<table>
<thead>
<tr>
<th>Title</th>
<th>GLProbs: Aligning multiple sequences adaptively</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>YE, Y; Cheung, DWL; Wang, YD; Yiu, SM; Zhan, Q; Lam, TW; Ting, HF</td>
</tr>
<tr>
<td>Citation</td>
<td>IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2015, v. 12 n. 1, p. 67-78</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2015</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/204716">http://hdl.handle.net/10722/204716</a></td>
</tr>
<tr>
<td>Rights</td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
GLProbs: Aligning Multiple Sequences Adaptively

Yongtao Ye, David Wai-lok Cheung, Yadong Wang, Siu-Ming Yiu, Qing Zhang, Tak-Wah Lam, and Hing-Fung Ting

Abstract—This paper introduces a simple and effective approach to improve the accuracy of multiple sequence alignment. We use a natural measure to estimate the similarity of the input sequences, and based on this measure, we align the input sequences differently. For example, for inputs with high similarity, we consider the whole sequences and align them globally, while for those with moderately low similarity, we may ignore the flank regions and align them locally. To test the effectiveness of this approach, we have implemented a multiple sequence alignment tool called GLProbs and compared its performance with about one dozen leading alignment tools on three benchmark alignment databases, and GLProbs’s alignments have the best scores in almost all testings. We have also evaluated the practicability of the alignments of GLProbs by applying the tool to three biological applications, namely phylogenetic trees construction, protein secondary structure prediction and the detection of high risk members for cervical cancer in the HPV-E6 family, and the results are very encouraging.

Index Terms—Multiple sequence alignment, progressive alignment, hidden Markov model, phylogenetic analysis, secondary structure prediction

1 INTRODUCTION

The similarity of a set of biological sequences often implies functional similarity or suggests divergence from a common ancestor, and a common way to find out how similar the sequences are is to align them, i.e., to organize homologous positions across different sequences in columns. This method of multiple sequence alignment also helps biologists isolate relevant regions in the sequences, and identification of these regions is important to various analyses such as protein secondary structure prediction and phylogenetic tree construction. During the last two decades, there were a lot of software tools developed for multiple sequence alignment; however, all of them have their own weaknesses and perform poorly on some particular types of inputs. In general, when the similarity of the sequences falls below 25 percent, the accuracies of most multiple sequence alignment tools drop considerably.

This paper introduces a simple, but surprisingly effective approach for improving the quality of multiple sequence alignment. Note that there are two major ways to align multiple sequences: either globally or locally. If the sequences in a family are similar, there will be few ambiguous positions and we should take the whole sequences into consideration and align them globally. If the sequences are not similar, they may still contain similar patterns (e.g., motifs) over some local regions. In such case we should align them locally to find these similar local regions. Furthermore, sequences with different similarities have different characteristics and properties, and we may exploit them to help us align the sequences more accurately. This suggests the following adaptive approach for improving the alignment quality: we first decide the similarity of the sequences, and if they are similar, we focus on global alignments, otherwise we focus on local alignments.

This adaptive approach has a fundamental difficulty: we need to know the correct multiple sequence alignment in order to determine the similarity of the sequences. For example, the similarity scores given in the benchmark alignment databases BALiBASE [1], OXBench [2] and SABMark [3] are computed based on some given multiple sequence alignments, whose correctnesses have been verified by reliable biological methods. To overcome this difficulty, we suggest a natural measure to estimate the sequence similarity without the need to know the correct alignment. Our measure is based on the percent identity (PID) of a pair of sequences, which is the percentage of the number of homologous positions in the alignment of these two sequences. To estimate the similarity of a family of sequences, we align every pair of the sequences, compute their PIDs, and use the average of these PIDs to estimate the similarity of the sequences. We believe that for sequences with high similarity, most pairs should be unambiguous and have high PIDs, and thus the average PID will be high; otherwise the average PID will be low.

