The Similarity Metric in Various Contexts

William Heath and William Rice

March 31, 2018

Abstract

Kolmogorov Complexity has proved to be useful in comparing the information content of symbolic strings. In particular, a universal metric called "*The Similarity Metric*" has been defined in terms of Kolmogorov Complexity that is optimal for arbitrary strings [1]. In our paper, we define an alternative metric for the cases when the strings are not arbitrary. We describe an approach for when we have some prior knowledge about the structure of the strings. We introduce three different contexts in which comparisons between strings might need to be made and verify the performance of our approach against *The Similarity Metric* via MATLAB simulations. In the last context, our approach does not define a metric, but we demonstrate its practical utility nevertheless. Other context-based similarity metrics have been explored for strings (as in [2]), but unlike these, our metric is based on *The Similarity Metric*, which means it takes full advantage of Kolmogorov Complexity.

1 Introduction

1.1 Quantifying Information

In dealing with information systems, we often need to quantify and compare the information content of two symbolic strings. In biology, for example, the similarity between genetic sequences can tell us about specimen relatedness or evolutionary distance. In Natural Language Processing, similarity between documents can determine authorship or relatedness of ideas. In linguistics, similarity can describe the evolution of languages [3].

Information theory essentially emerged at once when Claude Shannon published *A Mathematical Theory of Communication* in 1948 [4]. Information theory deals with the storage, measurement, and transmission of information. Two major areas of information theory used in this study are statistical and algorithmic information theory.

1.2 Statistical Information Theory

Information content can be quantified statistically based on the likelihood of receiving a given signal or message from some transmitting channel or source. In practice, the likelihood is taken to be the relative frequency of that particular signal compared to others when the source has transmitted multiple signals. The information content of a signal depends on its *entropy*, which is a measure of its uncertainty given all possible messages that *could* have been sent; the less common a message is, the more information it is said to contain [4]. With strings, the "source" is each position in the string, and the possible messages are the symbols used. In the case of binary

strings, the symbols are 0 and 1, and maximal uncertainty for a given position occurs when either symbol is equally likely.

Quantifying entropy from a collection of strings is source-dependent. For example, an ensemble of strings may represent a single message being sent multiple times over a noisy channel for redundancy. Another case could be that the ensemble is a set of related messages coming from more than one source.

1.3 Algorithmic Information Theory

Algorithmic Information is concerned with the semantic information contained in a string irrespective of any specific interpreter. Intuitively, a message contains more information if it is difficult to describe (or reproduce according to a list of deterministic instructions, as in an algorithm). The length of the shortest such description should correspond to the amount of information in that message [3]. The use of a "shortest description" came to be in the 1960s when Ray Solomonoff, Andrey Kolmogorov, and Gregory Chatin co-discovered algorithmic complexity. Unlike the source-dependent statistical approach, algorithmic complexity is an inherent property of the string [1].

In essence (and in practice), the process of "describing" a string is compression. Anything that can be compressed to a shorter description has, relatively speaking, low complexity, or information content (e.g. segment repeats, palindromes) [5]. The degree of incompressibility for a string is conventionally considered to be a proxy for the information content of that string. Segments of higher complexity are said to have greater information content since they cannot be compressed to as short of a description. The problem with this approach is that sometimes complexity results from noise, making the string appear to contain more information than it actually does. This issue is the main motivation for our project. While information content in one string is given by its short description, similarity between two strings is essentially quantified by the length of the shortest code that can change one string into the other.

1.4 Creating a Context

After understanding how information is quantified and compared in each of the two approaches, it is evident that both have powerful applications. However, they both have disadvantages that depend on the application context. Here, we address those concerns and clarify the direction of this study.

Statistical information theory is too restrictive. Consider three different biological signals:

1. The binding pocket on an enzyme consisting of 30 amino acid residues specified by 30 codons
2. The 90 nucleotides coding for the amino acids above
3. The 3000 nucleotides for the gene that encodes the whole enzyme mentioned in (1)

As we move down the list, there appears to be more and more information. In the context of a catalytic function, the binding pocket is the most important, so it may be the case that not all segments in the whole gene are relevant (3). Comparing the 90 nucleotides in two different strings would account for degeneracy in the third place of the codon, which can be considered irrelevant

information (2). The statistical approach, in this case, is most applicable to the simplified version in (1).

