A Parallel Algorithm for Three-Profile Alignment Method

Che-Lun Hung
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
allen@sslab.cs.nthu.edu.tw

Chun-Yuan Lin
Department of Computer Science and Information Engineering, Chang Gung University
Taoyuan, Taiwan
eyulin@mail.cgu.edu.tw

Yeh-Ching Chung
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
ychung@cs.nthu.edu.tw

Chuan Yi Tang
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
cytsang@cs.nthu.edu.tw

Abstract—Profile-profile alignment is an important technique in the computational biology filed. Several profile–profile alignment methods have been proposed to improve the sensitivity and the alignment quality compared with other sequence–sequence and profile–sequence methods. An increasing number of studies indicated that the three-way alignment may provide additional information or more accurate alignment result than the pair-wise alignment does. Therefore, we propose the dynamic programming based three-profile alignment method, TPA, at first to align three profiles simultaneously. The time and space complexities of TPA are $O(n^2)$ and $O(n^3)$, respectively. To reduce the complexities of TPA, we further develop the parallel version of TPA, PTPA, which achieves $O(n^2/p)$ time and $O(n^3/p)$ space complexities, where $p$ is the number of the processor. In the case study I, the result presented that PTPA can find more conserve candidates than those by the profile-profile alignment method (CLUSTALW). In the case study II, we applied the PTPA to the Feature Amplified Voting Algorithm (FAVA) to analysis the Amidohydrolase superfamily. Several amino acid residues those were known to be related to the function or the structure of mammalian imidase are identified by PTPA-FAVA.

Keywords—multiple alignment; profile alignment; three-way alignment; parallel computing; parallel sequence alignment

I. INTRODUCTION

Alignment of proteins is the scientific method to assist the study of homology and evolutionary events [1-3]. A pair-wise alignment of a novel sequence with a sequence of known function or structure can help to identify homologous positions and regions. The similarity of the distantly related proteins can be hard to detect from the primary sequence alone, although they may share a common fold and function [4]. Multiple alignments of related proteins can provide more information about the family, indicating patterns of conservation or variation at each residue position. Therefore, alignments for two multiple-sequence alignments (profiles) or statistical models of such alignments have been the important applications in the computational biology. A profile is statistical model of a multiple alignment. The profile typically contains the estimated probability of each amino acid type at each position. Traditionally, the sum of pairs multiple alignment can be classified into three groups: sequence–sequence methods [5,6], profile–sequence methods [7-10] and profile-profile methods [11-16]. Profile-profile alignment methods have been reported that the alignment accuracy and the homolog recognition are improved over profile–sequence and sequence–sequence methods. Profile–profile methods have been used in the genome annotation and the protein classification [17-20].

Recently, the three-way alignment method has been applied to compare three sequences in many applications. The theoretical analyses indicated that the simultaneous comparison of three sequences, rather than two ones, increases the alignment power to distinguish significant matches [21,22]. Murata et al. [23] presented a three-way alignment algorithm with a constant gap penalty function, which leads to more accurate and significant similarities than the pair-wise alignment approach for three copper-containing proteins, namely plastocyanin, stellacyanin and cucumber basic blue protein. M.S. Rosenberg [24] indicated that the improvement in accuracy of adding a third sequence to a pair-wise alignment and that the improvement depends on the evolutionary distance. The analyses of the mammalian phylogeny [20-22] indicated that the alignment accuracies of human and mouse sequences can be enhanced if a third species with an evolutionary distance that is similar to the Capuchin (Cebus albifrons) or the blind mole rat (Spalax judaei) is applied in the alignment.

Therefore, we propose the three-profile alignment method, TPA, which is extended from the dynamic programming based three-sequence alignment method [25] at first to provide the different insight to the profile-profile alignment method. The time and space complexities of TPA are $O(n^3)$ and $O(n^7)$, respectively, by adopting the Hirschberg’s algorithm [26]. In order to reduce the high computation complexity and space requirements, we utilize the parallel computers to approach this goal. We develop a parallel three-profile alignment algorithm, PTPA, which...
adapts the concepts of the previous work [27]. PTPA requires $O(n^3/p)$ time and $O(n^2/p)$ space complexities whose are optimal, where $p$ is the number of processors. In the case study I, we evaluate the performance of PTPA in the three *Enterovirus* types. The results shows that PTPA can find more conserve candidates than that by the profile-profile alignment method, such as CLUSTALW [28]. Therefore, PTPA may offer more different information than the profile-profile alignment.

