Compression- based Algorithms for Comparing Fragmented Genomic Sequences

Ramez Mina, Dhundy Bastola, Hesham H. Ali
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE, USA
Email: {rmina,dkbastola,hali}@unomaha.edu

***Abstract*—Sequence comparison is a fundamental tool in bioinformatics research since it helps to distinguish one sequence from another in terms of structure and function. Typically, methods such as global or local alignment are the preferred tools to measure a distance between sequence samples. Although they are often suitable tools for differentiation work, they could give erroneous results when the sequence data includes sequencing errors, gaps, repeats, and trans-locations which interfere with alignment methods. Next Generation sequence assembly tasks produce an enormous number of contigs and are reliant on alignment technologies to correctly place adjacent contigs together in the final sequence. If these alignment methods are confused by interruptions (i.e., fragmentation, gaps, mismatches or other blemishes) in the sequence data, then the assembly task may not be successful. We therefore suggest that sequence comparison can be successfully performed using alignment-free technologies and sequence compression methods which are less sensitive to inherent faults in sequencing tasks. In this paper, we evaluate different compression complexities and describe the use of compression algorithms for comparing biological sequence data. We analyze algorithm performance using protein sequence data and mitochondrial genomes with differing levels of interruption. Mitochondria is small dataset but is a well studied medium and is suitable to describe the effectiveness of the Lempel-Ziv complexity, Kolmogorov complexity using Lempel-Ziv-Welch, and Kolmogorov complexity using the Huffman coding schemes. We conclude our study by showing that sequence comparison via compression techniques is largely successful and could be a major help to high-throughput next-generation sequencing projects.**

***Keywords*-compression algorithms; Kolmogorov complexity; Lempel-Ziv complexity; tree path difference; next generation sequencing;**

I. INTRODUCTION

At the core of bioinformatics research is the comparison of sequence data. Since the 1970's, computational tools implementing local and global alignments were recommended methods to detect alterations between sequences. ClustalW and ClustalX [1] are example of such tools widely used in comparative work. However, they are not always appropriate for expansive orders of data. For this reason, heuristics such as Blast [2] and Blat [3] are alternative approaches.

Sequencing technology as well as the sequence assembly algorithms are continiously evolving. The alignment algorithms of complexity $o(n^2)$, which are used to determine the read placements for genome construction, are often slow to produce results [4]. Furthermore, as the sequencing technologies begin to produce longer reads, these algorithms may soon become obsolete and make way for other forms of sequence comparison. More importantly, the analysis by these alignment methods may be in accurate due to sequence noise such as, mutations, trans-locations and similar natural sequence altering factors [5].

Alignment-free methods are becoming increasing popular due to increased number of sequencing projects. These methods do not depend on base-by-base comparison but, instead, depend on the comparison of distributions of elemental frequencies in the sequences. For instance, the similarity of two sequences are determined by comparing frequency distributions. The generation of the frequencies and their comparison, typically a task of linear complexity, may easily take less time to run than to employ a traditional alignment algorithm of a $o(n^2)$ complexity. During a sequencing project where much data must be applied to alignment algorithms to determine adjacent reads in a genome, there is clearly a mounting demand to spend less time in the alignment bottleneck.

A. Next Generation Sequencing

Next generation sequencing, a major advancement of the Sanger sequencing technologies of the 1980's are able to generate as much data in 24 hours as several hundred Sanger-type DNA capillary sequencers [6]. They also produce a variety of different sizes of reads [7] [8]. When these reads are placed together in the correct order then a genome can be constructed. However, gaps often appear in the scaffolding that must be manually filled-in using reference sequences. This process can take a long time and could result in many inaccuracies in the completed genome.

Although recent research introduced alignment-based methods for the next generation sequencing as in Schatz M. C. [9], these new techniques did not eliminate the process of predicting the gaps between the fragments. Therefore the alignment-based method would still be a time-consuming and inaccurate approach. In particular, sequence alignment may fail to identify the distance between genomes, as the filled gaps are based on references that could be incorrect and produce inaccuracies. Therefore, alternative methods which are able to deal with reads of different sizes and orders are in demand such as alignment-free methods. These

 ISBN: 978-1-61208-260-8

methods have been highlighted in the last two decades and have attracted much attention to address the abundance of data from bioinformatics research [10].

Compression-based techniques for comparing biological sequences would be a better method for comparing these reads. The fact that compression-based techniques generally run in linear time and are capable of identifying the distance between groups of reads by an analysis of elemental frequencies may be able to create more accurate results and help to expedite the completion of assembly projects. In this paper, we intend to support our hypothesis that compression-based techniques are comparable with alignment-based methods. We provide evidence from experimental results on mitochondrial data-sets that fragmented sequence data is able to be conveniently processed for sequence comparison by compression algorithms based on Lempel-Ziv and Lempel-Ziv-Welsh, Kolmogorov and nearest-neighbor clustering.