We see that when more subtle changes count, the statistical approach loses it validity. Consider comparing two English documents, which has been done using the algorithmic approach. As mentioned earlier, a universal metric based on Kolmogorov Complexity has been discovered and applied to many different fields that compare information. This approach depends on algorithmic information theory. It is best to use when no prior knowledge is known about the strings in question. The issue arises when some high Complexity segment of information may not be relevant (i.e. it is due to noise).

Kolmogorov Complexity treats noise as if it were signal. Our goal is to remedy this restriction on algorithmic complexity by defining three different contexts of comparison and describing suitable approaches for each context.

2 Definitions and General Methods

We begin by defining the terms used in the algorithmic approach and move into applying them to metrics. These metrics are used to calculate the distance between two strings. This section concludes with a discussion of computable functions and general simulation methods used in MATLAB.

Definition 1. The **Kolmogorov Complexity** $K(x)$ of an object x is the length of the shortest input program x^* from which a Universal Turing Machine (UTM) computes x.

Definition 2. The **Conditional Kolmogorov Complexity** $K(x|y)$ of an object x given another object y is defined to be the length of the shortest program x^* that can be given to a universal Turing machine along with y to compute x.

Definition 1 gives the complexity of a single string, where $K(x) = |x^*|$. In looser terms, we can say that x is the output of some shortest coding program x^*, where the input is the empty string which is used to construct x^*. Definition 2 gives a similar result, except the input string is y instead of the empty string such that $K(x|y) = |x^*|$. One could say y is being used as a reference, or template, for describing x.

Definition 3. A **metric** on a set X is a function $d : X \times X \to [0, \infty)$ such that $\forall\ a, b, c \in X$, the following properties hold:

1. $d(a, b) > 0$ unless $x = y$ then $d(a, b) = 0$
2. $d(a, b) = d(b, a)$ (symmetric)
3. $d(a, c) \leq d(a, b) + d(b, c)$ (triangle inequality)

The definition of a metric gives rise to an application of Kolmogorov Complexity in measuring the information distance, between two strings. This comparison is based on the strings' complexities.

$$d(x, y) = 1 - \frac{K(x) - K(x|y)}{K(xy)} \tag{1}$$

It has been proven that the Normalized Information Distance (NID) given by equation (1) is a metric[6]. The numerator, $K(x) - K(x|y)$ represents the mutual information between the two strings while the denominator serves as a normalization factor. Applications such as reconstructing a phylogenetic tree from RNA, detecting plagiarism, data mining, and deriving a language tree have been made and reproduced by other sources using the NID.

$$d(x,y) = \frac{max\{K(x|y^*), K(y|x^*)\}}{max\{K(x), K(y)\}} \tag{2}$$

The similarity metric (2) is an improvement on the NID. It has been proven that it is a universal metric, which means it is optimal for any two arbitrary objects [1]. It utilizes the maximum conditional Kolmogorov Complexity between two strings, which is also dependent on the shortest program of the input string (i.e. y^* instead of y). Kolmogorov Complexity is noncomputible (Chaitin 1960) [5]. By definition, it is the shortest program (x^*) that can be read by a UTM to output x. The noncomputibility issue arises in "providing" an all-encompassing algorithm for a UTM to read. An algorithm provided to a UTM may compress the object x, but there is always another way to compress the information further, resulting in a shorter program. Therefore, we say that the Kolmogorov Complexity of an object is an idealized value [5].

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \tag{3}$$

The normalized compression distance (NCD) is an approximation of the similarity metric [1]. Instead of using Kolmogorov Complexity, a real algorithm is used to find the complexity of an object (this is denoted by $C(x)$ instead of $K(x)$). Compression effectiveness depends on both the algorithm and the nature of the object [5]. This study uses the Lempel-Ziv algorithm, the algorithm used in the gzip compression program, to calculate the complexity of a given string.

