Local Rank Distance

Radu Tudor Ionescu
Department of Computer Science
University of Bucharest
14 Academiei, Bucharest, Romania
E-mail: raducu.ionescu@gmail.com

***Abstract*—Researchers have developed a wide variety of methods for string data, that can be applied with success in different fields such as computational biology, natural language processing and so on. Such methods range from clustering techniques used to analyze the phylogenetic trees of different organisms, to kernel methods used to identify authorship or native language from text.**

Results of such methods are not perfect and can always be improved. Some of these methods are based on a distance or similarity measure for strings, such as Hamming, Levenshtein, Kendall-tau, rank distance, or string kernel. This paper aims to introduce a new distance measure, termed Local Rank Distance (LRD), inspired from the recently introduced Local Patch Dissimilarity for images. Designed to conform to more general principles and adapted to DNA strings, LRD comes to improve over state of the art methods for phylogenetic analysis.

This paper shows two applications of LRD. The first application is the phylogenetic analysis of mammals. Experiments show that phylogenetic trees produced by LRD are better or at least similar to those reported in the literature. The second application is to identify native language of English learners. By working at character level, the proposed method is completely language independent and theory neutral. In conclusion, LRD can be used as a general approach to measure string similarity, despite being designed for DNA.

***Keywords*-rank measure; ordinal measure; string distance; closest string; DNA comparison; phylogenetic analysis; native language identification.**

I. Introduction

Researchers have developed a wide variety of methods for string data, that can be applied with success in different fields such as computational biology, natural language processing and so on. Such methods range from clustering techniques used to analyze the phylogenetic trees of different organisms, to kernel methods used to identify author or native language from text.

The phylogenetic analysis of organisms is one of the most important problems in biology. When a new organism is discovered, it needs to be placed in a phylogenetic tree in order to determine its class, order, family and species. But, the phylogenetic trees of already known organisms, obtained with different methods, are still disputed by researchers. For example, there is no definitive agreement on either the correct branching order or differential rates of evolution among the higher primates, despite the research in this area. Joining human with chimpanzee and the gorilla with the orangutan is currently favored, but the alternatives that group humans with either gorillas or the orangutan rather than with chimpanzees also have support [1].

The results of several state of the art techniques, such as those that obtain phylogenetic trees or those used for DNA sequence comparison, are inaccurate from a biological point of view, and can always be improved. Some of these methods are based on a distance measure for strings. Popular choices for recent techniques are the Hamming distance [2, 3], edit distance [4], Kendall-tau distance [5], rank distance [6, 7], and many others [8]. This paper aims to introduce a new distance measure, termed Local Rank Distance (LRD), inspired from the recently introduced Local Patch Dissimilarity for images [9]. Designed to conform to more general principles and adapted to DNA strings, LRD is more suitable to be used instead of other distances, from a biological point of view. Phylogenetic analysis experiments show that the use of LRD enables significant improvements over several state of the art methods.

This paper shows that LRD can also have applications in natural language processing. Using words is natural in text analysis tasks like text categorization (by topic), authorship identification and plagiarism detection. Perhaps surprisingly, recent results have proved that methods handling the text at character level can also be very effective in text analysis tasks. In [10] string kernels were used for document categorization with very good results. String kernels were also successfully used in authorship identification [11, 12, 13]. For example, the system described in [13] ranked first in most problems and overall in the PAN 2012 Traditional Authorship Attribution tasks. If it is reasonable to treat texts just as sequences of symbols (strings) from a practical point of view, then maybe it is also reasonable to use LRD in conjunction with different (kernel-based) learning methods for text analysis tasks. Indeed, this paper presents an application of LRD for native language identification, showing that LRD can be used as a general approach to measure string similarity, despite being designed for DNA.

The paper is organized as follows. LRD and its principles are described in section II. Section III presents the LRD definition. An algorithm to compute LRD is presented in section IV. Experiments on phylogenetic analysis are described in section V, while experiments on native language identification are described in section VI. Finally, the conclusions are drawn in section VII.

II. Approach and Discussion

The development of Local Rank Distance (LRD) is based on Local Patch Dissimilarity [9], which is a generalization of rank distance for two-dimensional input (digital images). Rank distance [14] has applications in many different fields such as bioinformatics [6, 15], computational linguistics and computer science. Rank distance can be computed fast and benefits from some features of the edit (Levenshtein) distance [16]. Despite of having such broad applications, rank distance it is not always adequate for specific data types. Local Rank Distance comes from the idea of adapting rank distance for string data, in order to capture a better similarity (or dissimilarity) between string objects, such as DNA sequences or text.

The distance between two strings can be measured with rank distance by scanning (from left to right) both strings. First, characters need to be annotated with indexes in order to eliminate duplicates. For each annotated letter, rank distance measures the offset between its position in the first string and its position in the second string. Finally, all these offsets are summed up to obtain the rank distance. Intuitively, the rank distance computes the total non-alignment score between two sequences.

