2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing

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
cyulin@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
cytang@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 $O(n^3)$ and $O(n^2)$, respectively. To reduce the complexities of TPA, we further develop the parallel version of TPA, PTPA, which achieves $O(n^3/p)$ time and $O(n^2/p)$ 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 $O(n^2)$, 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

978-0-7695-3739-9/09 $25.00 © 2009 IEEE
DOI 10.1109/IJCBS.2009.41

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 (α sequences), the group of sequences divergent from the target sequence (β 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 $O((\alpha\beta)n^3)$ by performing $(\alpha\beta)$ 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 the case study II, PTPA-FAVA is used to predict the functional sites of the target sequence (rat imidase). The time complexity of PTPA-FAVA is $O(n^3/p)$. The result shows that the functional residues of a rat imidase are identified by PTPA-FAVA, and they are further confirmed by experimental references and available structural information.

II. ALGORIGHM

A. TPA algorithm

The previous work [25] has presented a dynamic programming based algorithm of three-way alignment utilizing the variable gap penalty, and the time and space complexities are $O(n^3)$ and $O(n^2)$, respectively. In this paper, the first proposed three-profile alignment algorithm, TPA, is extended from the previous work, to align three profiles simultaneously. Let P_1, P_2 and P_3 be three profiles with the profile lengths L_1, L_2 and L_3, respectively. Each profile has one or more sequences. All of these sequences are over an alphabet set Σ. The alphabet set Σ contains various symbols to represent biology sequences, such as 20 symbols for protein sequences. The symbol "-" denotes a "gap" in the sequence alignment. The basic idea to calculate the score at each column for three profiles is illustrated in Figure 1. The score of each column is calculated according to the pair profiles or the gap penalty. The definition of scoring pair profiles, P_1-P_2, is shown as follows:

$$Sp_{12}(i,j)=\frac{\sum_{a=1}^{m}\sum_{b=1}^{n}\left(W_a\times W_b\times M\left[r_{1(a,i)}\right]\left[r_{2(b,j)}\right]\right)}{m\times n},$$

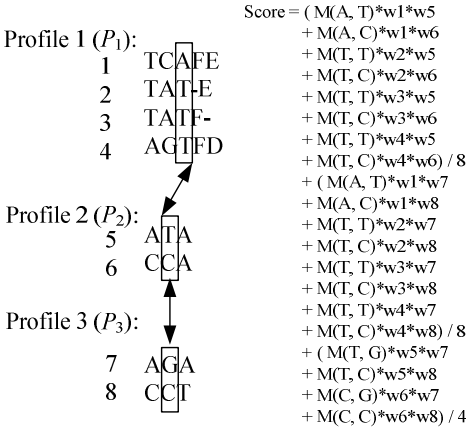

Figure 1: The scoring scheme for comparing three columns from three profiles.

where Sp_{12} is the score at the ith and jth 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 $r_{1(a,i)}$ as the residue at ith column for sequence a in P_1. M is the value of the substitute matrix for $r_{1(a,i)}$ and $r_{2(b,j)}$. 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_3 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 ith column of P_1, jth column of P_2 and kth column of P_3. The one-gap column is that the gap is at the column of each sequence in one of three profiles. The two-gap column is that the gap is at the column of each sequence in two of three profiles. The $S(i, j, k)$ can be computed along with the auxiliary matrices according to the recurrences. The matrices G, E and F save the scores that the one-gap columns open at kth column of P_3, jth column of P_2 and ith column of P_1, respectively. The matrices H, I and J save the scores that ith column of P_1, jth column of P_2 and kth column of P_3 match two-gap columns, respectively. The definitions of auxiliary matrixes are listed as follows.

$$S(i,j,k)=\begin{cases}0 & \text{if } i=0, j=0 \text{ and } k=0\\ -\left(GOP_2(j,k)+\sum_{l=1}^{l=i}GEP_2(j,k)\right) & \text{if } i>0, j=0 \text{ and } k=0\\ -\left(GOP_2(i,k)+\sum_{l=1}^{l=j}GEP_2(i,k)\right) & \text{if } i=0, j>0 \text{ and } k=0\\ -\left(GOP_2(i,j)+\sum_{l=1}^{l=k}GEP_2(i,j)\right) & \text{if } i=0, j=0 \text{ and } k>0\\ \max\left[G(i,j,k),H(i,j,k),I(i,j,k)\right] & \text{if } i>0, j>0 \text{ and } k=0\\ \max\left[E(i,j,k),H(i,j,k),J(i,j,k)\right] & \text{if } i>0, j=0 \text{ and } k>0\\ \max\left[F(i,j,k),I(i,j,k),J(i,j,k)\right] & \text{if } i=0, j>0 \text{ and } k>0\\ \max\begin{bmatrix}E(i,j,k),F(i,j,k),G(i,j,k),\\ H(i,j,k),I(i,j,k),J(i,j,k),\\ S(i-1,j-1,k-1)\\ +Sp(i,j)+Sp(i,k)+Sp(j,k)\end{bmatrix} & \text{if } i>0, j>0 \text{ and } k>0\end{cases}$$