B. Alignment-free Methods

Earlier work has been done to evaluate compression-based techniques for the comparison of mitochondrial genomes, such as the works by [11] and [12].

However, to the best of our knowledge, comparing mitochondrial genomes with interrupted or incomplete data has not yet been addressed. Here we employ data from mitochondria because it is generally of a convenient size and is generally agreed upon to cover a large breadth of sequence structure and form which may encompass many of the kinds of obstacles encountered in nuclear DNA. This paper is based on the hypothesis that, compared to alignment-based techniques, compression-based techniques will provide a better measure to determine the relatedness between genomes, which are constantly being subjected to various natural events such as rearrangements, inversion, and trans-location. Additionally, there are many genome sequences that show sequence assembly errors, many sequences that are incomplete from their unordered fragments. Therefore these events, whether natural or simply associated with the sequencing technology, may seriously affect the development of software solutions used in the automation of the genome assembly and sequence comparison process.

Consequently, closing the genomic sequencing is one of the most time-consuming steps in the entire genome sequencing and annotation pipeline. Therefore, the need for computer algorithm(s) that accommodate the features of the data and help to overcome the limitations associated with the data is highly desirable. Alignment-free methods are suited for this work since they analyze and compare elemental frequencies across sequences. Their results can be conveniently described by trees of relatedness. Therfore, in the present study, we use a mathematical method to compare these trees, which allows for a comparison of results obtained with different alignment-free or compression-based techniques for the same data-set. We used a standard algorithm for tree comparison with a modified representation of the results in order to normalize them.

C. Background on Compression

The development of data compression techniques in computer science was motivated by the need to reduce network traffic when transmitting large amounts of data. In addition, storage was also a lending factor to this development. Compression methods from computer science became popular research topics in bioinformatics research when it was noted that DNA, appearing random, could not be easily efficiently compressed by Gzip or Bzip2 [13]. DNA has since been shown not to be as random as previously thought [14], and can be applied to compression techniques using only two bits. It was established that DNA had a syntax for coding genes [15] and furthermore that this information could be applied to compression techniques. These techniques derived elemental frequencies from the syntax to be compared with other kinds of sequence data.

Lempel and Ziv, along with Kolmogorov, introduced the concept of compression complexity, which later became the gateway for introducing the Lempel-Ziv compression technique in 1976. It is the complexity of a sequence that enables us to evaluate whether or not a particular compression algorithm is applicable. In [13] and [16] it was discussed that, in higher eukaryotes, biological sequences have tandem repeats and multiple copies of genes, which make them a good subject for compression techniques. In addition to these properties, DNA sequences are rich with other properties that are hidden within the sequences. These properties could be useful for compression since they include natural evolutionary events such as random mutations, translocation, cross-overs, and reversal events. In [16] it was discussed how compression would address such properties and take advantage of them to compress the sequences. The compression would then reflect the relatedness between the sequences. By concatenating two sequences we would be able to compress them effectively if they share common information.

D. Kolmogorov Complexity

For any two sequences x and y, we define conditional Kolmogorov complexity, $K(x|y)$, as the shortest binary program that computes x in terms of y [4]. Also, the Kolmogorov complexity of a sequence x we defined as $K(x)$ or $K(x|\lambda)$, where λ signifies an empty string. We also define the information distance ID between two sequences x and y as, $ID(x,y) = max\{K(x|y), K(y|x)\}$

The Kolmogorov complexity of a sample of information, such as text, is a measure of the computational resources needed to specify the sample. Kolmogorov theory is a concept more than a measure and does not offer a metric value that could be used in constructing a tree of relatedness. The Universal Similarity Metric (USM) was thus implemented to

 ISBN: 978-1-61208-260-8

measure the complexity of Kolmogorov, where we represent the compression of sequence x by $C(x)$ and the compression of sequence x appended by sequence y by $C(yx)$. Three practical approximations of Kolmogorov were suggested, namely:

- Universal Compression Distance/Dissimilarity (UCD)
- Normalized Compression Distance/Dissimilarity (NCD)
- Compression Distance/Dissimilarity (CD).

