DESIGNING A TIGHTER SEARCHING SPACE FOR PAIRWISE GLOBAL SEQUENCE ALIGNMENTS OVER MULTIPLE SCORING SYSTEMS

Changjin Hong and Ahmed H. Tewfik

Department of Electrical and Computer Engineering, University of Minnesota
200 Union St. SE, Minneapolis, MN 55455, USA
{hongcj92,tewfik}@ece.umn.edu

ABSTRACT

The need to repeat alignments on a pair of amino acid residue sequences to select an appropriate scoring function and detect a significance of score arises in genomics and proteomics. While computing the alignments obtained through a set of typical scoring matrices with the corresponding default gap cost, we observe that many aligned segments are shared with a reference global alignment. We show that parameters extracted from the search for an alignment corresponding to a good scoring system can be used to predict the deviation of alignments computed with respect to different scoring matrices. By training sample pairs of protein sequences from SCOP 1.71 *of the ASTRAL database, we build the approximated probability distribution of distance from a node on a reference path to the alignments based on other scoring schemes with respect to the proposed parameters. We show that the overall computational cost to perform alignments using 'three' scoring matrices and the proposed method can be reduced to* 11% *of a normal Needleman-Wunch global alignment with an average* 92% *accuracy.*

KEYWORDS: dynamic programming, heuristic searching space, global sequence alignment, scoring matrices

1. INTRODUCTION

When investigating an unknown biological sequence from a living organism looking for its origin, we usually seek which previously identified sequences are similar. In general, we perform a matching procedure based on Dynamic Programming (DP) on the sequence of interest against known sequences in a Database. The naive DP approach is the Needleman-Wunch (NW)procedure, a global sequence alignment where the entire sequence is involved in the alignment [1]. The problem of calculating an edit distance between two sequences is a simple version of NW. The corresponding sequence similarity could serve as evidence of structural and functional conservation, as well as of evolutionary link [2]. In particular, the outcome of the aligning procedure is highly affected by the scoring scheme we selected. There are several scoring matrices such as BLOSUM, PAM, and Gonnet [3]. Suppose a scoring matrix you select yields a certain alignment score for the pair of sequence. However, what would you say about this score if, with the same scoring matrix, any randomly selected two sequences return a score as large as the score? This problem is called the statistical significance of sequence alignment score. Once we model a probability distribution of an alignment score assessed by its P-value [4], we can define a cutoff score such that we can safely claim that any higher score can be observed among truly highly closed sequences by chance. The NW procedure must be updated in order to find an application-dependent good scoring scheme that yields a meaningful inference in the sense of biological similarity. This is computationally very expensive.

High performance in computational biology has been addressed in depth. However, no prior work has highlighted the relationship among result of alignments with respect to the multiple scoring matrices and how different alignments relate to each other. For example, a heuristic DP [7, 8, 9] has a limitation because of high computational overhead with on-line prediction of a threshold for promising entries. Gap parametric alignment [11] is only defined for the dependency of a gap model. Both approximate alignments [10] and suboptimal alignments [12] focus on providing a reliability of an optimal alignment in $O(N^3)$ operations (N is a sequence length), which is infeasible. Sparse DP [13] is not relaxed in relatively remote homologue family of sequences.

We list out a notation used in this work to keep our discussion succinct in the following table.

SYMBOL	MEANING
(M,N)	(length(s_1),length(s_2))
S_k	scoring matrix k
g_O	open gap cost in an affine gap model
g_E	gap extension cost in an affine gap model
q_k	alignment by a reference S_k
$q_{k'}$	alignments by $S_{k'}$ other than S_k
Υ	searching space for $\{q_{k'}\}$ into DP mat.
η	percentage identity
λ	diagonal line from $(0,0)$ to (M,N) in DP mat.
u	index of q_k's path state
α	distance from $q_k(\tau)$ to λ
β	max. distance from $q_k(\tau)$ to $q_{k'}(\tau)$
ν	searching band offset at $q_k(u)$
ϕ, ψ	(conditional) probability distribution function

We select a reference scoring matrix and perform an alignment. From the reference alignment, we infer which parameter may control the similarity among those alignments when we repeat the same job over multiple scoring matrices. This inference is accomplished through a statistical model from training a data set of pairs of alignments. Having parameters suggested by the statistical distribution, we design a novel tighter searching space around the reference path in a DP matrix. In section 2, we briefly review DP for sequence alignment and then we define our problem. In section 3, our novel work is detailed in off-line and on-line computational step of alignments. Then, the performance of our proposed algorithm is validated through experimental results in section 4. In section 5, we discuss a future work and limitation.

