DNA SEQUENCE ALIGNMENT USING THE MATCHING PURSUIT DECOMPOSITION

Lakshminarayan Ravichandran[1], *Antonia Papandreou-Suppappola*[1], *Andreas Spanias*[1],

Zoé Lacroix[1,2], *and Christophe Legendre*[1]

[1]Dept. of Electrical Engineering, Arizona State University, Tempe AZ 85287, USA
[2]Pharmaceutical Genomics Division, Translational Genomics Research Institute (TGen)
13400 E Shea Blvd, Scottsdale AZ 85259, USA

ABSTRACT

Sequence alignment is the positioning of primary biological sequences, such as DNA, RNA and protein sequences, to identify regions of similarity in large databases. Common signal processing techniques include cross-correlations in time or frequency. However, these techniques can result in many misalignments when capturing a grouping in local or repetitive portions of the sequence. We propose a time-frequency based alignment technique using the matching pursuit decomposition method and a mapping algorithm. The aim of this alignment technique is to identify local and global alignments more efficiently and with greater precision than existing methods. Its success is based on the fact that sequence elements are mapped to unique Gaussian basis atoms that uniformly sample the time-frequency plane.

1. MOTIVATION

The goal of sequence alignment is to determine whether there are any sequences in the public databases that are similar to a given query sequence. In general, sequence alignment is the arrangement of primary sequences of DNA (deoxyribose nucleic acid), RNA (ribonucleic acid) or protein elements, in order to identify regions of similarity. By studying the similarity between a new gene sequence and sequences of known structure, one can infer the functionality of the new gene. An idealistic approach towards aligning two DNA sequences would be to search for an exact match within the two sequences. However, a sequence alignment tool must consider mutations due to cloning, sequencing errors, variations in the nucleotides, and insertions and deletions in the query sequence.

Dynamic algorithms such as the Smith-Waterman and Needleman-Wunsch methods, and heuristic computational approaches such as FASTA and BLAST [1], have been developed for sequence alignment. Signal processing algorithms have also been used based on cross-correlations of the sequences as a measure of similarity [2–6]. Often, the cross-correlation is obtained using the fast Fourier transform (FFT) to reduce the computational complexity. These approaches lack robustness to partial sequence mismatches and have an ambiguous misalignment reading when applied to periodic sequences [5]. Furthermore, these techniques do not perform well in finding local alignments.

We propose a DNA sequence alignment method using a time-frequency (TF) analysis technique based on the matching pursuit decomposition (MPD) algorithm. The MPD is a TF technique used in signal processing to decompose signals into a weighted linear combination of highly localized basis functions from a complete dictionary [7]. Although very popular in audio and video processing applications, the MPD has not been used before in sequence alignment. In our algorithm, query sequence and database sequence are mapped to Gaussian basis atoms based on the nucleotide composition of the sequences. We use the MPD frequency-shift parameter to represent the type of element in a sequence and the time-shift parameter to represent the position of the element in the sequence. Then, using the iterative MPD algorithm, we identify the highly-correlated regions of the query and database sequences as to be aligned. The similarity measure (number of identical nucleotides in the two sequences) is the sum of the coefficients obtained by the MPD algorithm. Repetitive patterns in the two sequences can be aligned using this method; this is not performed efficiently when BLAST is used.

This paper is organized as follows. In Section 2, we describe some alignment scenarios. In Section 3, we provide a general description of the MPD algorithm and the proposed sequence alignment algorithm. In Section 4, we analyze the performance of the proposed algorithm and compare it to BLAST.

2. SEQUENCE ALIGNMENT SCENARIOS

The essence of sequence alignment is to find regions of similarity between two or more sequences. If the similarity is captured over the entire length of the two sequences, it is called *global alignment*. If the similarity is captured over smaller local portions of the two sequences, it is called *local alignment*. In the alignment problem, a short query sequence $q(n)$ is to be aligned with a long database sequence $d(n)$.

