*Blast*ing Out of the Box:

Saad Mneimneh Weigang Qiu

October 9, 2007

Abstract

Identification of homologous sequences is a fundamental problem in bioinformatics. Increasing the statistical power of detecting distantly related homologs will greatly help biomedical researchers in genome annotation, such as identifying functions and structures of an uncharacterized protein based on a known homolog. We propose an approach to increase the sensitivity of Blast in detecting homologs. Unlike existing methods, this approach relies on reconstructing an ancestor sequence, which gives a biological justification for homology detection based on the first principles of sequence evolution. Our preliminary studies using exploratory, empirical, and theoretical analyses showed the validity and promise of this novel approach, called "REWIND".

1 Introduction and Significance

Two proteins are homologs if they have the same biological origin. Homology detection is therefore one of the most important tools for the study of evolution. However, given a query protein, biologists often rely on *Blast* (Basic Local Alignment Search Tool) [1] to discover other proteins similar in their amino acid sequence to the query sequence. Blast reports such similarities by assigning scores based on statistical techniques. A homolog protein may then be considered to be one that achieves a high score given the query. However, while Blast is an efficient tool to search for highly similar sequences, a homolog that diverged considerably by evolution may not achieve a high enough score with Blast, and hence may not be reported. Furthermore, the construction of evolution trees (phylogeny) often demonstrates that evolutionary close proteins are not necessarily those who exhibit the highest sequence similarity [2] (see Section 2 for a toy example).

Blasting out of the box is our proposal to remedy the above limitation of Blast. The central idea is the following: **Instead of Blasting the query protein against a database, we Blast the database using an ancestral sequence**. In most cases, this ancestor is unknown. Therefore, given a query protein, we obtain similar proteins using Blast. From these proteins, we reconstruct the **ancestral** protein sequence. Such an ancestor is fictitious and, therefore, does not necessarily belong to the database or the query dataset, hence the title of our proposal. A new search using Blast and the ancestor as query is now likely to reveal more homologs. The procedure can then be iterated until no more homologs are identified. We call this novel homology-searching approach **REWIND**. [1]

This project is **significant** in at least three aspects. First, homology identification is a fundamental problem in bioinformatics because homologs are the basis for evolutionary inference, domain and motif identification, as well as protein structure identification. The success of our approach will therefore have a high impact to virtually all fields of bioinformatics. Second, our approach has the potential to shift the paradigm of homology-searching based on scoring matrices (like BLAST) to homology-searching based on the first principles of molecular evolution. Although scoring matrices used in BLAST are implicitly evolution-based, such indirect approach looses much of the information that could have been captured more fully by using models of sequence evolution. Third, the main challenge in homology-searching is the identification of distantly-related homologs. Examples are bacterial species that diverged over a billion years ago, fungal species that diverged over 500 million years ago, and protein molecules that share the same 3D fold but have diverged for far too long to retain any significant sequence similarity. Our approach is expected to have the most impact in identifying remote homology. In addition, since this project will be a collaboration between an biologist (Qiu) and a computer scientist (Mneimneh), it will have a high impact to promote bioinformatics research and education in CUNY.

[1] The word refers to an analogy by Steven Jay Gould, an eminent evolutionary biologist, who often use "rewinding the tape of evolution" to illustrate historical contingencies in evolution. Here, we are *rewinding* extant sequences to their ancestors.

2 Background

2.1 Evolution vs. similarity: a toy example

The construction of evolution trees often attempts to minimize the number of changes needed to explain evolution across all branches of the tree. More generally, a change from one amino acid to another contributes a certain weight (obtained statistically, either from experimental data or by probabilistic optimization methods).

Consider the following amino acid sequences:

$$A = EFGHIK$$

$$B = QYGHIK$$

$$C = QFTHIK$$

$$D = EYGNLR$$

Sequences A and B are the most similar sequences with a difference of two amino acids only. However, an evolution tree is likely to place A and B apart from each other to minimize the total number of changes (or weighted changes) across all branches of the tree. This is illustrated below showing a total of seven changes:

