoma, 8 for Colon cancer and 6 for DiGeorge syndrome. A half of each of these groups was selected as the training corpus, and we ran the classification process on the remaining half. The results, for $n = 1, \ldots, 10$ and function d_4 were again excellent for $n > 7$ (they are given in Table 5). Figure 15 shows results of clustering of the genes using the clustering methods 1 and 2. As we can see, the results are not so positive for this kind of problem, but they are a good starting point for adapting the technique for this domain. In this case method 1 gave better results. We, also, would like also to test and adapt our method for computational prediction of microRNA, as well as computational prediction of exons and introns in eukaryotic genomes.

n	1	2	3	4	5	6	7	8	9	10
	66.7%	66.7%	66.7%	66.7%	88.9%	88.9%	66.7%	100%	100%	100%

Table 5: Classification results for human genes

10 Conclusions

In this paper we addressed the problems of automatic isolate classification, and clustering, i.e., 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 [10], while we changed the key ingredient of the technique — the dissimilarity function. For the clustering problem we presented two novel algorithms.

We tested the techniques on the corpus of 445 HIV-1, 18 HIV-2 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. Additionaly the method is tested on different corpora with more distantly related genomes and with corpus which contains mixed prokaryotic and eukaryotic genomes with length more then 10 Kb. The presented experimental results suggest that the proposed techniques can be successfully used. Compared with other methods on the same corpora, the method produced better results.

Our future plans include improving and testing the techniques on other corpora and different problem (one such preliminary test with human diseases genes is presented in Section 9). We believe that the proposed technique can be used in many practical applications in biological and medical research and practice.

11 Acknowledgments

We are grateful to prof. dr Gordana Pavlović-Lažetić, prof. dr Christian Blouin, dr Miloš Beljanski, dr Michael Rebhan and Aleksandra Nestorović, for very useful suggestions and feedback on the preliminary version of this paper.

References

[1] N. Saitou and M. Nei, The neighbor-joining method: A new method for reconstructing phylogenetic trees, *Mol. Biol. Evol.* 4 (1987) 406–425.

[2] Chenna, Ramu, Sugawara, Hideaki, Koike, Tadashi, Lopez, Rodrigo, Gibson, Toby J, Higgins, Desmond G, Thompson, and Julie D, Multiple sequence alignment with the clustal series of programs, *Nucleic Acids Res* 31 (2003) 3497–3500.

[3] J. Felsenstein, Phylip - phylogeny inference package (version 3.2), *Cladistics* 5 (1989) 164–166.

[4] A. Edwards and L. Caalli-Sforza, The reconstruction of evolution, *Annals of Human Genetics* 27 (1963) 105–120.

[5] J.H. Camin and R.R. Sokal, A method for deducing branching sequences in phylogeny, *Evolution* 19 (1965) 311–327.

[6] O. Trelles, C. Ceron, H.C. Wang, J. Dopazo and J.M. Carazo, New phylogenetic venues opened by a novel implementation of the dnaml algorithm, *Bioinformatics* 14 (1998) 544–545.

[7] J. Felsenstein, Taking variation of evolutionary rates between sites into account in inferring phylogenies, *Journal of Molecular Evolution* 53 (2001) 447–455.

[8] G.J. Olsen, H. Matsuda, R. Hagstrom and R. Overbeek, Fastdnaml: a tool for construction of phylogenetic trees of dna sequences using maximum likelihood, *Comput Appl Biosci* 10 (1994) 41–48.

[9] Ji Qi, Hong Luo and Bailin Hao, CVTree: a phylogenetic tree reconstruction tool based on whole genomes, *Nucleic Acids Research* 32 (2004) 45–47.

[10] 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.

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

[12] J.L. Wisniewski, Effective text compression with simultaneous digram and trigram encoding, *Journal of Information Science* 13 (1997) 159–164.

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

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

[15] El-Nasan Adnan, Sirharsha Veermachaneni and George 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.

