

## Feedback Algorithm and Web-Server for Protein Structure Alignment

ZHIYU ZHAO,<sup>1</sup> BIN FU,<sup>2</sup> FRANCISCO J. ALANIS,<sup>2</sup> and CHRISTOPHER M. SUMMA<sup>1</sup>

### ABSTRACT

We have developed a feedback algorithm for protein structure alignment that uses a series of phases to improve the global alignment between two protein backbones. The method implements a self-improving learning strategy by sending the output of one phase, the global alignment, to the next phase as an input. A web portal implementing this method has been constructed and is freely available for use at <http://fjpsa.cs.uno.edu/>. Based on hundreds of test cases, we compare our algorithm with three other, commonly used methods: CE, Dali, and SSM. Our results show that, in most cases, our algorithm outputs a larger number of aligned positions when the ( $C_\alpha$ ) *RMSD* is comparable. Also, in many cases where the number of aligned positions is larger or comparable to the other methods, our learning method is able to achieve a smaller ( $C_\alpha$ ) *RMSD* than the other methods tested.

**Key words:** protein structure alignment, protein structure comparison, rigid body transformation, root mean square deviation (*RMSD*).

### 1. INTRODUCTION

**P**ROTEIN STRUCTURE ALIGNMENT attempts to compare the structural similarity between protein backbone chains. A protein molecule can have one or more protein chains, and each chain consists of a series of amino acid residues connected by peptide bonds. Protein structural similarity can be used to infer evolutionary relationships or in classifying protein structures into more generalized groups. Typically, in protein structure comparison process, each protein chain is represented by an ordered set of three-dimensional (3-D) points where each point corresponds to an alpha-carbon ( $C_\alpha$ ) atom in an amino acid residue. To compare the structural similarity between these “backbone” representations, a protein structure alignment algorithm seeks an optimal transformation by which chains are matched as closely as possible. An alignment is characterized by (1) how many positions are matched, (2) where these positions are, and (3) how well they are matched. Once an alignment is determined, (1) and (2) are available. For (3), a transformation-based alignment algorithm usually calculates ( $C_\alpha$ ) *RMSD*, the root mean square distance between aligned positions.

The alignment problem is non-trivial—in fact, the problem of finding the optimal global alignment between protein structures has been shown to be NP-hard (Lathrop, 1994; Godzik, 1996). Therefore, there

---

<sup>1</sup>Department of Computer Science, University of New Orleans, New Orleans, Louisiana.

<sup>2</sup>Department of Computer Science, University of Texas–Pan American, Edinburg, Texas.

have been a number of protein structure alignment algorithms presented in the past years (Chew et al., 1999; Falicov, 1996; Fischer et al., 1992; Holm et al., 1993; Ilyin et al., 2004; Kolodny et al., 2004; Krissinel et al., 2004; Lessel et al., 1994; Madej et al., 1995; Ortiz et al., 2002; Shindyalow et al., 1999; Singh et al., 1997; Taylor et al., 1989; Taylor, 1999; Yuzhen et al., 2004; Ye et al., 2004; Yona et al., 2005; Zhang et al., 2005; Zhao et al., 2007), among them Dali (distance matrix based method) (Holm et al., 1993), SSM (secondary structure matching) (Krissinel, 2004), CE (the combinatorial extension method) (Shindyalow et al., 1999), and FATCAT (protein structure alignment based on flexible transformation) (Yuzhen et al., 2004) are commonly used. We have developed a feedback algorithm for pairwise protein structure alignment, and our web alignment tool is available for public access. Our algorithm is named SLIPSA, which stands for Self Learning and Improving Protein Structure Alignment. SLIPSA is self learning in that it has a feedback loop which sends the current alignment result back to its input in order to learn a better result in the next stage. In addition, SLIPSA accepts any reasonable upper-bound ( $C_\alpha$ ) *RMSD* value as one of the inputs, and outputs an alignment result with an ( $C_\alpha$ ) *RMSD* never greater than that value. Like CE, Dali, and SSM, the SLIPSA alignment method is based on rigid body transformation, as opposed to flexible transformation-based algorithms such as the one described in Yuzhen et al. (2004).

Our paper is organized as follows: Section 2 presents the SLIPSA algorithm; Section 3 describes the framework and procedures used in SLIPSA; Section 4 reports the experimental results of SLIPSA and compares it with some well-known algorithms such as CE, DaliLite (the pairwise version of Dali), and SSM, each of them having a public website; and Section 4.3 discusses the results and concludes the paper.

