Comparison of Sequence Similarity Measures for Distant Evolutionary Relationships

Abhishek Majumdar, Peter Z. Revesz
Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE. USA

Abstract—*Sequence similarity algorithms are used to reconstruct increasing large evolutionary trees involving increasingly distant evolutionary relationships. This paper proposes two sequence similarity algorithms, called the Greedy Tiling and the Random Tiling algorithms, that are both based on the idea of tiling one sequence by parts of another sequence. Experimental comparisons show that the new algorithms are better at detecting distant evolutionary relationships than the Needleman-Wunsch sequence similarity algorithm.*

Keywords: bioinformatics; Needleman-Wunsch; protein; sequence; similarity.

1. Introduction

Sequence similarity in genetics is often used to identify homologous genes, that is, genes which have evolved from a common ancestry. Similarly, sequence similarity of proteins allows identification of homologous proteins whose encoding genes evolved from a common ancestor. Therefore, similarly to the case of genes, biologists can describe an evolutionary hierarchy of proteins.

There are several ways of measuring the similarity between pairs of proteins [1]. Most protein similarity algorithms are based on the alignment of the sequences of the amino sequences. Such sequence similarity algorithms include Needleman-Wunsch [4], Smith-Waterman [9], and its extension by Gotoh [2]. Other protein similarity measures consider the 3-D structure of the proteins, especially the binding sites of the proteins, to determine their similarity [5,8]. In this paper we are only interested in sequence similarities because while sequence information is commonly available in databases because the 3-D structure of most proteins is still unknown [10].

Although sequence similarity plays a major role in genetics, there is little information about the relative reliability of various similarity measures, which is a general problem in data integration [7]. This project proposes two novel sequence similarity algorithms, called the Greedy Tiling and the Random Tiling algorithms, and compares their effectiveness with older similarity measures in recreating the evolutionary hierarchy of related proteins. Both of the tiling algorithms implement the tiling similarity measure that was non-algorithmically defined by Revesz [6] based on the idea of tiling one sequence by parts of another sequence.

This paper is organized as follows. Section 2 describes two new algorithms for finding the tiling similarity of two sequences. Section 3 describes experimental results. Section 4 analyses the results. Finally, Section 5 concludes the paper.

2. Implementations of Tiling Similarity

Revesz [6] introduced the tiling similarity measure, which is based on the idea of tiling one sequence with parts of the other sequence. The tiling similarity value depends on finding the optimal tiling and is an intractable problem for large sequences. Nevertheless, we give below two algorithms that in many cases give a good approximation of the optimal tiling. Our approximation algorithms, called Greedy Tiling and Random Tiling, both run efficiently even on large sequences.

2.1 Greedy tiling

Given as input two protein sequences X and Y with length(Y) $\leq$ length(X), *GreedyTiling* tries to reconstruct Y using segments, called *tiles*, from X. This is done using the algorithm with the following pseudocode:

```
Greedy Tiling $GreedyTiling(X, Y, Tiling)$
1. $X' = X$
2. $Y' = Y$
3. $Tiling = \emptyset$
4. $i = 0$
5. while $Y'$ is not empty do
6.     $i = i + 1$
7.     Smith-Waterman($X', Y', x_i, y_i$)
8.     $X' = X' - x_i$
9.     $Y' = Y' - y_i$
10.    if $x_i$ and $y_i$ are subsequences of $X$ and $Y$ then
11.        $Tiling = Tiling \cup \{(x_i, y_i)\}$
12.    else split $x_i$ and $y_i$ into proper subsequences
13.        $x_i = x_i' \mid x_i'' \mid \ldots$
14.        $y_i = y_i' \mid y_i'' \mid \ldots$
15.        $Tiling = Tiling \cup \{(x_i', y_i'), (x_i'', y_i''), \ldots\}$
16.    end-if
17.end-while
```

The above algorithm assumes that we have the function Smith-Waterman(X, Y, x, y) that finds the best locally matched segments x in X and y in Y, when given as input the sequences X and Y. LCS^* repeatedly calls the Smith-Waterman algorithm to find the longest common subsequences between X', the remaining X, and Y', the remaining Y. In each iteration, the pair of longest common subsequences x_i of X' and y_i of Y' are added to the set of tiles and deleted from X' and Y'. Each segment x_i and y_i is inspected whether it is a proper subsequence of X and Y, respectively, or a concatenation of two or more parts of X and Y. Accordingly x_i and y_i are broken up into its constituent components as necessary and added to the tiles as a set of pairs. This process is repeated iteratively until Y' is empty.