Rank distance is based on mathematical principles that are not suited for specific input data, such as images or DNA sequences. The reason is that rank distance was designed to work with rankings (ordered sets of objects). Consider the following example to understand how rank distance works on text and how it can be adapted to specific input data. For two strings s_1 and s_2, the characters are first annotated. The annotation step is necessary to transform the strings into (full or partial) rankings. Now, rank distance can be computed between annotated strings $\bar{s_1}$ and $\bar{s_2}$.

Example 1: If $s_1 = CCGAATACG$ and $s_2 = TGACTCA$, the annotated strings are $\bar{s_1} = C_1C_2G_1A_1A_2T_1A_3C_3G_2$ and $\bar{s_2} = T_1G_1A_1C_1T_2C_2A_2$. The rank distance between s_1 and s_2 is

$$\begin{aligned}\Delta_{RD}(s_1, s_2) = \Delta_{RD}(\bar{s_1}, \bar{s_2}) = |1-4| + |2-6| + |3-2| \\ + |4-3| + |5-7| + |6-1| + x + x + x + x,\end{aligned}$$

where x represents the offset of characters that cannot be matched (because they are missing in the other string), such as A_3 or C_3.

In order to compute rank distance on strings, a global order is introduced by the annotation step. Notice that there are annotated characters in $\bar{s_1}$ (such as A_3 or C_3) that have no matching characters in $\bar{s_2}$. In practice, conventions are made to replace x (the offset of unmatched characters) with the maximum possible offset between two characters, or the average offset. To reduce the effect of missing characters, strings can be annotated both from left to right and from right to left. Some of these approaches are studied in [15]. Another idea with results similar to previous approaches is to consider circular DNA sequences [17]. However, these mathematical tricks are not natural from a biological point of view. Thus, one may ask whether this global order is really necessary or whether the global order is defined in the right way (for example, should strings be annotated from right to left instead of left to right). One can argue that strings can be annotated from left to right and from right to left, and the two distances obtained after annotation can be summed up. But, in order to define rank distance for specific input data (digital images, for example), answering such questions becomes difficult. One would have to ask which is the first pixel of the image, then which is the second one? There is a very large number of possibilities to define a global order on the pixels of an image. Local Patch Dissimilarity solves this problem by simply dropping the annotation step and replicating only the local phenomenon and how rank distance is computed on strings. Therefore, pixels or patches have no global order to conform to. This will also be the case of Local Rank Distance, where the annotation step is dropped, and characters in one string are simply matched with the nearest similar characters in the other string. The advantage of using no annotation immediately becomes clear. In Example 1, A_3 remains unmatched in the case of rank distance, while in the case of LRD, A_3 from $\bar{s_1}$ will be matched with A_2 from $\bar{s_2}$, as in Example 2, since there are no annotations.

If long DNA strings (as found in nature) are considered, that contain only characters in the alphabet $\Sigma = \{A, C, G, T\}$, one can observe that measuring the non-alignment score between characters that might be randomly distributed (with a uniformly distributed frequency) is not too relevant, because scores between different strings will almost be the same. Therefore, important information encoded in the DNA strings might be lost or not completely captured by a distance measure that works at character level (this is the case of Hamming or rank distance, for example). There is a similar situation in image processing. Instead of analyzing images at pixel level, computer vision researchers have developed techniques that work with patches [18, 19]. To extract meaning from image, computer vision techniques, including Local Patch Dissimilarity, look at certain features such as contour, contrast or shape. It is clear that these features cannot be captured in single pixels, but rather in small, overlapping rectangles of fixed size (e.g., 10×10 pixels), called patches. A similar idea was introduced in the early years of bioinformatics, where k-mers (also known as n-grams or substrings) are used instead of single characters. There are recent studies that use k-mers for the phylogenetic analysis of organisms [20], or for sequence alignment [21]. Analyzing DNA at substring level is also more suited from a biological point of view, because DNA substrings may contain meaningful information. For example, genes are encoded by a number close to 100 base pairs, or codons that encode the twenty standard amino acids are formed of 3-mers. LRD should also be designed to compare DNA strings based on k-mers and not on single characters.

It is interesting to note that LRD does not require an alignment of the strings by the addition of insertions and deletions, unlike other commonly used phylogenetic analysis methods. The most common methods in phylogeny deal with aligned strings, and the alignment process is itself complicated. Removing this step greatly speeds up the calculation of the distance between two strings.

Both rank distance [14] and LRD are related to the rearrangement distance [22]. The rearrangement distance works with indexed k-mers and is based on a process of converting a string into another, while LRD does not impose such global rules. LRD focuses only on the local phenomenon present in DNA: strings may contain similar k-mers that are not perfectly aligned.

