International Journal of Foundations of Computer Science
Vol. 21, No. 1 (2010) 51–59

DOI: 10.1142/S012905411000712X

AN ALGORITHM AND APPLICATIONS TO SEQUENCE ALIGNMENT WITH WEIGHTED CONSTRAINTS

YUNG-HSING PENG, CHANG-BIAU YANG*, KUO-TSUNG TSENG
and KUO-SI HUANG

Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
**cbyang@cse.nsysu.edu.tw*

Received 31 October 2008
Accepted 16 June 2009
Communicated by Omer Egecioglu

Given two sequences S_1, S_2, and a constrained sequence C, a longest common subsequence of S_1, S_2 with restriction to C is called a *constrained longest common subsequence* of S_1 and S_2 with C. At the same time, an optimal alignment of S_1, S_2 with restriction to C is called a *constrained pairwise sequence alignment* of S_1 and S_2 with C. Previous algorithms have shown that the constrained longest common subsequence problem is a special case of the constrained pairwise sequence alignment problem, and that both of them can be solved in $O(rnm)$ time, where r, n, and m represent the lengths of C, S_1, and S_2, respectively. In this paper, we extend the definition of constrained pairwise sequence alignment to a more flexible version, called *weighted constrained pairwise sequence alignment*, in which some constraints might be ignored. We first give an $O(rnm)$-time algorithm for solving the weighted constrained pairwise sequence alignment problem, then show that our extension can be adopted to solve some constraint-related problems that cannot be solved by previous algorithms for the constrained longest common subsequence problem or the constrained pairwise sequence alignment problem. Therefore, in contrast to previous results, our extension is a new and suitable model for sequence analysis.

Keywords: Algorithm; longest common subsequence; sequence alignment; weighted constraint.

1991 Mathematics Subject Classification: 22E46, 53C35, 57S20

1. Introduction

Given a sequence S, a subsequence $\bar{S}$ of S can be obtained by deleting zero or more characters from S. Given two sequences S_1 and S_2, a longest common subsequence (LCS) of S_1 and S_2, is a longest sequence $\bar{S}'$ such that $\bar{S}'$ is a subsequence of both S_1 and S_2. Finding an LCS of two or more sequences is a well-known problem that has been widely studied for several decades [1, 7, 11, 21], because of its great influence on sequence analysis.

*Corresponding author.

Aside from the traditional LCS problem, some variants, such as the mosaic LCS problem [9, 12] or the merged LCS problem [8], have been proposed to fit different applications. Among these variants, the constrained longest common subsequence (CLCS) problem has drawn much attention [3, 4, 6, 11, 14, 18, 22]. The CLCS problem was delivered by Tsai [18], who also proposed an $O(rn^2m^2)$ algorithm for solving this problem. As a tool for measuring the similarity of two sequences, the CLCS problem is more flexible than the traditional LCS problem [10, 21], because the output sequence of the former problem contains a user-defined constrained sequence. Therefore, CLCS is then extended to the constrained pairwise sequence alignment (CPSA) [17], which is a new concept that allows biologists to define some single residues or nucleotides as constraints that must be covered in the output alignment. Afterwards, four groups of researchers proposed improved algorithms independently [3, 4, 14, 22]. In their articles, an $O(rnm)$-time algorithm was proposed for solving the CPSA problem, which greatly improves Tsai and Tang's results [17, 18]. Recently, Gotthilf *et al.* [6] showed that the CLCS problem with multiple constrained sequences is an NP-hard problem that cannot be approximated. In addition, they gave an approximate algorithm with linear time for another variant, the CLCS problem with multiple input sequences but only one constrained sequence. From Gotthilf's result, one can see that the constrained multiple sequence alignment (CMSA) problem [13, 17] is also NP-hard.

However, one should note that most of the previous results [4, 6, 13, 14, 17, 18, 22] force the output sequence (or alignment) to cover all constraints. This may be too rigorous for applications in which some constraints might not exist in the output sequence. An obvious example is to compute the similarity of two articles with regard to some given keywords that do not necessarily exist in both articles. As a result, Arslan and Eğecioğlu first extended CLCS to another flexible version [3], which we call the *distance constrained longest common subsequence* (DCLCS) problem, where the number of ignored constraints is allowed to a degree d. For example, given two sequences S_1="cccccggaga", S_2="aggaaccccc" and C="ag", we have "aga" as their CLCS. By setting $d = 1$, we have "ggaa" as the CLCS of S_1 and S_2, which can be verified by individually ignoring the constraint 'a' and 'g'. Furthermore, by setting $d = 2$, one can see that the CLCS would be "ccccc", which happens to be the longest common subsequence of S_1 and S_2. For the DCLCS problem, Arslan and Eğecioğlu gave an $O(drnm)$-time and $O(drm)$-space algorithm [3].