3 Context I: Noise-Signal Differentiation

3.1 Construction

The first context we consider assumes we know which part of a string codes for signal and which part codes for noise. We only consider comparisons between strings that have the same signal positions and the same length. To specify the "signal positions," we associate with each string of length l a **relevance vector** containing ones at the positions corresponding to signal in the string and zeros at the noise positions. For example, a binary string x and its associated relevance vector might look like this:

$$x = \overbrace{1\ 1\ 0\ 1\ 0 \ldots 1\ 0\ 0\ 1\ 0}^{noise}\ \overbrace{\mathbf{1\ 1\ 1\ 0}}^{signal}\ \overbrace{0\ 1\ 1\ 0\ 0 \ldots 0\ 1\ 1}^{noise}\ \overbrace{\mathbf{0\ 0\ 1\ 1\ 0\ 1}}^{signal}$$
$$\mathbf{p} = (0,0,0,0,0,0,\ldots 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,\ldots 0,0,0,1,1,1,1,1,1)$$

We implement the Similarity Metric in this context as follows. First, define a function

$$\phi_p : S_{l,p} \to S_r$$

where $S_{l,p}$ is the set of all binary strings of length l whose associated relevance vector is $\mathbf{p}$, and S_r is the set of strings of length r where r is the number of ones in $\mathbf{p}$. The function ϕ_p is defined by

$\phi_p(x) = x_r$ where x_r is the concatenated relevant (signal) part of x. Then, define a metric d_p^* over the equivalence classes of strings in $S_{l,p}$ with equivalence given by

$$x \sim y \iff \phi_p(x) = \phi_p(y).$$

The metric is defined as

$$d_p^*(X, Y) = d(\phi_p(x), \phi_p(y)) \tag{4}$$

where X and Y are equivalence classes in $S_{l,p}$ and x is any representative in X. In this way, only the signal is counted in the comparison. We expect this metric to improve the Similarity Metric by removing perturbation due to noise.

To be sure the function in (4) is a metric, we prove it satisfies the criteria given by definition 3.

Claim: d_p^* is a metric.

Proof. To eliminate the trivial cases, let $l > 0$ and suppose p contains at least one nonzero component.

1. Let X, Y be distinct equivalence classes in $S_{l,p}$. Then because $X \neq Y$, we know $\phi_p(x) \neq \phi_p(y)$ for all $x \in X$ and $y \in Y$. Since d is a metric, $d(\phi_p(x), \phi_p(y)) > 0$. So $d_p^*(X, Y) > 0$.

   Next suppose X is an equivalence class in $S_{l,p}$ and that $X = Y$. Then $\phi_p(x) = \phi_p(y)$ for all $x \in X$ and $y \in Y$. Since d is a metric, $d(\phi_p(x), \phi_p(y)) = 0$. Therefore $d_p^*(X, Y) = 0$ also. So d_p^* satisfies the first criterion.

2. Let $X, Y \in S_{l,p}$. Then $d_p^*(X, Y) = d(\phi_p(x), \phi_p(y)) \underset{(d \text{ is a metric})}{=} d(\phi_p(y), \phi_p(x)) = d_p^*(Y, X)$ for all $x \in X$ and $y \in Y$. So d_p^* satisfies symmetry.

3. Let $X, Y, Z \in S_{l,p}$. Then $d_p^*(X, Z) = d(\phi_p(x), \phi_p(z)) \overset{(d \text{ is a metric})}{\leq} d(\phi_p(x), \phi_p(y)) + d(\phi_p(y), \phi_p(z)) = d_p^*(X, Y) + d_p^*(Y, Z)$ for all $x \in X$, $y \in Y$, and $z \in Z$. So d_p^* satisfies the triangle inequality.

   Therefore, d_p^* is a metric over the equivalence classes in $S_{l,p}$.