$$G(i,j,k)=\max\begin{Bmatrix}S_{(i-1,j-1,k)}-GOP_1(k),\\ G_{(i-1,j-1,k)}\end{Bmatrix}+Sp(i,j)-GEP_1(k) \quad \text{if } i>0, j>0$$

$$E(i,j,k)=\max\begin{Bmatrix}S_{(i-1,j,k-1)}-GOP_1(j),\\ E_{(i-1,j,k-1)}\end{Bmatrix}+Sp(i,k)-GEP_1(j) \quad \text{if } i>0, k>0$$

$$F(i,j,k)=\max\begin{Bmatrix}S_{(i,j-1,k-1)}-GOP_1(i),\\ F_{(i,j-1,k-1)}\end{Bmatrix}+Sp(j,k)-GEP_1(i) \quad \text{if } j>0, k>0$$

$$H(i,j,k)=\max\begin{Bmatrix}S_{(i-1,j,k)}-GOP_2(j,k),\\ H_{(i-1,j,k)}\end{Bmatrix}-GEP_2(j,k) \quad \text{if } i>0$$

$$I(i,j,k)=\max\begin{Bmatrix}S_{(i,j-1,k)}-GOP_2(i,k),\\ H_{(i,j-1,k)}\end{Bmatrix}-GEP_2(i,k) \quad \text{if } j>0$$

$$J(i,j,k)=\max\begin{Bmatrix}S_{(i,j,k-1)}-GOP_2(i,j),\\ H_{(i,j,k-1)}\end{Bmatrix}-GEP_2(i,j) \quad \text{if } k>0$$

An entry (i, j, k) denotes that three columns of three profiles, respectively; ith column of the P_1, jth column of the P_2, kth 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). GOP_1 and GOP_2 are denoted as the gap opening penalty for one-gap column and two-gap column, respectively. For example, $GOP_1(i)$ return the value of one gap opening penalty if one-gap column is opened at the ith column of P_1. GEP_1 and GEP_2 are denoted as the gap extension penalty for one-gap column and two-gap column, respectively. In these matrices, the gap penalty function can be changeable. For example, *GOP* and *GEP* can return fixed value at any column position (affine gap penalty strategy). Conversely, *GOP* and *GEP* can return variant values at any column position (variable gap penalty strategy). In this work, the affine gap penalty strategy is used for the experiments.

Once $S(L_1, L_2, L_3)$ is computed, an optimal alignment of best score $S(L_1, L_2, L_3)$ can be found by a trace back procedure. Since the entire matrices have to be kept, the trace back procedure requires $O(L_1L_2L_3)$ space. For reducing the space requirement to $O(L^2)$, the divide-and-conquer algorithm of D.S. Hirschberg [26] is also applied into TPA. The main idea that underlies the divide-and-conquer approach is the division of the original three-profile alignment into two smaller three-profile alignments. Each smaller alignment can be further divided into another two ones and so on recursively until no further partitioning is possible. The position that divides the alignment is called the "middle point" (i_{mid}, j_{mid}, k_{mid}) on an optimal path from $S(0, 0, 0)$ to $S(L_1, L_2, L_3)$. The optimal three-profile alignment is obtained by merging the series of the computed middle points. In TPA, there are four phases, *initial*, *forward*, *reverse*, and *middle point*. In the *initial* phase, the scores at each column of three pair profiles are calculated and save to the three matrixes. To find the middle point, k_{mid} is set to $\left\lfloor \frac{L_3}{2} \right\rfloor$. In the *forward* phase, the matrices, S, E, F, G, H, I, J, are computed, where $0 \le i \le L_1$, $0 \le j \le L_2$ and $0 \le k \le k_{\text{mid}}$. Let R, T, U, V, W, X and Y be the set of scores that are similar to the scores given above, where R corresponds to S, T to E, U to F, V to G, W to H, X to I and Y to J. In the *reverse* phase, these matrices, R, T, U, V, W, X, Y, are also computed, where $0 \le i \le L_1$, $0 \le j \le L_2$ and $0 \le k \le k_{\text{mid}}$. Oppositely, the matrices in the reverse phase are computed in a decreasing order of indices. In the *middle point* phase, the middle point (i_{mid}, j_{mid}, k_{mid}) is found by using the optimal paths from the score $S(0, 0, 0)$ to the score $S(L_1, L_2, k_{\text{mid}})$ and from the score $R(L_1, L_2, k_{\text{mid}})$ to the score $R(0, 0, k_{\text{mid}})$ obtained in the *forward* and the *reverse* phases, respectively. The middle point of an optimal alignment of P_1, P_2 and P_3 is one which has the score