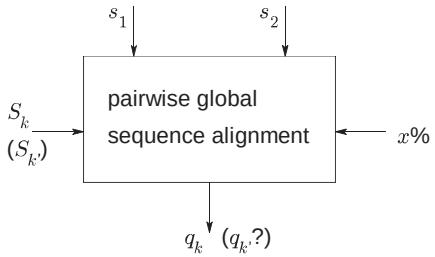

Figure 1: Dynamic Programming with updated input configurations. Is $q_{k'}$ is same as q_k? If not, should we repeat Eq.1?

2. PROBLEM DEFINITION

Given two sequences $s_1[1\cdots M]$and $s_2[1\cdots N]$ where M and N corresponds to the length of each sequence and $s_x[i]$ is one of the 20 amino acid residues (i.e., A, R, N,...,Y, V), a global sequence alignment is defined as transforming one into the other such that the similarity score is maximized with respect to two scoring rules. The first is a scoring matrix that is defined as a similarity between two residues in the evolutionary distance (i.e., BLOSUM50, BLOSUM62, PAM120, and PAM240). The second is a gap model, helping create alignments that better conform to underlying biological models as the model must specify how to weight gaps depending on either single or consecutive mutations so as to reflect their biological meanings [9]. In our problem, we use an affine gap model [15]. Let g_O, g_E, and $S_k(s_1[i],s_2[j])$ be an opening gap, an extension gap, and a similarity score between a residue $s_1[i]$ and $s_2[j]$ from a scoring matrix S_k respectively. For a number of either single or consecutive gaps, j, the affine gap model is defined as $w_j = g_O + g_E \cdot j$. The DP solution guarantees an optimal alignment which can be represented into a sequence of states such as (mis)match, insertion, and deletion by the following equations [17],

$$d_{(i,j)} = \max\{d_{(i-1,j-1)} + S_k(s_1[i],s_2[j]), \max_{k\geq 1}\{d_{(i,j-k)} - w_k\} \\ , \max_{l\geq 1}\{d_{(i-1,j)} - w_l\}\} \quad (1)$$

with a zero boundary condition of $d_{(i,0)} = -g_O - i\cdot g_E$ and $d_{(0,j)} = -g_O - j\cdot g_E$. In Eq.1, note that $d_{(i,j)}$ represents the best score between two suffix $s_1[1\cdots i]$ and $s_2[1\cdots j]$.

Assuming that we obtain q_k, say, a reference optimal alignment of s_1 and s_2 with respect to S_k, we are asked to find an updated alignment for the same pair of sequences, given a different scoring matrix, say, $S_{k'}$. This procedure can be simplified as shown in Fig.1, when we look at s_1, s_2, S_k, and x% (alignment confidence level) as input parameters and both q_k and its similarity score as outputs. In analogy to the scenario of updated scoring matrix as an input, the authors investigated how the previous optimal alignment can be updated efficiently when we face with changes on multiple residues of sequences [16].

In general, all we can do is to repeat the recursion over all the searching space in a DP matrix for every S_k. In our problem, however, we are curious to know how different a new alignment, $q_{k'}$, would be from the previous alignment q_k and examine changes in the alignment score. Especially, our primary objective is to design a new DP process box in order to save a computational cost under the typical situation of long sequences[1] and multiple scoring matrices, instead of starting from scratch, while controlling the trade-off between the alignment quality and speed.

3. PROPOSED SEARCHING SPACE OVER MULTIPLE SCORING SCHEMES

It is frequently observed that $q_{k'}$s overlap with a large number of segments of q_k. The evidence can be easily recognized in Fig2. Not surprisingly, this observation is obvious because all the scoring schemes have a common property that large positive score are assigned to matched residues. In other words, scoring matrices have a high correlation with each other. As a consequence, the framework of our proposed approach is built on a heuristic DP method to estimate a searching bound after a reference alignment is established. In general, this approach, called a pruning DP, gets rid of unpromising nodes having sum of forward score with expected backward score lower than a certain threshold such that the node would unlikely be part of q_k [8]. While computing $q_{k'}$s, however, a reasonable bound approximated from backward score based on a geometry cost model cannot be found because the affine gap model leads inconsistency with the actual cost. Alternatively, when we consider using a backward intermediary distance from q_k instead of that of $q_{k'}$, we face the problem of not only carrying $O(NM)$ space memory but also having a difficulty in selecting a threshold value for a searching bound. Evaluating the updated sum distance to design a searching path is computational overhead.

