LZW Mutual Information Applied to Biological Sequence Alignments

Noah S. Peterson
Dept. of Biology, Drake University, Des Moines, IA, 50311, USA

Sequence alignment is a fundamental concept in bioinformatics that involves comparing two or more biological sequences, such as nucleotide or amino acid sequences, to identify regions of similarity or difference. However the length of these sequences can mean the the time to get results for these calculations can be arduous. In this paper we look at definitions for Mutual Information using Lempel-Ziv-Welch Compression Algorithm which is an extremely fast and information lossless method of compressing data. We believe that these definitions may provide an efficient heuristic for alignment algorithms such as Clustal Omega.

Contents

I. BACKGROUND

A. Sequence Alignments

Sequence alignment is a fundamental concept in bioinformatics that involves comparing two or more biological sequences, such as nucleotide or amino acid sequences, to identify regions of similarity or difference.

Sequence alignment is an important tool in bioinformatics because it enables researchers to identify and analyze important biological features that are not apparent from the raw sequence data. For example, sequence alignment can reveal the presence of functionally important regions, such as protein domains or regulatory regions, within a sequence. Sequence alignment is also essential for studying the evolutionary relationships between different organisms or genes. By comparing the sequences of homologous genes or proteins from different organisms, researchers can reconstruct the evolutionary history of those genes or proteins and gain insights into how they have diverged over time. Finally, sequence alignment is a critical component of many bioinformatics applications, such as identifying disease-causing mutations, predicting the structure and function of proteins, and designing new drugs[1].

There are two main types of sequence alignment: pairwise and multiple sequence alignment. Pairwise alignment compares two sequences at a time, while multiple sequence alignment compares three or more sequences simultaneously. Pairwise alignment is commonly used for identifying homologous regions between two sequences, while multiple sequence alignment is used for identifying conserved regions across multiple sequences[1].

Overall, sequence alignment is a crucial technique in bioinformatics that has broad applications across a wide range of biological research areas thus having faster or more efficient methods of calculating these alignments or heuristics for them would be very beneficial to the bioinformatics community at large.

1. Clustal Omega

Clustal Omega is a multiple sequence alignment algorithm that can align a large number of protein or nucleotide sequences quickly and accurately. It uses a progressive alignment approach, where sequences are first clustered based on their similarity and then aligned in a hierarchical manner [2].

Clustal Omega uses a combination of techniques to achieve fast and accurate multiple sequence alignments. These techniques include heuristic algorithms, progressive alignment, hidden Markov models (HMMs), and seeded guide trees[2].

The basic steps of the Clustal Omega algorithm are as follows [3]:

1. Pairwise alignment: The algorithm starts by aligning every pair of sequences to generate a similarity score matrix. This is typically done using a rapid heuristic algorithm such as k-mer alignment.

2. Guide tree construction: The similarity score matrix is then used to build a guide tree, which represents the evolutionary relationships between the sequences. This is done using a neighbor-joining algorithm or another method that can handle large datasets.

3. Multiple sequence alignment: The sequences are then aligned in a hierarchical manner, following the branches of the guide tree. This is done using progressive alignment, where the sequences are first aligned pairwise, and then the resulting alignments are combined to create larger and larger alignments. The final alignment is produced by merging all the pairwise and multiple sequence alignments.

4. HMM refinement: The final alignment is then further refined using HMMs. An HMM is a statistical model that represents the probability distribution of a sequence and can be used to identify conserved regions and insertions/deletions (indels) in

the alignment. The HMMs are generated from the final alignment and used to iteratively refine the alignment until convergence.

5. Post-processing: The final alignment is then post-processed to remove poorly aligned regions and to improve the overall quality of the alignment.

Seeded guide trees are a variation of the Clustal Omega algorithm that uses an additional step to improve the accuracy of the guide tree. In seeded guide trees, a set of representative sequences is chosen based on their evolutionary diversity, and these sequences are used to construct an initial guide tree. The remaining sequences are then aligned to this initial guide tree, and the resulting alignment is used to construct a more accurate guide tree. This process is repeated until convergence[3].

The combination of progressive alignment, HMM refinement, and seeded guide trees allows Clustal Omega to achieve fast and accurate multiple sequence alignments, even for large and diverse data sets. Clustal Omega is one of the most widely used alignment tools in bioinformatics and has been used in numerous studies across different fields of biology[2]. Because of its efficiency and standard use in the industry we thought this would be a very good algorithm to compare LZW Mutual information with for this project.

B. Computing Mutual Information Using Lempel-Ziv-Welch Compression

1. *Lempel-Ziv-Welch Compression Algorithm*

The Lempel-Ziv-Welch Compression Algorithm (LZ) is an information lossless method of data compression created by Abraham Lempel, Jacob Ziv, and Terry Welch. It is currently used in Unix based compression utilities and GIF image files [4].