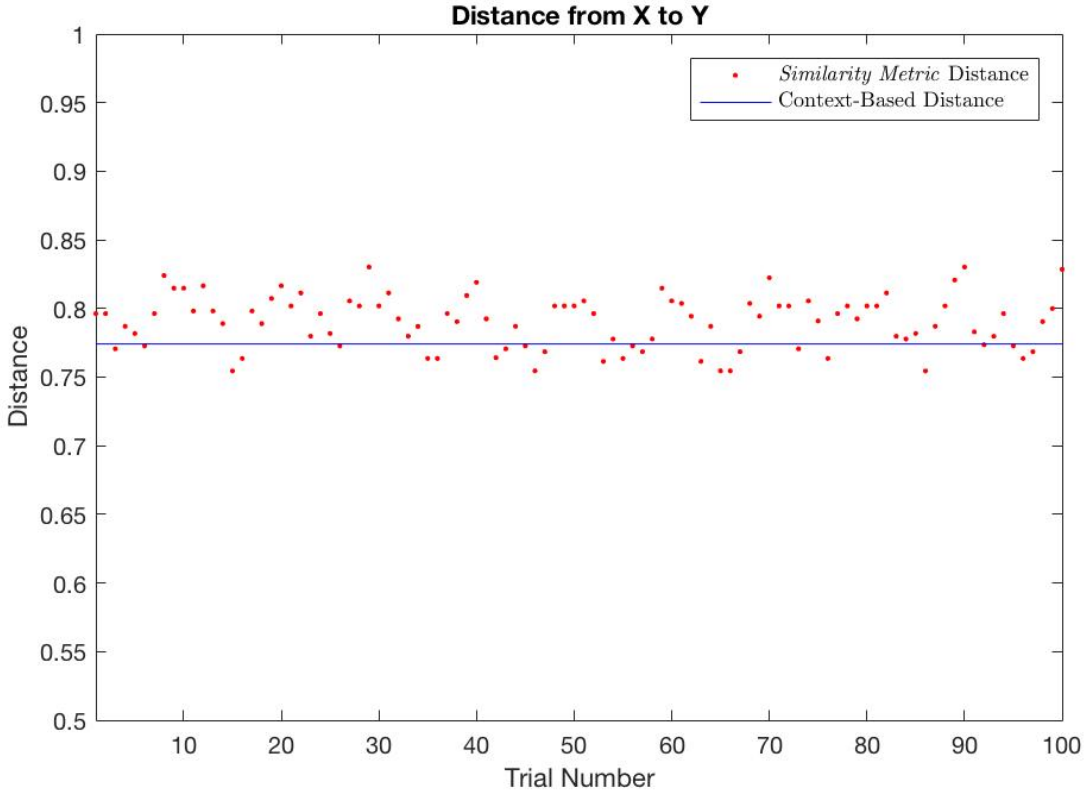
□

3.2 Validation

To demonstrate that this metric is, in fact, an improvement on the Similarity Metric, we randomized strings of the same length in MATLAB and specified which parts of the strings would serve as signal and which parts would be noise. Equivalence classes were constructed by randomizing the designated noisy positions of an ensemble of strings whose values were identical at signal positions.