Instead monitoring the updated forward distance, we simply focus on q_k in order to derive hints for a potential 'static' searching bound, Υ. Note that we want to emphasize the term, static, since the bound can be decided during the calculation of q_k.

3.1 Two Design Parameters (η,α_{max})

Assuming that we are allowed to perform a normal NW of Eq.1 based on S_k, then we infer which information that can be obtained from q_k would be closely related to the deviation of $\{q_{k'}\}$ from q_k. Again, this information should be computable from q_k to avoid an overhead before starting to compute any $q_{k'}$.

Here, we introduce two off-line parameters. First, we observe that the more matches show up in a pair alignment, the more likely $q_{k'}$ would traverse those highly conserved segments. Then, Υ probably shrinks into a narrow band around those segments. In order to describe this information, a percentage identity[2]is given to represent the degree of inherent similarity between two sequences [14]. Let's define this parameter as η. Second, if those highly matched segments are spread out, forming a large set of suboptimal alignments which is called as 'a twilight zone' in bioinformatics, then η alone would not be perfect to predict a shape for Υ. For example, let λ be a diagonal line connected from $(0,0)$ to (M,N) in the DP matrix and define α as a perpendicular distance from q_k to λ. Then, as q_k holds a larger α_{max}, we frequently observe that $q_{k'}$s either sit farther from λ or sweep a larger space around q_k.

[1]The median length of the protein annotated among Eukaryotes and Bacteria is 361 and 267 respectively

[2]In our work, it defines as (num. of matched residues)$\cdot$100/min$\{M,N\}$%

3.2 Phase 1:Off-line building a probability distribution model

Once we identify a parameter set (η, α_{max}), our objective is to design Υ along with q_k, based on probabilistic model of alignments from a group of pairs having the similar parameter set. Accordingly, it is required to build the quantified correlation between (η, α_{max}) and its corresponding Υ.

Let's consider a node (r,c) on q_k. Denote $\beta(u = r+c)$ as a maximum distance from the node on $q_{k'}$ to q_k. Parameter u is defined as a state index of q_k and satisfies the inequality, $1 \leq u = (r'+c') \leq U \leq (M+N)$. Here, U indicates the length of q_k. In Fig. 2, note that β^+ is obtained from an upper alignment and β^- from a lower one. The performance of computational cost will highly depend on how close the searching offset we can predict is to these β's. Let (v_u^+, v_u^-) be a searching band offset at u on q_k. Now, suppose we are given a conditional probability distribution, $\psi_{k'}(\beta|\alpha)$ for each S'_k. If v^+ is greater than β^+ *and* nu^- is smaller than β^- for each u, the Υ to be build by the chain of offsets would cover all alignment paths, $\{q_{k'}\}$, correctly. In Fig. 2, for example, an upper bound of Υ fails to include a part of q_{b50}. When we want more than $x\%$ confidence level, our objective is to find a Υ such that $\{(v_u^+, v_u^-)\}$ satisfies

$$\begin{aligned} &\min\{\Sigma_{u=1}^{U}|v_u^+ - v_u^-|; \\ &x\% \leq \prod_{u=1}^{U}(\phi_{k'}(\alpha)\int^{v_u^+}\psi_{k'}(\beta|\alpha)d\beta \\ &+(1-\phi_{k'}(\alpha))\int^{v_u^-}\psi_{k'}(\beta|\alpha)d\beta)\} \end{aligned} \tag{2}$$

Here, the $\phi_{k'}(\alpha)$ is the probability that any $q_{k'}$ is above q_k or not. When α is equal to 0, the probability is approximately 0.5. In reality, it is infeasible to model both $\psi_{k'}$ and $\phi_{k'}$ with respect to each continuous variable α since it is hard to get unbiased samples for each specific α from a training data set. This situation becomes even worse since we need repeat this for all scoring schemes. Furthermore, the exact distribution must exist foreach parameter domain of (η, α_{max}).