Other methods of encoding often create a dictionary of encoding which the decoding device must use to reassemble the original object. What makes LZ so efficient is that it creates its compression dictionary iteratively in such a way that it can be decoded iteratively with out actually sending the compression dictionary to the decoder [5] [6] [7].

2. *Mutual Information*

Mutual information is a measure of the dependence or correlation between two random variables that quantifies the amount of information one variable provides about the other. It is often used in machine learning, statistics, and information theory to identify relationships between variables and can be used to measure the similarity between two detests. The mutual information between two variables is zero if they are independent, and it increases as the variables become more correlated or dependent. [8][9].

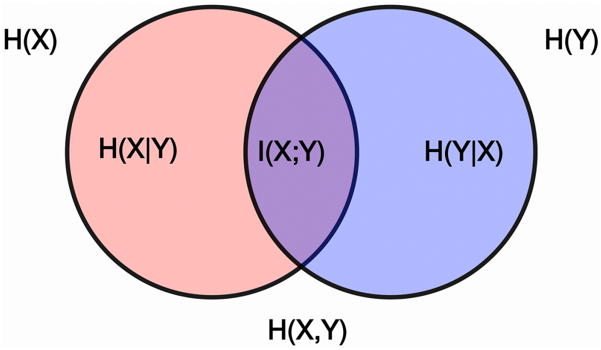

FIG. 1: Visual representation of the mutual information of two objects [10]

The fallowing are definitions of mutual information [9]:

$$I(X : Y) = H(X) + H(Y) - H(X, Y) \quad \text{(I.1)}$$

$$I(X : Y) = H(X) - H(X|Y) \quad \text{(I.2)}$$

Note that $I(X : Y)$ is the mutual information of objects X and Y. $H(X)$ is the information content of Shannon entropy of the object X. $H(X, Y)$ is the joint entropy i.e. in information contained in both object X and Y. Finally $H(X|Y)$ is the conditional information i.e. the information in Y that is not in X.

3. Mutual Information using LZ

We generally relate the information content of an object or sequence to the to its length when compressed. The idea being that when an object is compressed we have taken out all of the redundant information so we can see its true information content. [11]. Thus our project we used a few definitions for Mutual Information using LZW based off the definitions from equations I.1 and I.2 [14].

$$I(X:Y) = \rho_{LZ}(X) + \rho_{LZ}(Y) - \rho_{LZ}(X+Y) \quad (1)$$

$$I(X:Y) = \rho_{LZ}(X)+\rho_{LZ}(Y)-\rho_{LZ}(Zip(X,Y)) \quad (2)$$

$$I(X:Y) = \rho_{LZ}(X) + \rho_{LZ}(Y) - \rho_{LZCrossed}(X,Y) \quad (3)$$

$$I(X:Y) = \rho_{LZ}(X) - \rho_{LZCrossed}(X|Y) \quad (4)$$

ρ_{LZ} denotes the compression ratio of LZW on a sequence which is calculated by dividing the length of the encoding by the length of the original string. Note there are a few subdefinitions I would like to clarify:

1. $(X+Y)$ means that we are concatenating the sequences X and Y.
2. $Zip(X,Y)$ means that we are zipping X and Y i.e. we are making a new sequence by adding the next character from the sequences in an alternating order, if one of the sequences runs out we just add the rest of the other sequence to the end.
3. $\rho_{LZCrossed}(X,Y)$ is the some of the compression ratio of X given the LZ dictionary of Y and the compression ratio of Y given the LZ dictionary of X.
4. $\rho_{LZCrossed}(X|Y)$ compression ratio of X given the LZ dictionary of Y.

II. PROJECT OVERVIEW AND METHODS

Overall we believe that LZW Mutual Information would be most comparable to a pairwise alignment sequence because it can only compare two sequences at once, at least with the current definitions. However, given the large amount of time and computation power needed to compute large numbers of these alignments we decided it would be best to test these definitions against Clustal percent identity as it is much faster to compute. We chose to compare the Mutual Information with the Percent Identity because how well the sequences line up should be fairly similar to to the shared information content.

First we chose the set of sequences we would be testing. We ended up with Alpha Tubulin protein sequences because these are both long enough for LZW to run efficiently and conserved enough that we should be able to get a decent idea of the correlation for them. We then ran a mutable alignment on these sequences using the EBI online Clustal Omega Multiple Sequence Alignment Tool https://www.ebi.ac.uk/Tools/msa/clustalo/ [15] and got the Percent Identity matrix for all possible pares of our sequences.

We then compared each sequence using our Mutual Information definitions and built a Mutual Information matrix with values corresponding to those of the Percent Identity matrix.