To study the applicability this average PID measure and the effectiveness of our approach, we have developed an adaptive alignment tool called GLProbs. Since the objective of our study is not to develop another alignment tool for the general public, we did not implement GLProbs from scratch. Instead, we developed GLProbs by modifying the open source codes of MSAProbs-0.9.7. To sketch our modification here, we note that MSAProbs [4] is a progressive alignment...
tool [5] based on the pair-Hidden Markov model (pair-HMM). Similar to other progressive alignment tools, MSAProbs aligns the sequences iteratively using pairwise alignments, and the most related sequences are aligned first and the more distant ones are aligned later. When aligning two sequences $x$ and $y$, the substitution scores used by MSAProbs are computed based on the posterior probability $Pr(x_i \sim y_j | x, y)$, which is the probability that positions $x_i$ and $y_j$ of $x$ and $y$ are matched under the condition that the pair-HMM has generated an alignment for $x$ and $y$ [6]. GLProbs modifies the method to compute these probabilities, and we will give the details in Section 2. Below, we highlight the main point.

Given a family of sequences as input, GLProbs first aligns every pair of the sequences and compute their PIDs, and the average PID will help GLProbs determine how to compute the posterior probabilities $Pr(x_i \sim y_j | x, y)$ as follows:

- $PID > 40\%$. It uses the standard three-state global pair-HMM ([7], [8]) to generate the posterior probabilities.
- $25\% < PID \leq 40\%$. It uses a modification of the local pair-HMM to generate the posterior probabilities (more details in Section 2.1).
- $PID \leq 25\%$. In this case, the sequences are so different that there may not have clear conserved local regions, and we are not sure what the right way is to align them. For this case, GLProbs resorts to consensus: it computes several posterior probabilities using different models, and then uses their root-mean-squares as the final probabilities.

We have compared GLProbs with many leading multiple sequence alignment tools including ten using the progressive method: MSAProbs [4], Probalign [8], ProbCons [6], ClustalW [9], ClustalΩ [10], COBALT [11], CONTRAAlign [12], MAFFT [13], MUSCLE [14], and T-Coffee [15], and three using the non-progressive method: Align-m [3], DIALIGN-PFAM [16], and PicXAA [17], using the benchmark alignment databases BAliBASE, OXBench and SABmark. GLProbs achieved the highest alignment accuracy and was statistically ranked as the best. In particular, GLProbs outperformed the other tools significantly for divergent sequences. For example, GLProbs got a 26 percent improvement of TC score over ClustalW for families of sequences in OXBench with similarity between 0-20 percent. We have also compared these tools on three biological applications, namely protein secondary structure prediction, phylogenetic analysis, and HPV-E6 protein analysis. Our results show that GLProbs had better performance as well. Details of these empirical comparisons will be given in Section 3. For verification of our results, GLProbs can be downloaded via the link http://glprobs.sourceforge.net, and the benchmark alignment databases can be accessed from http://www.drive5.com/bench [18].

We have also studied whether the adaptive approach is effective for the non-progressive method. We have modified the pair-HMM based non-progressive alignment tool PicXAA [17], just like what we have done to the progressive alignment tool MSAProbs. We call the resulting tool PicXAA-AD, which uses our adaptive approach to generate the posterior probabilities. We have compared the performance of PicXAA and PicXAA-AD using BAliBASE, OXBench and SABmark. The results given in Section 4 show that PicXAA-AD had significant improvements.

2 METHODS

In this section, we will first describe a modification of the local pair-HMM, which helps us compute the posterior probabilities when PID is between 25 and 40 percent. Then, we give the other details of GLProbs.

2.1 Coupling Local and Random Pair-HMMs

Local pair-HMM, as shown in Fig. 1a, has been used in some earlier alignment tools such as ProDA [19] and CONTRAAlign (local model) [12], but these tools may return poor results even for families with moderately low similarity. The main reason is that the model also gives scores to the leading and tailing flanking regions (i.e., the unaligned segments at the beginning and at the end of the local alignment). To make the alignment process focus on local conserved regions, we need to remove the “noises” of these flanking regions. To this end, GLProbs makes a modification of the model by applying the following standard technique of coupling local pair-HMM with a random pair-HMM, and using the log-odds ratios derived from the two models to determine the posterior probabilities (see [7], [20] for more details).