In order to overcome this difficulty in practice, three simplified approximations are incorporated into finding Υ. First, we fix the mostly suggested affine gap model to the corresponding scoring matrix in Table 1. Those parameters are recommended by Vingron and Waterman, Mount, and Pearson for a global alignment [6]. In particular, we employ BLOSUM62 as our reference scoring matrix in this work. Second, we make categories with the combination of (η, α_{max}), grouping continuous values into a set of grid points. For example, we choose five distinct groups for η from 18.5 to 70% and also four groups for α_{max} from 0 to 50% where α_{max} is computed as a ratio to an average length of two sequences. This configuration yields twenty $\psi(\beta|\alpha;(\eta,\alpha_{max}))$'s. Unfortunately, the distribution is unknown. Therefore, a kernel density estimator `ksdensity` from MATLAB® is used to fit each distribution. For the sake of space complexity, we construct distributions for only several confidence values (i.e., $x\%$ is .83, .88, .93, .98). Finally, instead of storing the ψ for each α in a (η, α_{max}), we only focus on three ψ's. The first is $\psi(\beta|\alpha = .5\alpha_{max})$ for generating smallest one out of either β^+ or β^- at $.5\alpha_{max}$ and the second is $\psi(\beta|\alpha = \alpha_{max})$ at α_{max}. The third ψ is a distribution for β_{max} which is the maximum β in all the u.

Table 1: Default affine gap cost of scoring matrices.

S_k \ cost	g_E	g_O
BLOSUM50	10	2
BLOSUM62	7	1
PAM120	16	4
PAM250	11	1

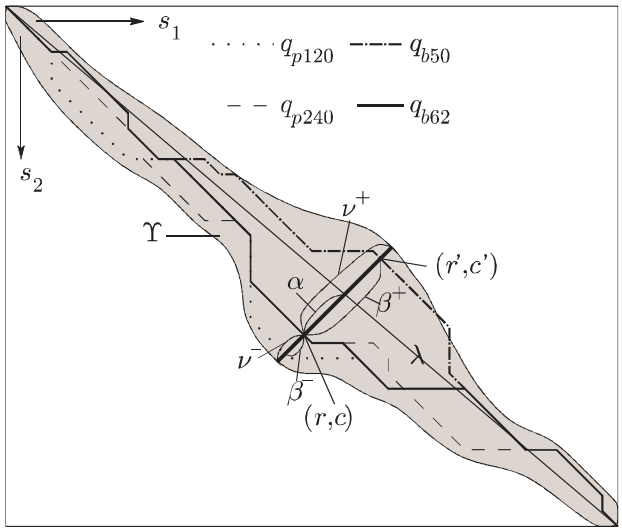

Figure 2: Overlapped alignments w.r.t. all different scoring matrices. Note that $q_{b62}^{1,2}$ is a reference alignment based on BLOSUM62. The primary objective is to find a potential searching space that would contain all $\{q_{k'}\}$within a tighter bound.

The last step in particular plays a crucial role in capturing a Υ for $q_{k'}$s for the on-line computation. The reason why those ψ's are chosen will become clear in the following discussion.

3.3 Phase 2: Proposed on-line searching space

In this section, we describe the method we use to estimate a Υ such that we can control the computational performance and the alignment quality. Once we identify (η, α_{max}) from q_k, we lookup the three $\psi(\beta|\alpha;\eta,\alpha_{max})$'s that correspond to the parameter in order to evaluate repeated alignments for $\{S_{k'}\}$. The ψ's suggest a Υ design parameter set, $(\beta_{max}, \beta(.5\alpha_{max}), \beta(\alpha_{max}))_x$, which satisfies a confidence level $x\%$ of interest. The easiest way to define a Υ is simply to offset q_k with β_{max} into both the upper right and lower left direction respectively, since we do not know which direction $q_{k'}$s would follow. This configuration of Υ is exactly to go back to the procedure used to derive ψ during the off-line training phase.