Next we plotted the Mutual Information results against the Clustal Percent Identity values. We found that linear trend lines seamed to have the highest correlation. We used the R^2 value of these trend lines as our metric of correlation because, as the coefficient of determination it measures the variance in our Mutual Information Values with respect to our Percent Identities.

III. RESULTS

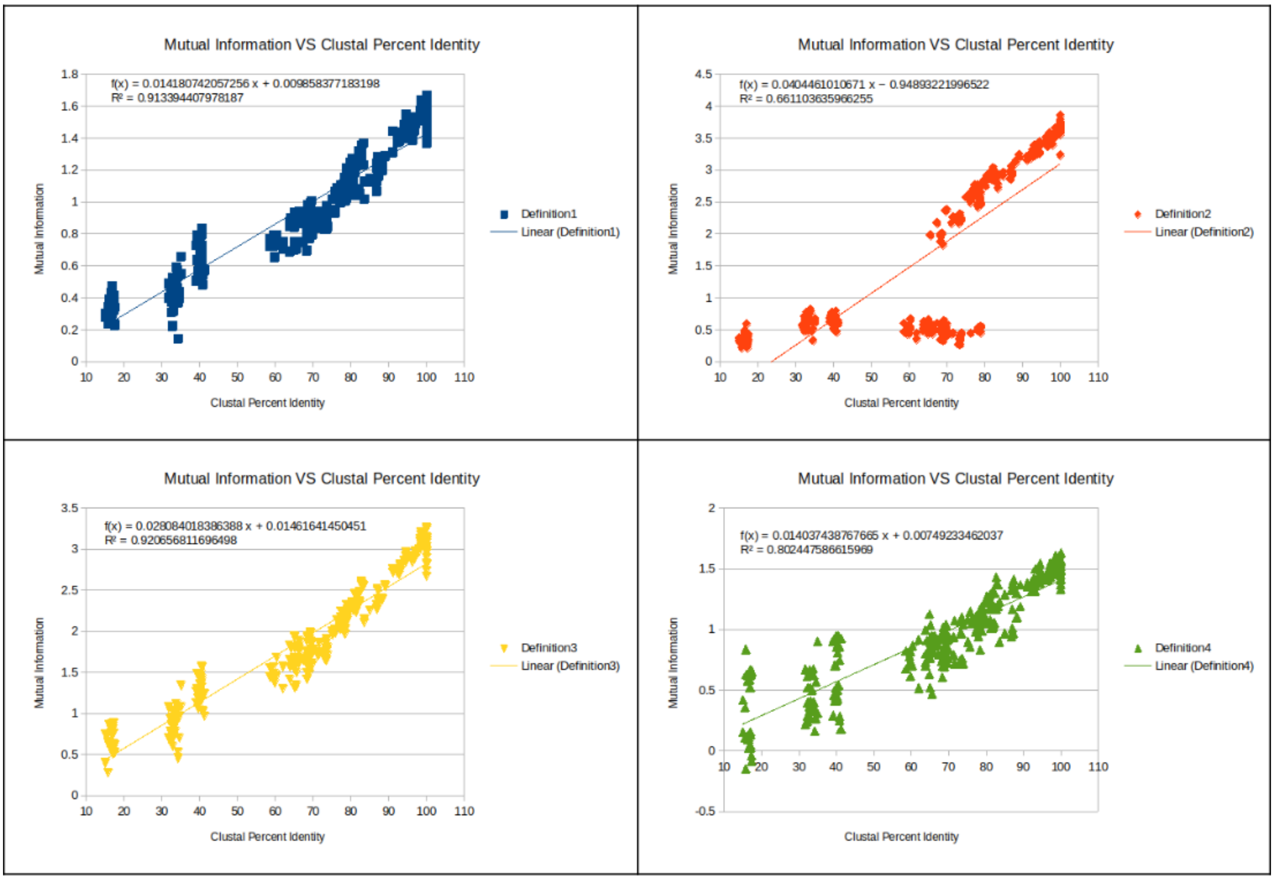

FIG. 2: Shows the Mutual Information Calculated by our Definitions vs the Percent Identity calculated by the Clustal Omega algorithm. Each graph has a corresponding trend-line, trend-line equation, and R^2 value

IV. CONCLUSIONS

Overall we it would seam that Definitions 1 and 3 had the best results both with R^2 values just above 0.9. Definition 4 had a decent R^2 value of about 0.8 however it had a a fair amount of vertical variation meaning that at least in its current state it would not work as a very good heuristic. Finally Definition 4 had the lowest R^2 with about 0.66, we believe that this is mostly due to the group of values ranging from 60 to 70 Percent Identity that are still around 0.5 Mutual Information, if this anomaly can be studied further it is possible this definition be adjusted to account for this; however, as it stands Definition 2 would not make a good heuristic for the Clustal Percent Identity.