III. Local Rank Distance Definition

To compute the LRD between two DNA strings, the idea is to sum up all the offsets of similar k-mers between the two strings. For every k-mer in one string, the LRD approach is to search for a similar k-mer in the other string. First, LRD looks for similar k-mers in the same position in both DNA strings. If those k-mers are similar, it sums up 0 since there is no offset between them. If the k-mers are not similar, it starts looking around the initial k-mer position in the second string to find a k-mer similar to the one in the first DNA string. If a similar k-mer is found during this process, the offset between the two k-mers is summed up to the total distance. The search goes on until a similar k-mer is found or until a maximum offset is reached. Note that the maximum k-mer offset must be set a priori and should be proportional to the size of the alphabet and to the lengths of the DNA strings. Using the maximum offset parameter ensures that the search is limited by a fixed window centered on the initial k-mer position. Finding similar matches beyond this window is costly and it may also bring unwanted noise in the process. A similar search limitation brought improvements in terms of accuracy and speed for Local Patch Dissimilarity [9, 23]. LRD is formally defined next.

Definition 1: Let $S_1, S_2 \in \Sigma^*$ be two strings with symbols (k-mers) from the alphabet Σ. Local Rank Distance between S_1 and S_2 is defined as:

$$\begin{aligned}\Delta_{LRD}(S_1, S_2) &= \Delta_{left} + \Delta_{right}\\ &= \sum_{x_s \in S_1} \min_{x_s \in S_2} \{|pos_{S_1}(x_s) - pos_{S_2}(x_s)|, m\}\\ &+ \sum_{y_s \in S_2} \min_{y_s \in S_1} \{|pos_{S_1}(y_s) - pos_{S_2}(y_s)|, m\},\end{aligned} \tag{1}$$

where x_s and y_s are occurrences of symbol $s \in \Sigma$ in strings S_1 and S_2, $pos_S(x_s)$ represents the position (or the index) of the occurrence x_s of symbol $s \in \Sigma$ in string S, and $m \geq 1$ is the maximum offset.

A string may contain multiple occurrences of a symbol $s \in \Sigma$. LRD matches each occurrence x_s of symbol $s \in \Sigma$ from a string, with the nearest occurrence of symbol s in the other string. A symbol can be defined either as a single character, or as a sequence of characters (k-mer). Overlapping k-mers are also permitted in the computation of LRD, since there is no restriction that tells where k-mers should start or end in a DNA string. Notice that in order to be a symmetric distance measure, LRD must consider every k-mer in both strings. Offsets between matched k-mers are computed with the euclidean distance based on the L_1-norm. Thus, offsets are always non-negative integer values.

To understand how LRD actually works, consider Example 2 where LRD is computed between strings s_1 and s_2 from Example 1 using 1-mers (single characters).

Example 2: Let s_1 and s_2 be defined as in Example 1, and $m = 10$ be the maximum offset. The LRD between s_1 and s_2 is given by:

$$\Delta_{LRD}(s_1, s_2) = \Delta_{left} + \Delta_{right}$$

where the two sums Δ_{left} and Δ_{right} are computed as follows:

$$\begin{aligned}\Delta_{left} &= \sum_{x_s \in s_1} \min_{x_s \in s_2} \{|pos_{s_1}(x_s) - pos_{s_2}(x_s)|, 10\}\\ &= |1-4| + |2-4| + |3-2| + |4-3| + |5-3|\\ &+ |6-5| + |7-7| + |8-6| + |9-2| = 19\\ \Delta_{right} &= \sum_{y_s \in s_2} \min_{y_s \in s_1} \{|pos_{s_1}(y_s) - pos_{s_2}(y_s)|, 10\}\\ &= |1-6| + |2-3| + |3-4| + |4-2| + |5-6|\\ &+ |6-8| + |7-7| = 12.\end{aligned}$$

In other words, Δ_{left} considers every 1-mer from s_1, while Δ_{right} considers every 1-mer from s_2. Observe that $\Delta_{LRD}(s_1, s_2) = \Delta_{LRD}(s_2, s_1)$.

LRD replicates how Local Patch Dissimilarity works on images. As Local Patch Dissimilarity is adapted to images, LRD is based on principles that make it more suited to DNA sequences. LRD essentially measures the non-alignment score between k-mers. Both LRD and Local Patch Dissimilarity can be considered as extended versions of rank distance [14], which are adapted to specific input data.

IV. Local Rank Distance Algorithm

Algorithm 1 computes the LRD between DNA strings seq_1 and seq_2 using k-mers. In this algorithm, k-mers are paired when they match exactly. Another solution that is more flexible, is to compute the similarity between k-mers using the Hamming distance. In this case, a threshold should be used to determine which k-mers are similar and which are not. This will allow LRD to pair k-mers even in the presence of different mutations in DNA. Using Hamming distance between k-mers also implies a more computationally heavy algorithm.