$Score = max\{S(i, j, k) + R(i, j, k), E(i, j, k) + U(i, j, k) + GOP_1(j), F(i, j, k) + V(i, j, k) + GOP_1(i), J(i, j, k) + Y(i, j, k) + GOP_2(i,j)\}$

After the middle point (i_{mid} , j_{mid} , k_{mid}) has been determined, the optimal alignments of subsequences from the score of (0, 0, 0) to the score of (i_{mid} , j_{mid} , k_{mid}) and from the score of (i_{mid} + 1, j_{mid} + 1, k_{mid} + 1) to the score of (L_1, L_2, L_3) can be recursively computed. The result of this optimal alignment is obtained by merging the series of the computed middle points. The pseudo code of the TPA is presented as below.

Pseudo code of TPA

calculating Sp_{12}, Sp_{13} and Sp_{23} // Phase 1
Call align (0,0,0,L_1,L_2,L_3)

Procedure align (int x_1, int y_1, int z_1, int x_2, int y_2, int z_2) {
1. $k_{\text{mid}} = z_2/2$;
2. Find an optimal path from (x_1, y_1, z_1) to (x_2, y_2, k_{mid}); // Phase 2
3. Find an optimal path from (x_2, y_2, z_2) to (x_1, y_1, k_{mid}); // Phase 3
4. Find a middle point (i_{mid} , j_{mid} , k_{mid}); // Phase 4
5. align(x_1, y_1, z_1, i_{mid} , j_{mid} , k_{mid});
6. align(i_{mid}+1, j_{mid}+1, k_{mid}+1, x_2, y_2, z_2); }

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 rank_*0 *to rank_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 Sp_{12}, Sp_{13} and Sp_{23}, respectively.
3. Set $k_{\text{mid}} = \left\lfloor \frac{L_3}{2} \right\rfloor$;
4. Allocate 8 matrices and each matrix has $(L_1+1)\times(\left\lfloor \frac{L_2}{p} \right\rfloor+1)$ elements and a few arrays for processor from *rank_*0 to *rank_p*-2.

In this phase, k_{mid} is set to $\left\lfloor \frac{L_3}{2} \right\rfloor$ 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_{mid}) {
  Processor *rank*_0:
    Step 1: Compute matrixes: 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 $1 \le i \le p-2$ **:**
    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 }

The pseudo code of *reverse* phase is shown as below:

Reverse phase
For(k=*L to* k=*kmid***) {**
  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_i*, for $p-2 \ge i \ge 1$ **:**
    Step 1: Receive the scores of last columns of R, U, V and Y from rank_(i-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_(*i+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 }

After above two phases are completed, these matrices are merged to find the middle point. In each processor, a middle point is found and is send to processor *rank*_0 as a candidate middle point. Processor *rank*_0 determines the middle point with the best score among these candidate middle points. The pseudo code of middle point 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 $O(n^2)$ 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 $O(n^2)$ as the three-way alignment algorithm.

In PTPA, each matrix of each processor has $(L_1+1)\times(\left\lfloor \frac{L_2}{p} \right\rfloor+1)$ elements except the last (rank_p-1) processor. The last processor allocates $2\times(L_1+1)\times(\left\lfloor \frac{L_2}{p} \right\rfloor+1)$ 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 $O((L_1L_2)/p)$.

III. EXPERMENTS

A. Case study I: comparison of Three-profile alignment and Profile-Profile alignmnet in Enterovirus

In this case study, three types of Enterovirus are utilized to evaluate the performance of PTPA. These three types are *Polioviruses* Types 1 (PV1), *Coxsackieviruses* 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.