In mathematical terms, we have the following,

$$UCD(x,y) = \frac{max\{|C(xy)| - |C(x)|, |C(yx)| - |C(y)|\}}{max\{|C(x)|, |C(y)|\}}$$

$$NCD_1 = \frac{\{|C(xy)| - min\{|C(x)|, |C(y)|\}\}}{max\{|C(x)|, |C(y)|\}}$$

$$CD(x,y) = \frac{min\{|C(xy)|, |C(yx)|, |C(x)| + |C(y)|\}}{|C(x)| + |C(y)|}$$

$$then,$$

$$NCD(x,y) = min\{NCD_1(x,y), NCD_1(y,x)\}$$

E. Lempel-Ziv Complexity

Consider the sequence $S = AACGTACC$. Some of its histories, or fashions of grouping and placing adjacent components of the sequence together as defined by [5], are the following:

1) $H(S) = A \cdot A \cdot C \cdot G \cdot T \cdot A \cdot C \cdot C$
2) $H(S) = A \cdot AC \cdot G \cdot T \cdot A \cdot C \cdot C$
3) $H(S) = A \cdot AC \cdot G \cdot T \cdot ACC.$

The exhaustive history, presented by the same authors, is defined as the history where no substring has a repetition, and no substring can be found in the whole sequence before this substring. This means that if a substring is chosen at the i^{th} position, then the sequence of characters before this position will not contain an occurrence of this substring. A mathematical representation for this concept could be the resulting number of components which making up an entire sequence, here called a *unique history*. Upon examining the histories in the previous example, we note that the first two cannot be exhaustive histories since 'A' and 'C' are repeated, but the third one is exhaustive. The Lempel-Ziv-complexity is defined as the least exhaustive history of a sequence and is noted as $C(sequence)$ implying the number of components in a an exhaustive history of a sequence. Consider the following three sequences:

- $S = AACGTACCATTG$
- $R = CTAGGGACTTAT$
- $Q = ACGGTCACCAA$

With the component words, separated by dots making up the entire sequence, the exhaustive histories for the sequences are the following:

- $HE(S) = A \cdot AC \cdot G \cdot T \cdot ACC \cdot AT \cdot TG$
- $HE(R) = C \cdot T \cdot A \cdot G \cdot GGA \cdot CTT \cdot AT$
- $HE(Q) = A \cdot C \cdot G \cdot GT \cdot CA \cdot CC \cdot AA$

The total number of components of these exhaustive histories are the following:

- $c(S) = c(R) = c(Q) = 7$

The exhaustive histories for SQ and RQ are:

- $HE(SQ) = A \cdot AC \cdot G \cdot T \cdot ACC \cdot AT \cdot TG \cdot ACGG \cdot TC \cdot ACCAA$
- $HE(RQ) = C \cdot T \cdot A \cdot G \cdot GGA \cdot CTT \cdot AT \cdot ACG \cdot GT \cdot CA \cdot CC \cdot AA$
- $c(RQ) = 12$ and $c(SQ) = 10$

This implies that S is closer to Q than R is to Q, which is evident by the following:

- $S = AACGTACCATTG$
- $Q = ACGGTCACCAA$
- $Q = ACGGTCACCAA$
- $R = CTAGGGACTTAT$

Lempel-Ziv complexity itself is not a distance measure between sequences. It is instead a form of distance measurement.

Distance measure 1:
$d(S,Q) = max\{c(SQ) - c(S), c(QS) - c(Q)\}$

Distance measure 2:
$d^*(S,Q) = \frac{max\{c(SQ)-c(S), c(QS)-c(Q)\}}{max\{c(S), c(Q)\}}$

Distance Measure 3:
$d_1(S,Q) = c(SQ) - c(S) + c(QS) - c(Q)$

Distance Measure 4:
$d_1^*(S,Q) = \frac{c(SQ)-c(S)+c(QS)-c(Q)}{c(SQ)}$

These distances would be the same as the scoring values of any sequence alignment method and would be used in building the tree of relatedness of the data-set. Notice that the shorter the numerical distance, the closer the pair sequences are to each other.

The rest of this paper is organized as follows:

- **Section 2**. Two different compression techniques are tested with different parameters, namely Kolmogorov complexity and Lempel-Ziv complexity, on a nucleic acid sequence (mitochondria) and on protein sequences from different species. Random incomplete genome fragments are then generated with different percentages, where these fragments could be ordered or disordered. Trees are then generated for both compression-based techniques and for multiple sequence alignment, a method of comparing similarity across more that two sequences.
- **Section 3**. These trees are compared against the standard tree, which serves as a reference for each data-set, while calculating the distance between the two trees.
- **Section 4**. We present our conclusions regarding the usefulness of the compression-based algorithms for sequence comparison.

 ISBN: 978-1-61208-260-8

II. METHODOLOGY

We start with the experimental design, then collect the data-sets, and finally apply the steps of each experiment to evaluate our hypothesis.

A. Experimental Design

The experiments consisted of three phases including data-set assembly, scoring matrices compilation and calculation, and evaluation of results.