With reference to Fig. 1a, the three states $M$, $X$, $Y$ in the local pair-HMM are for generating an alignment. When visited, $M$ emits a pair of characters $(x_i, y_j)$, which means that $x_i$ is aligned to $y_j$ in the alignment to be generated, and $X$ and $Y$ emit (character, gap) pairs, which means that the characters emitted are unaligned. The states $RX_1$, $RY_1$, and $RX_2$, $RY_2$, which all emit (character, gap) pairs, are for producing the unaligned flanking regions. A path from the “Begin” state to the “End” state in the model determines an alignment of $x$ and $y$ by emitting the character pairs from the states along the path, and the probability that the alignment is generated by the model is given by the product of the emission probabilities and transmission probabilities.
involved. The random pair-HMM in Fig. 1b generates an alignment similarly; the states $RX$ and $RY$ emit (character, gap) pairs, and they generate two unaligned sequences.

To eliminate the noises of the flanking regions, we use log-odds ratio, which is defined to be the logarithm of the ratio between the probabilities for the alignment generated by the local pair-HMM and those by the random pair-HMM[7]. Note from Fig. 1a that the unaligned flanking regions generated by the local pair-HMM are in fact generated by two random pair-HMMs with states $RX_1$, $RY_1$, and with states $RX_2$, $RY_2$. Thus, when computing the log-odds ratio, the probabilities contributed by those terms coming from the unaligned flanking regions will be canceled out by those contributed by the corresponding terms coming from the random model.

We now explain how we determine the emission and transition probabilities. For the local pair-HMM, we use some common substitution matrices to determine the state emission probabilities, and use the unsupervised EM method to determine the state transitions probabilities. For the random pair-HMMs, we introduce an innovative method to determine the transition probability $\eta$ (see Fig. 1b). We observe that (i) if the sequences are less similar, the local conserved region will be shorter, and thus the unaligned flanking regions in their alignment will be longer, and (ii) if $\eta$ is small, it is more likely to postpone the local alignment (or the local alignment would end early) and thus the unaligned flanking regions will be longer. This motivates us to use different values of $\eta$ for different inputs as follows.

Note that GLProbs use the modified local pair-HMM, and thus needs the value of $\eta$, only when the PTT of the input sequences is in the range [0, 40 percent] (see Table 1). We partitioned this range into six different subranges, namely [0, 15 percent], [15, 20 percent], [20, 25 percent], [25, 30 percent], [30, 35 percent], [35, 40 percent], and for each of these subranges $R_t$, we prepare a data set $D_t$ of sequence families obtained from SABMark with similarities falling in this subrange. Then, we applied unsupervised EM on each $D_t$ to determine a value of $\eta$, and GLProbs will use this value when the input’s PTT falls in $R_t$.

### 2.2 Algorithm Design

GLProbs produces a multiple sequence alignment using the following six steps.

#### 2.2.1 Step 1: Determine the Model Used

For every pair of the input sequences, GLProbs finds their pairwise alignment and computes their percent identity, which is defined to be

\[
\text{PID} = \frac{N_{\text{Identity}}}{L_{\text{Alignment}}}
\]

where $N_{\text{Identity}}$ is the number of identities in the pairwise alignment, and $L_{\text{Alignment}}$ is the length of the alignment. As mentioned in Section 1, if PTT, the average PID, is greater than 40 percent, GLProbs uses global pair-HMM, and if PTT is in (25-40 percent), it uses the modified local pair-HMM to determine the posterior probabilities. We now explain how to handle the case when PTT is smaller than or equal to 25 percent.

For this case, GLProbs uses global pair-HMM and modified local pair-HMM to generate respectively the posterior probabilities $u = \Pr_{\text{global}}(x_i \sim y_j \mid x, y)$ and $v = \Pr_{\text{local}}(x_i \sim y_j \mid x, y)$. It also uses the double affine pair-HMM proposed in [21] to generate the probability $w = \Pr_{\text{aff}}(x_i \sim y_j \mid x, y)$.

Then, the posterior probability used in this case is given by

\[
\Pr(x_i \sim y_j \mid x, y) = \sqrt{u^2 + v^2 + w^2}/3.
\]

Table 1 summarizes our scheme.