Motivated by Arslan and Eğecioğlu's result, in this paper we propose another extension for CPSA, called *weighted CPSA* (WCPSA), where some of the given constraints might be ignored. Different from the previous version [15], in this paper we provide a general definition for WCPSA, and give a complete proof for our algorithm. Besides, an additional application of WCPSA is provided to give a fair comparison with Arslan and Eğecioğlu's result.

The rest of this paper is organized as follows. In Section 2, we give some annotations to WCPSA, and show that WCPSA can be solved with $O(rnm)$ time and $O(rn)$ space. In Section 3, we give some constraint-related problems that can be solved by WCPSA, showing the flexibility of WCPSA over DCLCS. Finally, Section 4 concludes our results.

2. Sequence Alignment with Weighted Constraints

Suppose we are given three sequences S_1, S_2, and C, where $|S_1| = n$, $|S_2| = m$, and $|C| = r$ represent the lengths of S_1, S_2, and C, respectively. For any given sequence S, let $S[i]$ denote the ith character in S. Also, let $S[i,j]$ denote the substring of S that starts with the ith character and ends with the jth character. Let $\sigma[k]$ and $\delta[k]$ denote the gain and penalty (weights) when $C[k]$ is included and excluded in the final output, respectively, for $1 \leq k \leq r$. Let $f(S_1[i], S_2[j])$ be the score for aligning $S_1[i]$ with $S_2[j]$, $f(S_1[i], `-')$ be the gap penalty for aligning $S_1[i]$ with a gap, $f(`-', S_2[j])$ be the gap penalty for aligning $S_2[j]$ with a gap, for $1 \leq i \leq n$ and $1 \leq j \leq m$. In general, $\delta[k]$, $f(S_1[i], `-')$ and $f(`-', S_2[j])$ should be of negative values, since they represent penalty scores. The *weighted constrained pairwise sequence alignment* (WCPSA) problem is to find the constrained alignment with the maximum score based on the above scoring scheme. For ease of comparison, we rewrite the definitions of the CLCS, DCLCS, CPSA and WCPSA problems in Definitions 1, 2, 3, and 4, respectively.

Definition 1. [18] ***The CLCS (constrained LCS) problem:*** *Given S_1, S_2, and C, find the longest sequence S' that contains C as a subsequence, where S' is a common subsequence of S_1 and S_2.*

Definition 2. [3] ***The DCLCS (distance constrained LCS) problem:*** *Given S_1, S_2, C, and a nonnegative integer d, find the longest sequence S' that contains a subsequence C', where S' is a common subsequence of S_1 and S_2, and C' is a subsequence in C with $|C'| \geq (r-d)$.*

Definition 3. [4] ***The CPSA (constrained pairwise sequence alignment) problem:*** *Given S_1, S_2, C, $f(S_1[i], S_2[j])$, $f(S_1[i], `-')$, and $f(`-', S_2[j])$, $1 \leq i \leq n$ and $1 \leq j \leq m$, among all alignments of S_1 and S_2 that contain C as a matched subsequence, find an alignment that has the maximum alignment score.*

Definition 4. [This paper] ***The WCPSA (weighted CPSA) problem:*** *Given S_1, S_2, C, $f(S_1[i], S_2[j])$, $f(S_1[i], `-')$, $f(`-', S_2[j])$, $\sigma[k]$, and $\delta[k]$, $1 \leq i \leq n$, $1 \leq j \leq m$ and $1 \leq k \leq r$, find an alignment of S_1 and S_2 that has the maximum alignment score under this weighted scoring scheme.*

By extending previous formula of CPSA [4], we can obtain an optimal WCPSA. Let $R(k,i,j)$ denote the score for an optimal WCPSA of $S_1[1,i]$, $S_2[1,j]$, and $C[0,k]$, where $1 \leq i \leq n$, $1 \leq j \leq m$, and $0 \leq k \leq r$. Note that we treat $C[0]$ as an empty character. Our algorithm for WCPSA is given as follows.