TABLE 1. THE COMPARISON OF THE CANDIDATE CONSERVE SITES BY PTPA AND CLUSTALW IN ENTEROVIRUS

	EV71-PV1	EV71-CA16	PV1-CA16	Total
PTPA	43	54	43	40
CLUSTALW	39	54	42	35

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

PTPA-FAVA selected residues	FAVA selected residues	Corresponding residues in 1GKQ[1]	MSA predicted residues[3]	Functional annotations base on 1GKQ
His69[2]	**His69[2]**	**His61[2]**	**His61[2]**	**Metal coordinate**
Ala34	Ala34	Arg30		Quaternary structure
His67	**His67**	**His59**	**His59**	**Metal coordinate**
Lys159	**Lys159**	**Lys150 (Kcx)**		**Metal coordinate**
His248	**His248**	**His239**	**His239**	**Metal coordinate**
Asp326	**Asp326**	**Asp315**		**Metal coordinate**

[1]MSA result according to the literature [31].
[2]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

No. of processors / Sequence length	1(TPA)	2	4	8
1000×1000×1000	179	153	102	70
1500×1500×1500	634	523	405	222
2000×2000×2000	1566	1297	725	411

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 $O((\alpha\beta)m_{\max}^3)$ by performing $(\alpha\beta)$ times three-way alignments where α and β 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 $m_{\max}$ 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 $O(m_{\max}^3/p)$.

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/PTPA into more interesting applications.

REFERENCES

[1] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, "A model for evolutionary change in proteins," *Atlas of Protein Sequence and Structure*, vol. 5, pp. 345-358, 1978.

[2] A. Elofsson, "A study on protein sequence alignment quality," *Proteins*, vol. 46, no. 3, pp. 330-339, Feb.2002.

[3] B. Wallner, H. Fang, T. Ohlson, J. Frey-Skott, and A. Elofsson, "Using evolutionary information for the query and target improves fold recognition," *Proteins*, vol. 54, no. 2, pp. 342-350, Feb.2004.

[4] S. E. Brenner, C. Chothia, and T. J. Hubbard, "Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships," *Proc. Natl. Acad. Sci. U. S. A*, vol. 95, no. 11, pp. 6073-6078, May1998.

[5] W. R. Pearson, "Rapid and sensitive sequence comparison with FASTP and FASTA," *Methods Enzymol.*, vol. 183, pp. 63-98, 1990.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403-410, Oct.1990.

[7] R. L. Tatusov, S. F. Altschul, and E. V. Koonin, "Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks," *Proc. Natl. Acad. Sci. U. S. A*, vol. 91, no. 25, pp. 12091-12095, Dec.1994.

[8] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389-3402, Sept.1997.

[9] K. Karplus, C. Barrett, and R. Hughey, "Hidden Markov models for detecting remote protein homologies," *Bioinformatics*, vol. 14, no. 10, pp. 846-856, 1998.

[10] M. Gribskov, M. Homyak, J. Edenfield, and D. Eisenberg, "Profile scanning for three-dimensional structural patterns in protein sequences," *Comput. Appl. Biosci.*, vol. 4, no. 1, pp. 61-66, Mar.1988.

[11] T. Ohlson and A. Elofsson, "ProfNet, a method to derive profile-profile alignment scoring functions that improves the alignments of distantly related proteins," *BMC. Bioinformatics*, vol. 6, p. 253, 2005.

[12] G. Yona and M. Levitt, "Within the twilight zone: a sensitive profile-profile comparison tool based on information theory," *J. Mol. Biol.*, vol. 315, no. 5, pp. 1257-1275, Feb.2002.

[13] O. N. von, I. Sommer, and R. Zimmer, "Profile-profile alignment: a powerful tool for protein structure prediction," *Pac. Symp. Biocomput.*, pp. 252-263, 2003.

