Learning Significant Alignments: An Alternative to Normalized Local Alignment

Eric Breimer and Mark Goldberg [*]

Computer Science Department, Rensselaer Polytechnic Institute, 110 Eight Street, Troy NY 12180, USA, breime@cs.rpi.edu

Abstract. We describe a supervised learning approach to resolve difficulties in finding biologically significant local alignments. It was noticed that the $O(n^2)$ algorithm by Smith-Waterman, the prevalent tool for computing local sequence alignment, often outputs long, meaningless alignments while ignoring shorter, biologically significant ones. Arslan *et. al.* proposed a $O(n^2 \log n)$ algorithm which outputs a *normalized local alignment* that maximizes the degree of similarity rather than the total similarity score. Given a properly selected normalization parameter, the algorithm can discover significant alignments that would be missed by the Smith-Water algorithm. Unfortunately, determining a proper normalization parameter requires repeated executions with different parameter values and expert feedback to determine the usefulness of the alignments. We propose a learning approach that uses existing biologically significant alignments to learn parameters for intelligently processing sub-optimal Smith-Waterman alignments. Our algorithm runs in $O(n^2)$ time and can discover biologically significant alignments without requiring expert feedback to produce meaningful results.

1 Background

Local sequence alignment is an essential technique for identifying similarity between biological sequences [6, 8, 10]. The Smith-Waterman algorithm [15] is considered the standard tool for computing local sequence alignment. However, it was noticed (see, [2, 4]) that the algorithm has two essential flaws. It often combines two or more segments of high similarity and aligns internal segments that are not related; the *mosaic effect*, see Fig. 1 (ii). Occasionally, it finds long alignments with a high score and misses shorter ones with a higher degree of similarity; the *shadow effect*, see Fig. 1 (i).

A number of attempts have been made to correct the flaws of the Smith-Waterman algorithm, including unsuccessful approaches that were abandoned [11, 14], approaches that are computationally expensive [12, 17], and approaches that require sensitive heuristics [18, 3]. Further attempts were made to consider length in the computation of alignments [5, 13, 16]. The most recent and successful approach was proposed by Arslan et. al. [4]. This approach seeks to maximize

[*] Supported by a grant from Rensselaer Polytechnic Institute.

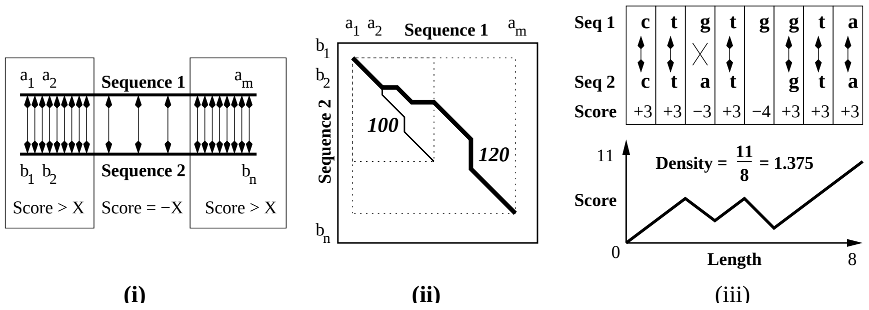

Fig. 1. (i) Mosaic effect: Two high scoring segments are joined into a single alignment that is not biologically significant. (ii) Shadow effect: Sub-optimal alignments with high degree of similarity are ignored in favor of longer alignments. (iii) Alignment Density: Alignment score divided by the length of the alignment.

the *normalized score* S_N defined by

$$S_N = \frac{S}{l+N},$$

where S is the conventional alignment score, l is the sum of the lengths of the two aligned segments, and N is the *normalization parameter* used to control the degree of normalization. Arslan et. al. [4] designed an $O(n^2 \log n)$ algorithm for computing normalized local alignment (*nla*) that uses the fractional programming technique developed in [9]. Unfortunately, the algorithm is very sensitive to the value of N. If N is too small, the algorithm is indistinguishable from an exact matching algorithm, whereas if N is too large, the algorithm suffers from the same negative side effects of the Smith-Waterman algorithm. The useful range for N is input-dependent and the relationship between the input and the appropriate value of N is generally unclear [4]. Thus, applying the algorithm requires *guessing* a preliminary value for N, and obtaining meaningful alignments may require repeated execution of the algorithm with a varying value of N. Repeatedly executing the *nla*-algorithm and manually evaluating the results is tedious and time consuming. For large scale applications, a more automated and efficient process is still needed.