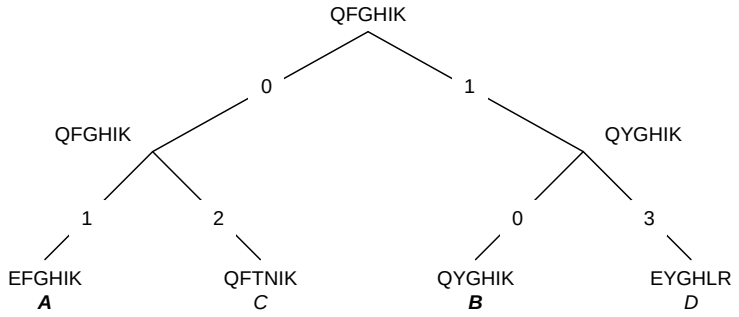

Figure 1: Evolution tree of toy example

More importantly, if C is our query protein, D may not be reported by Blast because C and D are highly divergent in their sequences (a total difference of six amino acids). However, the ancestor sequence shown above is more similar to D with a difference of four amino acids. Using Blast with the ancestor sequence as query will not only find A, B, and C, but also possibly D. Note that this ancestor could have been reconstructed from A, B, and C only (connect the ancestor to B by a direct branch and ignore the rest of the branches to its right). Note that one approach (see transitive closure in Section 3) would be to use A and/or B as query instead of the ancestor to possibly identify D as homolog (a difference of three amino acids). However, reversing the roles of C and D reveals an even stronger similarity between the ancestor and the possibly newly detected sequence (only a difference of two amino acids). This is generally true whenever the rates of evolution in the tree are high [3][otu], i.e. we expect that the ancestor is closer to a distant (undiscovered) homolog than any of the discovered leaves is to that homolog. Moreover, based on a preliminary analytical study of a random evolution tree, the number of changes from the ancestor to a leaf is not larger than that from any leaf to another, if leaves are distant enough (which is the interesting case).

2.2 Related work

The literature if full of attempts to enhance the sensitivity of Blast. We list here the two main approaches.

2.2.1 Transitive closure

This approach uses the similar sequences reported by Blast as a query set for a subsequent round. This is repeated until no more new sequences can be found [6]. There are two major problems with this approach: First, the query set may become large, yielding to a much longer search. Second, sensitivity is increased at the expense of a larger number of false positives (explosion), where potentially the whole database may become part of the query set. We expect that using the ancestor sequence as query will exhibit a smaller number of false positives, even when the procedure is iterated, and especially when the evolution rates are high.

2.2.2 Substitution matrix modification

This approach is based on modifying the statistics underlying the scoring mechanism of Blast. Blast uses a symmetric 20×20 matrix S, called substitution matrix, that assigns a score S_{ij} for substituting amino acid i with amino acid j. Using a statistical theory of similar sequences [4], [5], such a matrix can be constructed from *real biological data* of similar proteins (sequences of amino acids). However, this construction is highly dependent on the data at hand.

It is often the case that the distribution amino acids in the proteins of interest is very different from that of the *real biological data* used to construct the substitution matrix. Therefore, the substitution matrix is modified to take into account this new distribution. Blast-gtQ [6] is an example of such a modification and deals with low complexity proteins where the frequency of the Serine and Threonine amino acids is relatively high.

In another approach, similar sequences reported by Blast are used as the *real biological data* to recompute the substitution matrix. An example of this approach is psi-Blast [7]. To achieve even higher sensitivity, psi-Blast computes a different substitution matrix for each specific position in the query sequence, based on the distribution of amino acids for that position in the reported similar sequences. This procedure is then iterated, and new sequences reported by Blast are used to recompute position specific matrices in the same way. Thus the name psi-Blast (position specific iterated Blast), which is known for its high sensitivity.

None of the approaches described above, however, accounts for evolution. This makes our approach significantly unique, and biologically more relevant. However, one may wonder, for instance, whether the iterative nature of psi-Blast corrects the statistics enough to account for the evolutionary aspect of the proteins. We doubt that this is the case (but this is to be explored). Can psi-Blast replace our evolution based approach? This is answered in the following section by showing that our approach enhances the sensitivity of psi-Blast.

3 Preliminary Results

3.1 A proof of concept

While the toy example of Section 2 gives enough motivation for our REWIND approach, we need to justify that such an approach actually works. Moreover, we need to demonstrate that other alternatives (which are not evolution based), like psi-Blast for instance, do not provide a good substitute for our approach. In fact, we show in this section that the REWIND approach can be used within psi-Blast to further enhance its sensitivity.

We constructed an evolution tree for outer surface vaccine target proteins (ospA) from four Lyme disease bacteria (*Borrelia burgdorferi*) strains PKo, PBi, DN127, and B31, using PHYLIP [8]. This type of tree is called maximum parsimony and is based on minimizing a weighted function of the total number of changes.