#### 2.2.2 Step 2: Compute the Pairwise Distances

As in MSAProbs, GLProbs uses the posterior probabilities obtained in Step 1 to determine the substitution scores, and then applies a dynamic programming algorithm (without gap penalty) to compute, for every pair $x, y$ of the input sequences, the maximum expected accuracy

\[
E(x, y) = \max_a \left\{ \sum_{x_i \sim y_j \in a} \Pr(x_i \sim y_j \mid x, y) \right\}
\]

of $x$ and $y$, where the maximum is over all possible alignments $a$ of $x$ and $y$ (for details, see [7]). Then, it computes their pairwise distance $d_{xy}$, which is given by

\[
d_{xy} = 1 - \frac{E(x, y)}{\min\{|x|, |y|\}}.
\]

#### 2.2.3 Step 3: Construct a Guide Tree

Based on the pairwise distances computed in Step 2, GLProbs applies the greedy linear heuristic UPGMA [22] to construct a guide tree. During the construction, it uses the following definition of distance between two clusters of sequences: Suppose that the cluster $C_k$ is the union of the two disjoint clusters $C_i$ and $C_j$. Then for any other cluster $C_l$, the distance $d_{kl}$ between $C_k$ and $C_l$ is defined recursively to be

\[
d_{kl} = \frac{d_{kl}[C_i] + d_{kl}[C_j]}{|C_i| + |C_j|},
\]

where $|C_i|$ and $|C_j|$ are the number of sequences in $C_i$ and $C_j$, respectively.

#### 2.2.4 Step 4: Transform the Probabilities for Consistency

For every pair of input sequences $x$ and $y$, GLProbs makes the following adjustment of their posterior probabilities

\[1\text{. The double affine pair-HMM is similar to the three-state global model, except that there is an extra pair of gap states for long insertions and deletions.}]

### TABLE 1

<table>
<thead>
<tr>
<th>Category</th>
<th>PTT</th>
<th>Model</th>
<th>Posterior Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divergent</td>
<td>≤ 25%</td>
<td>Combination of global, modified local and double affine pair-HMMs</td>
<td>$\sqrt{u^2 + v^2 + w^2}/3$</td>
</tr>
<tr>
<td>Medium</td>
<td>25%-40%</td>
<td>Modified local pair-HMM</td>
<td>$u$</td>
</tr>
<tr>
<td>Similar</td>
<td>&gt; 40%</td>
<td>Global pair-HMM</td>
<td>$u$</td>
</tr>
</tbody>
</table>
Pr(\(x_i \sim y_j \mid x, y\)), trying to make them more consistent with the other sequences in final alignment (see [6] for more discussion).

Let \(P_{xz}\) and \(P_{zy}\) be the posterior probabilities matrices for sequence pairs \(x, z\) and \(z, y\), i.e., \(P_{xz}^{ij} = Pr(\(x_i \sim z_j \mid x, z\))\), and similarly for \(z, y\). Then the adjusted posterior probabilities matrix \(P_{0xy}\) for \(x, y\) is given as follows:

\[
P_{0xy}^{ij} = \frac{1}{|S|} \sum_{z \in S} P_{xz}^{ij} P_{zy}^{ij},
\]

where \(S\) is the set of input sequences. These adjusted posterior probabilities will be used to determine the substitution scores in Step 5.

Remark. Note that our transformation is somewhat different from that used in MSAProbs, the predecessor of GLProbs; while MSAProbs uses weighted consistency transformation, we use an unweighted one because we find that the weighted version is not significantly better.

2.2.5 Step 5: Obtain a Weighted Progressive Alignment

This step obtains a multiple sequence alignment of the input by performing weighted profile-profile alignments iteratively according to the order suggested by the guide tree constructed in Step 3. To avoid biased sampling of sequences, we first apply the method in [9] to calculate, for each sequence \(x\), a weight \(w_x\), which is dependent on \(x\)’s distance from the root of the guide tree. Then the substitution score matrix for the alignment of two profiles \(K\) and \(L\) is given by

\[
\text{Score}(K_i, L_j) = \sum_{x \in K_i, y \in L_j} w_x w_y \frac{Pr^i(x_i \sim y_j \mid x, y)}{\sum_{x \in K_i, y \in L_j} w_x w_y},
\]

where \(K_i\) and \(L_j\) are the \(i\)th and \(j\)th column of the profiles \(K\) and \(L\), and \(x_i\) and \(y_j\) are the characters of \(x\) and \(y\) in \(K_i\) and \(L_j\), respectively. Note that the probabilities \(Pr^i(x_i \sim y_j \mid x, y)\) are given by Step 4.