Example 1: Suppose we have the following two sequences: $X = WARICDFLRE$ and $Y = FIREICEWAR$.

In the first iteration, the Smith-Waterman algorithm finds between $X' = X$ and $Y' = Y$ the best local alignment to be $x_1 = WAR$ and $y_1 = WAR$, which are proper subsequences of X' and Y'. Hence x_1 and y_1 are added as a pair to $Tiling$ and deleted from X' and Y' to yield $X' = ICDFLRE$ and $Y' = FIREICE$, respectively.

In the second iteration, the best matching segments are $x_2 = FLRE$ and $y_2 = FIRE$, which are also proper subsequences of X' and Y'. Deleting those yields $X' = ICD$ and $Y' = ICE$.

In the third iteration, the best matching segments are $x_3 = ICD$ and $y_3 = ICE$, which are also proper subsequences of X' and Y'. Deleting y_3 from Y' will make it empty. Hence the algorithm terminates. Hence in this case, LCS^* will return the following:

$$Tiling = \{(WAR, WAR), (FLRE, FIRE), (ICD, ICE)\}$$

As a measure of the similarity between X and Y, we use the *tiling similarity*, or *TS*, measure of Revesz [6], which is defined as follows:

$$TS(X, Y) = \frac{\sum_{i=1}^{i=n} s_i}{n}$$

where n is the number of segments used for reconstruction and s_i is the similarity score between tiles x_i and y_i. For example, if we use the BLOSUM62 similarity matrix, then:

$$s_1 = sim_{BLOSUM62}(WAR, WAR) = 20$$

$$s_2 = sim_{BLOSUM62}(FLRE, FIRE) = 18$$

$$s_3 = sim_{BLOSUM62}(ICD, ICE) = 15$$

Hence the tiling similarity will be:

$$TS(X, Y) = \frac{20+18+15}{3} = \frac{53}{3} = 17.66$$

2.2 Random tiling

The second algorithm uses a randomized approach to find the different segments/tiles required for the reconstruction of sequence Y. It randomly breaks up sequence X into tiles of different lengths. Then filters out a select few using a constraint for a valid range of tile-length. Finally it uses this selected set of tiles (say $x_1, x_2, x_3.....x_n$) to match the different portions of Y. Since this approach is randomized the entire process needs to be iterated an arbitrary number of times, each time with a set of randomly generated tiles, and the tiling with the highest tiling similarity score selected. Below we give only the pseudocode of the basic algorithm that needs to be repeated.

```
Random Tiling $RandomTiling(X, Y, Tiling)$
1. Split $X$ into a random set of tiles $T(X)$
2. $Y_U = Y$
3. $Tiling = \emptyset$
4. while $Y_U$ is not empty and longer than the shortest tile do
5.     $BestScore = -100$
6.     for each tile $x_i \in T(X)$ do
7.         $y_m$ = prefix of $Y_U$ with $length(x_i)$
8.         if $BestScore < sim(x_i, y_m)$ then
9.             $BestScore = sim(x_i, y_m)$
10.            $BestPair = (x_i, y_m)$
11.        end-if
12.    $Tiling = Tiling \cup BestPair$
13.end-while
```

In each iteration we begin the tile-matching from the leftmost end of Y. Let Y_U denote unmatched section of Y. Clearly, initially Y_U = Y. For each tile x_i from the tile set we match it with left most segment y_m of Y_U which is of same length as x_i. We always select the tile which gives the highest matching score. In the next iteration we update Y_U by deleting from it the initial segment y_m. Then we continue the tile-matching process with the updated Y_U. This iteration is carried out from left to right until Y is fully matched. In the last iteration, if there is a case that the length of the current Y_U is less than the length of smallest tile then that remaining Y_U is matched with gaps.