$$R(k,i,j) = max \begin{cases} R(k,i-1,j) + f(S_1[i], `-\text{'}) \\ R(k,i,j-1) + f(`-\text{'}, S_2[j]) \\ R(k,i-1,j-1) + f(S_1[i], S_2[j]) \\ R(k-1,i-1,j-1) + f(S_1[i], S_2[j]) + \sigma[k] \\ \qquad\qquad\qquad\qquad \text{if} \;\; S_1[i] = S_2[j] = C[k] \\ R(k-1,i,j) + \delta[k], \end{cases}$$

with boundary conditions
$R(0,0,0) = 0,$
$R(0,0,j) = \sum_{t=1}^{j} f(`-\text{'}, S_2[t]),$ for $1 \le j \le m$
$R(0,i,0) = \sum_{t=1}^{i} f(S_1[t], `-\text{'}),$ for $1 \le i \le n$
$R(k,0,0) = \sum_{t=1}^{k} \delta[t],$ for $1 \le k \le r$
$R(k,0,j) = \sum_{t=1}^{k} \delta[t] + \sum_{t=1}^{j} f(`-\text{'}, S_2[t]),$ for $1 \le k \le r,\; 1 \le j \le m$
$R(k,i,0) = \sum_{t=1}^{k} \delta[t] + \sum_{t=1}^{i} f(S_1[t], `-\text{'}),$ for $1 \le k \le r,\; 1 \le i \le n$
$R(k,i,j) = -\infty,$ for any $k < 0$ or $i < 0$ or $j < 0$.

Lemma 5. *In the above formula, R(k,i,j) computes the score for an optimal WCPSA of $S_1[1,i]$, $S_2[1,j]$ and $C[0,k]$.*

Proof. The proof can be briefly done by discussing whether $C[k]$ is ignored. If $C[k]$ is ignored, then the optimal solution relies on $R(k-1,i,j)$. Therefore, considering the penalty, we have $R(k,i,j) = R(k-1,i,j) + \delta[k]$, which is the last case in our formula. Except for the last case, one can see that the remaining cases discuss all situations where $C[k]$ is not ignored, which are similar to the formula of CPSA [4]. The first case $R(k,i-1,j) + f(S_1[i], `-\text{'})$ aligns $S_1[i]$ with a gap, and forces $C[k]$ to be covered in some alignment of $S_1[1,i-1]$ and $S_2[1,j]$. The second case $R(k,i,j-1) + f(`-\text{'}, S_2[j])$ aligns $S_2[j]$ with a gap, and forces $C[k]$ to be covered by some alignment of $S_1[1,i]$ and $S_2[1,j-1]$. The third case $R(k,i-1,j-1) + f(S_1[i], S_2[j])$ aligns $S_1[i]$ with $S_2[j]$, and forces $C[k]$ to be covered in some alignment of $S_1[1,i-1]$ and $S_2[1,j-1]$. Finally, the fourth case $R(k-1,i-1,j-1) + f(S_1[i], S_2[j]) + \sigma[k]$ discusses the situation that $C[k]$ is covered by $S_1[i]$ and $S_2[j]$. One can see that all situations are considered in this formula, which means the lemma holds. □

According to our algorithm, the WCPSA problem can be solved with $O(rnm)$ time and space. However, note that an optimal solution in WCPSA may not cover the entire constrained sequence. Therefore, to show that WCPSA is an extension to CPSA, we still need to prove that CPSA is a special scheme of WCPSA. The main idea of our proof is to select proper values for σ and δ, which guarantee that an optimal WCPSA is also optimal in CPSA.

Lemma 6. *The CPSA problem can be linearly reduced to the WCPSA problem.*

Proof. Let $u = max\{n,m\}$ and $v = max\{|f(S_1[i], `-\text{'})|, |f(`-\text{'}, S_2[j])|, |f(S_1[i], S_2[j])|\}$, for $1 \le i \le n$ and $1 \le j \le m$. Let $\delta[k] = 0$ and $\sigma[k] = x$, for $1 \le k \le r$. For any alignment $\bar{A}$ of S_1 and S_2, we assume that it obtains score αx from covering α constraints, and obtains score p from the function f. Clearly, $\bar{A}$ has score $\alpha x + p$ if it is measured with WCPSA. Note that for any alignment $\bar{A}$, we have $|\bar{A}| \leq 2u$, where $|\bar{A}|$ denotes the length of $\bar{A}$. Since v denotes the maximum absolute value obtained from f, we conclude that $-2uv \leq p \leq 2uv$. Based on this conclusion, by setting x to $4uv + 1$, one can see that any alignment with maximum WCPSA score must first maximize α (the number of covered constraints), and then maximize p (the alignment score). Therefore, we can obtain α in $O(1)$ time by rounding $\frac{R(r,n,m)}{4uv+1}$. That is, we can set each $\sigma[k]$ to $4uv + 1$, and check if α equals to r after we obtain $R(r, n, m)$. If α equals to r, then an optimal WCPSA is also an optimal CPSA, because the obtained WCPSA covers all constraints, and has maximized its score in CPSA. At the same time, there is no solution for CPSA if $\alpha < r$. Hence, the CPSA problem can be reduced to the WCPSA problem. □