2.2.6 Step 6: Final Refinement

This step tries randomly to improve the accuracy of the alignment. It is executed only for input with \(\text{PID} < 70\) percent because we found from our empirical testings that this step does not help for sequences with high similarity.

During this step, we iteratively divide the multiple sequence alignment into two random groups (each sequence will be assigned to the two groups with equal probability), and we re-align them using the standard unweighted profile-profile procedure to see if we can make any improvement. Given an input family of \(N\) sequences, we stop the iterations when one of the following conditions is true:

- There are \(2N\) iterations in which we cannot find any improvement.
- We have iterated \(4N\) times.

GLProbs returns the best multiple sequence alignment it has seen during all the iterations as its final output.

3 EVALUATION RESULTS

We have conducted a comprehensive comparison of GLProbs with thirteen other leading MSA tools using standard benchmark alignment databases, and GLProbs had the best scores in all but one case. Furthermore, we have studied the practicability of GLProbs by using it in two biological analyses, namely phylogenetic analysis and protein secondary structure analysis. We have also used GLProbs to detect high risk members for cervical cancer in the HPV-E6 protein family. Our results showed rather convincingly that GLProbs has great potential for real applications.

3.1 Performance Comparison Using Benchmark Databases

We have compared GLProbs with 13 other leading multiple sequence alignment tools, including 10 based on the progressive method: MSAProbs 0.9.7, Probalign 1.4, CONTRAAlign(local) 2.01, ProbCons 1.12, MUSCLE 3.8.31, MAFFT 7.031, COBALT, ClustalW 2.1, Clustal Omega 1.1.0 and T-Coffee 9.03, and three on the non-progressive method: DIALIGN-PFAM, PicXAA(PHMM) 1.02, Alig-nm 2.3. Running with default parameters, these tools were used to align families of \(N\) sequences in three popular benchmark alignment databases, namely OXBench 1.3, SABmark 1.65 and BAliBASE 3.0 and we compared the
sum-of-pairs scores (SP) and total column scores (TC) of their alignments.

Table 2 shows the results for the OXBench 1.3 database. The table is divided into five categories according to the input families' similarities given in the databases. For example, the two columns under the category “ALL (0-100 percent)” show the average SP and average TC scores over all the input families in the database, while the two under “(0-20 percent)” are those for families with similarity between 0 and 20 percent. Note that GLProbs achieved the highest average SP and TC scores for all five categories. In particular, for the category (0-20 percent), which corresponds to families of divergent sequences, GLProbs had the most improvement. For example, as shown in

![Fig. 2. GLProbs's improvement on the average TC scores over MSAPros and ClustalW for OXBench.](image)

Table 3 shows the results for SABmark 1.65 and BAliBASE 3.0. Again, for these two benchmarks, GLProbs achieved the highest average SP and TC scores, except

![Fig. 3. Scalability of alignment quality.](image)

The P-values were computed by the Friedman rank test between GLProbs and other tools. Entries of MSAPros and Probalign are shown on OXbench [left]/SABmark [middle]/BAliBASE [right] respectively. And others are all smaller than 0.001 on three separate benchmarks. The difference is considered significant if P-values < 0.05.
TF105311 is the reference phylogenetic tree. The name of the tools stand for the phylogenetic trees computed through the MSAs constructed by themselves.

Fig. 4. Phylogenetic trees of TF105311.
for the (0-30 percent) category in BAliBASE, in which Probalign’s TC score is better.2

As in many previous studies on MSA tools (e.g., [4], [6], [8], [14]), we assessed the statistical significance of the differences in the SP and TC scores over each benchmark by performing the Friedman rank test [24] for GLProbs against each of the thirteen MSA tools we have evaluated. Table 4 shows the resulting P-values, and since all P-values in the table are less than 0.05, we may conclude that GLProbs has achieved a statistically significant improvement of accuracy over the MSA tools involved in the evaluation.

We note that the families of sequences used in our comparisons are not large. For example, the families in BAliBASE have each no more than 142 sequences, and those in OXBench no more than 122. To evaluate the performance of GLProbs for larger families, and to study how its accuracy degrades when the size of families increases, we used GLProbs to align large families of sequences, which were generated by the following method suggested in [10], [25]: We enlarged a particular domain family, namely, the Kunitz family [26] by adding more and more sequences from the benchmark database HomFam [10], which contains big families in the same domains. Fig. 3 shows GLProbs’ average SP and TC scores (restricted to the Kunitz domain, in which we have the reference alignments), and it also shows the scores obtained by six other MSA tools. Note that the scores of GLProbs are the highest regardless of the family sizes, and more importantly, the drops of GLProbs’ scores are rather mild when we increase the family sizes.