The Lempel-Ziv-Welch and Huffman compression algorithms which rely on prefix coding, were the seeds for Kolmogorov complexity metrics. The Lempel-Ziv complexity has its own algorithm to measure the complexity before seeding it to the metrics, implemented with a modified algorithm published by Borowska et al. [17]. All four of the distance measures were calculated for Lempel-Ziv complexity.

B. Dataset Collection

Our data-sets varied according to the experiment. The first experiment used both protein and whole genome mitochondrial sequences (CK-36-PDB and AA-15-DNA). The other four experiments used only a mitochondrial data-set (AA-15-DNA). These two data-sets were used to test the viability of compression techniques in comparing biological sequences. This data has been previously used and consists of 36 protein domains in the amino acid sequence set and the genome data consists of complete DNA sequences 15 different organisms [4].

The second experiment focused on comparing incomplete sequences, containing only 10 - 90 percent of total genome sequences, and the start positions varied as shown (Figure 1). The third experiment evaluated incomplete genomes made from separate segments, but the total length contained 10 - 90 percent of the whole genomes (Figure 2). The fourth experiment explored genomes that were 10 - 100 percent incomplete, containing several shuffled fragments combined together (Figure 3). The fragments were placed in random order using the Fisher-Yates algorithm, an algorithm which generates a random permutation of a finite set [18]. The fifth experiment dealt with sequences with variants. The variants were obtained with different percentages and reflected point-mutations seen in nature.

C. Sequence Comparison

For each experiment, multiple sequence alignment was used to analyze each data-set. To accomplish this, the MUSCLE [19] package, a software used to compute the multiple sequence alignment for protein and nucleotide sequences, was also employed.

The comparison between trees was accomplished by estimating the path-length-difference metric as described in Felsenstein [20]. For this, a matrix was constructed for each tree. The size of the matrix is m^2, where m is the number

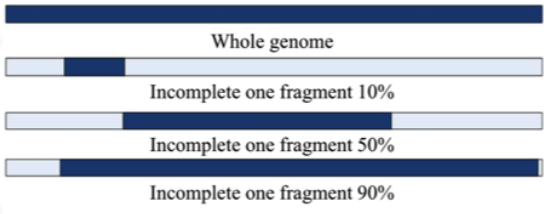


Figure 1. Cartoon diagram depicting the imperfection in genome sequence generated by choosing different fragment lengths from the original whole genome sequence.

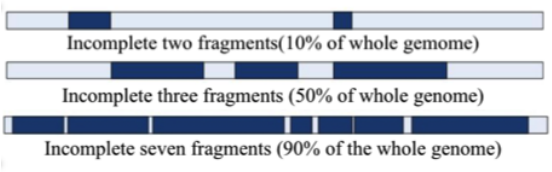
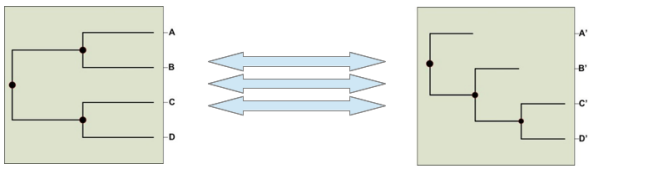


Figure 2. Cartoon diagram depicting the imperfection in genome sequence generated by choosing fragments from different regions of the whole genome.

Figure 3. Cartoon diagram depicting the imperfection in genome sequence generated by choosing fragments of different length and order in the whole genome

of tree leaves (the species), and each cell in the matrix has the number of branches that separates the species of the corresponding row and column. The squared difference was computed between each cell in a matrix and its representative in the gold standard tree matrix. The distance was then calculated by finding the square root of the sum of the cells where we took care not to include duplicate values. The distance was normalized by dividing the distance by the summation of the distances between each of the cells in the gold standard tree.



$$Distance = \sqrt{(AB - A'B')^2 + (AC - A'C')^2 + (AD - A'D')^2 + (BC - B'C')^2 + (BD - B'D')^2 + (CD - C'D')^2}$$
$$= \sqrt{(2-3)^2 + (4-4)^2 + (4-4)^2 + (4-3)^2 + (4-3)^2 + (2-2)^2}$$
$$= \sqrt{1^2 + 0^2 + 0^2 + 1^2 + 1^2 + 0^2} = \sqrt{1+0+0+1+1+0} = \sqrt{3}$$

Figure 4. Two hypothetical trees. We show the calculations to determine tree distance between reference and generated trees.