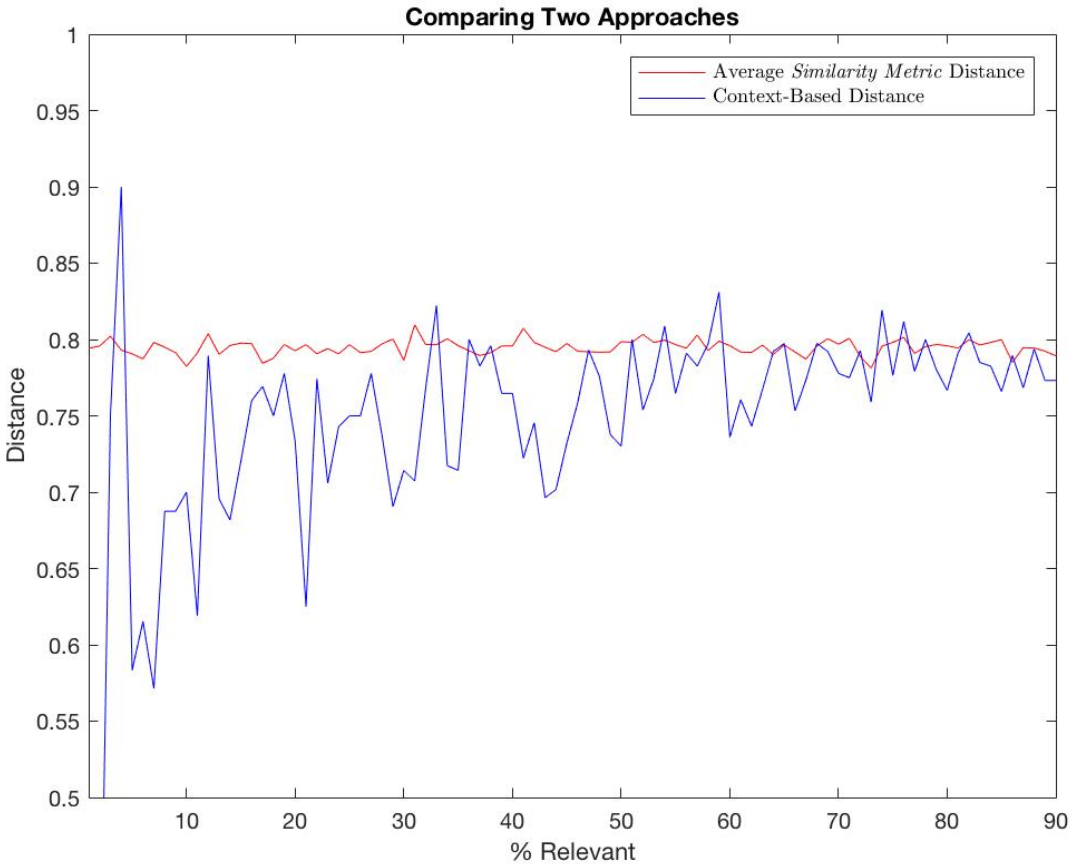
As Kolmogorov Complexity is noncomputable, we applied the same method as above to the NCD (rather than the to Similarity Metric) to calculate information distance between two strings. A

Lempel-Ziv based compressor acted as the estimator for Kolmogorov Complexity. As expected, the Similarity Metric returned a variety of distances for strings in the same equivalence classes while d_p^* returned just one consistent distance.



In this case (strings with 60% noise), we attribute the NCD's tendency to return larger distances than d_p^* to the improbability of two randomized strings accidentally being similar. In general, we expect the NCD between two randomized strings to be high because the class of strings that are highly compressible is much smaller than the class of strings that are incompressible, so the probability that a randomized string happens to be highly compressible is low. To be precise, for a binary string of length n, there are at most 2^n shorter descriptions.

While the above figure shows multiple *Similarity Metric* values and compares them to the single context-based metric value for those two equivalence classes at a fixed signal-to-noise ratio, the following figure shows how the two metrics compare for varying signal-to-noise ratios. The average *Similarity Metric* distance is the average over all 100 measurements between the 10 strings in one class and the 10 strings in the other.