[14] A. R. Panchenko, "Finding weak similarities between proteins by sequence profile comparison," *Nucleic Acids Res.*, vol. 31, no. 2, pp. 683-689, Jan.2003.
[15] R. Sadreyev and N. Grishin, "COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance," *J. Mol. Biol.*, vol. 326, no. 1, pp. 317-336, Feb.2003.
[16] R. C. Edgar and K. Sjolander, "COACH: profile-profile alignment of protein families using hidden Markov models," *Bioinformatics*, vol. 20, no. 8, pp. 1309-1318, May2004.
[17] K. Pawlowski, B. Zhang, L. Rychlewski, and A. Godzik, "The Helicobacter pylori genome: from sequence analysis to structural and functional predictions," *Proteins*, vol. 36, no. 1, pp. 20-30, July1999.
[18] K. Pawlowski, L. Rychlewski, B. Zhang, and A. Godzik, "Fold predictions for bacterial genomes," *J. Struct. Biol.*, vol. 134, no. 2-3, pp. 219-231, May2001.
[19] J. G. Henikoff, E. A. Greene, S. Pietrokovski, and S. Henikoff, "Increased coverage of protein families with the blocks database servers," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 228-230, Jan.2000.
[20] V. Kunin, B. Chan, E. Sitbon, G. Lithwick, and S. Pietrokovski, "Consistency analysis of similarity between multiple alignments: prediction of protein function and fold structure from analysis of local sequence motifs," *J. Mol. Biol.*, vol. 307, no. 3, pp. 939-949, Mar.2001.
[21] S. F. Altschul and D. J. Lipman, "Protein database searches for multiple alignments," *Proc. Natl. Acad. Sci. U. S. A*, vol. 87, no. 14, pp. 5509-5513, July1990.
[22] E. H. Margulies, C. W. Chen, and E. D. Green, "Differences between pair-wise and multi-sequence alignment methods affect vertebrate genome comparisons," *Trends Genet.*, vol. 22, no. 4, pp. 187-193, Apr.2006.
[23] M. Murata, J. S. Richardson, and J. L. Sussman, "Simultaneous comparison of three protein sequences," *Proc. Natl. Acad. Sci. U. S. A*, vol. 82, no. 10, pp. 3073-3077, May1985.
[24] M. S. Rosenberg, "Multiple sequence alignment accuracy and evolutionary distance estimation," *BMC. Bioinformatics*, vol. 6, p. 278, 2005.
[25] CL Hung, CY Lin, YC Chung, and CY Tang, "Introducing variable gap penalties into three-sequence alignment for protein sequences," *First IEEE International Workshop on BioComputing*, 2008.
[26] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, vol. 18, no. 6, pp. 341-343, 1975.
[27] CY Lin, CT Huang, YC Chung, and CY Tang, "Efficient Parallel Algorithm for Optimal Three-Sequences Alignment," *The International Conference on Parallel Processing, IEEE Computer Society.*, 2007.
[28] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4673-4680, Nov.1994.
[29] CH Lee, YT Lin, CY Tang, and YS Yang, "Identify Amino Acid Candidates Critical for Function of Rat Imidase by Cross-Reference Voting in Imidase Super Family," *ACM Symposium on Applied Computing, Bioinformatics Track*, pp. 127-134, 2003.
[30] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Natl. Acad. Sci. U. S. A*, vol. 89, no. 22, pp. 10915-10919, Nov.1992.
[31] G. J. Kim and H. S. Kim, "C-terminal regions of D-hydantoinases are nonessential for catalysis, but affect the oligomeric structure," *Biochem. Biophys. Res. Commun.*, vol. 243, no. 1, pp. 96-100, Feb.1998.
[32] L. Holm and C. Sander, "An evolutionary treasure: unification of a broad set of amidohydrolases related to urease," *Proteins*, vol. 28, no. 1, pp. 72-82, May1997.
[33] C. H. Wu, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z. Z. Hu, R. S. Ledley, K. C. Lewis, H. W. Mewes, B. C. Orcutt, B. E. Suzek, A. Tsugita, C. R. Vinayaka, L. S. Yeh, J. Zhang, and W. C. Barker, "The Protein Information Resource: an integrated public resource of functional annotation of proteins," *Nucleic Acids Res.*, vol. 30, no. 1, pp. 35-37, Jan.2002.
[34] C. Notredame, D. G. Higgins, and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment," *J. Mol. Biol.*, vol. 302, no. 1, pp. 205-217, Sept.2000.
[35] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Res.*, vol. 32, no. 5, pp. 1792-1797, 2004.
[36] C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," *Genome Res.*, vol. 15, no. 2, pp. 330-340, Feb.2005.
[37] N. K. Williams, M. K. Manthey, T. W. Hambley, S. I. O'Donoghue, M. Keegan, B. E. Chapman, and R. I. Christopherson, "Catalysis by hamster dihydroorotase: zinc binding, site-directed mutagenesis, and interaction with inhibitors," *Biochemistry*, vol. 34, no. 36, pp. 11344-11352, Sept.1995.
[38] J. B. Thoden, G. N. Phillips, Jr., T. M. Neal, F. M. Raushel, and H. M. Holden, "Molecular structure of dihydroorotase: a paradigm for catalysis through the use of a binuclear metal center," *Biochemistry*, vol. 40, no. 24, pp. 6989-6997, June2001.

APPENDIX

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

Enterovirus type (Profile)	**Accession No.**
Enterovirus Type 71 (EV71)	EU864507.1, EU812515.1, EU364841.1, EU703814.1, EU703813.1, EU703812.1, EU131776.1, EF373576.1, EF373575.1, DQ341368.1
Polioviruses Types 1 (PV1)	V01149.1, EF682359.1, EF682358.1, EF682357.1, EF682356.1, AF538843.1, AF538842.1, AF538841.1, AF538840.1, V01148.1
Coxsackieviruses Type 16 (CA16)	EU812514.1, EU262658.1, AY790926.1, AF177911.1, U05876.1