This research was partially supported by the National Science Foundation (grants IIS 0431174, IIS 0551444, and IIS 0612273).

2.1. Complete Alignment

Complete alignment occurs when the query sequence is similar in its entirety, or up to a small number of mismatches, to a portion of the long database sequence. The aligned region can occur anywhere in the database sequence. Let the database sequence be composed of p subsequences, i.e., $d(t) = \{d_1(t), \ldots, d_p(t)\}$, where the i^{th} subset of characters in the sequence $d(n)$ is represented by the continuous time signal $d_i(t)$. Let each sub-sequence be composed of Q characters. Each of these sub-sequences is different by one character (maximum overlap) as shown in Figure 1. The mapped time distance between consecutive nucleotides is τ_s, such that the duration of $d_i(t)$ is $Q\tau_s$. The best match for $q(t)$ from the sub-sequence $d_i(t)$, where $i = 1, \ldots, p$, needs to be found.

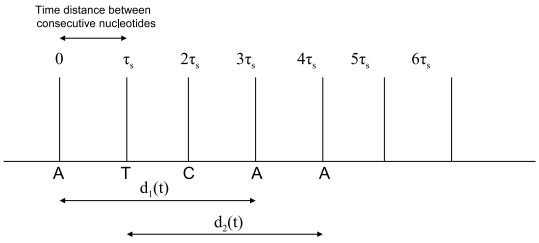

Figure 1. Sub-sequences $d_1(t)$ and $d_2(t)$ for the database sequence $d(t)$.

2.2. Local Alignment

In the local alignment case, portions of the query sequence (query sub-sequences) are aligned with the database subsequence. A best match is not found for the entire sequence, however portions of the sequence have matches in the database sequence at different positions. If the subsequences are of small length, a large number of alignments will occur. Thus, there is a need for a threshold which defines the minimum length of acceptable alignment. The threshold should allow for a small number of mismatches, if the best local alignments are to be obtained. Note that we do not take into account the gaps required for insertions and deletions.

3. MPD SEQUENCE ALIGNMENT ALGORITHM

3.1. Matching Pursuit Decomposition Algorithm

The MPD algorithm expands a signal $x(t)$ into a linear combination of basis functions called atoms, which are selected form a dictionary D. The atoms in the dictionary are defined as:

$$g_{n,k,l}(t) = g(\frac{t-\tau_n}{a_l})e^{-j2\pi f_k t}, \tag{1}$$

where τ_n is the n^{th} time shift, f_k is the k^{th} frequency shift and a_l is the lth scale change. The basic atom is a Gaussian signal $g(t) = e^{-\pi t^2}$, and the range of values of n, k, and l depend on how finely we sample the TF plane. The advantage of using the Gaussian atoms is that they are the most concentrated signals in both time and frequency [7].

The decomposed signal is given by

$$x(t) = \sum_{i=0}^{N-1} \alpha_i g_i(t) + r_N(t) \tag{2}$$

where N is the number of iterations, α_i are the expansion coefficients,

$$\alpha_i = \int_{\tau_s} r_i(t) g_i^*(t) dt, i = 0, \ldots, N-1 \tag{3}$$

and $r_i(t)$ denotes the residue function after the i^{th} iteration, with the initial residue taken as the signal itself. The i^{th} selected atom $g_i(t)$ is chosen as the atom that resulted in the maximum correlation between any dictionary atom and the i^{th} residue signal,

$$g_i(t) = \arg\max_{n,k,l} \int_{\tau_s} r_i(t) g^*_{n,k,l}(t) dt \tag{4}$$

The MPD is an iterative process that yields a sparse decomposition; if the signal is matched to the basis functions, the MPD requires only the first few atoms to obtain a good approximation of the signal [7]. The procedure steps are summarized as follows:

1. Initialize the residue vector: $r_0(t) = x(t)$
2. (a) For iterations $i = 0, \ldots, N-1$, compute the correlation (inner product) between $r_i(t)$ and every atom $g_{n,k,l}(t)$ in the dictionary D:

   $$\forall g \in D : \Lambda_{n,k,l} = |\langle r_i, g_{n,k,l} \rangle|$$

   where $\langle r_i, g_{n,k,l} \rangle = \int_{\tau_s} r_i(t) g^*_{n,k,l}(t) dt$

   (b) Search for the atom that resulted in the highest correlation value:

   $$g_i(t) = \arg \max_{g(t) \in D} \Lambda_{n,k,l}$$

   (c) Subtract the weighted atom from the residue:

   $$r_{i+1}(t) = r_i(t) - \alpha_i g_i(t)$$

   where α_i is computed as in (3).
3. The iterations are terminated when the desired level of accuracy is reached in terms of the extracted number of atoms or in terms of the energy ratio between the original signal and the current residue $r_i(t)$.

3.2. MPD Sequence Alignment

Consider, sub-sequences for both the database sequence $d(t)$ and the query sequence $q(t)$:

$$\begin{aligned} d(t) &= \{d_1(t), \ldots, d_p(t)\} \\ q(t) &= \{q_1(t), \ldots, q_r(t)\}, r \leq p. \end{aligned} \tag{5}$$

Our aim is to find a match for $q_j(t)$ in $d_i(t)$ ($i = 1, ..., p$ and $j = 1, ..., r$). If the length of the match exceeds the

threshold, then $d_i(t)$ and $q_j(t)$ are considered as possible local alignment pairs. All combinations of $d_i(t)$ and $q_j(t)$ that satisfy the threshold are considered as cases of local alignment. Though the direct cross-correlation algorithm is fairly simple, the intensity of computations and the number of variables used to store the positions and similarity measures for each alignment makes the method an unsuitable candidate for this alignment case. Also, computational methods such as BLAST compromise on the quality of alignments to maintain time-efficiency. We propose here to use the MPD algorithm, which provides an additional mapping parameter to characterize the position of the element in a sequence.

3.3. Sequence Alignment Algorithm

From the original MPD algorithm, we use TF-shifted Gaussian atoms as the dictionary elements. For our alignment purposes, we do not need the scaling transformations in (1); note that the dictionary maintains its completeness [7]. Thus, the dictionary elements become

$$g_{n,k}(t) = g(t - n\tau_s)e^{j2\pi kFt} \tag{6}$$

where k = 1, 2, 3, 4, F corresponds to the mapped frequency separation, and τ_s corresponds to the mapped time distance between the nucleotides. The frequency shifts in the MPD dictionary represent the four nucleotides. In particular, the frequency shift k = 1 represents the character A, k = 2 represents C, k = 3 represents G, and k = 4 represents T. The time shift parameter n of the dictionary element is used to represent the position of a nucleotide in a sequence. For example, n = 3 indicates that the position of the third element in the sub-sequence is position three. This is demonstrated in Figure 2. For example, the sequence $\{ATCA\}$ can be represented in terms of Gaussian atoms as $\{g_{1,1}(t), g_{2,4}(t), g_{3,2}(t), g_{4,1}(t)\}$.

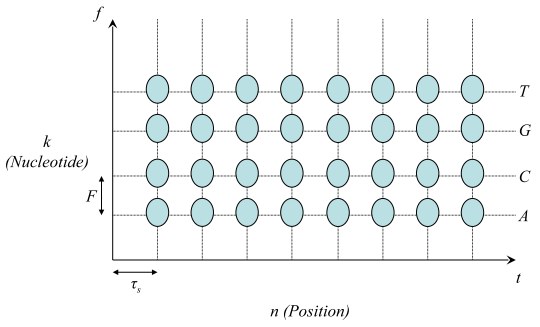

Figure 2. Gaussian signals for the DNA nucleotides in the TF-plane based on their position in a sequence.