[16] T. De Heer, Experiments with syntactic traces in information retrieval, *Information Storage Retrieval* 10 (1974) 133–144.

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

[18] 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.

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

[20] 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.

[21] 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.

[22] V.V. Solovyev and K.S. Makarova, A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localization, *Comput Appl Biosci.* 9 (1993) 17–24.

[23] B.Y. Cheng, J.G. Carbonell and J. Klein-Seetharaman, Protein classification based on text document classification techniques, *Proteins* 58 (2005) 955–970.

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

[25] Dictionary of Cancer Terms, National Cancer Institute. On-line at (last access Mar. 2005): http://www.cancer.gov/dictionary.

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

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

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

[29] The Natural History Museum, Nature Navigator, On-line at (last access Mar. 2005): http://internt.nhm.ac.uk/jdsml/naturenavigator/naturenamed/index.dsml.

[30] 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.

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

[32] K. Pekrun, R. Shibata, T. Igarashi, M. Reed, L. Sheppard, P. A. Patten, W. P. C. Stemmer, M. A. Martin and N. Soong, Evolution of a human immunodeficiency virus type 1 variant with enhanced replication in pig-tailed macaque cells by dna shuffling, *Journal of Virology* 76 (2002) 2924–2935.

[33] R.R. Sokal and C.D. Michener, A statistical method for evaluating systematic relationships, *Univeristy of Kanas Scientific Bulletin* 28 (1958) 1409–1438.

[34] 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.

[35] M. Li, J. H. Badger, C. Xin, S. Kwong, P. Kearney and Haoyong Zhang, An Information Based Sequence Distance and Its Application to Whole Mitochondrial Genome Phylogeny, *Bioinformatics* 17 (2001) 149–154.

[36] S. Karlin and C. Burge, Dinucleotide relative abundance extremes: a genomic signature, *Trends in Genetics* 11 (2000) 283–290.

[37] J Qi, B Wang and BI. Hao, Whole proteome prokaryote phylogeny without sequence alignment: a k-string composition approach, *Journal of Mol Evol.* 58 (2004) 2924–2935.

[38] J.R. Quinlan, C4.5 Release 8, http::/www.rulequest.com/Personal/c4.5r8.tar.gz.

[39] A. McCallum, Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering, http::/www-2.cs.cmu.edu/ mccallum/bow/.

[40] G. Stuart, K. Moffett and RF. Bozarth, A whole genome perspective on the phylogeny of the plant virus family tombusviridae, *Archives of Virology* 149 (2004) 1595–1610.

[41] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, *Biological sequence analysis*, Cambridge Univeristy Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2003.

[42] Online Mendelian Inheritance in Man, OMIM (TM), McKusick-Nathans Institute for Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD), 2000. URL: http://www.ncbi.nlm.nih.gov/omim/.