For the remainder of this paper, an alignment that captures biologically significant similarity is called a *motif.* In practice, motifs are identified by expert biologists. Coincidental alignments without biological significance are called *padding.* The training data consists of pairs of sequence segments where all motifs are known. Thus, given any alignment, we can identify the specific segments that are motifs and those that are padding. Degree of similarity, also called *density,* refers to the alignment's score divided by the alignment's length (see Fig. 1 (iii)). Alignment density, although similar to normalized score, does not include a normalization parameter and defines length to be the number of aligned symbol pairs.

```
while $i < m_A$ and $j < m_B$
    if ($A_i^x == B_j^x$)
        if ($A_i^y == B_j^y$) Mark that $A_i$ and $B_j$ overlap
        $i = i + 1$
        $j = j + 1$
    else if ($A_i^x > B_j^x$) $j = j + 1$
    else $i = i + 1$
```

Fig. 2. Algorithm for computing the overlap between two alignments A and B of lengths m_A and m_B respectively. Each alignment is represented as a sequence of index pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ where (x_i, y_i) indicates that the symbols a_{x_i} and b_{y_i} from input sequences $a = a_1 \ldots a_n$ and $b = b_1 \ldots b_n$ are aligned.

2 Learning Approach

Our learning approach uses input sequences with known motifs to train an algorithm to align and extract these motifs with high probability. It is expected that the motifs possess certain generalizable characteristics that the trained algorithm can use to perform well on similar inputs. This expectation is not unreasonable for biological sequences given that the motifs arise from underlying processes of mutation and conservation that are common among all inputs. We describe a system that learns a strategy for exploring sub-optimal Smith-Waterman alignments and learns parameters for processing these alignments to identify motifs and ignore padding. The main goals are to train an algorithm that is superior to the Smith-Waterman algorithm in its ability to align biologically similar regions and to improve upon the $O(n^2 \log n)$ runtime of the *nla* algorithm.

2.1 Sub-optimal Alignments

It is well known (see [2]) that the alignment's degree of similarity is an important measure in identifying biologically significant similarity. Alignments with the highest degree of similarity may not be captured by the maximum scoring Smith-Waterman alignment, but are often captured by sub-optimal alignments [19]. Given a training input, we use a modification of the Smith-Waterman algorithm similar to that proposed in [7] to efficiently output the non-overlapping maximal scoring alignments to see if a motif is discovered, *i.e.*, contained within a sub-optimal alignment. The precise locations of overlap between any two alignments can be computed in linear time using the algorithm in Fig. 2.

In practice, alignments that arise from coincidence rarely score higher than motifs. A Smith-Waterman alignment that contains a motif will naturally score higher than the motif alone, otherwise the algorithm would have returned the motif and ignored the lower scoring alignment. If a motif is embedded in a sub-optimal alignment, it will typically be among the top k scoring alignments, where k is the number motifs in the input. If the number of expected motifs is known, we can limit the number of computed alignments to improve the efficiency of training or the efficiency of the learned algorithm.

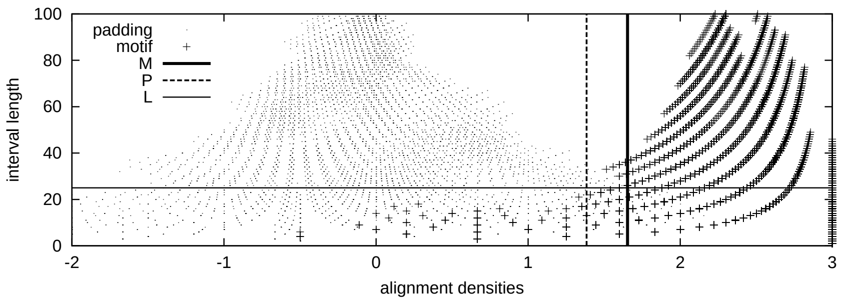

Fig. 3. Alignment densities: Given a training sample, we examine the top k Smith-Waterman alignments using different sampling interval lengths. The left and right clusters represent intervals contained within padding and motif respectively.