 ISBN: 978-1-61208-260-8

Consider the two trees in Figure 4, where the tree on the left represents the gold standard tree (species A, B, C, and D), and the second tree on the right represents the output tree of an algorithm (species A', B', C', and D'). The scoring matrices of Table I were calculated by summing the edges between two nodes in a tree. The distance between these two trees was calculated by finding the mean root square, noted in Figure 4. Their distance, $\sqrt{3}$, is normalized by division of the sum of the distances between the species in the gold standard tree. The calculation of this sum of distances is the following: (AB + AC + AD + BC + BD + CD) = (2 + 4 + 4 + 4 + 4 + 2) = 20. The normalized distance between the two trees is $\frac{\sqrt{3}}{20} \approx 8.66\%$.

Table I
SCORING MATRIX FOR TREES IN FIGURE 4

	A	B	C	D		A'	B'	C'	D'
A	0	2	4	4	A'	0	3	4	4
B	2	0	4	4	B'	3	0	3	3
C	4	4	0	2	C'	2	3	0	2
D	4	4	2	0	D'	4	3	4	0

III. RESULT AND ANALYSIS

The results show the performance of compression-based methods over different kinds of data-sets. To evaluate these methods over data-sets with different features that reflected imperfection in the input sequence data, we started with a data-set that was error-free (according to NCBI), then we picked a genomic data-set and manufactured data-sets with errors to incorporate imperfection in the sequence data.

Results are shown for the phylogenies generated from the compression-based methods from multiple sequence alignment (Table II). The purpose of having results from multiple sequence alignment is to evaluate whether compression-based methods were similar, worse, or better than multiple sequence alignment with data-sets of different quality. These results are the distances between the calculated trees and the gold standard tree. These distances reflect the quality of clustering for the species, based on the pair-wise distances generated from the methods, (i.e., the scoring matrices). The column labeled as *Variant*, lists the different distance measures which calculated for the first experiment. Table II shows the results from trees created by neighbor-joining methods, and also the UPGMA methods.

Shaded cells in Table II indicate the cases where compression-based algorithm performed better than multiple sequence alignment

A. Analysis of Datasets with No Errors

The first experiment determined the feasibility of using compression-based algorithms in phylogenetic analysis of sequence data. The goal was to test the algorithms against regular data-sets that are error-free and helps to evaluate

Table II
COMPARISON OF COMPRESSION ALGORITHMS AND MULTIPLE SEQUENCE ALIGNMENT FOR THE PROTEIN DATASET CK-36-PDB IN EXPERIMENT 1

Test		**Protein data-set CK-36-PDB**	
Algorithm	Variant	Neighbor-Joining	UPGMA
Kolmogorov using Huffman coding	CD	2.395244	3.169468
	NCD	2.328382	2.264505
	UCD	2.328382	2.264505
Kolmogorov using LZW compression	CD	2.176959	2.165911
	NCD	2.210704	2.215544
	UCD	2.305268	2.238781
Lempel - Ziv complexity	Dist 1	2.345943	2.280642
	Dist 2	2.330589	2.219562
	Dist 3	2.26719	2.287058
	Dist 4	2.272324	2.306048
Multiple Sequence Alignment		2.370071	1.937603

whether these methods are capable of measuring the distances of normal data-sets. We compare the results obtained from various versions of compression-based sequence comparison with results obtained from multiple sequence alignment. In this experiment, two data-sets were used: a set of protein sequences and a set of complete mitochondrial genomes. The gold standard trees for both data-sets were available to provide the base line comparison. Tables II and III display the results for the first experiment. The shaded cells reveal the compression techniques that surpassed multiple sequence alignment. In the protein data-set (Table III), the consistently desirable results were derived from UPGMA clustering using the scoring matrices of both Kolmogorov and Lempel-Ziv complexities.

In the mitochondrial data-set (Table III), only Lempel-Ziv outperformed multiple sequence alignment. These results clearly indicate that compression-based sequence comparison provides a valid measure of similarity for biological sequences.

These measurements are comparable to the ones produced by multiple sequence alignment and outperform alignment in several instances. It is also clear that a careful selection of the clustering algorithm, compression methods, and associated distance measure can improve the overall results. As in Table II, shaded cells in Table III indicate outcomes that are better than multiple sequence alignment.

The purpose here was two-fold: first, to determine if the imperfection in the quality of sequence data and the choice of compression-based methods used impacted the outcome, and second, to determine which method would be a better solution for the type of imperfection in the data-sets. For this purpose, the mitochondrial genomes were used, incrementally removing percentages of genome and using

 ISBN: 978-1-61208-260-8

Table III
COMPARISONS OF COMPRESSION ALGORITHMS AND MULTIPLE SEQUENCE ALIGNMENT FOR THE MITOCHONDRIAL GENOME DATASET AA-15-DNA, IN EXPERIMENT 1

Test		Mitochondrial Genome data-set AA-15-DNA	
Algorithm	Variant	Neighbor-Joining	UPGMA
Kolmogorov using Huffman coding	CD	7.871585	7.871585
	NCD	7.871582	7.871582
	UCD	7.871582	7.871582
Kolmogorov using LZW compression	CD	3.034474	3.034474
	NCD	2.797647	2.797647
	UCD	2.878755	2.878755
Lempel-Ziv complexity	Dist1	1.554705	1.357058
	Dist2	1.554705	1.357058
	Dist3	1.554705	1.357058
	Dist4	1.554705	1.357058
Multiple Sequence Alignment		1.5547053	1.878762

an algorithm to randomly choose the starting position of the remaining genome (refer back to Figure 1).