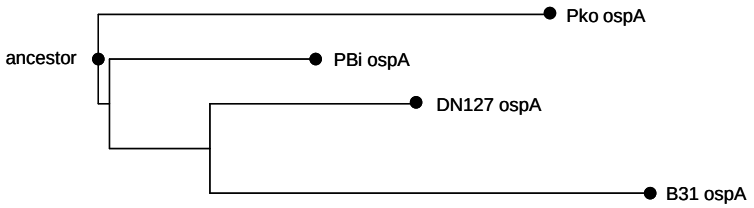

Figure 2: Evolution tree for four proteins. The length of a branch represents the distance (weighted number of changes) along that branch.

We performed Blast and psi-Blast with each of PKo, PBi, DN127, and the ancestor being the query sequence against the B31 genome as the database (containing 1,639 protein sequences). Table 1 shows that the constructed ancestor sequence has a better statistical significance for its similarity to B31 than PKo and PBi in Blast and psi-Blast (DN127 being evolutionary the closest to B31).

Table 1 suggests that the ancestor is more likely to identify remote homologs when used instead of the query. Note that the proteins in the above example are closely related, and we expect even better improvement of ancestor over query when proteins are highly divergent (see Figure 3 below).

3.2 Power analysis

We ask how much improvement in the power of detecting homologs is expected from the REWIND approach. The probability that an amino acid remains unchanged by evolution is given by a Poisson distribution and is equal to e^{-D}, where D is the average evolutionary rate of change. By reconstructing an ancestor, the REWIND approach potentially

	statistical significance (smaller is better)	
query	Blast	psi-Blast
PKo	$2 \cdot 10^{-72}$	$6 \cdot 10^{-91}$
PBi	$3 \cdot 10^{-77}$	$2 \cdot 10^{-92}$
DN127	$1 \cdot 10^{-80}$	$2 \cdot 10^{-94}$
ancestor	$7 \cdot 10^{-80}$	$2 \cdot 10^{-94}$
ancestor *	$6 \cdot 10^{-79}$	$6 \cdot 10^{-94}$

* B31 ospA excluded during construction

Table 1: Statistical significance of Blast and psi-Blast for the shown queries against B31 genome as the database. Statistical significance should not be compared across columns because Blast and psi-Blast use different scoring mechanisms.

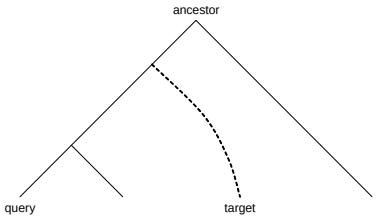

Figure 3: Ancestor is evolutionary closer to target

reduces this rate by half, so that the above probability increases to $e^{-D/2}$. The lower limit of sequence identity that can be reliably detected by BLAST as homolog is about 25%. At this level of sequence identity, regular BLAST detects homologs with an evolution rate of $D = -ln(0.25) = 1.38$, while the evolution rate detectable by REWIND would be $D = -2ln(0.25) = 2.76$, twice that of BLAST. This latter distance corresponds to a sequence identity value of 6.25%. Therefore, REWIND could theoretically lower the threshold of sequence identity for homolog identification to only about 6%.

4 Project Design

4.1 Reconstructing the ancestor

Many tools for constructing phylogeny are available, such as the one we used in Section 4. However, these tools are likely to represent a bottleneck for the Blast search. We propose here two approaches to reconstruct the ancestor based on heuristics. After all, it is not the tree itself that we are particularly interested in, and an approximate ancestor is likely to serve our purpose.

Given the result of Blast, we first construct an alignment of the highly scoring sequences found, by placing those sequences against the common template of the query to match the highly scoring segments. A cartoon example is illustrated in Figure 4.

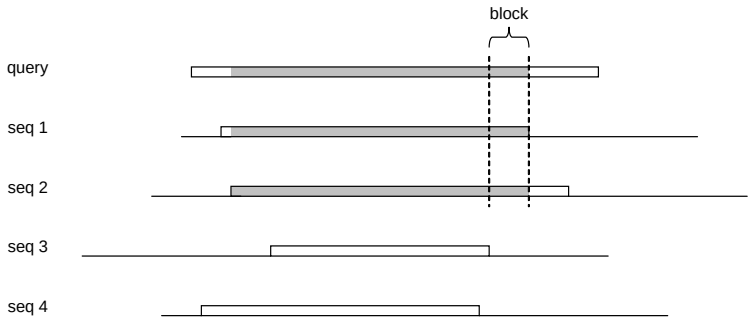

Figure 4: Alignment of Blast result. Lines represent sequences, and rectangles represent segments of sequences that score highly with the query. The shaded partial alignment of query, seq 1, and seq 2, is used to recompute the block specific substitution matrix.