The Feature Amplified Voting Algorithm (FAVA) [29] has been proposed to search for functional key residues in an Amidohydrolase superfamily. FAVA predicts the functional residues at a target sequence by comparing three different groups: one target sequence, the group of sequences similar to the target sequence ($\alpha$ sequences), the group of sequences divergent from the target sequence ($\beta$ sequences). FAVA calculates and sums the score for each residue triplet at the alignment of three sequences iteratively, and finds the key residues with higher scores. The time complexity of FAVA algorithm is $O((a+b)n^2)$ by performing $(a+b)$ times optimal three-way alignments. For PTPA, the three-profile (three groups of sequences) can be defined by users for various applications/observations; therefore, PTPA can be integrated into FAVA (PTPA-FAVA) to reduce the computation cost. In this case study II, PTPA-FAVA is used to predict the functional sites of the target sequence (rat imidase). The results shows that PTPA can find more conserve candidates than that by the profile-profile alignment method, such as CLUSTALW [28]. Therefore, PTPA may offer more different information than the profile-profile alignment.

![Figure 1: The scoring scheme for comparing three columns from three profiles.](image)

where $Sp_{i,j}$ is the score at the $i$th and $j$th columns on $P_1$ and $P_2$, respectively. $P_1$ has $m$ sequences and $P_2$ has $n$ sequences. $W_a$ and $W_b$ are the sequence weights for sequence $a$ in $P_1$ and sequence $b$ in $P_2$. The residue is denoted by $r_{i,j}$ as the residue at $i$th column for sequence $a$ in $P_1$. $M$ is the value of the substitute matrix for $r_{i,j}$. There are many substitute matrices have been proposed to improve the accuracy of alignments, such as BLOSUM [30] and PAM [1]. These substitute matrices were included into TPA. Similarly, the definitions of scoring pair profiles, $P_1-P_2$ and $P_2-P_3$, are similar to that of pair profiles $P_1-P_2$.

Let $S(i, j, k)$ be the score of an optimal alignment at $i$th column of $P_1$, $j$th column of $P_2$ and $k$th column of $P_3$. The one-gap column is that the gap is at the column of each sequence in two of three profiles. The two-gap column is that the gap is at the column of each sequence in one of three profiles. The columns open at $i$th column of $P_1$, $j$th column of $P_2$ and $k$th column of $P_3$ match two-gap columns, respectively. The definitions of auxiliary matrices are listed as follows.

\[
S(i, j, k) = \begin{cases} 
0 & \text{if } i = 0, j = 0 \text{ and } k = 0 \\
- \left( \mathcal{G}(i, j, k) + \sum_{l=1}^{m} \mathcal{M}(i, j, l) \right) & \text{if } i > 0, j = 0 \text{ and } k = 0 \\
- \left( \mathcal{G}(i, j, k) + \sum_{l=1}^{m} \mathcal{M}(i, j, l) \right) & \text{if } i = 0, j > 0 \text{ and } k = 0 \\
- \left( \mathcal{G}(i, j, k) + \sum_{l=1}^{m} \mathcal{M}(i, j, l) \right) & \text{if } i = 0, j = 0 \text{ and } k > 0 \\
\max [G(i, j, k), H(i, j, k), J(i, j, k)] & \text{if } i > 0, j > 0 \text{ and } k = 0 \\
\max \left[ E(i, j, k), H(i, j, k), J(i, j, k) \right] & \text{if } i > 0, j = 0 \text{ and } k > 0 \\
\max \left[ F(i, j, k), H(i, j, k), J(i, j, k) \right] & \text{if } i = 0, j > 0 \text{ and } k > 0 \\
\max \left[ E(i, j, k), P(i, j, k), G(i, j, k), S(i, j, k) \right] & \text{if } i > 0, j > 0 \text{ and } k > 0 \\
\end{cases}
\]
An entry \((i, j, k)\) denotes that three columns of three profiles, respectively; \(i\)th column of the \(P_1\), \(j\)th column of the \(P_2\), \(k\)th column of the \(P_3\). From the definitions, \(S(i, j, k)\) is computed through the values of other matrices at the entry \((i, j, k)\). 