Upon examining the neighbor-joining method and UPGMA (Figures 5 and 6), Lempel-Ziv complexity surpassed multiple sequence alignment in all the trials (with both neighbor-joining and UPGMA clustering), except for in one case. Kolmogorov with Lempel-Ziv-Welch had viable results but was not competitive to Lempel-Ziv. These results showed that Lempel-Ziv complexity offered the most likelihood of revealing the similarities between the genomes. Despite the variations in the length of the genomes, Lempel-Ziv was able to address the dissimilarities between the sequences.

B. Analysis of Sequences Data with Incomplete Fragments that Are Not Continuous

This experiment was an expansion of the second experiment, where the genome was broken into several pieces, and the total size of the sequence was reduced to the same 10-90 percent but where each fragment was allowed to be a different random size (refer back to Figure 2).

Multiple fragments were then combined together and tested. The results obtained here with both the neighbor-joining method and UPGMA (Figure 7 and 8) mirrored the earlier results (in the second experiment) in that Lempel-Ziv complexity outperformed multiple sequence alignment in almost every percentile. Also, Kolmogorov using Lempel-Ziv-Welch compression and Kolmogorov, using Huffman coding, failed to perform better than multiple sequence alignment (results not presented).

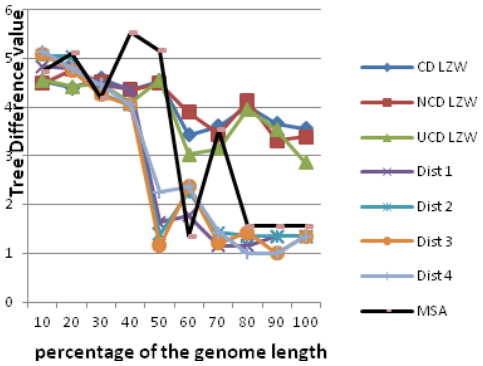


Figure 5. Analysis of the mitochondrial genomes using Neighbor-Joining.

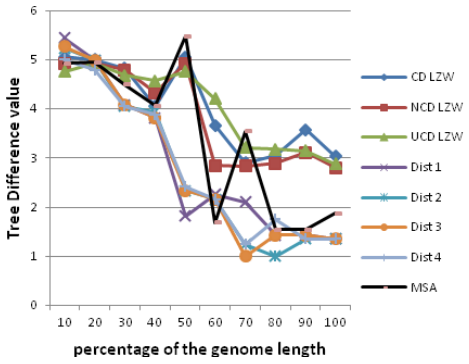


Figure 6. UPGMA clustering on the distances obtained with different algorithms.

C. Analysis of Datasets with Incomplete Fragments that Are Not Continuous and Not in Order

This experiment was designed to establish the goodness of fit of multiple sequence alignment and compression algorithms. The genomes for this experiment were cut into multiple fragments, randomly decreased in length to a total 10-100 percent of the original size, and then rearranged (refer back to Figure 3).

While the compression algorithms returned results similar to the previous experiments, multiple sequence alignment performed much worse (Figure 9). For the incomplete genomes less than 50 percent in length, Kolmogorov using Lempel-Ziv-Welch and Lempel-Ziv both surpassed multiple sequence alignment, but Kolmogorov was overtaken by multiple sequence alignment at 60 percent and above.

In this experiment where the data-set depicted translocation of DNA fragments, multiple sequence alignment performed very poorly and failed to detect the relatedness between the genomes. However, the compression-based method of Lempel-Ziv still detected the relationships among genomes and gave an accurate clustering. Even Lempel-Ziv-Welch was competitive with multiple sequence alignment in finding the right dissimilarities between the genomes.

 ISBN: 978-1-61208-260-8

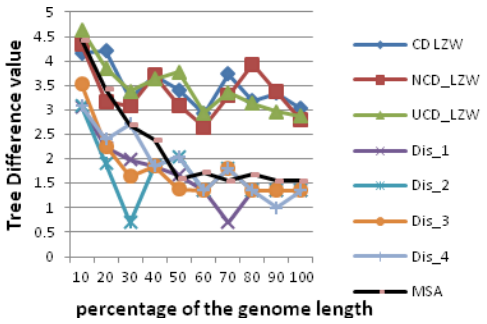


Figure 7. Analysis of sequence data of unequal length with data-set as shown in Figure 2 using neighbor-joining.