## 2. SLIPSA: AN ALGORITHM WITH FEEDBACK

SLIPSA can be traced to a prototype algorithm that we reported previously in Zhao et al. (2007), but the former has proceeded far beyond the latter in terms of maturity, stability, efficiency, and availability. The SLIPSA algorithm first searches all the locally similar sub-chain pairs from two protein backbone chains. Such sub-chain pairs are called local alignments. Next, consistent local alignments are grouped into global alignment candidates called “double-center stars,” and a currently optimal global alignment is chosen from all the candidates. Then this output is sent back to its own input in order to learn from itself. We call this a feedback. Such feedback is repeated to obtain improved results, until finally an optimal alignment (i.e., a result with as many as possible aligned  $C_\alpha$  pairs and an acceptable ( $C_\alpha$ ) *RMSD*). SLIPSA can also learn from other algorithms when they are available.

### 2.1. General algorithmic concepts

As shown in Figure 1, local alignments are discovered by checking the distance difference between corresponding  $C_\alpha$  pairs. A local alignment  $L = (i, j, l)$  is defined as the longest consecutive stretch of  $C_\alpha$  pairs starting from position  $i$  in protein backbone chain  $S = p_1 \cdots p_n$  and position  $j$  in backbone chain  $S' = q_1 \cdots q_m$  and having length  $l$ , such that  $|d(p_{i+u}, p_{i+v}) - d(q_{j+u}, q_{j+v})| \leq 2\epsilon$  for any  $0 \leq u, v \leq l-1$  and  $u \neq v$ , where  $d(p, q)$  is the Euclidean distance between two 3-D points  $p$  and  $q$ , and  $\epsilon$  is a small constant. A local alignment has to be long enough to make sense.

After local alignments are discovered, they are organized into groups. Ideally, only consistent local alignments should be added to the same group. Suppose there are two local alignments  $L_1 = (i_1, j_1, l_1)$  and  $L_2 = (i_2, j_2, l_2)$ , the point set  $P = \{p_{i_1}, \dots, p_{i_1+l_1-1}, p_{i_2}, \dots, p_{i_2+l_2-1}\}$  is all the aligned points in the first chain, including those in  $L_1$  and  $L_2$ , and  $Q = \{q_{j_1}, \dots, q_{j_1+l_1-1}, q_{j_2}, \dots, q_{j_2+l_2-1}\}$  is all the aligned points in the second chain, also including those in  $L_1$  and  $L_2$ . We say that local alignments  $L_1$  and  $L_2$  are consistent if, after applying a rigid body transformation to  $Q$ , the ( $C_\alpha$ ) *RMSD* between  $P$  and transformed  $Q$  is small enough. In other words, if we have a set of local alignments, we conclude that all these local alignments are consistent if all the local alignments share a common rigid body transformation which makes them consistent with each other. Therefore, a global alignment can be defined as such a set of consistent local alignments with a common transformation and an acceptable ( $C_\alpha$ ) *RMSD*.

The consistency relationship between local alignments can be represented as a graph. Given  $A_L = \{L_1, L_2, \dots, L_w\}$  where each  $L_u = (i_u, j_u, l_u)$  ( $1 \leq u \leq w$ ) is a local alignment. A graph  $G = (V, E)$  is defined accordingly, where each local alignment is a vertex of the graph,  $V = A_L$  is the vertex set and  $E$  is

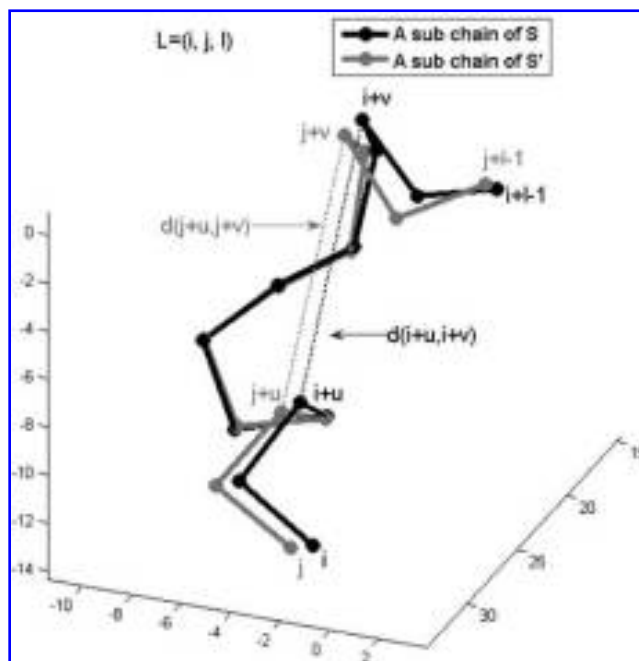


FIG. 1. Local alignment  $L = (i, j, l)$ .

