A Tiling Algorithm-Based String Similarity Measure

PETER Z. REVESZ
Computer Science and Engineering
University of Nebraska-Lincoln Lincoln,
NE, USA

Abstract—This paper describes a similarity measure for strings based on a tiling algorithm. The algorithm is applied to a pair of proteins that are described by their respective amino acid sequences. The paper also describes how the algorithm can be used to find highly conserved amino acid sequences and examples of horizontal gene transfer between different species.

Keywords—amino acid, drug discovery, string similarity, tiling

Received: May 14, 2021. Revised: August 3, 2021. Accepted: August 5, 2021. Published: August 10, 2021.

1. Introduction

There are many problems that require finding similarities between pairs of strings. One such problem occurs in drug discovery. Drug discovery is a process when one searches for chemicals, usually some kind of proteins, that are similar to known drugs [2]. Once the similar proteins are identified, they are tested whether they have a similar effect in vivo than the known drug. If the effect is similar or even enhanced while there are fewer side-effects, then the similar protein may be an alternative drug to the known one. Of course, the testing process is slow in first testing the drug on animals, then on volunteer human patients, and after careful testing whether the potential new drug seems to give some benefits over the current one, then it goes through an approval process before it can be marketed. Often, dozens of potential alternative drugs are tested at various levels before one of them passes all the tests and can be legally sold as either prescription or non-prescription medicine.

In this paper, we focus only on the first step of the drug discovery process, namely, on the problem of finding proteins that are similar to a known protein. There are many approaches to the problem of testing protein-protein similarity. These approaches fall into two major categories. The first category is where the two proteins' amino acid sequences are compared with each other. The second category is where the proteins' actual chemical structures are compared with each other. The second approach is applicable only if we actually know the structure of the proteins. Unfortunately, that is not always the case. For example, if we have a DNA sequence of some organism, then it is possible to identify the protein encoding sequences on that DNA with high accuracy using computer algorithms. Then one can predict the amino acid sequence of the protein from the protein encoding sequences by the use of the standard amino acid encoding table, which gives the corresponding amino acid for each triplet of nucleotides.

There are some algorithms which also try to predict the chemical structure of the protein, which is called the protein folding problem. The chemical structure of a protein is largely responsible for its chemical behavior. Intuitively, the way a protein folds is like a string being tied into a huge knot. The little crevices in the protein fold or the knot are where other chemical structures could establish some connections. The effectively interacting other chemical structures fit into the protein's crevices as well as the proper keys fit into a lock. Biological evolution seems to have shaped the proteins of each organism such that the proteins within the organism interact efficiently and properly with each other.

The problem with the protein folding prediction algorithms is that they are less accurate than reliable for the purpose of drug discovery. Hence it still seems more reasonable to approach finding the protein-protein similarities by first investigating the sequence similarities and then developing the potential proteins in a laboratory for further chemical testing. Hence, in our paper we focus on the problem of string similarity.

While the string similarity algorithm presented in the paper was primarily motivated by the issue of efficient drug discovery, it is also applicable to other string searching problems. For example, plagiarism checkers also use string searches to test the similarities among written texts.

The outline of our paper is the following. Section II describes a similarity measure based on a greedy partial tiling algorithm. Section III discusses related work, gives some conclusions and describes future work.

2. Similarity Measure Based on Greedy Partial Tiling

We designed the following greedy partial tiling algorithm that finds the best matching segment pairs between two strings. Let *Si* and *Si'* be strings for any i > 0. We assume that we have the following functions:

subset(S_1, S_2)

which is true if and only if S_1 can be obtained after cutting out segments of S_2. For example, if $S_1 = AD$ and $S_2 = ABCD$, then *subset*(S_1, S_2) is true because S_1 can be obtained by cutting out the segment *BC* from the middle of S_2. Further, the function

slice(S_3, S_2)

returns the result of slicing out string S_3 from S_2. For example, if $S_2 = ABCD$ and $S_3 = BC$, then *slice*(S_3, S_2) = *AD*. If there are multiple copies of S_3 within S_2, then all of the copies are cut from S_2. If *subset*(S_1', S_1) and *subset*(S_2', S_2) are true, then the function

closest-pair(s_1, S_1', S_1, s_2, S_2', S_2)

returns the closest pair of segments s_1 of both S_1' *and* S_1 and s_2 of both S_2' *and* S_2 as scored by some string similarity scoring function. For example, if the strings are amino acid sequences, then we can use the PAM matrix, which is a common similarity measure between pairs of amino acids. Finally, the function

score(S_1, S_2)

returns the similarity score between strings S_1 and S_2. We consider a similarity score over 20 as significant.