3.3.1. MPD algorithm for complete alignment

For complete alignment, we create a dictionary of Gaussian atoms $g_{n,k}(t)$, k = 1, 2, 3, 4 and n = 1, ..., Q, where k is the frequency shift parameter, n is the time shift parameter, and Q is the length of the query sequence. We consider the database signal $d(t) = \{d_1(t), \ldots, d_p(t)\}$ and the query signal $q(t) = \sum_{n=1}^{Q} g_{n,k}(t)$. The MPD complete alignment algorithm is outlined in Table 1.

Table 1: MPD Complete Alignment Algorithm

for $i = 1$ to p
  let $r_0(t) = d_i(t)$
    {*Initialize the residue vector*}
  $\xi_0^i = 0$
    {*Initialize the variable to store correlation value*}
  for $l = 0$ to $Q - 1$
    $\Lambda_{n,k} = \langle r_l, g_{n,k} \rangle = \int_{\tau_s(n-1))}^{\tau_s(n)} r_l(t) g_{n,k}^*(t) dt$
      {*Compute the inner product between the residue and all elements in the dictionary*}
    $g_l(t) = \arg \max_{g_{n,k}(t) \in D} \Lambda_{n,k}$
    {*Search for the atom yielding the highest correlation*}
    $r_{l+1}(t) = r_l(t) - \alpha_l g_l(t)$
      {*Subtract the weighted atom from the residue*}
    $\xi_l^i = \xi_l^i + \alpha_l$
      {*Update the correlation value*}
  end for
end for
$\hat{d}(t) = \arg \max_{i=1:p} \xi_{Q-1}^i$

3.3.2. MPD algorithm for local alignment

The use of the MPD algorithm for the local alignment is based on the following steps. The sub-sequence $q_j(t)$ of the query sequence (whose minimum length is specified by the user) is mapped using Gaussian atoms. The dictionary is formed by all Gaussian atoms needed to map $q(t)$ in (5). The length of the dictionary is extended as the length of q_j increases. An increment of 1 provides the best alignment, however it makes the algorithm computationally expensive. Based on the accuracy required, the value of the increment is defined. The extension of the dictionary is continued until the best possible alignment is obtained, i.e., $\hat{d}_j(t) = \arg \max_i \xi_{Q_j-1}^i$ and the minimum threshold condition is satisfied. Other alignments with lower similarity scores are also considered and stored for future use. This is the local alignment performed on a portion of the query data.

The unaligned portion of the query signal is taken as the new query signal. The above steps of alignment are repeated to find the best alignment of the query signal with the database signal. This process is repeated until the end of the query sequence is reached. Once the entire query signal is aligned with the data signal, the aligned sequences are stored in order of the similarity scores, with information about the position of the aligned portions in the query and database sequences.

4. RESULTS

A hundred nucleotide sequences from *S. Cerevisiae* and *E. Coli* form the sequence database for the evaluation of the local alignment schemes. Fifty cases (χ= 1, ..., 50) of query sequences were taken and the database sequences were aligned with the query sequences. The sequences were aligned using the *bl2seq* algorithm of BLAST and the proposed MPD based alignment technique. In order to

compare the two algorithms, the following scoring scheme is considered. For an alignment of length L, $1/L$ is the reward for the correct alignment (as in BLAST), $-2/L$ is the penalty for the wrong alignment, and $-1/L$ is the penalty for not capturing an alignment that is captured by BLAST. The score is obtained by:

$$\text{Score} = (\tfrac{1}{L})M_1 - (\tfrac{2}{L})M_2 - (\tfrac{1}{L})M_3,$$

where M_1 is the count of nucleotides captured correctly in the alignment, M_2 is the count of nucleotides captured incorrectly in the alignment, and M_3 is the count of nucleotides not captured in the alignment. This score is considered for every alignment produced by BLAST. If the MPD-based alignment algorithm captured a major alignment (length of alignment greater than 50) that was not captured by BLAST, the score was incremented by one. The corresponding score results are shown in Figure 3, a zoomed plot of the scores for χ = 21, . . ., 30 is shown in Figure 4.