\[ G(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GOP}(i) \text{, if } i > 0, j > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(k) \text{, if } i > 0, j > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GOP}(0) \text{, if } i > 0, j > 0 \end{array} \right\} + \text{Sp}(i, j) - \text{GEP}(k) \text{ if } i > 0, j > 0 \]

\[ E(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GOP}(j) \text{, if } i > 0, k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(j) \text{, if } i > 0, k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(j) \text{, if } i > 0, k > 0 \end{array} \right\} + \text{Sp}(i, k) - \text{GEP}(j) \text{ if } i > 0, k > 0 \]

\[ F(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GEP}(i) \text{, if } j > 0, k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GOP}(i) \text{, if } j > 0, k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GOP}(i) \text{, if } j > 0, k > 0 \end{array} \right\} + \text{Sp}(j, k) - \text{GEP}(i) \text{ if } j > 0, k > 0 \]

\[ H(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GEP}(j, k) \text{, if } i > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(j, k) \text{, if } i > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(j, k) \text{, if } i > 0 \end{array} \right\} \]

\[ I(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GEP}(i, k) \text{, if } j > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(i, k) \text{, if } j > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(i, k) \text{, if } j > 0 \end{array} \right\} \]

\[ J(i, j, k) = \max \left\{ \begin{array}{l} S_{i-1,j-1} - \text{GEP}(i, j) \text{, if } k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(i, j) \text{, if } k > 0 \noalign{\vskip 2pt} S_{i-1,j-1} - \text{GEP}(i, j) \text{, if } k > 0 \end{array} \right\} \]

The pseudo code of the TPA is presented as below.

Procedure align \( (i_1, j_1, k_1, i_2, j_2, k_2) \) { 
1. \( k_{\text{mid}} = \frac{k_2}{2} \);
2. Find an optimal path from \((i_1, j_1, k_1)\) to \((i_2, j_2, k_2)\);
3. Find an optimal path from \((i_2, j_2, k_2)\) to \((i_1, j_1, k_1)\);
4. Find a middle point \((i_{\text{mid}}, j_{\text{mid}}, k_{\text{mid}})\);
5. \( i_{\text{mid}} = i_{\text{mid}} + 1, j_{\text{mid}} = j_{\text{mid}} + 1, k_{\text{mid}} = k_{\text{mid}} + 1 \);
6. \( i_{\text{mid}} = i_{\text{mid}} + 1, j_{\text{mid}} = j_{\text{mid}} + 1, k_{\text{mid}} = k_{\text{mid}} + 1 \);
}

B. **PTPA algorithm**

The critical cost of TPA is the computation of the auxiliary matrices. Hence, this part should be paralleled to reduce the computation complexity and the space requirement. For each matrix in TPA, it is divided into \( p \) parts, where \( p \) is the number of processors. As TPA, PTPA also has four phases: initial, forward, reverse and middle point phases. The pseudo code of initial phase is shown below:

**Initial phase:**

1. Initialize \( p \) processors from \( 0 \) to \( p - 1 \).
2. Reading input profiles \( P_1, P_2 \) and \( P_3 \) with sequence lengths \( L_1, L_2 \) and \( L_3 \), respectively.
3. Compute \( S_{P_{12}}, S_{P_{13}} \) and \( S_{P_{23}} \), respectively.
4. Set \( k_{\text{mid}} = \frac{L_1}{2} \);
5. Allocate \( 8 \) matrices and each matrix has \((L_1 + 1)\times(\frac{L_1}{p} + 1)\) elements and a few arrays for processor from \( 0 \) to \( p - 2 \).
In this phase, $k_{\text{mid}}$ is set to $\left\lfloor \frac{L_0}{2} \right\rfloor$ initially and the matrices are allocated to each processor dynamically. Since input sequences may be large, dynamically allocating memory for matrices is very important. In TPA, the matrices computations are divided into two phases, forward and reverse. To implement PTPA, only eight matrices, $S$, $E$, $F$, $J$, $R$, $U$, $V$ and $Y$ need to be compute in these two phases. These matrices store the values computed according to the recurrences defined before. The pseudo code of forward phase is shown as below:

Forward phase
For ($k=0$ to $k=k_{\text{mid}}$) {
Processor rank $= 0$
Step 1: Compute matrices: $S$, $E$, $F$ and $J$
Step 2: Send the scores of last columns of $S$, $E$, $F$ and $J$ to processor rank $= 1$
Processor rank $i$, for $1 \leq i \leq p - 2$
Step 1: Receive the scores of last columns of $S$, $E$, $F$ and $J$ from rank $(i-1)$
Step 2: Compute $S$, $E$, $F$ and $J$
Step 3: Send the scores of last columns of $S$, $E$, $F$ and $J$ to rank $(i+1)$
Processor rank $(p-1)$
Step 1: Receive the scores of last columns of $S$, $E$, $F$ and $J$ from rank $(p-2)$
Step 2: Compute $S$, $E$, $F$ and $J$
}

Reverse phase
For ($k=L$ to $k=k_{\text{mid}}$) {
Processor rank $= (p-1)$
Step 1: Compute $R$, $U$, $V$ and $Y$
Step 2: Send the scores of last columns of $R$, $U$, $V$ and $Y$ to rank $(p-2)$
Processor rank $j$, for $2 \leq j \leq p - 1$
Step 1: Receive the scores of last columns of $R$, $U$, $V$ and $Y$ from rank $(j-1)$
Step 1: Compute $R$, $U$, $V$ and $Y$
Step 2: Send the scores of last columns of $R$, $U$, $V$ and $Y$ to rank $(j+1)$
Processor rank $= 0$
Step 1: Receive the scores of last columns of $R$, $U$, $V$ and $Y$ from rank $(p-2)$.
Step 2: Compute $R$, $U$, $V$ and $Y$
}

The pseudo code of reverse phase is shown as below:

Middle point phase
For (processor rank $= 1$ to rank $(p-1)${
Step 1: Find a middle point
Step 2: Send middle point to Rank_0
}
Processor rank $= 0$
Step 1: Find a middle point as a candidate point.
Step 2: Receive middle points from other processors as candidate middle points.
Step 3: Determine the middle point among these candidate middle points.

C. Complexity analysis
The dynamic programming approach for the three-sequence alignment requires $O(n^3)$ time and $O(n^3)$ space complexities by adopting the Hirschberg’s algorithm. Each matrices used in the forward and the reverse phases are reduced to two-dimensional matrices. In TPA, three additional two-dimensional matrices need to be computed in the initial phase for recording the scores of each pair profiles. Therefore, the time complexity of TPA can be written as $O(3n^2 + n^3)$, which is equivalent to $O(n^3)$. The space complexity of TPA is $O(n^3)$ as the three-way alignment algorithm.

In PTPA, each matrix of each processor has $(L_1+1) \times \left( \frac{L_2}{p} \right) + 1$ elements except the last $(rank_{p-1})$ processor. The last processor allocates $2 \times (L_1+1) \times \left( \frac{L_2}{p} \right) + 1$ elements for each matrix. Each processor takes $O((L_1L_2)/p)$ time and $O((L_1L_2)/p)$ space complexities in each matrices. Therefore, the time complexity of PTPA is $O((L_1L_2L_3)/p)$ and the space complexity of PTPA is $O((L_1L_2L_3)/p)$.