Using the above functions, the pseudocode of the greedy partial tilting algorithm can be expressed as shown in Fig. I.

Algorithm Greedy-Partial-Tiling (S_1, S_2)

1. total = 0
2. $Tiles_1$ = {}
3. $Tiles_2$ = {}
4. S_1' = S_1
5. S_2' = S_2
6. (s_1, s_2) = closest-pair(s_1, S_1', S_1, s_2, S_2', S_2)
7. k = score(s_1, s_2)
8. **while** k > 20 **do**
9. total = total + k
10. $Tiles_1$ = Tiles1 U {s_1}
11. $Tiles_2$ = Tiles2 U {s_2}
12. S_1' = slice(s_1, S_1')
13. S_2' = slice(s_2, S_2')
14. (s_1, s_2) = closest-pair(s_1, S_1', S_1, s_2, S_2', S_2)
15. k = score(s_1, s_2)
16. **end-while**
17. **return** (total, $Tiles_1$, $Tiles_2$)

Fig. I. THE GREEDY PARTIAL TILING ALGORITHM

The Greedy-Partial-Tiling algorithm is an iterative algorithm that repeatedly finds the next pair of segments that gives the highest score and slices them out from the strings. The algorithm exits the while loop only when there is no longer any pair with a similarity score of greater than 20.

Next, we illustrate the Greedy-Partial-Tiling algorithm for the case when the two inputs are the amino acid sequences of the proteins with the 1B54 and the 1RCQ identifiers in the Worldwide Protein Data Bank (PDB), as shown in the two topmost sequences in Fig. II.

In the first iteration of the while loop, the highest similarity score, 29, will be found between the two segments that are highlighted in light blue as shown in Fig. II.

In the second, third and fourth iterations, the most similar pairs of segments will be those that are highlighted in light brown, green and red, respectively. Note that the algorithm suggests as the most similar segment KVETIDSLKKAKKLN, but it cannot be used, because there is the missing light blue segment that we took out between the initial K and the following V. However, that pairing of the initial K with Q in the segment QLEAIERASLARPLN adds only one to the similarity score. Hence taking that away, we will get a similarity score that is 23, which is the maximum achievable for these strings if we respect the original sequence order.

We remark that our pseudocode ensures that the segment pairs always respect the original order, because the segments have to be segments not only of the current (already sliced) strings S_1' and S_2' but also of the original strings S_1 and S_2. The implementation of our pseudocode would require some backtracking until the best such pair is found. However, for simplicity, we used the implementation of the Smith-Waterman extension [5] of the Needleman-Wunsch algorithm [1] as provided at http://jalingeer.sf.net. Any other implementation of the Smith-Waterman algorithm could have been used as well.

Continuing in this manner for the next few iterations, we get the matching pairs as shown in Fig. III. At this point, any further attempt to find similar segment pairs that respect the original order returns a similarity score that is less than or equal to 20. Hence the Greedy-Partial-Tiling algorithm will exit the while loop and return the set of eight tiles that are shown in color in Fig. IV.

3. Conclusions and Future Work

The Greedy Partial Tiling algorithm is an improvement on earlier string similarity algorithms, including our earlier attempt described in Revesz [3]. The important improvement is to avoid selecting tiles that were not substrings of the original string. This improvement gives us a more reliable string similarity measure.

The increased reliability means that the new measure can be used to compare a set of proteins and to build a hypothetical evolutionary tree from the similarity scores obtained by the Greedy Partial Tiling algorithm that will be a more accurate reflection of the real biological evolution than the hypothetical evolutionary trees that could be built earlier [4]. We obtained some preliminary data on this, but a full demonstration of this idea remains an open problem that we may adress in an extended journal version of this paper.

References

[1] 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.