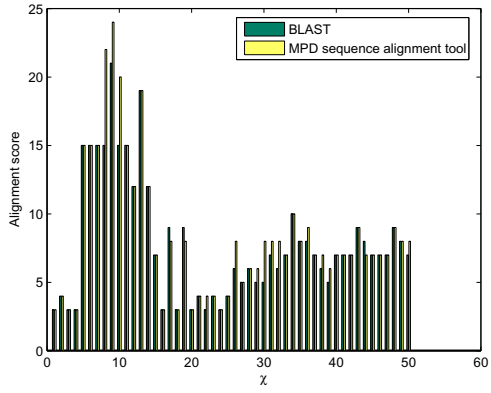

Figure 3. Scores of the MPD sequence alignment as compared to the scores of BLAST *bl2seq*; χ is the number of the query sequence tested.

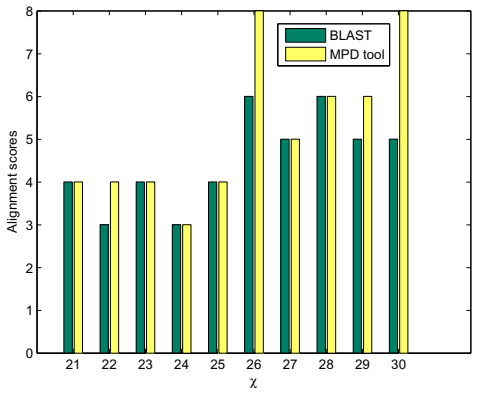

Figure 4. Zoomed version of Figure 3 for χ = 21,. . .,30

Note that the algorithm is successful in capturing the local alignments that are captured by BLAST in most cases and in a few cases, it captures additional alignments that are not captured by BLAST. Hence, the MPD-based sequence alignment algorithm shows an improved performance in capturing alignments. The improvement is due to the fact that BLAST does not consider periodic or repetitive patterns by default. Hence, if sequences with repetitive segments are to be aligned, the MPD-based alignment algorithm yields higher alignment performance than BLAST. The drawbacks of the time and frequency correlation alignment algorithms have also been overcome using this algorithm. In particular, less memory is needed by the MPD-based algorithm as the position of an element in a sequence is inherently stored by the Gaussian mapping.

5. CONCLUSION

In this paper, we proposed a sequence alignment algorithm based on mapping in the TF plane and using the MPD algorithm. The alignment was performed after mapping the sequences to Gaussian atoms based on the nucleotide type and position. The alignment algorithm was compared to BLAST, and an improvement in performance was observed for sequences with repetitive alignments. Note that further comparisons are needed with a higher number of query sequences as well as a different performance metric that takes into consideration the repetitive patterns.

6. REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool," *J. of Molecular Biology*, vol. 215, pp. 403–410, 1990.

[2] J. Felsenstein, S. Sawyer, and R. Kochin, "An efficient method for matching nucleic acid sequences," *Nucleic Acids Research*, vol. 19, pp. 133–139, 1982.

[3] E. Cheever, D. Searls, W. Karunaratne, and G. Overton, "Using signal processing techniques for DNA sequence comparison," in *Northeast Bioengineering Conference*, 27-28 March 1989, pp. 173–174.

[4] S. Rajasekaran, H. Nick, P. Pardalos, S. Sahni, and G. Shaw, "Efficient algorithms for local alignment search," *Journal of Combinatorial Optimization*, vol. 5, pp. 117–124, 2001.

[5] A. K. Brodzik, "A comparative study of cross-correlation methods for alignment of DNA sequences containing repetitive patterns," in *13th European Signal Processing Conference*, 2005.

[6] A. L. Rockwood, D. K. Crockett, J. R. Oliphant, and K. S. Elenitoba-Johnson, "Sequence alignment by cross-correlation," *Journal of Biomolecular Techniques*, vol. 16, pp. 453–458, 2005.

[7] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.