Since the uniform offset of β_{max} is applied to (v^-, v^+) without considering α, and $\psi(\beta|\alpha)$ varies with α, it turns out that there are lots of wasteful searching space. In most cases, we observe that the $q_{k'}$s encompass the space between q_k and λ as q_k diverges from λ. In other words, the probability that $q_{k'}$ goes by q_k, below it or above it highly depends upon α in this case. As a consequence, it is necessary to handle v^- and v^+ separately based on the following discussion. BLOSUM62 has smaller default gap penalty. Thus the scoring matrix itself can dominate the alignment path more

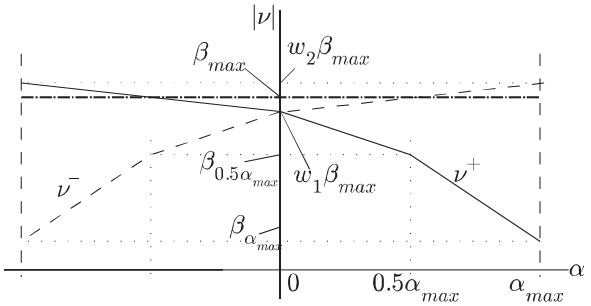

Figure 3: Two mapping functions for Υ. Method I uses an equivalent searching offset ($v_u^- = v_u^+ = \beta_{max}$) for $\forall u$. On the other hand, method II uses a varying offset (v_u^-, v_u^+) on u as a function of α calculated along with q_k.

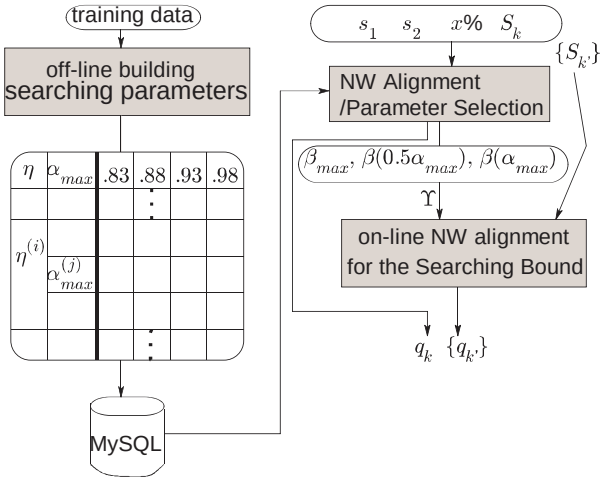

Figure 4: A flow graph from building a off-line resource from a training data to performing alignments with a test data.

than other effects, which means that q_k can go further from λ. This is the reason why this scoring scheme is selected in the sense that q_k is more informative. Also note that this is a global alignment which is highly affected by a geometry gap cost. Even If q_k stays at a distant u from λ showing conservation right there, $q_{k'}$ is also controlled by the $S_{k'}$'s gap cost. This observation is consistent with the fact that β is mostly retained from, a $q_{k'}$ that is within λ from q_k.

Taking advantage of these observations, we propose a simple technique to design a searching bar, (v_u^+, v_v^-), along with q_k at v. If q_k stays below λ, v^- has smaller offset while u^+ is designed based on β_{max} and vice versa. Thus, the former aims at saving computational cost and the latter is for higher accuracy. From a parameter set, we generate three linear functions for v. One is for governing larger v. For more refinement of accuracy, the bound toward λ from q_k is designed such that the v has the value between $w_1 \cdot \beta_{max}$ and $w_2 \cdot \beta_{max}$, where w_1 is deduced to be .8 and 1.1 for w_2 empirically. For example, as α is close to α_{max}, v is equal to $1.1\beta_{max}$. For shorter v, on the other hand, two separate linear functions are applied to save computational cost. Here, both $\psi(\beta|\alpha = .5\alpha_{max})$ and $\psi(\beta|\alpha = \alpha_{max})$ are used for this bound. This is based on the assumption of $\int \psi(\beta|\alpha_1) \leq \int \psi(\beta|\alpha_2)$, where $\alpha_1 \leq \alpha_2$. However, consider a smaller v which is derived by an linearized model of $\psi^{(y)}$ at α_y between $.5\alpha_{max}$ and α_{max}. At the same x% confidence level, this approximation is not always true, even if the assumption of monotonic function with respect to α greater than a specific value is still valid. However, finer grained linearized approximation leads to higher accuracy and more computational cost.

In Fig.3, we summarize how (v_u^+, v_u^-) can be calculated at each $u = (r, c)$ on q_k.