Example 2: Consider the following two sequences:
$X = ABCDEFGHIJKLOIYITB$ and
$Y = WUFGDJVMBKUG$.
We will reconstruct Y using tiles from X. Let the tiles obtained from X be x_1 = BCDE, x_2 = IJKL, x_3 = DEFGHI, x_4 = LOIYITB, and x_5 = AB. Initially Y_U = WUFGDJVM-BKUG. We start by matching each tile x_i with left-most portions of Y_U which is of same length as x_i. That is we match x_1 with WUFG, x_2 with WUFG, x_3 with WUFGDJ, x_4 with WUFGDJV and x_5 with WU. Say x_1 gives the best

matching. So now Y_U becomes DJVMBKUG. The above process is repeated again. That is x_1 matched with DJVM, x_2 with DJVM, x_3 with DJVMBK, x_4 with DJVMBKU and x_5 with DJ. Let the best tile be x_4. So now Y_U = G. This is matched with a gap as its length is less than that of x_5. So the reconstructed Y looks like:

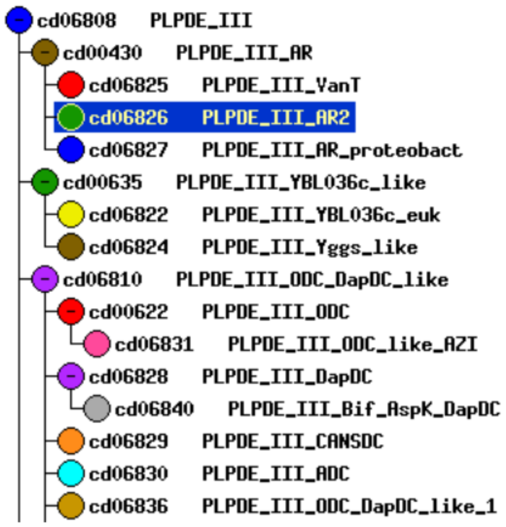
Y = $WUFG \mid DJVMBKU \mid G$

X = $BCDE \mid LOIYITB \mid -$

We can repeat the above process an arbitrary number of times and select the tiling which gives the highest tiling similarity score. Obviously, the more the basic algorithm is repeated, the higher tiling similarity is found. However, there is a trade-off between repetitions and increased tiling similarity values. There is a point where the increase in execution time may not be worth the diminishing chance of an increase in the tiling similarity score.

3. Experimental Results

In the experiments we focused on the Type III Pyridoxal 5-phosphate(PLP) dependent enzymes subfamily. This is important and well-studied subfamily is composed mainly of proteobacterial alanine racemases that help in the inter-conversion between L- and D-alanine, which is an essential component of the peptidoglycan layer of bacterial cell walls. Figure 1 shows a small portion of this subfamily hierarchy as described in the National Center for Biotechnology Information (NCBI) Conserved Domain Database [3].



cd06808 PLPDE_III
cd00430 PLPDE_III_AR
cd06825 PLPDE_III_VanT
cd06826 PLPDE_III_AR2
cd06827 PLPDE_III_AR_proteobact
cd00635 PLPDE_III_YBL036c_like
cd06822 PLPDE_III_YBL036c_euk
cd06824 PLPDE_III_Yggs_like
cd06810 PLPDE_III_ODC_DapDC_like
cd00622 PLPDE_III_ODC
cd06831 PLPDE_III_ODC_like_AZI
cd06828 PLPDE_III_DapDC
cd06840 PLPDE_III_Bif_AspK_DapDC
cd06829 PLPDE_III_CANSDC
cd06830 PLPDE_III_ADC
cd06836 PLPDE_III_ODC_DapDC_like_1

Fig. 1: Hierarchy tree.

Each node in Figure 1 is a also cluster of subsequences. That is, each node is composed of closely related bacterial genome sequences, which have a hierarchical relation among themselves as well. For instance, the node cd06825 is actually composed of nine sequences shown in Figure 2.