III. EXPERIMENTS
A. Case study I: comparison of Three-profile alignment and Profile-Profile alignment in Enterovirus
In this case study, three types of Enterovirus are utilized to evaluate the performance of PTPA. These three types are Poliovirus Type 1 (PV1), Coxsackievirus Type 16 (CA16) and Enterovirus Type 71 (EV71). Each type is regarded as a profile. The sequences of these three types are retrieved from National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov/), and the accessions of sequences are shown in Appendix. All of the tested sequences are the part (1A block) of the complete sequences. These three profiles are aligned simultaneously by PTPA. Three profile-profile alignments, EV71-PV1, EV71-CA16 and PV1-CA16, are done by the profile-profile alignment method, CLUSTALW, respectively. Table 1 shows the number of the candidate conserve sites found by PTPA and CLUSTALW for EV71, PV1 and CA16.
By splitting the resulting alignment of PTAP into three pair alignments, EV71-PV1, EV71-CA16 and PV1-CA16, 43, 54 and 43 conserve candidates can be found, respectively. For three profile-profile resulting alignments by CLUSTALW, only 39, 54 and 42, conserve candidates can be found, respectively. In this case study, the result shows that it may provide more different information than the profile-profile alignment by adding the third profile to do the alignment.

B. Case study II: An application of Three-Profile Alignment for imidase related proteins in Amidohydrolase superfamily

In the Amidohydrolase superfamily [32], some proteins are similar in the sequence but are divergent in the function and some are opposite. There were 156 protein sequences of Amidohydrolase superfamily found from the PIR database [33] by searching the rat imidase sequence (NP_113893). According to the sequence similarity and the biochemical properties, these sequences were clustered into five groups: I. imidase (imide-hydrolyzingenzyme from mammal); II. sequence related proteins (dihydropyrimidinase related proteins that contain higher than 50% sequence identity to mammalian imidase but without imidase activity); III. functionally identical enzymes (hydantoinase, or the imide-hydrolyzing enzyme from bacteria that contain 30-40% sequence identity to mammalian imidase); IV. functionally related enzymes (dihydroorotase, allantoinase, urease, and amidohydrolase that contain 25-48% sequence identity to mammalian imidase); and V. putative sequences (gene products with unknown function) that contain higher than 30% sequence identity to mammalian imidase.

Feature Amplified Voting Algorithm (FAVA) [29] was developed to identify the key residues in a target protein (interested sequence). There are two main steps in FAVA. The first step was to align the target protein, one of the functionally identical proteins and one of the sequence related proteins by the three-way alignment. In the second

### Table 1. The comparison of the candidate conserve sites by PTPA and CLUSTALW in enterovirus

<table>
<thead>
<tr>
<th></th>
<th>EV71-PV1</th>
<th>EV71-CA16</th>
<th>PV1-CA16</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTPA</td>
<td>43</td>
<td>54</td>
<td>43</td>
<td>40</td>
</tr>
<tr>
<td>CLUSTALW</td>
<td>39</td>
<td>54</td>
<td>42</td>
<td>35</td>
</tr>
</tbody>
</table>

Three profiles are utilized in this comparison: Polioviruses Types 1 (PV1), Coxsackieviruses Type 16 (CA16) and Enterovirus Type 71 (EV71). The accessions of each profile are listed in appendix. The alignments of EV71-PV1, EV71-CA16 and PV1-CA16 are abstracted from the alignment of EV71, PV1 and CA16 by PTPA, respectively. The alignments of EV71-PV1, EV71-CA16 and PV1-CA16 are directly aligned by CLUSTALW, respectively. The total number of candidate conserve sites of PTPA is found from the resulting alignment. The total number of candidate conserve sites of CLUSTALW is found by comparing three profile-profile resulting alignments, EV71-PV1, EV71-CA16 and PV1-CA16.