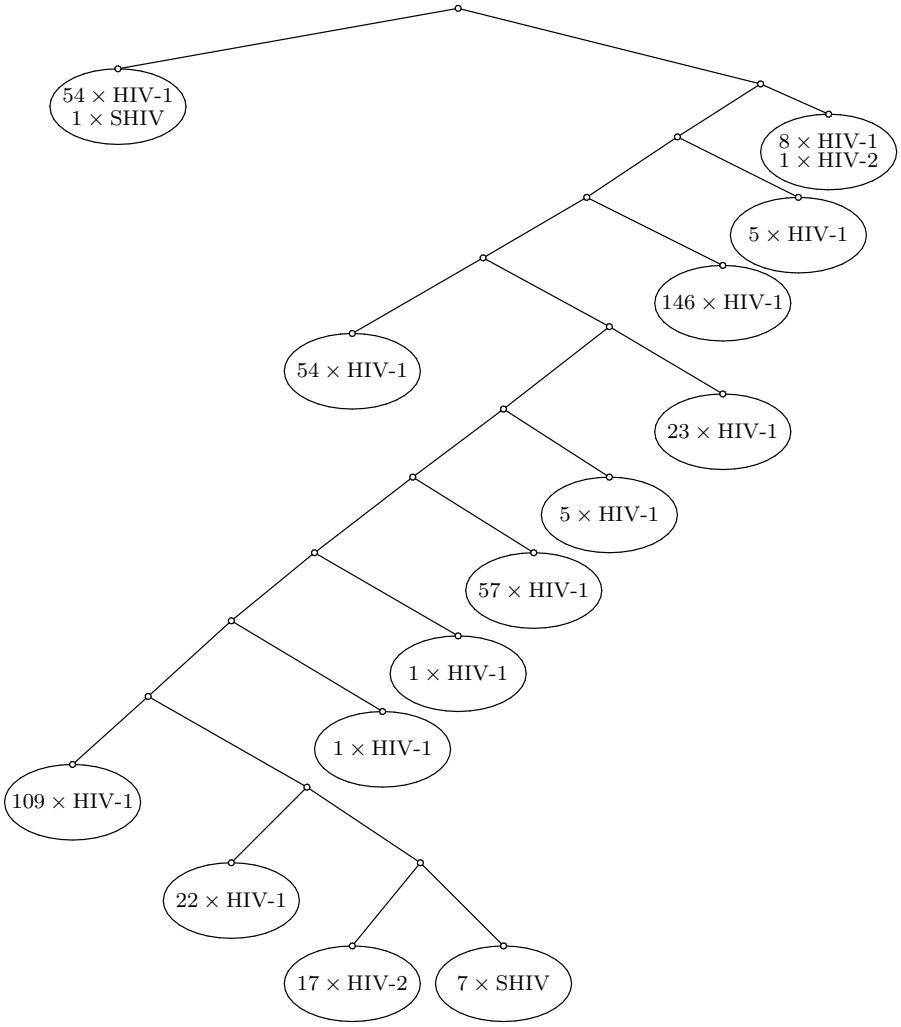

Figure 8: The phylogenetic tree of genomes from corpus 1 obtained by CVTree and NJ methods

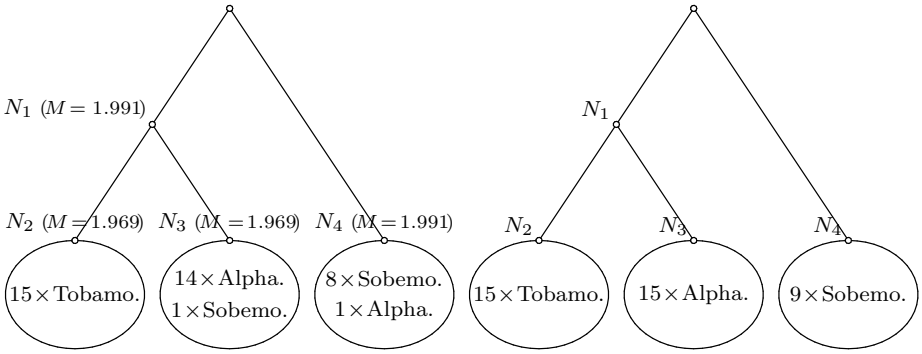

Figure 9: Clustering results for Tobamovirus, Alphavirus and Sobemovirus for Methods 1 (for threshold value 1.97) and Method 2, for n=10 and for dissimilarity function d_4)