4 Context II: A Degree of Relevance

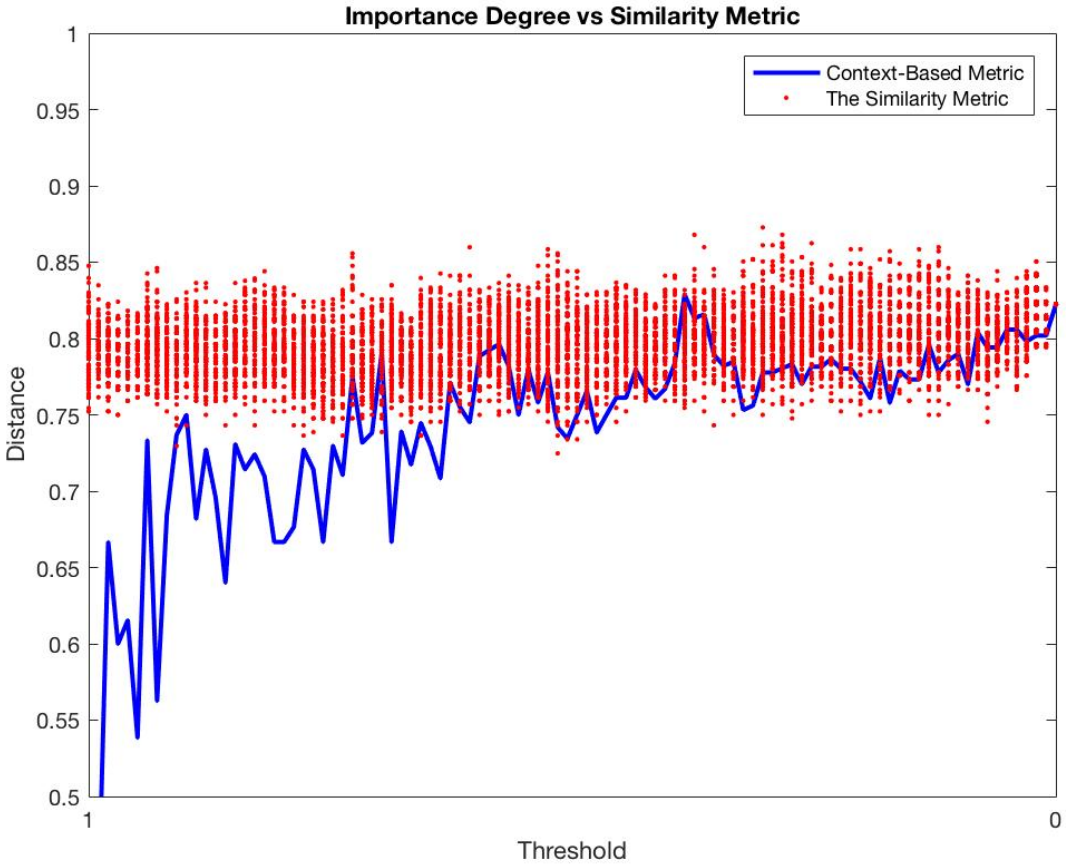
4.1 Construction

To generalize the structure of the message-carrying string, we will assume in this context that each position carries with it some weight of importance. The relevance vector will now consist of values between 0 and 1 rather than consisting only of the extremes. In this way, certain parts of the signal are deemed *more* important and some *less* important, but the distinction is no longer black and white. This introduces some difficulties in determining the equivalence classes. In order to construct equivalence classes, we will round the values of the relevance vector to 0 or 1 according to some threshold value. This then reduces to Context I and we proceed as before. While this generalization may at first seem to be of little interest, we can use it to learn about the structure of the signal by seeing how the distances vary in response to a changing threshold.

4.2 Validation

As in the first context, the NCD was used instead of the Similarity Metric for validation. The only difference in MATLAB was that the threshold needed to be applied before creating equivalence classes based on the signal parts. In the figure below, the vertical axis still represents distance, but now the horizontal axis represents the threshold value for that particular trial rather

than the explicit signal-to-noise ratio. Additionally, the different Similarity Metric distance values for each trial are plotted on a vertical spread rather than being averaged. For this demonstration, the threshold actually approximates the noise-to-signal ratio because the relevance vector was randomized. If the relevance vector were to contain a meaningful structure, it would change the general shape of its distance-vs-threshold graph. For example, a sudden rise in the graph could be indicative of the true threshold of "meaning" according to the interpreter in the communications system, but this type of analysis would depend on the application.