Figure 2 shows the gi version numbers (164602518, 44805037, etc.) which uniquely identify each sequence. The

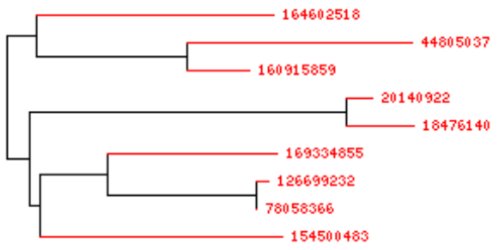


Fig. 2: cd06825 cluster

FASTA sequence description was also obtained from the NCBI website and used as input to our similarity algorithms and to the Needleman-Wunsch algorithm.

Both our algorithms take all possible combinations of two subsequences from the clusters to measure their tiling similarity scores (TSs). For example, the cd06825 cluster contains 36 pairs of sequences and yields as many similarity scores. Because of the large set of data, our experiments focused on the following five randomly selected clusters: cd06815, cd06817, cd06822, cd06825 and cd06826. The size of each cluster (in terms of number of constituent sequences) and the number of associated tiling similarity TS score combinations obtained is shown below in Table 1.

Table 1: Cluster details.

Cluster	Size	Score Combinations
cd06825	9	36
cd06826	11	55
cd06817	15	105
cd06822	36	630
cd06815	39	741

We define below the following relationship terminologies that are used in comparison of the similarity measures:

Siblings are sequences that are separated by only one evolutionary branching from a common ancestor in the evolutionary family tree. For instance, in Figure 2 sequences 44805037 and 160915859 are separated by a single evolutionary step from their common ancestor, hence they are siblings.

First cousins are sequences that are separated by at most two branching from a common ancestor in the evolutionary family tree. For instance, in Figure 2 sequences 164602518 and 44805037 are both at most two evolutionary steps distant from the common ancestor, which makes them first cousins.

Second cousins are sequences that are separated by at most three branching from a common ancestor in the evolutionary family tree. Again in Figure 2 sequences 164602518 and

20140922 are second cousins.

i^{th} **cousins** are sequences that are separated by at most (i+1) branching from the closest common ancestor.

In our experiments, we compared the Needleman-Wunsch algorithm [4], the greedy tiling, and the random tiling algorithms. We ran for each of the five above listed clusters each of the three algorithms. We calculate the similarity scores between siblings, first cousins and second cousins. For the calculation of the similarity scores, we used the common PAM250 substitution matrix [1] and a constant value of -8 as the gap-penalty. In addition, for the random tiling algorithm, the larger sequence X is always divided into 70 segments randomly to generate the available tiles $T(X)$. In the case of the random tiling algorithm, we ran the basic algorithm 1000 iterations before selecting the tiling that gave the highest score. The scores Needleman-Wunsch, the greedy tiling, and the random tiling are shown in Tables 2, 3, and 4.

Table 2: Needleman-Wunsch similarity scores.

Sequence	Siblings	First Cousin	Second Cousin
cd06815	860.20	704.88	653.56
cd06817	672.00	333.17	76.96
cd06822	496.50	115.25	143.17
cd06825	2050.33	-199.14	-1593.89
cd06826	987.75	985.43	815.75

Table 3: Greedy Tiling similarity scores.

Sequence	Siblings	First Cousin	Second Cousin
cd06815	472.87	353.09	321.17
cd06817	391.52	313.01	190.65
cd06822	357.96	236.28	175.46
cd06825	1499.11	327.42	128.24
cd06826	325.48	387.07	289.09

Table 4: Random Tiling similarity scores.

Sequence	Siblings	First Cousin	Second Cousin
cd06815	37.07	23.90	20.93
cd06817	56.23	44.63	42.90
cd06822	22.88	25.82	27.36
cd06825	68.57	49.08	83.88
cd06826	39.55	33.65	23.44

We also show the same results as a set of graphs in Figures 3, 4, and 5.