Remark. We have used GLProbs to align families of 4,000+ sequences, and it took around 8 hours to complete the alignments.

3.2 Biological Applications

To study the practicability of GLProbs, we have applied it in three biological analyses, and we summarise our findings as follows.

3.2.1 Phylogenetic Analysis

We have used GLProbs and four other MSA tools to align five families of protein sequences obtained from the TreeFam database [27]. Then, we passed the resulting alignments to the software tool MEGA5 [28] to reconstruct the phylogenetic trees of the families. Fig. 4 shows the reference phylogeny for one of the five families TF105311, as well as the corresponding phylogenetic trees reconstructed from the alignments obtained from the five MSA tools. We used the Robinson-Foulds (RF) distance [29] between the inferred trees and the references to assess the quality of the trees; the smaller the distances, the better the inferred trees. Table 5 summarizes the results, and we note that the phylogenetic trees derived from GLProbs’ alignments have the smallest distances in four of the five tests.

It should be noted that although RF distance is one of the commonly used metrics for assessing the quality of phylogenetic trees, it may lack discriminatory power under some circumstance [30], [31]. Thus, our study here only gave preliminary evidences that GLProbs may be better, and more comprehensive study is needed before we can make solid conclusion.

3.2.2 Secondary Structure Prediction

In this study, we applied GLProbs and four other alignment tools to predict the secondary structures of the two protein sequences Lipid Binding Protein (PDB ID: 1U27) [32] and Nuclear Receptor (PDB ID: 1LBD) [33], whose correct structures are widely agreed. For each of these two testing sequences, we first used the five MSA tools to obtain five multiple sequence alignments (of this sequence and those in the same family downloaded from Budd and Judge [34]), and then we passed these alignments to the tool JFPRED3 [35] to get five predictions of the secondary structures. As shown in Table 6, the predictions based on GLProbs’ alignments have the minimum number of wrongly predicted residues. For completeness, we also give details of the predictions in Fig. 5.

3.2.3 Analysis of the HPV-E6 Protein Family

Recently, the protein family HPV-E6 has been linked with cervical cancer [36], [37]. It was observed that the high risk members (for cervical cancer) in the HPV-E6 family contain a short stretch of four residues ETQ/V/L, which may play a key role in the binding of the PDZ domain, and the low risk HPV-E6 members do not have this stretch [38]. Moreover, it was recently proposed that for the high risk members, their ETQ/V/L stretches may be adjacent to an R residue [39].

We have used GLProbs to align HPV-E6 protein sequences sampled from the NCBI database (these sampled sequences have been studied in [40]) and the resulting alignment is given in Fig. 6. It is encouraging to note that GLProbs discovered a conserved region at the end of the alignment, and in that region, the high risk members contain the sketch of residues ETQ/V/L, and many of them are preceded by the residues R. On the other hand,

---

2 It is reported by Subramanian et al. [23] that “BAliBASE is heavily biased toward globally related protein families”, and we wonder whether this is the reason why Probalign, which is based on the global three-state pair-HMM design, has a better TC score.

---

TABLE 5

<table>
<thead>
<tr>
<th>TreeFamID</th>
<th>GLProbs MSAProbs MUSCLE T-Coffee ClustalW</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF101222(48)</td>
<td>0.69</td>
</tr>
<tr>
<td>TF105063(133)</td>
<td>0.82</td>
</tr>
<tr>
<td>TF105311(70)</td>
<td>0.64</td>
</tr>
<tr>
<td>TF105629(88)</td>
<td>0.62</td>
</tr>
<tr>
<td>TF105820(86)</td>
<td>0.72</td>
</tr>
</tbody>
</table>

The minimum distances in each row are shown in bold. The numbers of sequences in each query family are shown in the parentheses.