### Table 2. Functional annotations of the residues in rat imidase selected by PTPA-FAVA

<table>
<thead>
<tr>
<th>PTPA-FAVA selected residues</th>
<th>FAVA selected residues</th>
<th>Corresponding residues in 1GKQ</th>
<th>MSA predicted residues</th>
<th>Functional annotations base on 1GKQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>His69&lt;sup&gt;2&lt;/sup&gt;</td>
<td>His69&lt;sup&gt;2&lt;/sup&gt;</td>
<td>His61&lt;sup&gt;2&lt;/sup&gt;</td>
<td>His61&lt;sup&gt;2&lt;/sup&gt;</td>
<td>Metal coordinate</td>
</tr>
<tr>
<td>Ala34</td>
<td>Ala34</td>
<td>Arg30</td>
<td>Quaternary structure</td>
<td></td>
</tr>
<tr>
<td>His67</td>
<td>His67</td>
<td>His59</td>
<td>His59</td>
<td>Metal coordinate</td>
</tr>
<tr>
<td>Lys159</td>
<td>Lys159 (Kcx)</td>
<td>Lys150</td>
<td>Metal coordinate</td>
<td></td>
</tr>
<tr>
<td>His248</td>
<td>His248</td>
<td>His239</td>
<td>His239</td>
<td>Metal coordinate</td>
</tr>
<tr>
<td>Asp326</td>
<td>Asp326</td>
<td>Asp315</td>
<td>Metal coordinate</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> MSA result according to the literature [31].

<sup>2</sup> His69 selected by FAVA is the same residue as functional residue as His61 in 1GKQ. The rest of selected residues are corresponding to the residues in 1GKQ except Ala34.

### Table 3. The comparison of computation time for various numbers of processors and lengths of input profiles

<table>
<thead>
<tr>
<th>No. of processors</th>
<th>Sequence length</th>
<th>1(TPA)</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000x1000x1000</td>
<td></td>
<td>179</td>
<td>153</td>
<td>102</td>
<td>70</td>
</tr>
<tr>
<td>1500x1500x1500</td>
<td></td>
<td>634</td>
<td>523</td>
<td>405</td>
<td>222</td>
</tr>
<tr>
<td>2000x2000x2000</td>
<td></td>
<td>1566</td>
<td>1297</td>
<td>725</td>
<td>411</td>
</tr>
</tbody>
</table>

By splitting the resulting alignment of PTAP into three pair alignments, EV71-PV1, EV71-CA16 and PV1-CA16, 43, 54 and 43 conserve candidates can be found, respectively. For three profile-profile resulting alignments by CLUSTALW, only 39, 54 and 42, conserve candidates can be found, respectively. In this case study, the result shows that it may provide more different information than the profile-profile alignment by adding the third profile to do the alignment.

B. Case study II: An application of Three-Profile Alignment for imidase related proteins in Amidohydrolase superfamily

In the Amidohydrolase superfamily [32], some proteins are similar in the sequence but are divergent in the function and some are opposite. There were 156 protein sequences of Amidohydrolase superfamily found from the PIR database [33] by searching the rat imidase sequence (NP_113893). According to the sequence similarity and the biochemical properties, these sequences were clustered into five groups: I. imidase (imide-hydrolyzingenzyme from mammal); II. sequence related proteins (dihydropyrimidinase related proteins that contain higher than 50% sequence identity to mammalian imidase but without imidase activity); III. functionally identical enzymes (hydantoinase, or the imide-hydrolyzing enzyme from bacteria that contain 30-40% sequence identity to mammalian imidase); IV. functionally related enzymes (dihydroorotase, allantoinase, urease, and amidohydrolase that contain 25-48% sequence identity to mammalian imidase); and V. putative sequences (gene products with unknown function) that contain higher than 30% sequence identity to mammalian imidase.

Feature Amplified Voting Algorithm (FAVA) [29] was developed to identify the key residues in a target protein (interested sequence). There are two main steps in FAVA. The first step was to align the target protein, one of the functionally identical proteins and one of the sequence related proteins by the three-way alignment. In the second
step, a voting score (V-score) was given based on the previous assumption then the V-score was sum up in each comparison. These two steps were progressed iteratively until all triplets were compared. The time complexity of FAVA is $O(\alpha p m_{\text{max}}^3)$ by performing $(\alpha p)$ times three-way alignments where $\alpha$ and $p$ are numbers of proteins in a group of the functionally identical proteins (A proteins) and another group of the sequence related proteins (A proteins), respectively, and $m_{\text{max}}$ is the length of the longest sequence among all proteins. Therefore, PTPA can be integrated into FAVA (PTPA-FAVA). There are three steps for PTPA-FAVA. First, A and --A groups are separately aligned by progress multiple sequences alignment (MSA) methods, such as CLUSTALW [28], T-Coffee [34], MUSCLE [35], ProbCons [36] and etc. Second, these two groups and the target protein are aligned by PTPA. The last step is to calculate V-score for each column of alignments. The time complexity of PTPA-FAVA is $O(m_{\text{max}}^3 p^3)$.