2.2 Alignment Density

Due to the mosaic effect, high scoring alignments often contain two or more motifs separated by padding. A single motif may also be embedded in a large alignment with padding to the left and right of the motif. In practical input, the motifs have significantly higher alignment density compared to the padding. To sample density, we choose two alignment indices i and j $(i < j)$ and compute density d defined by

$$d = \frac{s(j) - s(i)}{j - i + 1}$$

where $s(i)$ and $s(j)$ are the alignment scores at position i and j, and $j - i + 1$ is defined at the *sampling interval length*. Given an alignment with known motifs, we can sample the density of the motifs and the padding to determine thresholds for discriminating the two. However, this density difference is not evident if the sampling interval length is very small. Small segments of the padding may have high density; similarly, small segments of the motif may possess low density. A large sampling interval may cover two or more motifs, which hinders the ability to identify the start and end of individual motifs. To use density thresholds effectively, one must carefully choose the sampling interval length.

By sampling and plotting segment densities using different interval lengths we can simultaneously detect the minimum interval length and density thresholds that adequately discriminates motif from padding. Figure 3 shows alignment density plotted over interval length. The right cluster represents intervals entirely contained within a motif, while the left cluster represents intervals entirely contained within the padding. Intervals that overlap motifs and padding are omitted from the plot. As the interval length increases the disparity between motif density and padding density increases. Our goal is to find the minimal interval length L that adequately discriminates motif from padding. Since very long intervals are undesirable, we can limit the length of the sampling intervals so that this process is efficient.

```
$length = 1$
$Sort\ m\ in\ ascending\ order\ according\ to\ density$
$Sort\ p\ in\ descending\ order\ according\ to\ density$
while $(m_j^d < p_k^d)$
    $length = length + 1$
    For all $m_i$, if $m_i^l < length$ then remove $m_i$
    For all $p_i$, if $p_i^l < length$ then remove $p_i$
    $j = |m| * (1 - r)$
    $k = |p| * (1 - r)$
$L = length$
$M = m_j^d$
$P = p_k^d$
```

Fig. 4. Algorithm for computing L, M, and P where r is the percentage of points that must achieve the thresholds, m_l^i and m_d^i are the length and density of the ith motif data point, and p_l^i and p_d^i are the length and density of the ith padding data point.

Minimum interval length (L)
The minimum length L such that all motif points above L fall to the right of some density threshold and all padding points fall to the left of that threshold.

Maximum motif threshold (M)
The maximum density M such that all motif points above L fall to the right of M.

Minimum padding threshold (P)
The minimum density P such that all padding points above L fall to the left of P.

These thresholds are relaxed so that a small percentage of outlying points can be ignored. Figure 4 shows the algorithm for computing L, P, and M. Preliminary experiments indicate that the thresholds accurately identify the start and end of the motifs. Moreover, these thresholds do not significantly change as the motif lengths and padding lengths of the training data are varied, or if the number of motifs or their relative positioning in the input sequences vary.

2.3 Learning

Our learning system uses the observations above to generate a local alignment algorithm. To generate an algorithm, the system requires at least one training sample and implements the following steps.

Learning Stage

1. **Find the motifs:** Given an input pair with k known motifs, output k maximal scoring Smith-Waterman alignments in order of total alignment score. Search each alignment for the known motifs and label each alignment accordingly.
2. **Obtain density statistics:** From the labeled alignments, scan each motif segment with varying intervals and obtain the motif density statistics. Likewise, scan each padding segment and obtain the padding density statistics.
3. **Compute thresholds:** Repeat steps one and two until the training data is exhausted or until some time or computation limit has exceeded. After training is complete compute L, M, and P accordingly.

Our algorithm requires $O(n^2)$ time and $O(n + k)$ space to output the score and the endpoints of k alignments. Each alignment can be generated using $O(n_i^2)$ time and $O(n_i)$ space, where n_i is the individual length of each alignment. If k is reasonably small, *i.e.*, $k < 200$, the total time to compute and sort all k alignments is only 2 or 3 times longer than the basic Smith-Waterman algorithm. Computing the overlap locations between the top k alignments and the known motifs is linear with respect to the sum of the lengths of the alignments. After labeling the sub-optimal alignments, the system samples the scoring densities of all the contiguously labeled segments and computes the thresholds. Once obtained, the thresholds define a customized algorithm for processing sub-optimal alignments. Assuming that the learned thresholds generalize well over a broad set of input, the new algorithm is expected to discover alignments that exhibit the same characteristics as the motifs used for training.