4. EXPERIMENTAL RESULTS

We follow a standard cross validation procedure. The flow chart for the overall algorithm is illustrated in Fig.4. In a training step, we sample pairs of protein sequences from the ASTRAL database to generate the corresponding probability distributions with respect to (η, α_{max}) [5]. We only collect samples having % identity from 18.5% to 70.0%. For testing data, the sequences are randomly sampled to be aligned with the remaining part of the database[3]. Since the global sequence alignment is mostly used to evaluate sequences among more closely related proteins, the samples of higher η are preferred. The average sequence length we worked with is varied from 70 to 1221. In our preliminary experiment, both alignment accuracy and computational cost are evaluated. A parameter set $(\beta_{max}, \beta(.5\alpha_{max}), \beta(\alpha_{max}))$ for designing Υ is normalized with respect to the average sequence length of two sequences. When we apply this parameter to a short sequence with a lower x%, its v is equal to 0. In order to avoid this case, threshold values for those parameters are assigned.

The varied offset design method (II) with α outperforms a simple half band searching offset method (I) with respect to three different confidence level from 88% to 98% in most η. When pairs of longer sequences are compared, the lower probability of successful Υ is generated. This method is shown in Eq.2. Sometimes, this is true for method I but not method II. This means that the accuracy of method II depends less on the length of sequence. Furthermore, in the lower accuracy experiment of Table 2, the overall accuracy of method II is dramatically improved. However, since less samples are collected in the category of less than 70% of η, which yields a biased distribution model, method II exhibits lower accuracy performance. Table 3 shows the comparison for the computational cost. Both method I and II for higher η consume significantly less searching space. Furthermore, this performance holds with longer sequences. As we expect, higher accuracy consumes more space. Method II saves even more searching space than method I while increasing the overall correctness of alignment.

In three different x% experiments, the proposed Υ of method II is reduced to an average 32.9% with overall 92.3% correct alignments for three scoring schemes, compared to a normal NW algorithm. The computational cost is calculated as a ratio of Υ to a whole DP matrix space. All entries on a correct alignment, $q_{k'}$, should be contained within Υ. Therefore, the correctness of Υ is defined as a ratio of a number of $S_{k'}$ satisfying the requirement to a total number of $S_{k'}$. Note that both methods complete all matching proce-

[3] A global alignment for randomly-selected pairs yields 20%. In order to obtain an uniform a number of samples over η, we take a different number of samples into hierarchical protein family clusters

Table 2: Correctness of proposed heuristic approaches. Refer to Fig.3 for definition of the method I and II.

η \ x%	.88		.93		.98		length
	I†	II‡	I	II	I	II	
18.5∼23.5	.821	.887	.861	.914	.910	.959	∼114
	.834	.930	.864	.939	.930	.947	∼164
	.794	.943	.830	.945	.900	.967	∼576
∼25.6	.738	.883	.814	.913	.889	.922	∼129
	.784	.930	.841	.931	.920	.937	∼237
	.843	.968	.890	.970	.935	.972	∼1221
∼27.7	.754	.898	.792	.910	.892	.928	∼148
	.800	.898	.864	.916	.955	.940	∼249
	.810	.926	.857	.927	.947	.947	∼1177
∼30.7	.799	.849	.843	.899	.937	.923	∼140
	.802	.873	.859	.906	.906	.944	∼246
	.785	.899	.865	.901	.933	.940	∼713
∼70.0	.883	.846	.913	.896	.946	.913	∼97
	.862	.924	.893	.938	.946	.941	∼187
	.887	.923	.931	.936	.957	.939	∼508

Table 3: Computational cost of proposed methods.

η \ x%	.88		.93		.98		length
	I†	II‡	I	II	I	II	
18.5∼23.5	.320	.316	.350	.347	.402	.403	∼114
	.303	.297	.332	.329	.385	.385	∼164
	.278	.276	.304	.307	.355	.360	∼576
∼25.6	.316	.304	.351	.339	.405	.393	∼129
	.303	.296	.337	.331	.393	.383	∼237
	.272	.284	.303	.318	.363	.369	∼1221
∼27.7	.323	.318	.354	.354	.434	.428	∼148
	.301	.301	.330	.333	.411	.407	∼249
	.272	.274	.303	.307	.386	.379	∼1177
∼30.7	.325	.317	.357	.352	.434	.422	∼140
	.317	.308	.350	.341	.425	.410	∼246
	.296	.289	.330	.322	.395	.384	∼713
∼70.0	.316	.306	.356	.343	.437	.418	∼97
	.228	.223	.259	.251	.330	.316	∼187
	.193	.192	.222	.221	.289	.280	∼508

dures over *three* $S_{k'}$'s at once. As a consequence, the overall searching space based on method II is dramatically reduced to $32.9/3 \approx 11\%$. As more scoring matrices are required to seek appropriate alignment parameters, this advantage can be significantly higher.