the edge set. Edge  $e_{uv}, e_{vu} \in E$  if and only if  $L_u$  and  $L_v$  are consistent. With this representation, grouping mutually consistent local alignments is equivalent to finding cliques in a graph, which is an NP-complete problem. A possible simplification to this problem is to look for “stars” rather than cliques in a graph. A star is a set of vertices including a center and all the other vertices that are connected to the center vertex. Since any clique must be included in some star, for our particular problem this simplification will not miss useful vertices. Figure 2 shows a graph, two cliques, and an example star. There are six stars in the graph since  $|V| = 6$ . They are  $Star_1 = Star_2 = Star_6 = \{L_1, L_2, L_5, L_6\}$ ,  $Star_3 = Star_4 = \{L_3, L_4, L_5\}$  and  $Star_5 = \{L_1, L_2, L_3, L_4, L_5, L_6\}$ . A set of all the unique stars is  $Stars = \{Star_1, Star_3, Star_5\}$ . Note that each star is finally a set of local alignments and each local alignment is a set of  $C_\alpha$  pairs.

For each unique star, a corresponding global alignment candidate is calculated by deleting badly aligned  $C_\alpha$  pairs involved in that star. Then all the candidates are compared and the optimal one is chosen. An example global alignment between protein chains 1ATP:E and 1PHK is shown in Figure 3, where  $N_{mat}$  is the number of aligned  $C_\alpha$  pairs,  $(C_\alpha)$  RMSD is the root mean square distance between the aligned pairs, and the rigid body transformation used to align the two chains is  $T$  (the translation vector) and  $R$  (the rotation matrix).

The “star” approach has been used in the earlier version of this algorithm (Zhao et al., 2007), which has one center for each of its stars and shows some instability for aligning large proteins. We introduce

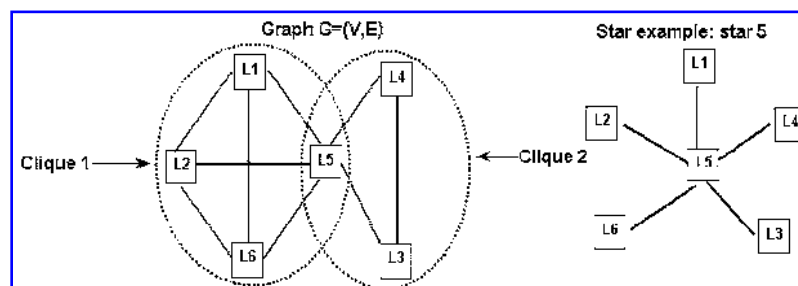


FIG. 2. An example star.