For solving the CLCS problem, the proper value of x in WCPSA is smaller. In the CLCS problem, we have $f(S_1[i], `-') = 0$, $f(`-', S_2[j]) = 0$, $f(S_1[i], S_2[j]) = 0$ if $S_1[i] \neq S_2[j]$, and $f(S_1[i], S_2[j]) = 1$ if $S_1[i] = S_2[j]$. Therefore, by setting $\delta[k] = 0$ and $\sigma[k] = x$, for $1 \leq k \leq r$, any solution in WCPSA will have its score $\alpha x + \beta$, where β denotes the number of matches in $\bar{A}$. In this case, we have $0 \leq \beta \leq u$. Hence, the proper value for x is $u + 1$, which is smaller than $4uv + 1$. Note that in this case, we have $\alpha = \lfloor \frac{R(r,n,m)}{u+1} \rfloor$.

By applying Hirschberg's algorithm [7], it is clear that the required space in our algorithm can be reduced to $O(rn)$. Therefore, WCPSA is an extension to both CLCS and CPSA, with the same time complexity and space complexity. The main difference between Arslan and Eğecioğlu's extension (DCLCS) and our WCPSA is the scoring scheme. In DCLCS, the scoring scheme cannot be written in a form linear to the number of covered constraints. However, from the proof of Lemma 6, we can clearly see that the scoring scheme in WCPSA is linear to α, which is a crucial property that enables the WCPSA problem to be solved in less time and space. To end this section, in the following we summarize our main result for the WCPSA problem.

Theorem 7. *It takes $O(rnm)$ time and $O(rn)$ space to solve the WCPSA problem, which is an extension to both the CLCS problem and the CPSA problem.*

3. Applications

Since some characters in C can be ignored, the length of C may be greater than n or m. This makes WCPSA feasible for more applications. A simple instance is the ratio $\frac{\alpha}{r}$, which can be used to measure the similarity between the input sequences and the constrained sequence. Besides, by setting proper parameters, WCPSA can be used to solve other constraint-related problems. In the following, we give three main applications to show the flexibility of WCPSA.

The first application is to find a sequence $\bar{S}'$ that covers the most constraints, where $\bar{S}'$ is a longest common subsequence of S_1 and S_2. Note that this problem is quite different from the CLCS problem, because the restriction is placed on the length of LCS. From the perspective of sequence analysis, this problem can be used as a model for filtering better alignments among all optimal alignments [19, 20]. Obviously, as the restriction varies, previous algorithms for CLCS and CPSA are no longer sufficient. However, one can see that by setting $f(S_1[i], S_2[j]) = u + 1$ for $S_1[i] = S_2[j]$, $f(S_1[i], S_2[j]) = 0$ for $S_1[i] \neq S_2[j]$, $f(S_1[i], `-') = 0$, $f(`-', S_2[j]) = 0$, $\sigma[k] = 1$, and $\delta[k] = 0$, for $1 \leq i \leq n$, $1 \leq j \leq m$, and $1 \leq k \leq r$, the score in WCPSA can be written as $\alpha + \beta(u + 1)$. Therefore, we can solve this problem in $O(rnm)$ time by using WCPSA, which is more efficient than adopting Arslan and Eğecioğlu's $O(drnm)$-time algorithm with $d = r$.