5. DISCUSSION

This work can contribute to assist in finding an appropriate scoring matrix efficiently when a global pairwise sequence alignments are needed usually at a high computational cost. By focusing on the reference alignment and building a probability density function to provide a correlation factor to design a tighter pruned searching space along with the the reference path, the proposed alignment has the advantage of avoiding on-line evaluation time of typical heuristic approach and is designed to be governed by highly controllable design parameters that provide a trade-off between an alignment accuracy and computational cost.

There will be a several extensions of our work. Our ψ possibly can include gap models as other input variables, since we use only fixed default gap costs. While we shift α_y between 0 and α_{max} rather than $.5\alpha_{max}$, it would be more interesting to find a position for ψ_{α_y} such that it can minimize an error of the linearized approximation for the other ψ. The proposed work has a computational overhead to calculate $\alpha(u)$ for each u of $O(M+N)$ operations. However, this overhead can be diminished with an additional parallel computation node that takes the dedicated computation on it since this does not affect on-line alignment.

6. ACKNOWLEDGEMENT

We thank Huzefa Rangwala and Aravindan Raghuveer for many suggestion and stimulating discussion.

REFERENCES

[1] S. B. Needleman et al., "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *J. Mol. Biol.* 48:443-53, 1970

[2] Y. Alexander et al., "Statistical Significance in Biological Sequence Analysis," *Briefings in Bioinformatics, doi:10.1093/bib/bbk001*, 2006

[3] D. W. Mount, "Bioinformatics: sequence and genome analysis," *Cold spring harbor laboratory press*, 2001.

[4] S. F. Altschul and B.W. Erickson, "Significance of Nucleotide Sequence Alignments," *Mol. Biol. Evol.*, 2:526-538, 1985

[5] J-M. Chandonia, G. Hon, L. L. Conte, N. Walker, P. Koehl, M. Levitt, and S. E. Brenner, "The ASTRAL compendium for sequence and structure analysis," *Nucleic Acids Research*, 28:254-56, 2000

[6] H. Pang, J. Tang, S-S Chen, and S Tao, "Statistical distributions of optimal global alignment scores of random protein sequences," *BMC Bioinformatics*, 6:257, 2005

[7] S. Altschul, W. Gish, W. Miller, E.W. Myers, and D. Lipman, "Basic Local Alignment Search Tool," *J. Mol. Biol.*, 215:403-10, 1990

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on SSC4,* 2:100-7, 1968

[9] D. Gusfield "Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology," *Cambridge University Press,* 1997

[10] T. Kahveci, V. Ramaswamy, H Tao, and T. Li, "Approximate global alignment of sequences," *Bioinformatics and Bioengineering, BIBE Fifth IEEE Symposium*, 81-8, 2005

[11] D. Gusfield, K. Balasubramanian, and D. Naor, "Parametric Optimization of Sequence Alginment," *Algorithmika*, 12:312-26, 1994

[12] M. Michael, C. Dieterich, and J. Stoye, "Suboptimal Local Alignments across Multiple Scoring Schemes (preview)", *Proceedings of WABI 2004*, LNBI 3240:99-110, 2004

[13] D. Eppstein, Z. Galil, R. Giancarlo, and G.F. Italiano., "Sparse dynamic programming," *1st ACM-SIAM Symp. Discrete Algorithms*, San Francisco, 513-22, 1990

[14] GPS Raghava and Geoffrey J Barton, "Quantification of the variation in percentage identity for protein sequence alignments," *BMC Bioinformatics*, 7:415, 2006

[15] S. F. Altschul and B. W. Erickson, "Optimal sequence alignment using affine gap costs," *Bull. Math. Biol*, 48:603-16, 1986

[16] C. Hong and A. H. Tewfik, " Handling Updates of a Pairwise Sequence Alignment," *ICASSP 2006 Proceedings*, 2:1104-06, 2006

[17] O. Gotoh, "An improved algorithm for matching biological sequences," *J. Mol. Biol.*, 162:705-08, 1982