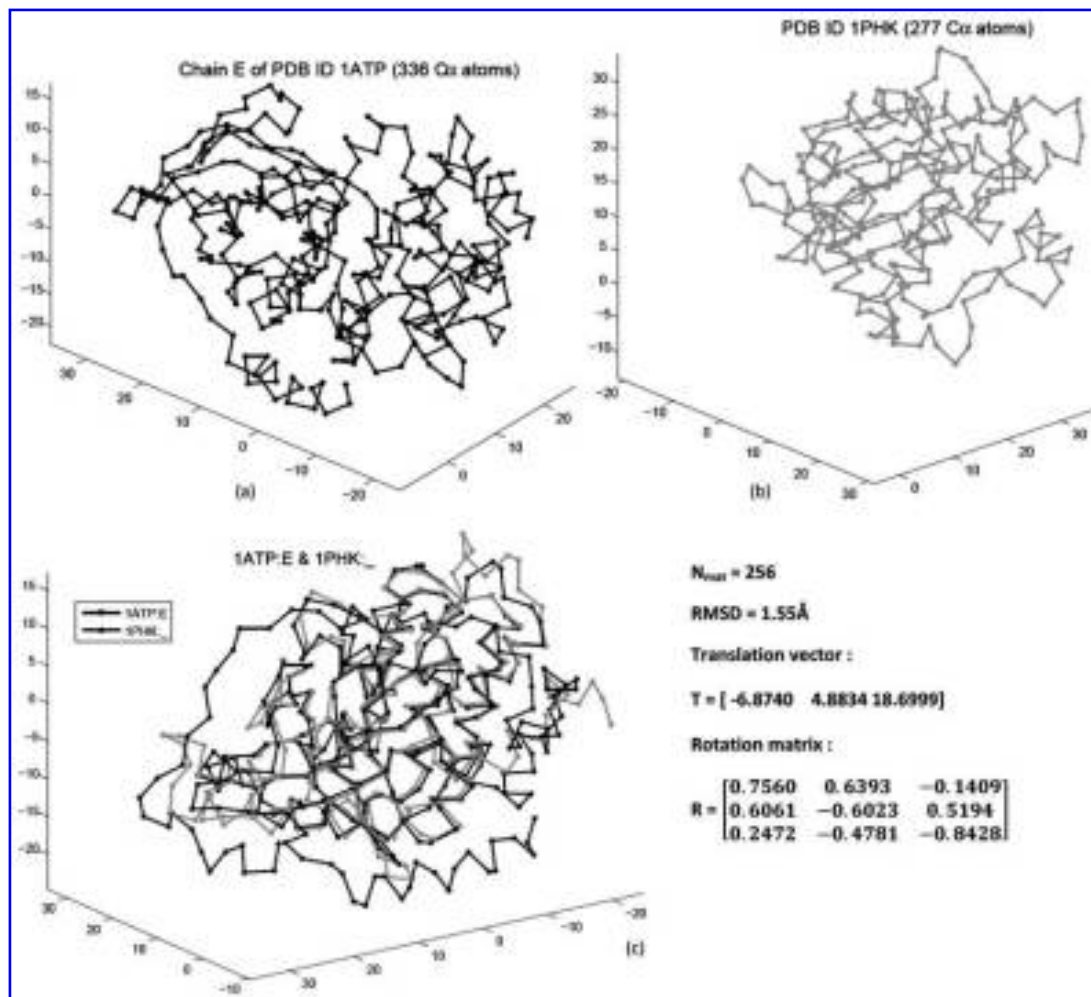


FIG. 3. A global alignment.

the double-center method to group the local alignments, and it is described in Section 2.2. This greatly improves the reliability of the algorithm. Another crucial new technical development of this paper is the learning strategy based on feedback, which is described in Section 2.3. The combination of two new methods greatly improves speed, reliability, and accuracy of the algorithm.

## 2.2. Introduction of double-center “stars”

The single-center star method is not flawless. It works well when the two protein chains match well or the chain diameters are small. However, we have found that it is less stable when the chains do not match very well or the chain diameters are large. This is caused by deleting badly matched  $C_{\alpha}$  pairs from each star, a method applied to obtain a global alignment candidate. When local alignments are grouped into an initial star, there may exist point pairs which do not match well. An initial transformation is calculated and the worst matched pair based on that transformation is first deleted, then the transformation is recalculated to select the second worst pair. This process is repeated. In this way, the well-matched pairs survive and the ( $C_{\alpha}$ )  $RMSD$  becomes smaller and smaller, until an acceptable ( $C_{\alpha}$ )  $RMSD$  is achieved. The effect of deleting bad point pairs relies on a good initial transformation, which in turn depends on the star center selection. With a single star center, the initial transformation has great freedom to move and rotate in the point pair deletion process, thus the deletion may go along a more unpredictable way. This is more obvious when the local alignments are relatively short, which usually happens when the chains do not match very

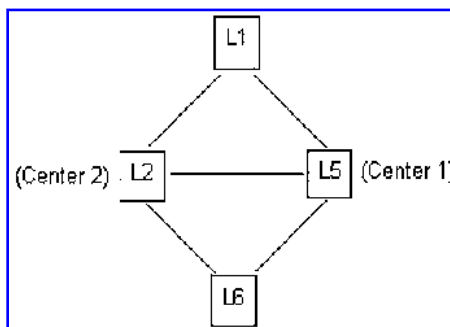


FIG. 4. A double-center star.

well or the chain diameters are large. Based on this observation, we consider grouping local alignments into double-center “stars.”

A “double-center star” is, as suggested by its name, a “star” with two centers. Each single-center star can be extended to a corresponding double-center star, while the latter is much more stable. In a single-center star, each local alignment consistent with the center is added to the star, while in a double-center star, a local alignment can be added only when it is consistent with both centers. The first center of a double-center star is exactly the one in a single-center star, and the second center is selected from that star. The selection of the second center satisfies the following conditions: (1) it is consistent with the first center; (2) it is long enough to make sense; (3) it is as far as possible from the first center. Figure 4 illustrates a double-center star corresponding to star 5 in Figure 2, supposing  $L_2$  is the second center.