Algorithm 1: Local Rank Distance

```
Input:
  $seq_1$ - a DNA sequence of $l_1$ bases;
  $seq_2$ - another DNA sequence of $l_2$ bases;
  $k$ - the size of the $k$-mers to be compared;
  $m$ - the maximum offset.
Initialization:
  $dist = 0$
Computation:
  for $i = 1$ to $l_1 - k + 1$ // compute $\Delta_{left}$
      get $k$-mer$_i$ at position $i$ in $seq_1$
      $j = 0$
      $found = false$
      while $j < m$ and $found == false$
          get $k$-mer$_{left}$ at position $i - j$ in $seq_2$
          get $k$-mer$_{right}$ at position $i + j$ in $seq_2$
          if $k$-mer$_i$ $==$ $k$-mer$_{left}$ or $k$-mer$_i$ $==$ $k$-mer$_{right}$
              $dist = dist + j$
              $found = true$
          endif
          $j = j + 1$
      endwhile
      if $found == false$
          $dist = dist + m$
      endif
  endfor
  for $i = 1$ to $l_2 - k + 1$ // compute $\Delta_{right}$
      get $k$-mer$_i$ at position $i$ in $seq_2$
      $j = 0$
      $found = false$
      while $j < m$ and $found == false$
          get $k$-mer$_{left}$ at position $i - j$ in $seq_1$
          get $k$-mer$_{right}$ at position $i + j$ in $seq_1$
          if $k$-mer$_i$ $==$ $k$-mer$_{left}$ or $k$-mer$_i$ $==$ $k$-mer$_{right}$
              $dist = dist + j$
              $found = true$
          endif
          $j = j + 1$
      endwhile
      if $found == false$
          $dist = dist + m$
      endif
  endfor
Output:
  $dist$ - the Local Rank Distance between $seq_1$ and $seq_2$.
```

Algorithm 1 needs two input parameters besides the two DNA strings. The k-mer size parameter represents the number of characters (bases) for the substrings involved in the computation of LRD. The maximum offset parameter determines the size of the search window. These parameters need to be adjusted with regard to the length of the DNA strings. For example, using 10-mers for DNA strings of 100 or 200 bases is not reasonable, since finding similar 10-mers in such short DNA strings is rare, if not almost impossible. But 10-mers are suited for mitochondrial DNA strings of $15.000 - 17.000$ bases. Notice that the maximum offset parameter should be adjusted accordingly. Using 10-mers and a maximum offset that is too small, such as 50, will result in finding almost no similar 10-mers in the search window. This happens because there are $|\Sigma|^k$ possible k-mers, where $|\Sigma|$ is the size of the alphabet $\Sigma = \{A, C, G, T\}$. In the case of 10-mers, it means that there are $4^{10} = 1.048.576$ combinations.

The analysis of the computational complexity of Algorithm 1 is straightforward. Let $l = \max\{l_1, l_2\}$ be the maximum length of the two strings. The complexity of the Algorithm 1 is $O(l \times m \times k)$, when using brute (linear) search to find similar k-mers. Using advanced string searching algorithms [24] the complexity can be reduced to $O(l \times m)$. If only 1-mers are considered, the computational complexity of LRD becomes linear as Hamming distance or rank distance. If approximate matches are allowed between k-mers, by using Hamming distance to compare k-mers for example, the time complexity remains $O(l \times m \times k)$, since computing the Hamming distance between two strings (or k-mers) cannot be done faster than linear time.

V. Phylogenetic Analysis

A. Dataset

LRD is evaluated on a important problem in bioinformatics, namely the phylogenetic analysis of mammals. In the experiment, mitochondrial DNA sequence genome of 22 mammals is used. The genomes are available for download in the EMBL database (http://www.ebi.ac.uk/embl/) using the accession numbers given in Table I. Mitochondrial DNA (mtDNA) is the DNA located in organelles called mitochondria. The DNA sequence of mtDNA has been determined from a large number of organisms and individuals, and the comparison of those DNA sequences represents a mainstay of phylogenetics, in that it allows biologists to elucidate the evolutionary relationships among species. In mammals, each double-stranded circular mtDNA molecule consists of $15,000$-$17,000$ base pairs. DNA from two individuals of the same species differs by only 0.1%. This means, for example, that mtDNA from two different humans differs by less than 20 base pairs. Because this small difference cannot affect the study, the experiments are conducted using a single mtDNA sequence for each mammal.

B. Experiment

Results on this dataset of 22 mammals are also reported by [7, 15]. Similar studies were also performed by [20, 25, 26]. In this work, a hierarchical clustering technique based on LRD, using the average linkage criterion, is tested on the 22 mammals dataset. Single or complete linkage criteria show similar results, but the average linkage seems to work best in combination with LRD.

Figure 1 shows the results obtained by the hierarchical clustering method based on LRD using different k-mer sizes. It is worth mentioning that despite Figure 1 shows only

Table I
THE 22 MAMMALS FROM THE EMBL DATABASE USED IN THE PHYLOGENETIC EXPERIMENTS. THE ACCESSION NUMBER IS GIVEN ON THE LAST COLUMN.