Figure 5 shows the post-processing algorithm for labeling alignments. The algorithm scans the alignment until the score begins increasing. From the start position of the increase j, the algorithm marks each successive position i as a motif if one of two conditions is satisfied: (i) the alignment density from i to $i+L$ exceeds the motif threshold M, or (ii) the alignment density from i to $i+L$ exceeds the padding threshold P and the alignment density from j to i exceeds the motif threshold M. The first condition indicates that i is the beginning of a significantly long alignment that satisfies the motif threshold. The second condition indicates that i is towards the end of a motif but is not yet at the beginning of a significantly long alignment that satisfies the padding threshold. We mark the position m where the maximum score occurs so that when the two conditions are violated we can mark every position from m to i as padding, since this segment decreases the score. Other post-processing algorithms were tested but were not as effective or efficient.

```
$j = 0$; $i = 1$
while ($i < m_A$)
    while ($A^s_i < A^2_j$)
        *Mark* $A_i$ *as padding*; $j = j + 1$; $i = i + 1$
    $m = 0$; $d_m = 0.0$; $w = min(i + L, m_A - 1)$
    $d_f = (A^s_w - A^s_i)/(w - i)$; $d_b = (A^s_i - A^s_j)/(i - j)$
    if ($d_f > M$)
    while ($d_f > M$ or ($d_f > P$ and $d_b > M$))
        *Mark* $A_i$ *as motif*
        if ($A^s_i > d_m$)
            $d_m = A^s_i$; $m = i$
        $i = i + 1$; $w = min(i + L, m_A - 1)$
        $d_f = (A^s_w - A^s_i)/(w - i)$; $d_b = (A^s_i - A^s_j)/(i - j)$
    $z = i - 1$
    while ($z > m$)
        *Mark* $A_i$ *as padding*; $z = z - 1$
    $j = i$; $i = j + 1$
```

Fig. 5. Algorithm for processing sub-optimal alignments where m_A is the length of alignment A and A^s_i is the alignment score at position i.

3 Experiments

We experimented on specific regions of the human[1] and mouse [2] genome where there exist highly conserved segments that have been identified as significant by biologists. We extracted a pair of sub-regions, one from the human and one from the mouse, which shared k motifs ranging in length from n_1 to n_2. We train on several samples, compute the thresholds, and cross-test to verify that the post-processing algorithm accurately identifies the known motifs of the training data. We then apply the learned thresholds to unseen samples with known motifs.

To discover a motif the algorithm must compute a contiguous alignment that contains 90% of a particular motif (called the *accuracy requirement*). To extract a motif the post-processing algorithm must identify a contiguous segment such that the length of overlap between the segment and the true motif is greater than 90% of the length of the segment (called the *precision requirement*).

We compare the output of our algorithm before and after post-processing. Before post-processing, our algorithm outputs the top k scoring alignments and computes how many motifs were discovered according to the accuracy requirement. After post-processing, zero or more segments may be identified as potential motifs. For each of these segments, we test to see if they satisfy both the accuracy requirement and the precision requirement. Our expectation is that every motif discovered before post-processing is precisely extracted by the post-processing algorithm. Table 1 summarizes the results of these experiments. In both train-

[1] Homo sapiens ATP-binding cassette, sub-family B (MDR/TAP), member 11
[2] Mus musculus ATP-binding cassette, sub-family B (MDR/TAP), member 11

Table 1. Training and Testing

Training

Trial	k	n_1–n_2	Discovered	Extracted
1	3	112–218	3/3	3/3
2	4	103–147	4/4	4/4
3	5	99–123	5/5	5/5
4	7	64–111	6/7	6/6
5	10	52–106	8/10	8/8
		Summary	26/29	26/26

Testing

Trial	k	n_1–n_2	Discovered	Extracted
1	3	118–204	3/3	3/3
2	4	114–185	4/4	4/4
3	6	103–157	6/6	6/6
4	8	66–128	7/8	7/7
5	10	64–105	9/10	8/9
		Summary	29/31	28/29

ing and testing combined, the post-processing algorithm extracted 54 out of 55 motifs that were discovered by the Smith-Waterman algorithm. It is important to note that the majority of these motifs (30 out of 55) were embedded into very long alignments that contained more than one motif.