The second application is to allow hierarchical constraints, which means the constraints can be classified into several levels according to their importance. By doing so, we can specify which constraints should be covered first, in case some constraints have to be ignored. Taking the two-level constraints for example, we have two kinds of constraints in group G_1 and G_2, respectively. Suppose that group G_1 is prior to group G_2, then the scoring scheme for the LCS can be written as $\alpha_1(u+1)^2+\alpha_2(u+1)+\beta$, where α_1 and α_2 denote the numbers of covered constraints in G_1 and G_2, respectively. Therefore, by setting $f(S_1[i], S_2[j]) = 1$ for $S_1[i] = S_2[j]$, $f(S_1[i], S_2[j]) = 0$ for $S_1[i] \neq S_2[j]$, $f(S_1[i], `-') = 0$, $f(`-', S_2[j]) = 0$, $\sigma[i'] = (u+1)^2$ for each $C[i'] \in G_1$, and $\sigma[j'] = (u+1)$ for each $C[j'] \in G_2$, any solution must first maximize α_1, then maximize α_2. Since α_1, α_2 and β are all nonnegative integers that never exceed $u + 1$, one can retrieve these three numbers by simply doing some $O(1)$-time division to $R(r, n, m)$. Therefore, the LCS problem with two-level constraints can be solved by adopting WCPSA. Nevertheless, for the LCS problem with h-level constraints, the weight for the highest level would be $(u + 1)^h$. This means we have to spend $O(h)$ time for doing one mathematic operation. Therefore, the time complexity grows into $O(hrnm)$, which will be $O(r^2nm)$ in the worst case. However, note that in hierarchical systems, the number of levels are usually limited. Hence, it is proper to assume that $h \ll max\{r, n, m\}$, which means in general, the required time for h-level constraints can be kept in $O(rnm)$. Therefore, WCPSA is an efficient model for handling hierarchical constraints. We also notice that DCLCS may be modified to handle hierarchical constraints. However, as one can see, DCLCS does not readily handle this case.

The third application of WCPSA is to solve the decision version of the DCLCS problem. That is, for a given integer d, we are to determine whether there exists any common subsequence (or alignment) of S_1 and S_2 that ignores no more than d constraints. Recall that by using WCPSA, we can obtain α (the maximum number of covered constraints) for the CLCS problem. Therefore, the decision version of the DCLCS problem can be answered by checking whether $(r - \alpha) \leq d$. It is clear that there exists a solution to the DCLCS problem if and only if $(r - \alpha) \leq d$. Hence, the decision version of Arslan and Eğecioğlu's extended problem can be solved in $O(rnm)$ time, rather than $O(drnm)$ time. The solution of WCPSA also indicates the minimum d that can be used to obtain a solution in the DCLCS problem. However, one should note that for a given d, Arslan and Eğecioğlu's $O(drnm)$-time algorithm is still the best known result for solving the DCLCS problem.

Aside from the above applications, one can replace each sequence of characters with a sequence of vocabularies. In this way, WCPSA can be used to compare articles with some user-specified keywords of different importance. Furthermore, the scoring function $f(S_1[i], S_2[j])$ in WCPSA can be replaced with other scoring matrices such as the PAM250 mutation matrix for proteins [16], which makes WCPSA more compatible with protein sequence alignment.

4. Conclusions and Future Studies

In contrast with Arslan and Eğecioğlu's DCLCS, our WCPSA is another flexible extension to CPSA. With our $O(rnm)$-time algorithm for WCPSA, some constraint-related problems can also be solved in $O(rnm)$ time by merely setting proper parameters. In Section 3, one can see the advantage of WCPSA over DCLCS. Therefore, WCPSA is a suitable model, which can further be used to design heuristic algorithms for multiple sequence alignment with constraints.

For future study, it is interesting and challenging to find out whether Arslan and Eğecioğlu's $O(drnm)$-time result for the DCLCS problem can be improved by manipulating parameters in WCPSA. Recently, Arslan extended C from a plain sequence to a sequence represented in regular expression, which can be used to find an optimal alignment guided by given motifs [2, 5]. Therefore, to consider weights on such C is also an interesting future study, which may have practical applications in bioinformatics.

Acknowledgments

The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] H.-Y. Ann, C.-B. Yang, C.-T. Tseng, and C.-Y. Hor, "A fast and simple algorithm for computing the longest common subsequence of run-length encoded strings," *Information Processing Letters*, Vol. 108, pp. 360–364, 2008.

[2] A. N. Arslan, "Regular expression constrained sequence alignment," Vol. 5(4), pp. 647–661, 2007.

[3] A. N. Arslan and Ömer Eğecioğlu, "Algorithms for the constrained longest common subsequence problems," *International Journal of Foundations of Computer Science*, Vol. 16(6), pp. 1099–1109, 2005.