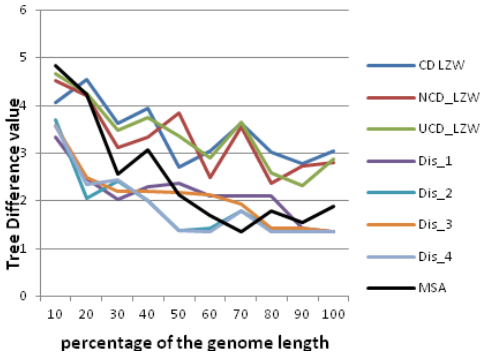


Figure 8. UPGMA clustering of the distances obtained with shown algorithms.

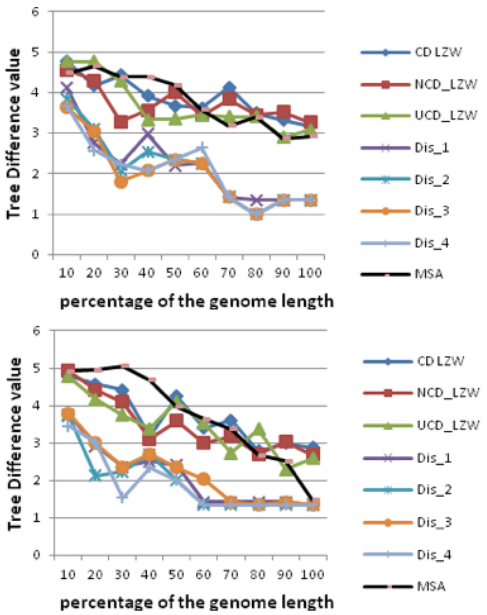


Figure 9. Analysis of sequence data of varying length with data-set as shown in Figure 3 using neighbor-joining (top figure) and UPGMA (bottom figure) clustering of the distances obtained with shown algorithms.

D. Analysis of Sequence Data-sets Containing Mutated Nucleotides

This experiment was designed to evaluate the performance of the compression-based methods on a mutated data-set. As the sequences mutate, it is difficult for methods like multiple sequence alignment to identify the relatedness among species. Mutations (point mutations) were taken with percentages of 1, 3, 5, and 7 percent. Comparison of the results to multiple sequence alignment was conducted in the same manner described earlier for sequence data with different fragment lengths and for measuring the distance between the resulting trees to the gold standard tree. The results obtained from this experiment are shown in Table IV.

As we can see, the shaded cells contain the results of Lempel-Ziv complexity, which performed relatively better than other compression or non-compression methods. With reasonable mutations, which typically would result in changes in the functions of the species but not in an evolution of the species itself, Lempel-Ziv complexity performed best and was able to detect the similarities among the species when compared to the multiple sequence alignment method. Kolmogorov complexity failed to detect similarities with this data-set.

IV. Conclusion

Compression-based techniques provide a viable alternative to multiple sequence alignment that is typically used to compare biological sequence data. In cases where the data-sets contained errors, gaps, or the arrangement of DNA fragments, compression-based techniques performed better than alignment in our experiments. Compression algorithms were also faster than alignment, particularly for large sequences. Of the three compression techniques examined in this study, Lempel-Ziv complexity performed the best in classifying the incomplete and highly imperfect data-sets.

To summarize these results, Lempel-Ziv complexity led in performance to the alignment-free techniques and even outperformed multiple sequence alignment in several cases. From the results obtained with the different experiments, we can see that compression techniques in general, and Lempel-Ziv in particular, were able to capture the relatedness among the input sequences and were less impacted by the incompleteness or rearrangement of the fragments.

V. Acknowledgment

The authors would like acknowledge the help by support staff in the UNO-Bioinformatics Core Facility, funded by the grants from the National Center for Research Resources (5P20RR016469) and the National Institute for General Medical Science (NIGMS) (8P20GM103427). We also like to thank Oliver Bonham-Carter for his help in editing and formatting of this manuscript.

 ISBN: 978-1-61208-260-8

Table IV
COMPARISON OF THE PERFORMANCE OF COMPRESSION AGAINST MULTIPLE SEQUENCE ALIGNMENT ON A MUTATED DATA-SET WITH MUTATION PERCENTAGES OF 1, 3, 5, AND 7 PERCENT.