Each local alignment in a single-center star is consistent with the center; however, this does not automatically guarantee that all the local alignments in the star are consistent. The consistency relationship is not necessarily transitive. To reduce the probability of adding inconsistent local alignments to the star, a double-center star accepts local alignments in a more prudent way. It rejects the local alignments originally surviving in the single-center star on a weak basis; therefore, local alignments in the double-center star are more likely to be those very good ones. To some extent, the presence of the second center has the effect of “extending” the local alignment in the first center. With such a long “local alignment” as the center, the star will be more stable because points in it have much less freedom to move or rotate. From another aspect, with this improvement, the extent to which the initial transformation will change along with the deletion of bad point pairs is reduced significantly—the initial transformation will be closer to the final one, and thus the deletion will cause less unpredictability. Furthermore, the filtering of unpromising local alignments reduces their negative contribution to the overall transformation (as well as the number of point pairs involved in the initial star), speeding up the deletion process and resulting in a faster and better global alignment.

### 2.3. Development of learning ability

*2.3.1. Self-learning.* As we have mentioned, good star centers produce promising stars and have a greater probability of generating good global alignments. However, thus far, the selection of star centers has been naive: any local alignment with sufficient length can be the center of a star. The double-center method helps remove some unpromising local alignments from a star when the first center is determined, but it contributes nothing to the selection of the first center. If the first center of a star can be selected intelligently rather than by arbitrarily picking up a local alignment, then the star may yield a better global alignment. This intelligence may be difficult to achieve without any a priori knowledge on the global structural similarity between the two chains. However, when such knowledge is available, it is possible to improve the alignment by way of a self-learning strategy.

Once a currently optimal global alignment is output, we are able to know approximately where the aligned positions are. We organize the consecutively aligned point pairs into groups, and each group of consecutive point pairs is called a global alignment segment. A global alignment segment looks exactly like a local alignment, while as a part of a good global alignment, it should be a good “local alignment.”

Here local alignment is quoted because global alignment segments are not output of the local alignment phase, although there is no substantial difference between both definitions. To take advantage of these good global alignment segments, we apply a feedback mechanism to teach our alignment algorithm how to improve itself. The self-learning is implemented via the iterative utilization of its own output. When a global alignment is ready, consecutive alignment segments are extracted, then each segment is used as a new star center and local alignments consistent with the center are added to its group. This global alignment phase is repeated with a few new stars obtained from the currently optimal global alignment, until the alignment output converges (i.e., no changes are found between two iterations).

*2.3.2. Learning from others.* When a global alignment from another algorithm is available, the global alignment segments in that result can work as initial star centers. These centers are likely to be better than our own local alignments because they are from an optimal alignment result obtained from another algorithm. With these centers, our global alignment searching starts from a very good jumping-off point, therefore it is possible to output a result better than without learning. Learning from other algorithms may be more effective in the cases our algorithm performs worse than others. When it performs better than other algorithms even without learning, this learning may be less necessary; however, it is never harmful, because if it results in a worse global alignment, its results can simply be disregarded. Therefore the combination of self-learning and learning-from-others will never output an alignment worse than the one of another algorithm. In the worst case it outputs nothing different. For this reason, our algorithm can also be used to improve the result of any other algorithm. We call this a refinement to that algorithm.

### 3. THE FORMAL DESCRIPTION OF SLIPSA

We give the formal description of SLIPSA. Combining the double-center star, the self-learning and the learning-from-others methods which use feedback, we greatly improve our earlier work (Zhao et al., 2007) and have found interesting results when comparing SLIPSA with some other algorithms. The SLIPSA framework is shown in Figure 5. This system takes six parameters: protein chains  $S$  and  $S'$ ,  $RMSD_{max}$  (a user-specified maximum ( $C_{\alpha}$ )  $RMSD$ ), distance constant  $\epsilon$ , minimum local alignment length  $l_{min}$ , and an optional external global alignment  $A_{G\_Ext}$ . Parameters  $S$  and  $S'$  are determined by the user,  $RMSD_{max}$  is either determined by the user or obtained from another algorithm,  $\epsilon$  and  $l_{min}$  are selected empirically according to the user input, and  $A_{G\_Ext}$  is either empty or also obtained from the external algorithm. The system outputs an optimal global alignment result consisting of  $A_G$  (a set of global alignment segments),  $(C_{\alpha}) RMSD$  (a value not greater than  $RMSD_{max}$ ) and  $F$  (a rigid body transformation corresponding to the final global alignment). The following sub-sections describe the details of the SLIPSA algorithm.