5 Context III: Relevance from Frequency

5.1 Construction

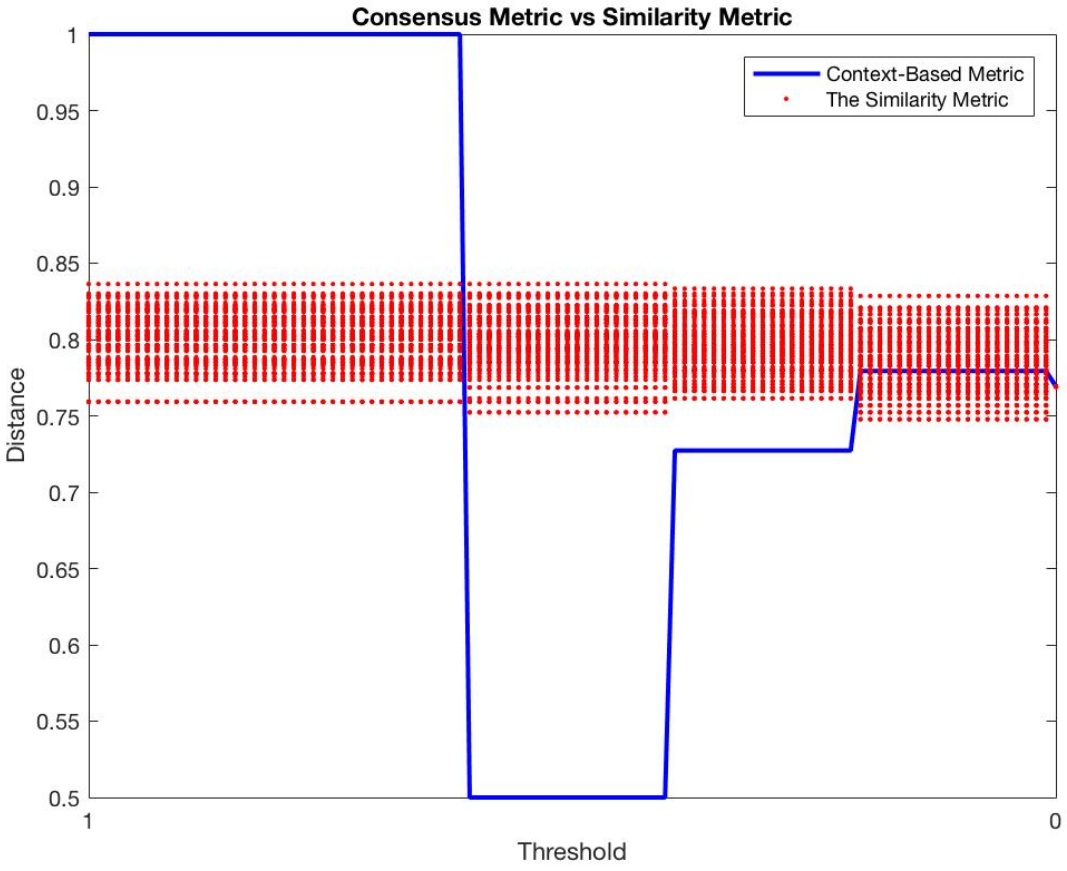
This context explores a possible application where the relevance vector is determined explicitly by an ensemble of strings. If a message is communicated over some noisy channel repeatedly for redundancy, the relevance vector can be defined (per position) to be the relative frequency of the mode for that position. That is, if in the 109^{th} position in a string, "0" appears seven of the ten times and a "1" is sent three times, the relevance vector's value at that position would be 0.7.[1] As in the previous contexts, a threshold is used to reduce the relevance vector to zeros and ones

[1]This has an analogue in genetics; certain positions in a given molecular sequence (e.g. DNA, RNA, or amino acid chains) might be more relevant to the biological signal if, in most instances of that sequence, the same symbols appear at those particular positions.

before creating equivalence classes. The strings are only compared at positions where neither is transmitting noise (according to the threshold).

5.2 Validation

In MATLAB, the equivalence classes for this context were generated as follows. First, two randomized ensembles of strings were created with each one representing a message sent with redundancy. Second, a relevance vector was generated from the frequency data.[2] Next, the relevance vector for one of the messages being compared was compared to the relevance vector of the other message while the threshold was applied, and only the parts where both vectors were above the threshold were considered to be the signal parts of the string (after all, we don't want to be comparing the noisy part of one string to the signal part of another). Finally, the strings in each ensemble were altered to match the mode at each signal position, and we proceed as we did in Context I. Note that in this context, we are no longer creating a metric because equivalence is only defined pairwise for each two things being compared.



The figure from the MATLAB simulation for this context has large jumps because the classes did not change except for at a few threshold values due to significant discretization of the positional frequency. The initial segment of the context-based metric at distance one was manually entered

[2]Because the strings were randomized, the frequency at each position in the relevance vector tended toward 0.5 as the size of the ensemble grew. For this reason, each message's ensemble was only made to have 10 members so that the frequencies' variation remained interesting for a simulation.

because for these threshold values, the strings were entirely noise.

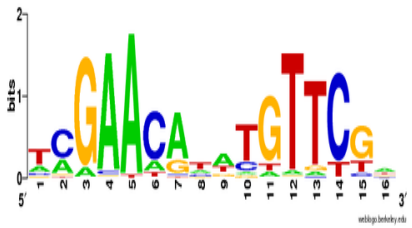
6 Conclusion and Future Work

We have demonstrated that corresponding sections of two binary strings can be used to construct equivalence classes, and those equivalence classes can be used in the Similarity Metric. The restrictions set on two objects determine the equivalence classes. This is left up to the user, where a high or low threshold value is used depending on the level of "relatedness" one desires for two things to be considered equivalent.