4 Discussion

When the normalization parameter N is carefully selected, the *nla* algorithm can potentially output significant alignments that would not be contained in any of the sub-optimal alignments produced by the Smith-Waterman algorithm. This occurs when significant alignments partially overlap the padding of two adjoined motifs (a result of the mosaic effect) or when significant alignments partially overlap a very long, insignificant alignment (a result of the shadow effect). By isolating padding and recomputing the alignment, it may be possible to discover alignments hidden in the padding between adjoined motifs. Adding another stage that identifies and isolates padding for further motif discovery may prove to be useful. It is important to note that such a stage would be impossible without an approach for discriminating motifs and padding.

While it is unclear how to discover significant alignments hidden by the shadow effect, it is important to consider the likelihood of such alignments in practical data. Although the *nla* algorithm appears promising, determine the proper range and granularity of N-values to compute these alignments is also unclear. One approach might be to apply supervised learning to automatically determine an appropriate range for N. Unfortunately, training on one sample requires the repeated execution of the $O(n^2 \log n)$ *nla* algorithm. In our approach, training on one sample requires only a single execution of our algorithm followed by post-processing, which together requires $O(n^2)$ time in practice.

While our algorithm does not eliminate all the flaws of the Smith-Waterman algorithm, it improves its discovery capabilities without adding severe computational costs. Our approach uses supervised learning to automatically generate thresholds for post-processing. We provide a system where existing knowledge of biological similarities can be used to automatically generate effective heuristics.

References

1. Alexandrov, N., Solovyev, V.: Statistical significance of ungapped alignments. Pacific Symp. on Biocomputing (1998) 463–472
2. Altschul, S., Erickson, B.: Significance levels for biological sequence comparison using nonlinear similarity functions. Bulletin of Mathematical Biology **50** (1988) 77–92
3. Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped Blast and Psi-Blast: a new generation of protein database search programs. Nucleic Acids Research **25** (1997) 3389–3402
4. Arslan, A., Eğecioğlu, Ö., Pevzner, P.: A new approach to sequence comparison: normalized sequence alignment. Proceeding of the Fifth Annual International Conference on Molecular Biology (2001) 2–11
5. Arslan, A., Eğecioğlu, Ö.: An efficient uniform-cost normalized edit distance algorithm. 6th Symp. on String Processing and Info. Retrieval (1999) 8–15
6. Bafna, V., Huson, D.: The conserved exon method of gene finding. Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Bio. (2000) 3–12
7. Barton, G.: An efficient algorithm to locate all locally optimal alignments between two sequences allowing for gaps. Computer Applications in the Biosciences **9** (1993) 729–734
8. Batzoglou, S., Pachter, L., Mesirov, J., Berger, B., Lander, E.: Comparative analysis of mouse and human DNA and application to exon prediction. Proc. of the 4th Annual Int. Conf. on Computational Molecular Biology (2000) 46–53
9. Dinkelbach, W.: On nonlinear fractional programming. Management Science **13** (1967) 492–498
10. Gelfand, M., Mironov, A., Pevzner P.: Gene recognition via spliced sequence alignment. Proc. Natl. Acad. Sci. USA **93** (1996) 9061–9066
11. Goad, W., Kanehisa, M.: Pattern recognition in nucleic acid sequences: a general method for finding local homologies and symmetries. Nucleic Acids Research **10** (1982) 247–263
12. Huang, X., Pevzner, P., Miller, W.: Parametric recomputing in alignment graph. Proc. of the 5th Annual Symp. on Comb. Pat. Matching (1994) 87–101
13. Oommen, B., Zhang, K.: The normalized string editing problem revisited. IEEE Trans. on PAMI **18** (1996) 669–672
14. Seller, P.: Pattern recognition in genetic sequences by mismatch density. Bull. of Math. Bio. **46** (1984) 501–504
15. Smith, T., Waterman, M.: Identification of common molecular subsequences. Journal of Molecular Biology **147** (1981) 195–197
16. Vidal, E., Marzal, A., Aibar, P.: Fast computation of normalized edit distances. IEEE Trans. on PAMI **17** (1995) 899–902
17. Zhang, Z., Berman, P., Miller, W.: Alignments without low-scoring regions. J. Comput. Biol. **5** (1998) 197–200
18. Zhang, Z., Berman, P., Wiehe, T., Miller, W.: Post-processing long pairwise alignments. Bioinformatics **15** (1999) 1012–1019
19. Zuker, M.: Suboptimal sequence alignment in molecular biology: alignment with error analysis. Journal of Molecular Biology **221** (1991) 403–420