TABLE 6

<table>
<thead>
<tr>
<th>TreeFamID</th>
<th>GLProbs MSAProbs MUSCLE T-Coffee ClustalW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1U27</td>
<td>19</td>
</tr>
<tr>
<td>1LBD</td>
<td>39</td>
</tr>
</tbody>
</table>

The minimum values in the rows are shown in bold.
the low risk members do not have ETQ[V/L] in the region. We have also used other MSA tools to align these HPV-E6 protein sequences and none of them discovered such conserved region. For example, Fig. 7 shows the alignment given by MSAProbs.

From GLProbs' alignment given in Fig. 6 we have some new observations. First, we note that almost all of the low risk members of HPV-E6 has a residue P preceding the conserved region at the end (the column pointed by the green arrow in Fig. 6); we suspect that this P

---

**Fig. 5. Secondary structure predictions of 1U27 and 1LBD.**

<table>
<thead>
<tr>
<th>Domain</th>
<th>1U27</th>
<th>1LBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH</td>
<td>129</td>
<td>238</td>
</tr>
<tr>
<td>SR</td>
<td>100</td>
<td>148</td>
</tr>
</tbody>
</table>

Panel A gives the sample details: 100 PH-domain sequences with less than 25% identity and 148 SR-domain sequences with 25%-40% identity. Panel B and Panel C are the secondary structure predictions of 1U27 and 1LBD respectively. 1U27_seq and 1LBD_seq are the query protein sequences. 1U27 and 1LBD denote the reference structures. The names of aligners stand for the structures predicted through the MSAs constructed by them. The characters ‘H’, ‘T’ and ‘-’ represent extended, helical and other types of secondary structure respectively.
residue may be involved in lowering the affinity between HPV-E6 and the PDZ-binding domain. Second, the low risk members HPV70, HPV40 and HPV43 are significantly different from the other low risk members; they have residues ET[Q/L/M][V/C] in the conserved region while most other low risk members have gaps, and the three members do not have P residue in the column pointed by the green arrow. It was reported that HPV70 is likely an intermediate state between high-risk and low-risk members [40], and we suspect that HPV40 and HPV43 members may also have similar status. We admit that these are wild guesses, and we leave them to the biologists to decide whether they are worthy of further investigation.

4 DISCUSSIONS

4.1 On the Running Time of GLProbs

Tables 2 and 3 in Section 3.1 show that GLProbs is not among the fastest MSA tools. To improve its efficiency, we have benchmarked the running times of each of its steps in Fig. 8. The figure shows clearly that the bottleneck is Step 4, the consistency transformation step, which takes up about 80 percent of the processing time. (Note that Fig. 8 lists the running time of the steps from bottom to top.) This is not surprising because to refine the posterior probabilities, this step needs to execute $O(N^3)$ (sparse) matrix multiplications, where $N$ is the number of input sequences. We note that these matrix multiplications are independent to each other, and they can be computed in parallel. An obvious extension of our work is to implement GLProbs on GPUs or other multi-cores machines, and we strongly believe that its running time will be reduced substantially.

4.2 Effectiveness of Our Adaptive Approach on Non-Progressive Method

Section 3 shows rather convincingly that our adaptive approach can improve the accuracy of the pair-HMM based progressive alignment tool MSAProbs. It will not be
surprising that the approach can improve other similar tools using progressive methods. A more interesting question is whether it can improve tools that use non-progressive methods. To find out the answer, we have studied the pair-HMM based non-progressive MSA tool PicXAA [17], which greedily builds up the alignment by successively adding the most probable residue pair to alignments.

Similar to what we have done to MSAProbs, we modified PicXAA so that it chooses adaptively some proper HMM models to compute the posterior probabilities according to the average PID of the input family. We call the modified tool PicXAA-AD. From Table 7, which shows the average SP and TC scores obtained by PicXAA and PicXAA-AD for aligning families in SABmark, OXBench and BAliBASE, we note that PicXAA-AD has significantly better scores, and we can conclude that our adaptive approach is rather general; it can improve the accuracy of both pair-HMM based progressive and non-progressive alignment methods.