Mammal	Latin Name	Accession
human	*Homo sapiens*	V00662
common chimpanzee	*Pan troglodytes*	D38116
pigmy chimpanzee	*Pan paniscus*	D38113
gorilla	*Gorilla gorilla*	D38114
orangutan	*Pongo pygmaeus*	D38115
Sumatran orangutan	*Pongo pygmaeus abelii*	X97707
gibbon	*Hylobates lar*	X99256
horse	*Equus caballus*	X79547
donkey	*Equus asinus*	X97337
Indian rhinoceros	*Rhinoceros unicornis*	X97336
white rhinoceros	*Ceratotherium simum*	Y07726
harbor seal	*Phoca vitulina*	X63726
gray seal	*Halichoerus grypus*	X72004
cat	*Felis catus*	U20753
fin whale	*Balaenoptera physalus*	X61145
blue whale	*Balaenoptera musculus*	X72204
cow	*Bos taurus*	V00654
sheep	*Ovis aries*	AF010406
rat	*Rattus norvegicus*	X14848
mouse	*Mus musculus*	V00711
North American opossum	*Didelphis virginiana*	Z29573
platypus	*Ornithorhynchus anatinus*	X83427

Table II
THE NUMBER OF MISCLUSTERED MAMMALS FOR DIFFERENT CLUSTERING TECHNIQUES ON THE 22 MAMMALS DATASET.

Method	Misclustered	Accuracy
Proposed by [15]	$3/22$	86.36%
Proposed by [7]	$3/22$	86.36%
LRD + sum of k-mers	$0/22$	100.00%

results obtained with even k-mer sizes (ranging from 2-mers to 10-mers) due to the lack of space, experiments with odd k-mer sizes (ranging from 3-mers to 9-mers) were also performed and the results are similar to those presented Figure 1. As the size of the k-mers grows, the dendrograms look better and better. For LRD with 8-mers (Figure 1(d)) and 10-mers (Figure 1(e)), perfect phylogenetic trees are obtained. In other words, mammals are clustered according to their biological orders: Primates, Perissodactylae, Cetartiodactylae, Rodentia, Carnivora, Metatheria, and Monotremata. The maximum offset of LRD ranges from 640 to 1000, and it is adjusted proportional to the size of k-mers used. Another dendrogram (Figure 1(f)) is obtained by summing up the Local Rank Distances with 6-mers, 7-mers, 8-mers, 9-mers and 10-mers, respectively. Again, mammals are perfectly clustered according to their orders. The idea of summing up distances obtained with different k-mers makes the hierarchical clustering method more robust. Table II shows the number of misclustered mammals of previously proposed techniques and that of the hierarchical clustering based on LRD with sum of k-mers. The best result with 100% accuracy is obtained by the hierarchical clustering based on LRD.

The accuracy level achieved by the clustering method based on LRD is better or at least comparable with state of the art methods proposed in similar studies [15, 20, 25, 26].

VI. NATIVE LANGUAGE IDENTIFICATION

A. Dataset

The dataset for the native language identification task is the TOEFL11 corpus [27]. This corpus contains 9900 examples for training, 1100 examples for development (or validation) and another 1100 examples for testing. Each example is an essay written in English by a person that is a non-native English speaker. The people that produced the essays have one of the following native languages: German, French, Spanish, Italian, Chinese, Korean, Japanese, Turkish, Arabic, Telugu, Hindi. For more details see [27].

The results presented in this work were also discussed in [28] and are obtained in the closed NLI Shared Task 2013 [29], where the goal of the task is to predict the native language of testing examples, only by using the training and development data. LRD treats documents or essays from this corpus as strings. Because LRD works at the character level, there is no need to split the texts into words, or to do any NLP-specific pre-processing. The only editing done to the texts was the replacing of sequences of consecutive space characters (space, tab, new line, and so on) with a single space character. This normalization was needed in order to not artificially increase or decrease the similarity between texts as a result of different spacing. Also all uppercase letters were converted to the corresponding lowercase ones.

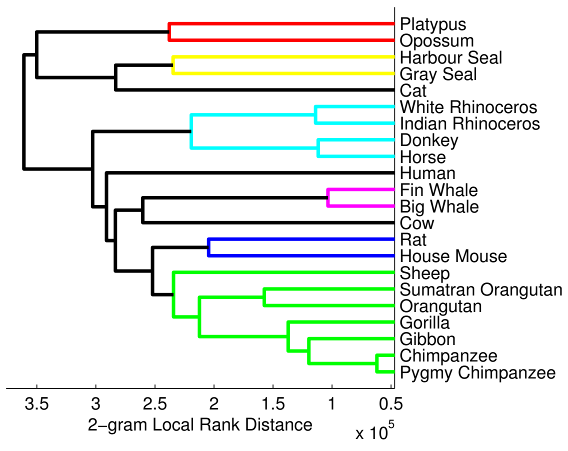
B. Learning Method

LRD measures the distance between two strings. Knowing the maximum offset (used to stop similar n-gram searching), the maximum LRD value between two strings can be computed as the product between the maximum offset and the number of pairs of compared n-grams. Thus, LRD can be normalized to a value in the $[0, 1]$ interval. By normalizing, LRD is transformed into a dissimilarity measure. LRD can be also used as a kernel, since kernel methods are based on similarity. The classical way to transform a distance or dissimilarity measure into a similarity measure is by using the Gaussian-like kernel [30]:

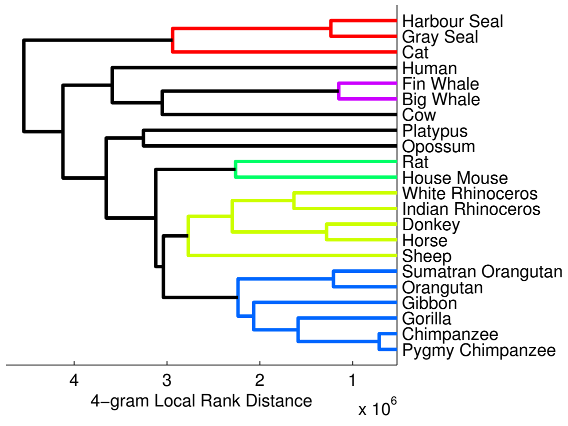
$$k(s_1, s_2) = e^{-\dfrac{LRD(s_1, s_2)}{2\sigma^2}}$$

where s_1 and s_2 are two strings. The parameter σ is usually chosen to match the number of features (characters) so that values of $k(s_1, s_2)$ are well scaled.

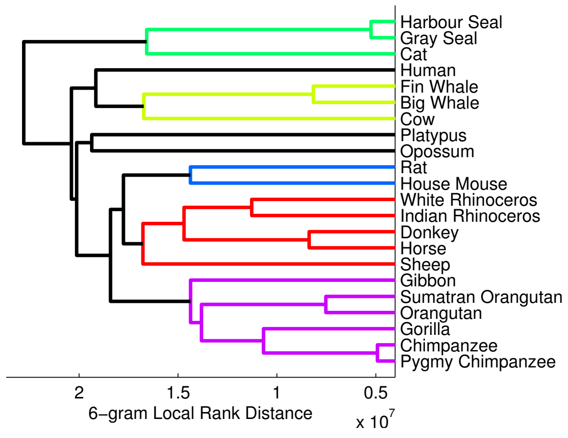
Next, state of the art learning methods such as the Support Vector Machines (SVM) or Kernel Ridge Regression (KRR) can be employed to do the learning task. Typically, these binary classifiers can solve multiclass problems by using common decomposing schemes such as: one-versus-all and one-versus-one. Preliminary experiments were conducted to



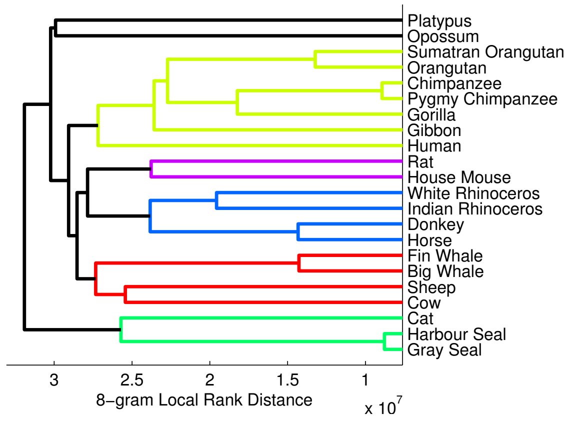
(a) Tree with LRD based on 2-mers



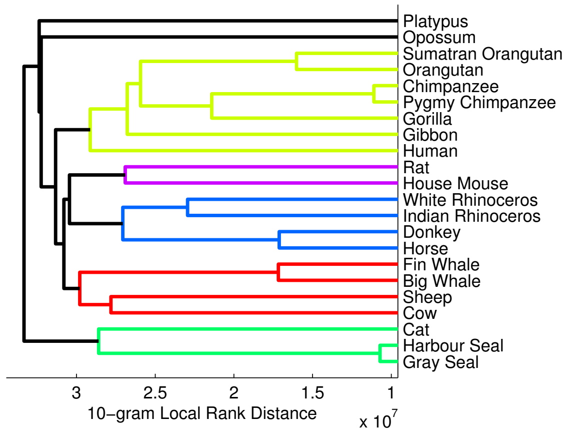
(b) Tree with LRD based on 4-mers



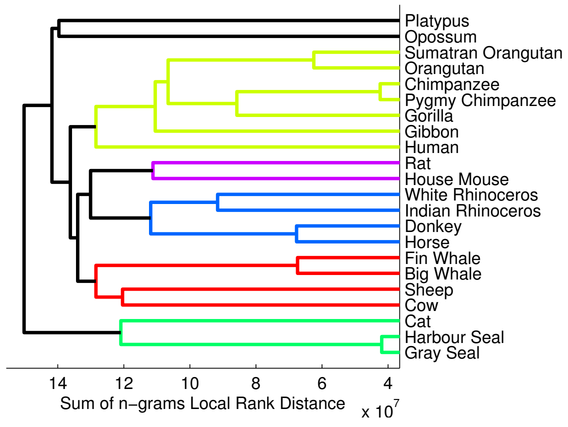
(c) Tree with LRD based on 6-mers



(d) Tree with LRD based on 8-mers



(e) Tree with LRD based on 10-mers



(f) Tree with LRD based on sum of k-mers (or n-grams)

Figure 1. Phylogenetic trees obtained for 22 mammalian mtDNA sequences using LRD based on k-mers of different size.