[2] R. Powers, J. Copeland, K. Germer, K. Mercier, 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.

[3] P. Z. Revesz, Introduction to Databases: From Biological to Spatio-temporal, Springer, 2010.

[4] 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.

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

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US

```
MSTGITYDEDRKTQLIAQYESVREVVNAEAKNVHVNENASKILLLVVSKLKPASDIQILYDHGVREFGENYVQELIEKAKLLPDDI
KWHFIGGLQTNKCKDLAKVPNLYSVETIDSLKKAKKLNESRAKFQPDCNPILCNVQINTSHEDQKSGLNNEAEIFEVIDFFLSEEC
KYIKLNGLMTIGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEVRIGTDIFGARPPKNEARII

MRPARALIDLQALRHNYRLAREATGARALAVIKADAYGHGAVRCAEALAAEADGFAVACIEEGLELREAGIRQPILLLEGFFEASE
LELIVAHDFWCVVHCAWQLEAIERASLARPLNVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEE
QLAAFSAASQGLEGEISLRNSPAVLGWPKVPSDWVRPGILLYGATPFERAHPLADRLRPVMTLESKVISVRDLPAGEPVGYGARYS
TERRQRIGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLL
CNLKRVPRVYSGA

Score: 29.0
jaligner_1       101 DLAKVPNLYS    110
                     :|.:||.:||
jaligner_2       346 NLKRVPRVYS    355

MSTGITYDEDRKTQLIAQYESVREVVNAEAKNVHVNENASKILLLVVSKLKPASDIQILYDHGVREFGENYVQELIEKAKLLPDDI
KWHFIGGLQTNKCKVETIDSLKKAKKLNESRAKFQPDCNPILCNVQINTSHEDQKSGLNNEAEIFEVIDFFLSEECKYIKLNGLMT
IGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEVRIGTDIFGARPPKNEARII

MRPARALIDLQALRHNYRLAREATGARALAVIKADAYGHGAVRCAEALAAEADGFAVACIEEGLELREAGIRQPILLLEGFFEASE
LELIVAHDFWCVVHCAWQLEAIERASLARPLNVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEE
QLAAFSAASQGLEGEISLRNSPAVLGWPKVPSDWVRPGILLYGATPFERAHPLADRLRPVMTLESKVISVRDLPAGEPVGYGARYS
TERRQRIGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLL
CGA

Score: 28.0
19 YESVREVVNAEA----------------------------------KNVHVNENASKILLLVVSKLKPASDIQILYDH   62
   |...||...|.|                                    :.:.:.|...:..:|::.....||::::::.|
17 YRLAREATGARALAVIKADAYGHGAVRCAEALAAEADGFAVACIEEGLELREAGIRQPILLLEGFFEASELELIVAH   93

MSTGITYDEDRKTQLIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKVETIDSLKKAKKLNESRAKFQPDCNPILCN
VQINTSHEDQKSGLNNEAEIFEVIDFFLSEECKYIKLNGLMTIGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSAD
FREAIRQGTAEVRIGTDIFGARPPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQLEAIERASLARPLNVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRAD
ELDCPRTEEQLAAFSAASQGLEGEISLRNSPAVLGWPKVPSDWVRPGILLYGATPFERAHPLADRLRPVMTLESKVISVRDLPAGE
PVGYGARYSTERRQRIGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQ
FGSIPYQLLCGA

Score: 28.0
jaligner_1       184 VRIGTDIFGARP    195
                     ||.|..::||.|
jaligner_2       130 VRPGILLYGATP    141

MSTGITYDEDRKTQLIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKVETIDSLKKAKKLNESRAKFQPDCNPILCN
VQINTSHEDQKSGLNNEAEIFEVIDFFLSEECKYIKLNGLMTIGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSAD
FREAIRQGTAEPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQLEAIERASLARPLNVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRAD
ELDCPRTEEQLAAFSAASQGLEGEISLRNSPAVLGWPKVPSDWFERAHPLADRLRPVMTLESKVISVRDLPAGEPVGYGARYSTER
RQRIGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLLCGA

Score: 24.00  (23.00 without the K/Q pair)
jaligner_1        56 KVETIDSLKKAKKLN     70
                     ::|.|:....|:.||
jaligner_2        27 QLEAIERASLARPLN     41
```