Given the obtained alignment, we define a block to consist of contiguous positions in the query that belong to the same set of highly scoring segments of the alignment. An example block is shown in Figure 4 where contiguous positions in the query belong to the first and second highly scoring segments. The alignment involving those segments (shaded in Figure 4) will be used to recompute a substitution matrix specific for that block, in a similar way to psi-Blast [7]. The alignment can then be viewed as a set of disjoint blocks, each with its block specific substitution matrix. The sequences within a block, together with the block specific substitution matrix, are then used to reconstruct the ancestor sequence for that block, i.e. for the corresponding part of the query. In the following discussion, an ancestor is implicitly understood as an ancestor for a particular block.

4.1.1 Consensus approach

Given the block specific substitution matrix, we seek a sequence that maximizes the average score with all the sequences within the block. We use that sequence, called consensus, as the ancestor. This approach is similar to [12], except that we use block specific scores, and we do not restrict a protein of the consensus sequence to be among those that appear in the block at the given position.

4.1.2 Spanning tree approach

Given the block specific substitution matrix, we compute the score for every pair of sequences within the block. This information can be represented by a complete weighted graph as follows: let every sequence be a node in the graph. Given any two nodes u and v, let the weight of edge (u, v) be the score of the corresponding pair of sequences. Given this graph, we find the maximum spanning tree, a tree that maximizes the weight of its edges. This is computationally efficient. We then transform the spanning tree into an evolution tree using the following procedure: We remove an edge from the tree. This will disconnect the nodes into two groups. We then recursively remove an edge from each group until we end up with all groups having one node each. This recursive grouping defines an evolution tree on the sequences. Maximum parsimony is then used to reconstruct the ancestor given the tree, also an efficient computation.

The order of edge removals determines the resulting evolution tree. Therefore, this is an area that must be investigated. We envision two possible strategies:

- SWEF: removes small weight edges first and, therefore, favors trees in which the highly scoring sequences are kept together. This strategy may induce a large deviation in the constructed ancestor because the highly scoring sequences are local in the tree.
- LWEF: removes large weight edges first and, therefore, favors trees in which highly scoring sequences are kept apart. This strategy, though not realistic, may exhibit better stability in the constructed ancestor, i.e. the ancestor will not deviate considerably from the highly scoring sequences because they are all over the tree.

The two above strategies may be good representatives for the tradeoff between sensitivity and false positives.

4.2 Validation

We will validate our REWIND approach using large benchmarking/standard datasets like Pfam [9], SCOP [10], and Aravind [11]. With these datasets, the goal is to detect remote homologs with increased sensitivity and the fewest possible false positives. The success of our approach can be claimed by an improved ROC curve (a graphical plot of true positives vs. false positives) over existing methods such as transitive closure and psi-Blast.

4.3 Implementation and Application Development

We will first publish our algorithm and validation results in a selected computational biology journal. A long term goal of the project is to develop a stand-alone application package (preferably web-based) of the REWIND approach of homology identification. However, implementation is beyond the scope of this project. We will seek external support for implementation and application development after completing this initial project of algorithm development and validation studies.

References

[1] Altschul, S. F., Gish W., Miller W., Myer W., and Lipman D. J. (1990). *J. Mol. Bio.*, 215, 403-410.

[2] some evolution reference.

[3] some OTU reference.

[4] Karlin S., Dembo A., and Kawabata T. (1990). *Ann. Stat.*, 18(2), 571-581.

[5] Karlin S., Altshul S. F. (1990). *Proc. Natl. Acad. Sci.*, 87, 2264-2268.

[6] Coronado J. E., Attie O., Epstein S. L., Qiu W-G., and Lipke P. (2006). *Eukaryotic Cell*, 628-637.

[7] Altschul S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W., and Lipman D. J. (1997). *Nucleic Acid Research*, 25(17), 3389-3402.

[8] Felsenstein J. (2005) PHYLIP phylogeny inference package, University of Washington.

[9] some reference for Pfam.

[10] some reference for SCOP.

[11] some reference for Aravind.

[12] Henikoff S. and Henikoff J. G. (1997). *Protein Science*, 6, 698-705.