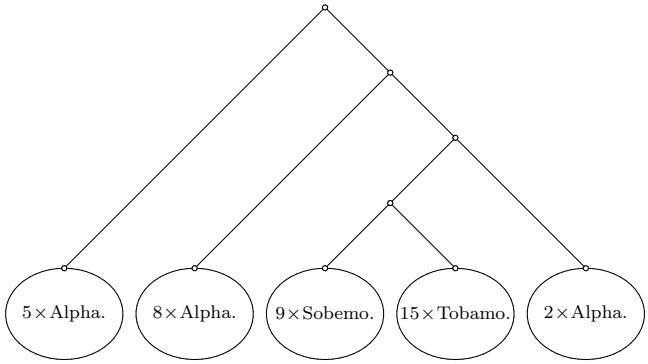

Figure 10: The phylogenetic tree of Tobamovirus, Alphavirus and Sobemovirus genomes obtained by multiple alignment (ClustalX) and NJ method

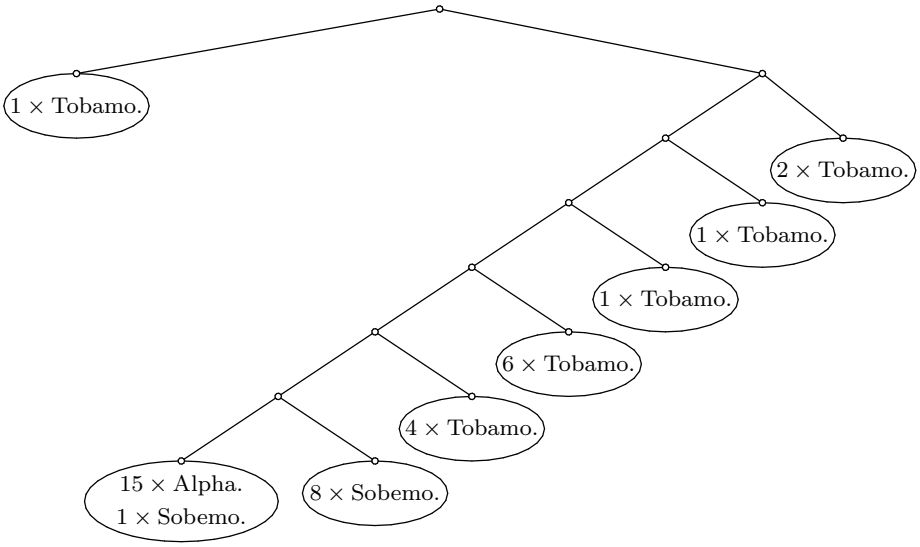

Figure 11: The phylogenetic tree of Tobamovirus, Alphavirus and Sobemovirus genomes obtained by CVTree and NJ methods

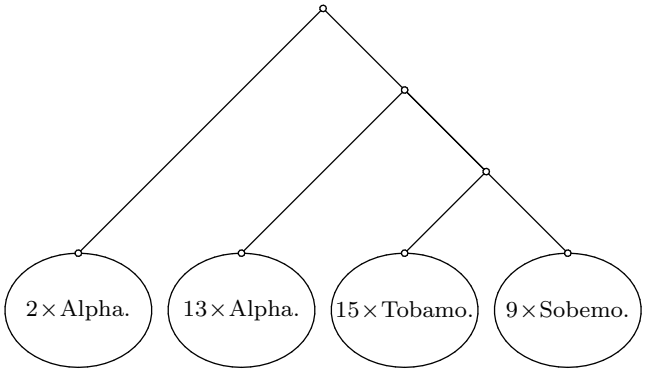

Figure 12: The phylogenetic tree of Tobamovirus, Alphavirus and Sobemovirus genomes obtained by multiple alignment(ClustalX) and parsimony method