Table III
ACCURACY RATES, USING 10-FOLD CROSS-VALIDATION ON THE TRAINING SET, OF LRD WITH DIFFERENT n-GRAMS, WITH AND WITHOUT NORMALIZATION. NORMALIZED LRD IS MUCH BETTER, AND n-GRAM COMBINATION IMPROVES ACCURACY.

Method	**Accuracy**
KRR + K_{LRD_6}	42.1%
KRR + K_{nLRD_4}	70.8%
KRR + K_{nLRD_6}	74.4%
KRR + K_{nLRD_8}	74.8%
KRR + $K_{nLRD_{6+8}}$	**75.4%**

Table IV
ACCURACY RATES OF THE KRR BASED ON $K_{nLRD_{6+8}}$ KERNEL ON DIFFERENT EVALUATION SETS.

Evaluation Set	**Accuracy**
CV Training	75.4%
Development	76.3%
CV Training + Development	75.7%
Test	75.8%

assess the performance of each learning method and each decomposing scheme. The best result was obtained by the one-versus-all KRR and it was selected it as the learning method.

C. Parameter Tuning for LRD Kernel

Parameter tuning for LRD kernel (K_{LRD}) was done by using 10-fold cross validation on the training data. First, it is important to note that the KRR based on LRD works much better with the normalized version of LRD (K_{nLRD}). Another concern was to choose the right length of n-grams. Several n-grams such as 4-grams, 6-grams and 8-grams that are near the mean English word length of 5-6 letters were evaluated. The tests show that the LRD kernels based on 6-grams (K_{nLRD_6}) and 8-grams (K_{nLRD_8}) give the best results. In the end, the LRD kernels based on 6-grams and 8-grams are combined to obtain even better results (see Table III). More precisely, the LRD kernels based on 6-grams and 8-grams, respectively, were summed up to obtain the kernel denoted by $K_{nLRD_{6+8}}$.

Finally, the maximum offset parameter m involved in the computation of LRD was chosen so that it generates search window size close to the average number of letters per document from the TOEFL 11 set. There are 1802 characters per document on average, and m was chosen to be 700. This parameter was also chosen with respect to the computational time of LRD, which is proportional to the parameter value. Table III shows the results of the LRD kernel with different parameters cross validated on the training set. For K_{nLRD}, the σ parameter of the Gaussian-like kernel was set to 1. The reported accuracy rates were obtained with the KRR parameter λ set to 10^{-5}.

D. Results and Discussion

The LRD kernel based on sum of 6-grams and 8-grams was tested using several evaluation procedures, with results shown in Table IV. First, the kernel was tested using 10-fold cross validation on the training set. Next, it was tested on the development set. In this case, the system was trained on the entire training corpus. Another 10-fold cross validation procedure was done on the corpus obtained by combining the training and the development sets. The folds were provided by the organizers [27]. Finally, the results of the system on the NLI Shared Task test set are also given in Table IV. For testing, the system was trained on the entire training and development set, with the KRR parameter λ set to $2 \cdot 10^{-5}$. Considering that LRD is inspired from biology and that it has no ground in computational linguistics, it performed very well, by standing in the top half of the ranking of all systems submitted to the NLI competition. By disregarding features of natural language such as words, phrases, or meaning, the LRD approach has an important advantage in that it is language independent. This experiment shows that the use of LRD is not limited to computational biology.

VII. CONCLUSION

A new distance measure, termed Local Rank Distance, was introduced in this paper. Designed to conform to more general principles and adapted to DNA strings, LRD comes to improve several state of the art methods for DNA sequence analysis. Phylogenetic experiments show that trees produced by LRD are better or at least comparable with those reported in the literature [7, 15, 20, 26]. An interesting fact in favor of LRD is that it has successfully been used for native language identification [28], achieving an accuracy of $75,8\%$ in the closed NLI Shared Task [29]. This shows that LRD can be used as a general approach to measure string similarity, despite being designed for DNA.

In future work, theoretical properties of LRD must be addressed, as a proof that LRD is a distance function is necessary. The Hamming distance could be used in future work to compare k-mers in the computation of LRD. This seems more appropriate from a biological point of view, in that it allows the pairing of k-mers with mutations. A faster version of LRD, that considers only the significant or the most frequent k-mers, is also of great interest for sequence alignment or related tasks. Significant k-mers could be those that encode genes, for example.

ACKNOWLEDGMENT

The author thanks the anonymous reviewers for helpful comments. The author expresses his deepest gratitude to Denis Enachescu and Marius Popescu, both from University of Bucharest, for guidance during different stages of this research. The author also thanks Liviu Dinu from University of Bucharest for many helpful comments and discussions.