[4] F. Y. L. Chin, A. De Santis, A. L. Ferrara, N. L. Ho, and S. K. Kim, "A simple algorithm for the constrained sequence problems," *Information Processing Letters*, Vol. 90, pp. 175–179, 2004.

[5] Y.-S. Chung, C. L. Lu, and C. Y. Tang, "Efficient algorithms for regular expression constrained sequence alignment," *Information Processing Letters*, Vol. 103, pp. 240–246, 2007.

[6] Z. Gotthilf, D. Hermelin, and M. Lewenstein, "Constrained LCS: hardness and approximation," *Combinatorial Pattern Matching, 19th Annual Symposium (CPM2008)*, Pisa, Italy, pp. 255–262, 2008.

[7] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, Vol. 18, pp. 341–343, 1975.

[8] K.-S. Huang, C.-B. Yang, K.-T. Tseng, H.-Y. Ann, and Y.-H. Peng, "Efficient algorithms for finding interleaving relationship between sequences," *Information Processing Letters*, Vol. 105(5), pp. 188–193, 2008.

[9] K.-S. Huang, C.-B. Yang, K.-T. Tseng, Y.-H. Peng, and H.-Y. Ann, "Dynamic programming algorithms for the mosaic longest common subsequence problem," *Information Processing Letters*, Vol. 102, pp. 99–103, 2007.

[10] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, Vol. 20(5), pp. 350–353, 1977.

[11] C. S. Iliopoulos and M. S. Rahman, "New efficient algorithms for the LCS and constrained LCS problems," *Information Processing Letters*, Vol. 106(1), pp. 13–18, 2008.

[12] G. A. Komatsoulis and M. S. Waterman, "Chimeric alignment by dynamic programming: algorithm and biological uses," *RECOMB '97: Proceedings of the first annual international conference on computational molecular biology*, New York, NY, USA, pp. 174–180, ACM Press, 1997.

[13] C.-L. Lu and Y.-P. Huang, "A memory-efficient algorithm for multiple sequence alignment with constraints," *Bioinformatics*, Vol. 21(1), pp. 20–30, 2005.

[14] C.-L. Peng, "An approach for solving the constrained longest common subsequence problem." http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search/search, Master's Thesis, Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, 2003.

[15] Y.-H. Peng, C.-B. Yang, K.-T. Tseng, and K.-S. Huang, "Sequence alignment with weighted constraints," *Proceedings of 2006 Symposium on Digital Life Technologies: Building a Safe, Secured and Sound (3S) Living Environment*, Tainan, Taiwan, pp. 36 (Abstract, full text in CD), 2006.

[16] R. M. Schwartz and M. O. Dayhoff., *Matrices for detecting distant relationships.* In M. Dayhoff, editor, Atlas of Protein Sequence and Structure, volume 5, pages 353-358. National Biomedical Research Foundation, Washington, DC, 1979.

[17] C. Y. Tang, C. L. Lu, M. D.-T. Chang, Y.-T. Tsai, Y.-J. Sun, K.-M. Chao,

J.-M. Chang, Y.-H. Chiou, C.-M. Wu, H.-T. Chang, and W.-I. Chou, "Constrained multiple sequence alignment tool development and its application to RNase family alignment," *Journal of Bioinformatics and Computational Biology*, Vol. 1, pp. 267–287, 2003.

[18] Y.-T. Tsai, "The constrained common sequence problem," *Information Processing Letters*, Vol. 88, pp. 173–176, 2003.

[19] K.-T. Tseng, C.-B. Yang, and K.-S. Huang, "The better alignment among output alignments," *Journal of Computers*, Vol. 3, pp. 51–62, 2007.

[20] K.-T. Tseng, C.-B. Yang, K.-S. Huang, and Y.-H. Peng, "Near-optimal block alignments," *IEICE Transactions on Information and Systems*, Vol. E91-D, No. 3, pp. 789–795, 2008.

[21] C.-B. Yang and R. C. T. Lee, "Systolic algorithms for the longest common subsequence problem," *Journal of the Chinese Institute of Engineers*, Vol. 10(6), pp. 691–699, 1987.

[22] D. C. T. Yu, "Efficient algorithms for constrained sequence alignment problems." http://ethesys.lib.pu.edu.tw/ETD-db/ETD-search/search, Master's Thesis, Department of Computer Science and Information Management, Providence University, Taichung, Taiwan, 2003.