4. Discussion of the Results

The essential difference between the Needleman-Wunsch and the tiling similarity measures is that the Needleman-Wunsch method is good for random mutations, insertions

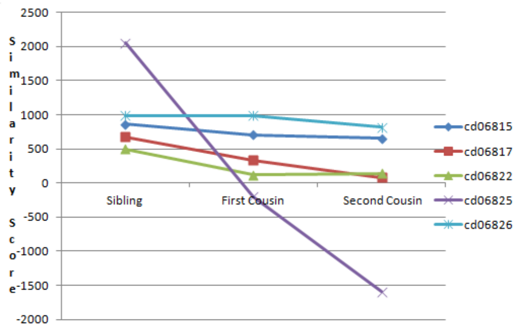


Fig. 3: Needleman-Wunsch similarity scores.

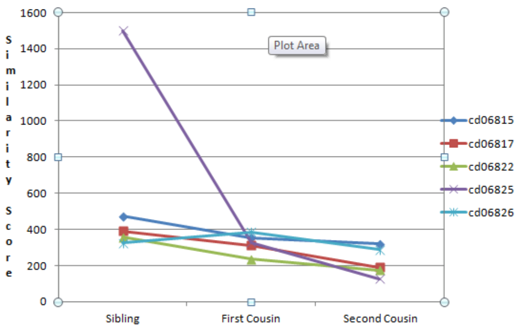


Fig. 4: Greedy Tiling similarity scores.

and deletions but is not good for reordering of parts of the sequences. In contrast, the tiling similarity measures are designed to be able to detect similarities in case of reordering. For example, recall that for the sequences $X = WARICDFLRE$ and $Y = FIREICEWAR$ a high tiling similarity was found in Example 1. In this case, it is possible to imagine a common ancestor $A = FLREICDWAR$ that branches and develops first as

$$FLREICDWAR \rightarrow_{\text{mutate L/I, D/E}} FIREICEWAR$$

and second as

$$FLREICDWAR \rightarrow_{\text{switch WAR/FLRE}} WARICDFLRE$$

yielding, therefore, Y and X, respectively.

Transpositions of parts of the genome are known to occur and would be reflected also in the amino acid sequences of the corresponding proteins. While mutations are expected to be much more frequent than such transpositions, they may not be enough to explain very distant evolutionary relationships because over large evolutionary distances some transpositions may also occur. The proteins we studied are considered ancient proteins because they help build the bacterial cell wall, which is an essential part of bacteria. Hence some transpositions may have occurred in various branches of this ancient evolutionary tree.

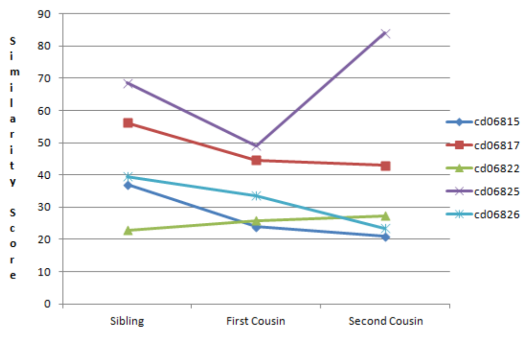


Fig. 5: Random Tiling similarity scores.

The natural expectation for all the similarity measures was the following:

1. All the similarity values were positive.

2. The average similarity among siblings was higher than among first cousins which was higher than among second cousins.

The **Needleman-Wunsch algorithm**, as shown in Figure 3, did not fulfil these expectations in three instances. In two instances, it gave a negative similarity value, namely for first and second cousins for cluster cd06825. In addition, for cluster cd06822 the similarity for first cousins was significantly less (115.25) than the similarity for second cousins (143.17).

The **greedy tiling method** gave only positive scores. The average scores for first cousins were always larger than the average scores for second cousins. The only anomaly was in cluster cd06826 where the average sibling similarity was slightly less (325.48) than the average first cousin similarity (387.07).

The **random tiling method** also gave only positive scores. The average scores for first cousins were less than the average scores for second cousins in the case of two clusters, namely, cd06822 and cd06825. In the case of cd6822, the average sibling similarity was also slightly less than the average first cousin similarity. Hence the random tiling method did not fulfil the expectations in three instances.

Therefore, our experiments suggest that the greedy tiling method is the most robust method, especially comparing larger evolutionary distances (first cousins versus second cousins). The random tiling method seems intermediate in performance. Probably it can be improved to be as good as the greedy tiling method by increasing the number of times its basic algorithm is repeated. Finally, the Needleman-Wunsch algorithm was good in comparing shorter evolutionary distances (siblings versus first cousins) but deteriorated considerably in comparing longer evolutionary distances (first cousins versus second cousins).