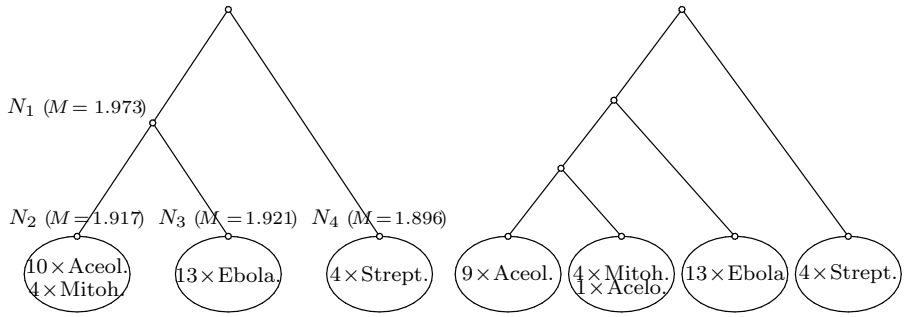

Figure 13: Clustering results for Ebola, Acelomata, Mithodnrion and Streptomyces for Methods 1 (for threshold value 1.92) and Method 2, for n=10 and for dissimilarity function d_4

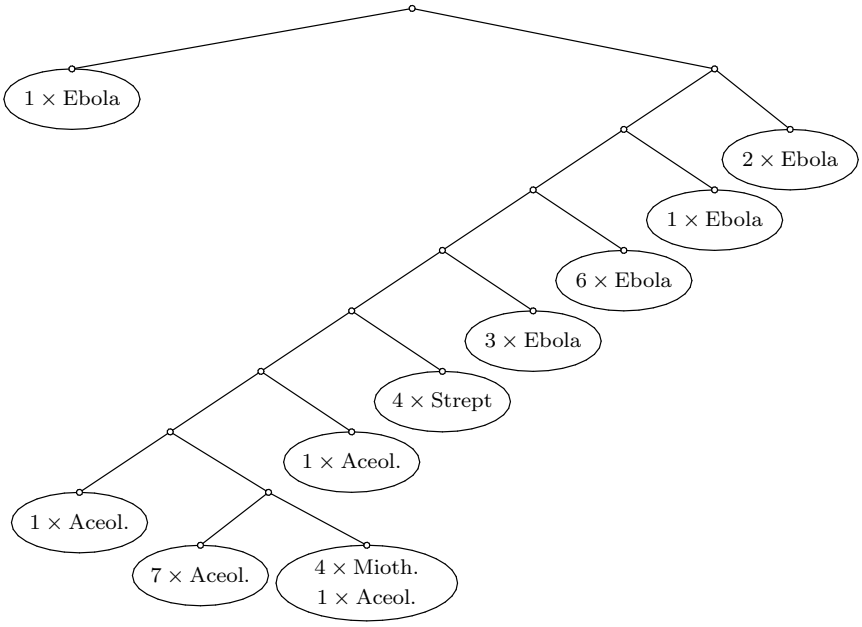

Figure 14: Clustering results for Ebola, Acelomata, Mithodnrion and Streptomyces obtained by CVTree and NJ methods

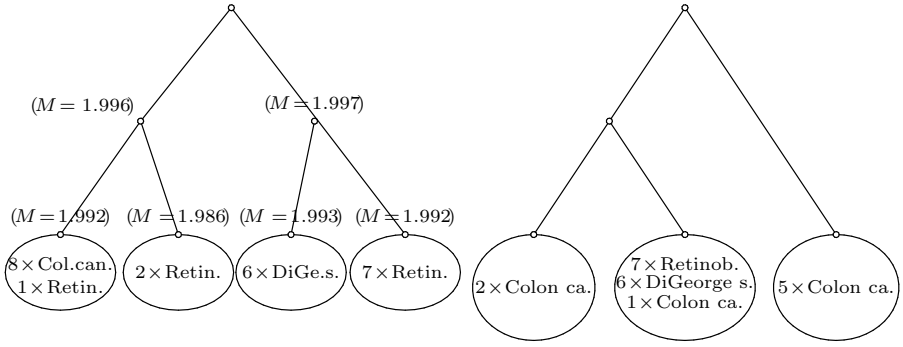

Figure 15: Clustering results for human disease genes: Retinoblastoma, Colon cancer, DiGeorge syndrome for Methods 1 (for threshold value 1.992) and Method 2, for n=10 and for dissimilarity function d_4