In this case study, rat imidase was a target sequence and DRPs (Group II proteins) were classified into --A proteins. Bacterial hydantoinases (Group III enzymes) were classified into A proteins. Dihydroorotase, allantoinase and other amidohydrolases (Group IV) were classified into another A proteins. The difference between Group III and Group IV enzymes was the degree of functional correlation to target sequence (rat imidase). A and --A groups were aligned by T-Coffee before performing PTPA. Following the above classification, the aligned groups were as the inputs to PTPA and two sets of scores which corresponded to each residue of target sequence were obtained by FAVA. Each residue in rat imidase obtains two sets of V-scores from Group II-Group III and Group II-Group IV vote. Six of ten amino acids, found by PTPA-FAVA and FAVA, were selected for further analysis when merging the two sets of top 10 high scores. The amino acids residues selected by PTPA-FAVA, FAVA and MSA analyses [31;37;38] and their corresponding locations in the protein with known crystal structures (PDB ID: 1GKQ) were summarized in Table 2. Two residues, Lys150 and Asp315, were not revealed by MSA analysis. In this case study, PTPA-FAVA successfully identified all the known important residues in rat imidase as FAVA.

C. Comparison of computation time

TPA and PTPA have been implemented by C and MPI+C, respectively. These two programs were tested in the Linux Cluster which is AMD Opteron 250 with a 2.4GHz CPU and 512MB memory. There are four test sets with different lengths for this comparison. Each test set has three profiles, and each profile has ten sequences. The runtime of PTPA with various numbers of processors and lengths of input sequences of profiles is summarized in Table 3. From Table 3, PTPA can reduce the computation time of TPA.

IV. CONCLUSIONS

Profile-profile alignment has been used in many applications of computational biology. Recently, an increasing number of studies indicated that the three-way alignment of sequences can offer further information or more accurate alignment than the pair-wise alignment does. Therefore, we proposed TPA to do the three-profile alignment, and developed the parallel version of TPA, PTPA, to reduce the time and space complexities of TPA. In the case study I, PTPA can find more conserve candidates than that by CLUSTALW in Enterovirus. Therefore, it may offer more information to assist biologists to analysis sequences by adding third profile to align three profiles simultaneously. In the case study II, we applied PTPA into FAVA (PTPA-FAVA) to present a specific application of three-profile alignment. PTPA-FAVA can reduce the time and space complexities of FAVA, and also can predict functional important amino acids in mammalian imidase. In the future, we will apply TPA into PTPA into more interesting applications.

REFERENCES

TABLE 4. THE LIST OF THE ACCESSIONS OF ENTEROVIRUS UTILIZED IN CASE STUDY 1

<table>
<thead>
<tr>
<th>Enterovirus type (Profile)</th>
<th>Accession No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterovirus Type 71 (EV71)</td>
<td>EU864507.1, EU812515.1, EU364841.1, EU703814.1, EU703813.1, EU703812.1, EU131776.1, EF373576.1, EF373575.1, DQ41368.1</td>
</tr>
<tr>
<td>Polioviruses Types 1 (PV1)</td>
<td>V01149.1, EF682359.1, EF682358.1, EF682357.1, EF682356.1, AF538843.1, AF538842.1, AF538841.1, AF538840.1, V01418.1</td>
</tr>
<tr>
<td>Coxsackieviruses Type 16 (CA16)</td>
<td>EU812514.1, EU262658.1, AV790926.1, AF177911.1, U05876.1</td>
</tr>
</tbody>
</table>