The experimental results suggest that the tiling similarity measure is better than the Needleman-Wunsch measure for distant evolutionary relationships. Intuitively, the reason seems to be that the tiling similarity allows transpositions of a subsequence on the genome. These transpositions may be only relatively rare evolutionary changes compared to random mutations, Nevertheless, if a significant number of transpositions accumulate in at least one branch of a large evolutionary tree, then the Needleman-Wunsch algorithm may be unable to detect them and give a low (even negative) similarity score for distantly related sequences. Based on the experimental results, we suspect that cluster cd06825 may contain some transpositions because the Needleman-Wunsch algorithm gave negative similarity scores for first counsins and second cousins, but both of the Greedy Tiling and the Random Tiling algorithms gave positive scores. Further, in the Random Tiling method the average similarity increased from first cousins to second cousins for the same cluster.

The above type of anomaly may be explained in an example. Suppose that in an evolutionary tree branch A has some transpositions that are not shared with its first cousin branch B and also not shared by A and B's second cousin branch C. In this case, the similarity between A and B, which is a first cousin similarity, could be lower than the similarity between B and C, which is a second cousin similarity. Hence if the evolutionary tree is extremely simple and has no other first cousin pairs and no other second cousin pairs beside A and C and B and C, then the average similarity among first cousins could be less than the average similarity among second cousins. The larger the evolutionary tree, the less likely such anomalies could occur. It is important to note that the cd06825 cluster is the smallest in size as shown in Table 1.

5. Conclusion and Future Work

We need to investigate further the reasons why the tiling similarity measure is better than the Needleman-Wunsch similarity measure for distant evolutionary relationships. In particular, it would be interesting to find actual examples of transpositions of subsequences within any of the clusters.

Another direction for further experiments would be to consider even larger evolutionary trees where we have enough data for third and fourth cousins. Experiments on such a larger data could show clearer the differences among the similarity measures. We suspect that the Needleman-Wunsch algorithm will perform even poorer on higher cousins but the tiling similarity algorithms will keep detecting well the more distant evolutionary relationships.

References

[1] R. Durbin and S. R. Eddy and A. Krogh and G. J. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.

[2] O. Gotoh, ”An improved algorithm for matching biological sequences,” *Journal of Molecular Biology*, vol. 162, no. 3, pp. 705–708, 1982.

[3] (March 20, 2012) The National Center for Biotechnology Information. [Online]. Available: http://www.ncbi.nlm.nih.gov/Structure/cdd/cddsrv.cgi?uid=143500

[4] S. B. Needleman and C. D. Wunsch, ”A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[5] R. Powers and J. Copeland and K. Germer and K. Mercier and V. Ramanathan and P. Z. Revesz, ”Comparison of Protein Active-Site Structures for Functional Annotation of Proteins and Drug Design,” *Proteins: Structure, Function, and Bioinformatics*, vol. 65, no. 1, pp. 124–135, 2006.

[6] P. Z. Revesz, *Introduction to Databases: From Biological to Spatio-Temporal*, Springer-Verlag, 2010.

[7] P. Z. Revesz and T. Triplet, ”Classification Integration and Reclassification using Constraint Databases,” *Artificial Intelligence in Medicine*, vol. 49, no. 2, pp. 79–91, 2010.

[8] M. Shortridge and T. Triplet and P. Z. Revesz and M. Griep and R. Powers, ”Bacterial Protein Structures Reveal Phylum Dependent Divergence,” *Computational Biology and Chemistry*, vol. 35, no. 1, pp. 24–33, 2011.

[9] T. F. Smith and M. S. Waterman, ”Identification of common molecular subsequences,” *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.

[10] T. Triplet and M. Shortridge and M. Griep and J. Stark and R. Powers and P. Revesz, ”PROFESS: a PROtein Function, Evolution, Structure and Sequence database,” *Database – The Journal of Biological Databases and Curation*, doi no. 10.1093/baq011, 2010.