		NJ (1 percent)		UPGMA (3 percent)		NJ (5 percent)		UPGMA (7 percent)	
Kolmogorov using Huffman coding	CD	7.184	7.872	7.184	7.872	7.184	7.872	7.184	7.872
	NCD	7.054	7.872	7.054	7.872	7.054	7.872	7.054	7.872
	UCD	7.054	7.872	7.054	7.872	7.054	7.872	7.054	7.872
Kolmogorov using Lempel-ZivW compressions	CD	3.201	3.266	3.443	3.152	3.643	3.097	3.696	3.009
	NCD	3.272	2.996	3.278	2.791	3.324	3.487	3.387	2.964
	UCD	3.41	3.041	3.128	2.99	3.278	2.707	3.537	3.003
Lempel and Ziv complexity	Dist1	1.357	1.357	1.357	1.774	1.858	2.101	2.054	2.276
	Dist2	1.357	1.357	1.357	1.357	1.357	1.53	1.357	1.357
	Dist3	1.357	1.357	1.357	1.774	1.357	2.7	1.159	2.276
	Dist4	1.357	1.357	1.357	1.542	2.017	1.426	1.357	1.357
Multiple Sequence Alignment		1.555	1.879	1.555	1.774	1.555	1.357	1.685	1.879

REFERENCES

[1] M. Larkin, G. Blackshields, N. Brown, R. Chenna, P. McGettigan, H. McWilliam, F. Valentin, I. Wallace, A. Wilm, R. Lopez *et al.*, "Clustal w and clustal x version 2.0," *Bioinformatics*, vol. 23, no. 21, pp. 2947–2948, 2007.

[2] S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman *et al.*, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[3] W. Kent, "Blatthe blast-like alignment tool," *Genome research*, vol. 12, no. 4, pp. 656–664, 2002.

[4] P. Ferragina, R. Giancarlo, V. Greco, G. Manzini, and G. Valiente, "Compression-based classification of biological sequences and structures via the universal similarity metric: experimental assessment," *BMC bioinformatics*, vol. 8, no. 1, p. 252, 2007.

[5] H. Otu and K. Sayood, "A new sequence distance measure for phylogenetic tree construction," *Bioinformatics*, vol. 19, no. 16, pp. 2122–2130, 2003.

[6] S. Schuster, "Next-generation sequencing transforms todays biology," *Nature*, vol. 200, no. 8, 2007.

[7] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in Bioinformatics*, vol. 11, no. 5, pp. 473–483, 2010.

[8] G. Zhang, I. Fedyunin, S. Kirchner, C. Xiao, A. Valleriani, and Z. Ignatova, "Fanse: an accurate algorithm for quantitative mapping of large scale sequencing reads," *Nucleic acids research*, vol. 40, no. 11, pp. e83–e83, 2012.

[9] M. Schatz, C. Trapnell, A. Delcher, and A. Varshney, "High-throughput sequence alignment using graphics processing units," *BMC bioinformatics*, vol. 8, no. 1, p. 474, 2007.

[10] S. Vinga and J. Almeida, "Alignment-free sequence comparisona review," *Bioinformatics*, vol. 19, no. 4, pp. 513–523, 2003.

[11] M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.

[12] D. Burstein, I. Ulitsky, T. Tuller, and B. Chor, "Information theoretic approaches to whole genome phylogenies," in *Research in Computational Molecular Biology*. Springer, 2005, pp. 992–992.

[13] X. Chen, S. Kwong, and M. Li, "A compression algorithm for dna sequences and its applications in genome comparison," in *Proceedings of the fourth annual international conference on Computational molecular biology*. ACM, 2000, p. 107.

[14] B. Behzadi and F. Le Fessant, "Dna compression challenge revisited: a dynamic programming approach," in *Combinatorial Pattern Matching*. Springer, 2005, pp. 85–96.

[15] Y. Neuman, "Meaning-making in language and biology," *Perspectives in biology and medicine*, vol. 48, no. 3, pp. 317–327, 2005.

[16] E. Rivalsy, O. Delgrangez, J. Delahayey, and M. Dauchety, "Compression and sequence comparison," 1994.

[17] M. Borowska, E. Oczeretko, A. Mazurek, A. Kitlas, and P. Kuc, "Application of the lempel-ziv complexity measure to the analysis of biosignals and medical images," *Annual proceedings of Medical Science*, 2005.

[18] R. Fisher, F. Yates *et al.*, "Statistical tables for biological, agricultural and medical research." *Statistical tables for biological, agricultural and medical research.*, no. Ed. 3., 1949.

[19] R. Edgar, "Muscle: multiple sequence alignment with high accuracy and high throughput," *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[20] J. Felsenstein *et al.*, *Inferring phylogenies*. Sinauer Associates Sunderland, 2004, vol. 2.

 ISBN: 978-1-61208-260-8