#### 3.1. Getting local alignments

The calculation of local alignments has been reviewed in Section 2.1. The procedure used to get local alignments can be from either Zhao et al. (2007) or other related papers (Yu et al., 2004). The procedure body is omitted.

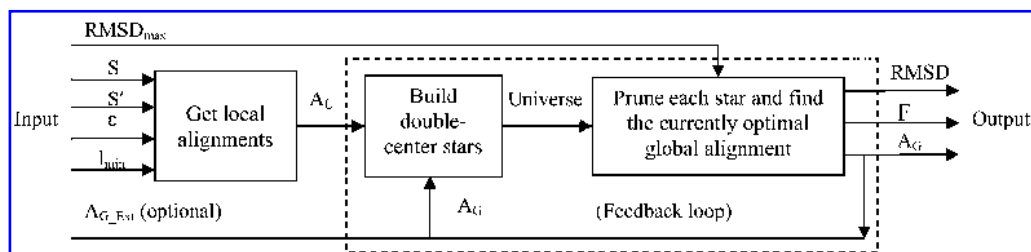


FIG. 5. The SLIPSA framework.

**Get-Local-Alignments**( $S, S', \epsilon, l_{\min}$ )

*Input:* protein backbone chains  $S = p_1 \cdots p_n$ ,  $S' = q_1 \cdots q_m$ , distance constant  $\epsilon$  and minimum local alignment length  $l_{\min}$ , where each  $p_i$  or  $q_i$  is a 3-D point corresponding to a  $C_\alpha$  atom in a protein backbone.

*Output:*  $A_L = \{L_1, L_2, \dots, L_w\}$ , a set containing all the local alignments of length  $\geq l_{\min}$  between  $S$  and  $S'$ .

3.2. *Building up stars from local alignments*

The improved procedure outputs double-center stars. The input is star centers from a set of global alignment segments, or from a local alignment set when the former is empty. The non-center nodes in a star are still chosen from the local alignment set.

**Build-Double-Center-Stars**( $A_L, A_G$ )

*Input:*  $A_L = \{L_1, L_2, \dots, L_w\}$  and  $A_G = \{L_{1'}, L_{2'}, \dots, L_{w'}\}$ , where  $A_L$  is a set of local alignments and  $A_G$  is a set of global alignment segments.

*Output:*  $Universe = \{Star_1, Star_2, \dots, Star_k\}$ , a set of all the unique double-center stars.

**begin**

$Universe \leftarrow \{\}$  (the empty set);

**if** ( $A_G = \{\}$ ) **then**  $A \leftarrow A_L$ ; **otherwise**  $A \leftarrow A_G$ ;

**for** (each local alignment  $L_u$  in  $A$ )

    find  $L_{u'}$ , the second center based on  $L_u$ , in  $A$ ;

$Star_u \leftarrow \{L_u, L_{u'}\}$ ;

**for** (each local alignment  $L_v$  in  $A_L$ )

**if** ( $L_v$  is consistent with both  $L_u$  and  $L_{u'}$ ) **then**  $Star_u \leftarrow Star_u \cup \{L_v\}$ ;

**end for**

**if** ( $Star_u \notin Universe$ ) **then**  $Universe \leftarrow Universe \cup \{Star_u\}$ ;

**end for**

**return**  $Universe$ ;

**end**3.3. *Finding a global alignment from the stars*

In each iteration of our algorithm, a global alignment is output and used as an input of the next iteration. We describe how to prune the set of aligned pairs in a star and obtain the global alignment which has an ( $C_\alpha$ ) *RMSD* not greater than that specified by the user. We refine a similar idea that is used in our another algorithm (Zhao et al., 2007) which does not use feedback.

**Prune-One-Star**( $Star, RMSD_{\max}$ )

*Input:* a *Star* and  $RMSD_{\max}$  (a user specified maximum *RMSD*).

*Output:* ( $A_S, RMSD_S, F_S, l_S$ ), where  $A_S = \{L_{1''}, L_{2''}, \dots, L_{w''}\}$  is a set of global alignment segments which share a common transformation  $F_S$  with  $RMSD_S \leq RMSD_{\max}$ , and  $l_S$  is the number of aligned point pairs in  $A_S$ .