![Fig. 7. Multiple sequence alignment of HPV-E6 proteins by MSAProbs.](image)

![Fig. 8. The running time of different steps of GLProbs.](image)

<table>
<thead>
<tr>
<th>TABLE 7</th>
<th>Average TC and SP Scores of PicXAA and PicXAA-AD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SABmark</td>
</tr>
<tr>
<td>SP</td>
<td>TC</td>
</tr>
<tr>
<td>PicXAA</td>
<td>59.37</td>
</tr>
<tr>
<td>PicXAA-AD</td>
<td>60.42</td>
</tr>
</tbody>
</table>

PicXAA-AD is a modified version of PicXAA [17] with incorporating our proposed adaptive scheme in Section 2. Columns show the average sum of pairs scores (SP) and total column scores (TC) multiplied by 100. The best results in each column are shown in bold.
5 Future Work

This paper has shown that we can improve the quality of the alignments by choosing adaptively the models for computing the posterior probabilities. An interesting question is whether we would get further improvement if we can choose adaptively not only the models but also the alignment methods (i.e., the progressive and the non-progressive methods). To get some insight, we have used GLProbs and PicXAA-AD to align families with different similarities. From the results shown in Fig. 9, we note that the non-progressive tool PicXAA-AD achieves better TC scores for families of divergent sequences (0-20 percent identity), and the progressive tool GLProbs achieves better scores for families with higher similarity. We believe that the non-progressive tool has better results for families of divergent sequences because it adds confident regions to its solutions greedily and thus has better chance to preserve the conserved regions, while the progressive tool is likely to insert gaps into these regions during the dynamic programming step. On the other hand, the non-progressive tool suffers from its greedy approach when aligning families with higher similarity because it cannot recover incorrect decisions made earlier, while the dynamic programming step of the progressive tool allows more globally optimal solutions. Therefore, the strength and weakness of the progressive and non-progressive methods are complementary, and developing an adaptive approach that combine the two methods will be a fruitful future work for further improving the alignment quality.

For another possible future work, we note that the average percent identity PID that GLProbs uses to measure the similarity of the input sequences may not be accurate under some situations; for example, when there are a significant number of outliers among the input sequences. To make improvement, we may also consider the variance of the PIDs when estimating the similarity. A comprehensive study is needed to find out how much this additional measure may help under different situations.

Acknowledgments

Tak-Wah Lam is partially supported by the GRF Grant HKU-713512E (10612042). Siu-Ming Yiu is partially supported by the GRF Grant HKU719611E. Tak-wah Lam and Hing-fung Ting are joint corresponding authors.

References


Yongtao Ye received the BSc degree from Sun Yat-sen University in 2012 and is currently working toward the MPhil degree in computer science from the University of Hong Kong. His current research interests include algorithms design and bioinformatics.

David Wai-lok Cheung received the MSc and PhD degrees in computer science from Simon Fraser University, Canada, in 1985 and 1989, respectively. Since 1994, he has been a faculty member in the Department of Computer Science, The University of Hong Kong. His research interests include database, data mining, database security and privacy. He was the program committee chairman of PAKDD 2001, program cochair of PAKDD 2005, conference chair of PAKDD 2007 and 2011, conference cochair of CIKM 2009, and conference cochair of PAKDD 2011.

Yadong Wang received the MS degree from the Harbin Institute of Technology, Harbin, China, in 1989. He is currently a professor in the School of Computer Science and Technology, Harbin Institute of Technology. He has contributed to more than 60 papers in scientific journals. His research interests include bioinformatics, machine learning and knowledge engineering. In bioinformatics field, his researches focus on analysis of high-throughput biomedical data (microarray, next generation sequencing data, clinical data), Personal genomics, and translational bioinformatics.

Siu-Ming Yiu is currently an associate professor in the Department of Computer Science, the University of Hong Kong. His research interests include computational biology, bioinformatics, and computer security.

Qing Zhang received the BSc and MEng degrees from the Department of Mathematics and School of Computer Science and Technology, respectively, in Harbin Institute of Technology (HIT), China. He is currently working toward the PhD degree in the School of Computer Science and Technology, HIT. His research interest includes bioinformatics.

Tak-Wah Lam received the PhD degree in computer science from the University of Washington in 1988. He is currently a professor with the University of Hong Kong. His research is mainly related to algorithms and bioinformatics.

Hing-Fung Ting received the PhD degree in computer science from Princeton University in 1992. He is currently an associate professor at the University of Hong Kong. His research is focus on the design and analysis of algorithms and bioinformatics.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.