Overall we believe that 1 and 3 could certainly make a good heuristic for Clustal Percent Identity and has good prospects for a general alignment heuristic. We would need more testing of different protein samples to say this for sure though.

V. FUTURE WORK

The original goal of this project was to test the LZW Mutual Information as a heuristic for the Needleman Wunsch Algorithm, an implementation of this algorithm was created and run specifically for this purpose; however due to time and computational constraints we where not able to complete these tests. Because LZW

Mutual Information runs in $O(nlog(n))$ time where as the Needleman Wunsch Algorithm runs in $O(n^2)$ time, being able to use the LZW algorithm as a heuristic would provide a great benefit and thus is top priority for the continuation of this research.

We also wish to test the use of these definitions as heuristics within algorithms such as step 1 in section I A 1.

Finally we noted that due to the variations in complexity between intron and exon sequences we believe that a simple compression ratio test on a data set of these sequences may prove interesting.

VI. ACKNOWLEDGEMENTS AND REFERENCES

I would like to acknowledge Dr. Adam Case for being my advisor for the CS side of this project. Also for his creation of k-balanced sequences that where used in the testing stage of this project. I would also like to acknowledge Dr. Jerry Honts for being my Advisor for the Biology side of this project and for putting together a set of proteins that had both a wide range of conservation and where long enough for LZ to run efficiently.

[1] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press. http://www.mcb111.org/w06/durbin_book.pdf

[2] Sievers, F., and Higgins, D. G. (2018). Clustal Omega for making accurate alignments of many protein sequences: Clustal Omega for Many Protein Sequences. Protein Science, 27(1), 135–145. https://doi.org/10.1002/pro.3290

[3] Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, Thompson JD, Higgins DG. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011 Oct 11;7:539. doi: 10.1038/msb.2011.75. PMID: 21988835; PMCID: PMC3261699.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3261699/

[4] Britannica, T. Editors of Encyclopaedia (2022, December 7). GIF. Encyclopedia Britannica. https://www.britannica.com/technology/GIF

[5] ZIV, JACOB, and ABRAHAM LEMPEL. "A Universal Algorithm for Sequential Data Compression - Duke University." Duke University, TRANSACTIONS ON INFORMATION THEORY, 1977, https://courses.cs.duke.edu/spring03/cps296.5/papers/ziv_lempel_1977_universal_algorithm.pdf.

[6] Ziv, Jacob, and Abraham Lempel. "Compression of Individual Sequences via Variable-Rate Coding." Compression of Individual Sequences via Variable-Rate Coding, Duke University, 1978, https://courses.cs.duke.edu/spring03/cps296.5/papers/ziv_lempel_1978_variable-rate.pdf

[7] Welch, Terry. "A Technique for High-Performance Data Compression - Duke University." a Technique for High-Performance Data Compression, Duke University, 1984, https://courses.cs.duke.edu/spring03/cps296.5/papers/welch_1984_technique_for.pdf

[8] Shannon, Claude E., and Warren Weaver. The Mathematical Theory of Communication. The Bell System Technical Journa, 1948. https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

[9] Fano, Robert M. "A Measure of Information." Transmission of Information: A Statistical Theory of Communication, Mit Press, S.l., 2003.

[10] Rita, Luis. Normalized Mutual Information.

6 May 2020, https://luisdrita.com/normalized-mutual-information-a10785ba4498

[11] Cover, T. M., and Joy A. Thomas. Elements of Information Theory. Second ed., Wiley, 2012.

[12] Madeira F, Pearce M, Tivey ARN, et al. Search and sequence analysis tools services from EMBL-EBI in 2022. Nucleic Acids Research. 2022 Apr:gkac240. DOI: 10.1093/nar/gkac240. PMID: 35412617; PMCID: PMC9252731.

[13] Shor, Peter. "Lempel-Ziv Notes Prof. Peter Shor - Math.mit.edu." Lempel-Ziv Notes, MIT, https://math.mit.edu/~djk/18.310/Lecture-Notes/LZ-worst-case.pdf.

[14] Peterson S, Noah. ”Mutual Information Using LZW-Compression”, CCSC 2023 April Central Planes Conference https://www.ccsc.org/centralplains/studentpapers/

[15] Madeira F1, Pearce M1, Tivey ARN1, Basutkar P1, Lee J1, Edbali O1, Madhusoodanan N1, Kolesnikov A1, Rodrigo Lopez ”Search and sequence analysis tools services from EMBL-EBI in 2022” Europe PMC 2022. https://europepmc.org/article/MED/35412617