**begin**

$A_S \leftarrow Star$ ;

$l_S \leftarrow$  the number of point pairs involved in  $A_S$ ;

calculate transformation  $F_S$  and  $RMSD_S$  for all the point pairs involved in  $A_S$ ;

**while** ( $RMSD_S > RMSD_{\max}$ )

    delete point pair ( $p, q$ ) with the largest  $d(p, F_S(q))$  in  $A_S$ ;

$l_S \leftarrow l_S - 1$ ;

```

        recalculate transformation  $F_S$  and  $RMSD_S$  for all the point pairs involved in  $A_S$ ;
    end while
    return  $(A_S, RMSD_S, F_S, l_S)$ ;
end

```

---

In the following function Find-Global-Alignment(), we apply the Prune-One-Star() procedure to each of the stars in the universe which is built from Build-Double-Center-Stars(). The alignment that contains the largest number of aligned pairs will be returned.

**Find-Global-Alignment**( $Universe, RMSD_{max}$ )

*Input:*  $Universe = \{Star_1, Star_2, \dots, Star_k\}$  and  $RMSD_{max}$  (a user specified maximum  $RMSD$ ).

*Output:*  $(A_G, RMSD, F)$ , where  $A_G = \{L_{1'}, L_{2'}, \dots, L_{w'}\}$  is a set of global alignment segments which share a common transformation  $F$  with  $RMSD \leq RMSD_{max}$ .

---

**begin**

sort  $Universe$  by a descending order of the number of 3-D point pairs involved in each star;

$l_{max} \leftarrow 0$ ;

**for** (each  $Star_u$  in  $Universe$ )

$(A_S, RMSD_S, F_S, l_S) \leftarrow \text{Prune-One-Star}(Star_u, RMSD_{max})$ ;

**if** ( $l_S > l_{max}$ ) **then**  $A_G \leftarrow A_S$ ;  $RMSD \leftarrow RMSD_S$ ;  $F \leftarrow F_S$ ;  $l_{max} \leftarrow l_S$ ;

**end for**

**return**  $(A_G, RMSD, F)$ ;

**end**

---

### 3.4. The feedback procedure

This is the main procedure of SLIPSA. It calls Get-Local-Alignments in the first step, then Build-Double-Center-Stars and Find-Global-Alignment are called repeatedly. A global alignment output by the current iteration serves as the input of the next iteration. The procedure terminates when the global alignment ceases to change (i.e., converges).

SLIPSA( $S, S', \epsilon, l_{min}, RMSD_{max}, A_{G\_Ext}$ )

*Input:*  $S, S', \epsilon, l_{min}, RMSD_{max}$  and  $A_{G\_Ext}$ , where  $A_{G\_Ext}$  can be either empty or a set of global alignment segments obtained from an external algorithm.

*Output:*  $(A_G, RMSD, F)$ .

---

**begin**

$A_L \leftarrow \text{Get-Local-Alignments}(S, S', \epsilon, l_{min})$ ;

$A_G \leftarrow A_{G\_Ext}$ ;

**do**

$A'_G \leftarrow A_G$ ;

$Universe \leftarrow \text{Build-Double-Center-Stars}(A_L, A'_G)$ ;

$(A_G, RMSD, F) \leftarrow \text{Find-Global-Alignment}(Universe, RMSD_{max})$ ;

**while** ( $A_G \neq A'_G$ );

**return**  $(A_G, RMSD, F)$ ;

**end**

---

When no external alignment is available, procedure SLIPSA is called by way of SLIPSA( $S, S', \epsilon, l_{min}, RMSD_{max}, \{\}$ ). When it is available, SLIPSA can be called as SLIPSA( $S, S', \epsilon, l_{min}, RMSD_{max}, A_{G\_Ext}$ ). We call this a refinement to external alignment  $A_{G\_Ext}$ . To independently test the performance of our algorithm, none of the experiments reported in section 4 uses any external alignment as our input.



### 4. EXPERIMENTAL ENVIRONMENT AND RESULTS

#### 4.1. Our web alignment tool

We have developed a web alignment tool based on the SLIPSA algorithm. The website is available for public access at <http://fpsa.cs.uno.edu/>. It is not only a SLIPSA alignment tool but also an alignment comparison tool between SLIPSA and DaliLite, CE, and SSM—some commonly used protein structure alignment algorithms with public websites.

The data used for protein alignment are the PDB files downloaded from the RCSB Protein Data Bank. The files were moved to the Worldwide Protein Data Bank (wwPDB) by the time we wrote this paper. As of January 2008, there were over 48,000 protein structures with over 90,000 chains discovered.