FIG. II. THE 1B54 PROTEIN (TOPMOST SEQUENCE) AND THE 1RCQ PROTEIN (SECOND SEQUENCE). THE FIRST, SECOND, THIRD, AND FOURTH ITERATIONS OF THE STRING SIMILARITY ALGORITHM FINDS THE SEGMENT PAIRS THAT ARE HIGHLIGHTED IN BLUE, BROWN, GREEN AND RED COLORS, RESPECTIVELY.

```
MSTGITYDEDRKTQLIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKESRAKFQPDCNPILCNVQINTSHEDQKS
GLNNEAEIFEVIDFFLSEECKYIKLNGLMTIGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTA
EPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEEQLA
AFSAASQGLEGEISLRNSPAVLGWPKVPSDWFERAHPLADRLRPVMTLESKVISVRDLPAGEPVGYGARYSTERRQRIGVVAMG
YADGYPRHAADGTLVFIDGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLLCGA

Score: 23.00
jaligner_1       108 KLNGLMTIGSWNVSHED    124
                     :|..:||:.|..:|..|
jaligner_2       125 RLRPVMTLESKVISVRD    141

MSTGITYDEDRKTQLIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKESRAKFQPDCNPILCNVQINTSHEDQKS
GLNNEAEIFEVIDFFLSEECKYISKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEEQLA
AFSAASQGLEGEISLRNSPAVLGWPKVPSDWFERAHPLADLPAGEPVGYGARYSTERRQRIGVVAMGYADGYPRHAADGTLVFI
DGKPGRLVGRVSMDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLLCGA

Score: 21.00
jaligner_1         4 GITYDEDRKTQL     15
                     |..|..:|:.::
jaligner_2       134 GARYSTERRQRI    145

MSTIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKESRAKFQPDCNPILCNVQINTSHEDQKSGLNNEAEIFEVI
DFFLSEECKYISKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEEQLA
AFSAASQGLEGEISLRNSPAVLGWPKVPSDWFERAHPLADLPAGEPVGYGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVS
MDMLTVDLTDHPQAGLGSRVELWGPNVPVGALAAQFGSIPYQLLCGA

Score: 21.00
jaligner_1        71 KSGLNNEAEIF      81
                     ::||.:..|::
jaligner_2       181 QAGLGSRVELW     191

MSTIAQGVREFGENYVQELIEKAKLLPDDIKWHFIGGLQTNKCKESRAKFQPDCNPILCNVQINTSHEDQEVIDFFLSEECKYI
SKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEPKNEARII

MRPARALIDLQALRHNDFWCVVHCAWQVWLKMDSGMHRVGFFPEDFRAAHERLRASGKVAKIVMMSHFSRADELDCPRTEEQLA
AFSAASQGLEGEISLRNSPAVLGWPKVPSDWFERAHPLADLPAGEPVGYGVVAMGYADGYPRHAADGTLVFIDGKPGRLVGRVS
MDMLTVDLTDHPGPNVPVGALAAQFGSIPYQLLCGA

Score: 21.00
jaligner_1       109 LKLSMGM     115
                     ||:..||
jaligner_2        30 LKMDSGM      36
```

FIG. III. THE FIFTH, SIXTH, SEVENTH, AND EIGHT ITERATIONS OF THE STRING SIMILARITY ALGORITHM.

```
MSTGITYDEDRKTQLIAQYESVREVVNAEAKNVHVNENASKILLLVVSKLKPASDIQILYDHGVREFGENYVQELIEKAKLLPD
DIKWHFIGGLQTNKCKDLAKVPNLYSVETIDSLKKAKKLNESRAKFQPDCNPILCNVQINTSHEDQKSGLNNEAEIFEVIDFFL
SEECKYIKLNGLMTIGSWNVSHEDSKENRDFATLVEWKKKIDAKFGTSLKLSMGMSADFREAIRQGTAEVRIGTDIFGARPPKN
EARII
```

FIG. IV. THE ORIGINAL 1B54 PROTEIN WITH THE SEGMENTS THAT CLOSELY MATCH CORRESPONDING SEGMENTS IN THE 1RCQ PROTEIN