REFERENCES

[1] R. Holmquist, M. M. Miyamoto, and M. Goodman, "Higher-Primate Phylogeny - Why Can't We Decide?"

Molecular Biology Evolution, vol. 3, no. 5, pp. 201–216, 1988.

[2] M. Chimani, M. Woste, and S. Bocker, "A Closer Look at the Closest String and Closest Substring Problem," *Proceedings of ALENEX*, pp. 13–24, 2011.

[3] F. Vezzi, C. D. Fabbro, A. I. Tomescu, and A. Policriti, "rNA: a fast and accurate short reads numerical aligner." *Bioinformatics*, vol. 28, no. 1, pp. 123–124, 2012.

[4] D. Shapira and J. A. Storer, "Large Edit Distance with Multiple Block Operations," *Proceedings of SPIRE*, vol. 2857, pp. 369–377, 2003.

[5] V. Y. Popov, "Multiple genome rearrangement by swaps and by element duplications," *Theoretical Computer Science*, vol. 385, no. 1-3, pp. 115–126, 2007.

[6] L. P. Dinu and R. T. Ionescu, "An Efficient Rank Based Approach for Closest String and Closest Substring," *PLoS ONE*, vol. 7, no. 6, p. e37576, 06 2012.

[7] ——, "Clustering based on Rank Distance with Applications on DNA," *Proceedings of ICONIP*, vol. 7667, pp. 722–729, 2012.

[8] J. Felsenstein, *Inferring Phylogenies*. Sunderland, Massachusetts: Sinauer Associates, 2004.

[9] L. P. Dinu, R. Ionescu, and M. Popescu, "Local Patch Dissimilarity for Images," *Proceedings of ICONIP*, vol. 7663, pp. 117–126, 2012.

[10] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[11] C. Sanderson and S. Guenter, "Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 482–491.

[12] M. Popescu and L. P. Dinu, "Kernel methods and string kernels for authorship identification: The federalist papers case," in *Proceedings of RANLP*, Borovets, Bulgaria, September 2007.

[13] M. Popescu and C. Grozea, "Kernel methods and string kernels for authorship analysis," in *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[14] L. P. Dinu, "On the classification and aggregation of hierarchies with different constitutive elements," *Fundamenta Informaticae*, vol. 55, no. 1, pp. 39–50, 2003.

[15] L. P. Dinu and A. Sgarro, "A Low-complexity Distance for DNA Strings," *Fundamenta Informaticae*, vol. 73, no. 3, pp. 361–372, 2006.

[16] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reverseals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[17] L. P. Dinu and F. Ghetu, "Circular Rank Distance: A New Approach for Genomic Applications," *DEXA Workshops*, pp. 397–401, 2011.

[18] T. S. Cho, S. Avidan, and W. T. Freeman, "The patch transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1489–1501, 2010.

[19] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein, "The PatchMatch Randomized Matching Algorithm for Image Manipulation," *Communications of the ACM*, vol. 54, no. 11, pp. 103–110, November 2011.

[20] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitanyi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[21] P. Melsted and J. Pritchard, "Efficient counting of k-mers in DNA sequences using a bloom filter," *BMC Bioinformatics*, vol. 12, no. 1, p. 333, 2011.

[22] A. Amir, Y. Aumann, G. Benson, A. Levy, O. Lipsky, E. Porat, S. Skiena, and U. Vishne, "Pattern matching with address errors: rearrangement distances," *Proceedings of SODA*, pp. 1221–1229, 2006.

[23] R. T. Ionescu and M. Popescu, "Speeding Up Local Patch Dissimilarity," *Proceedings of ICIAP*, vol. 8156, pp. 1–10, 2013.

[24] D. E. Knuth, J. H. Morris, and V. R. Pratt, "Fast pattern matching in strings," *SIAM Journal of Computing*, vol. 6, no. 2, pp. 323–350, 1977.

[25] Y. Cao, A. Janke, P. J. Waddell, M. Westerman, O. Takenaka, S. Murata, N. Okada, S. Paabo, and M. Hasegawa, "Conflict among individual mitochondrial proteins in resolving the phylogeny of Eutherian orders," *J. Mol. Evol.*, vol. 47, pp. 307–322, 1998.

[26] A. Reyes, C. Gissi, G. Pesole, F. M. Catzeflis, and C. Saccone, "Where Do Rodents Fit? Evidence from the Complete Mitochondrial Genome of Sciurus vulgaris," *Mol. Biol. Evol.*, vol. 17, no. 6, pp. 979–983, 2000.

[27] D. Blanchard, J. Tetreault, D. Higgins, A. Cahill, and M. Chodorow, "TOEFL11: A Corpus of Non-Native English," Educational Testing Service, Tech. Rep., 2013.

[28] M. Popescu and R. T. Ionescu, "The Story of the Characters, the DNA and the Native Language," *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 270–278, June 2013.

[29] J. Tetreault, D. Blanchard, and A. Cahill, "A report on the first native language identification shared task," *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 48–57, June 2013.

[30] J. S. Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.