The power of this method lies in measuring the similarity between two objects that are each sent multiple times over a noisy channel. One real-world application is observed using Shannon's Entropy (Equation 5) [4], which is used to construct sequence logos [7].

$$H(x) = -\sum_{b=a}^{t} P(x,b) log_2 P(x,b) \tag{5}$$

In the above equation, $P(x,b)$ is the probability of symbol b appearing at position x. The symbols $a, ..., t$ represent the four nucleotide symbols, A, C, T, and G, though they can be replaced with amino acid residue symbols for analysis at the protein level. Entropy, H, is measured in bits. The entropy represents the number of bits of information it takes *on average* to specify the symbol present at position x.



Sequence logos are constructed in a way similar to Context III. From a ensemble of multiple strings, a consensus sequence is determined based on the mode at each position. This consensus sequence becomes the largest, uppermost letters in the logo. Less frequent symbols are given heights corresponding to their frequencies; they appear shortened beneath the symbols in the consensus sequence. In a practical sense, Context III takes portions of the top entries found in a sequence logo, the most commonly occurring entries, and uses them in the similarity metric. A future direction would be to incorporate the less common symbols as well (the shorter logos found under the tallest), which would add to the amount of overall information, but would be weighted less to the overall similarity since there is a lower level of certainty in their average appearance.

Another direction would be to utilize a different metric in place of the ensembles found in context III, namely the Hamming Distance [8]. It would be modified to a *weighted* Hamming Distance. This metric counts the number of changes that must be made to make two strings the same. However, changes exclude insertions or deletions. Another metric, the Edit Distance, allows insertions, deletions, and reversals [9]. Applying these metrics to construct equivalence classes presents a problem: the operations are not transitive. In practical terms, X could be close enough to Y for them to be 'equal' and Y could be close enough to Z for them to be 'equal.' However, if

the perturbation is large enough between them, X could be too distant from Z for those two to be considered 'equal,' though they are both close to Y. If one were able to retain the three properties of equivalence classes, however, the similarity metric could then be used. This enables the user to take full advantage of Kolmogorov Complexity and analyze large amounts of information based on properties concerning their intrinsic algorithmic information.

7 Acknowledgements

We would like to thank Dr. Bernadette Mullins for her guidance in learning to conduct mathematics research, and Dr. Caleb Moxley for piquing our interest in information theory, which led to this project. We would also like to thank our sponsor, Dr. Sebastian Troncoso, for his guidance throughout the process and his feedback leading to the culmination of the project.

8 References

1. Li, M. et. al. "The Similarity Metric." *IEEE Transactions on Information Theory.* 2004 Dec. 50(12): 3250-3264

2. Adami, C., Cerf, N.J. "Physical Complexity of Symbolic Strings." *Physica D.* 1999 Apr. 137 (2000) 62-29.

3. Chen, X. et al. "Shared Information and Program Plagiarism Detection." *IEEE Trans. Information Theory*, July 2004 50(7).

4. Shannon, C.E. "A Mathematical Theory of Communication." *The Bell System Technical Journal.* 1948 Oct. Vol. 27 (pp. 379-423, 623-656).

5. Li, M., Vitanyi, P. "An Introduction to Kolmogorov Complexity and its Application," 2nd Ed. *Springer.* 1997.

6. Li, M. et. al. "An information-based sequence distance and its application to whole mitochondrial genome phylogeny." *Bioinformatics.* 2001 Feb. 17(2): 149-54.

7. Schneider, T. "Sequence Logos: a New Way to Display Consensus Sequences." *Nucleic Acids Research.* 1990. 18(20): 6097-6100.

8. Hamming, R.W. "Error Detecting and Error Correcting Codes. *Bell Labs Technical journal.* 1950 Apr. 2(29): 147.

9. Levenshtein, V.I. "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals." *Doklady Akademii Nauk SSSR.* 1965. 163(4): 845-848.