Our website is built on an Intel dual-Xeon 3-GHz PC server with 3-GB memory. The web development tools we used included Apache HTTP server with PHP support, ActivePerl, and MySQL database server. The SLIPSA algorithm is written in MATLAB. For the rigid body transformation method that we used in SLIPSA, see Umeyama (1991) and Eggert et al. (1995).

The work flow of our website is shown in Figure 6. Besides a maximum value for  $(C_\alpha)$  RMSD, it accepts either PDB IDs or user uploaded PDB files as input. It is optional to compare SLIPSA with DaliLite, CE, or SSM. When a comparing option is chosen, our tool automatically submits alignment request to and retrieves result from DaliLite, CE, or SSM website, and performs SLIPSA alignment according to the retrieved  $(C_\alpha)$  RMSD value. The website outputs the following alignment results. Beyond the first result listed, all others are optional depending on the user choices. Note that SLIPSA outputs  $A_G$  (a set of global alignment segments),  $(C_\alpha)$  RMSD and  $F$  (a rigid body transformation).

- (1)  $(A_G, RMSD, F)_{SLIPSA}$ : the SLIPSA result with a user specified  $RMSD_{max}$ .
- (2)  $(A_G, RMSD)_{DaliLite}$ : the DaliLite result retrieved automatically from its website.
- (3)  $(A_G, RMSD, F)_{DaliLite\_Comp}$ : the SLIPSA result with an RMSD retrieved from DaliLite website as input. This result is used to compare SLIPSA with DaliLite.
- (4)  $(A_G, RMSD)_{CE}$ : the CE result retrieved automatically from its website.
- (5)  $(A_G, RMSD, F)_{CE\_Comp}$ : the result used to compare SLIPSA with CE.
- (6)  $(A_G, RMSD)_{SSM}$ : the SSM result retrieved automatically from its website.
- (7)  $(A_G, RMSD, F)_{SSM\_Comp}$ : the result used to compare SLIPSA with SSM.

#### 4.2. Experimental results

We have collected 224 alignment cases to test the performance of our algorithm; the test cases were originally proposed by various papers for various testing purposes (Table 1). The alignment cases include nos. 1–20 (see Table III in Shindyalow et al., 1999), nos. 21–88 (see Table I in Fischer, 1996), no. 89 (see Tables I and II in Shindyalow et al., 1999), nos. 90–92 (see supplement to Table III in Shindyalow et al., 1999), no. 93 (see Fig. 5 in Shindyalow et al., 1999), nos. 94–101 (see Table IV in Shindyalow et al., 1999), nos. 102–111 (see Table V in Shindyalow et al., 1999), nos. 112–120 (see supplement to Table V in Shindyalow et al., 1999), nos. 121–124 (see Table VII in Shindyalow et al., 1999), nos. 125–143 (see

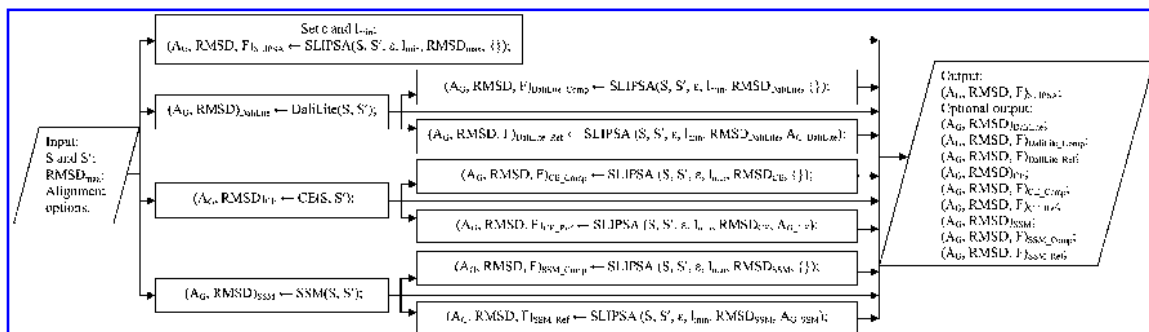


FIG. 6. The web alignment work flow.

TABLE 1. PDB CHAINS OF THE 224 TEST CASES

No.	Chain 1	Chain 2	No.	Chain 1	Chain 2	No.	Chain 1	Chain 2	No.	Chain 1	Chain 2
1	1ATP:E	1APM:E	57	2AK3:A	1GKY:_	113	1HLE:B	2ACH:B	169	1MBC:_	1PHN:A
2	1ATP:E	1CDK:A	58	1ATN:A	1ATR:_	114	1BBT:4	1TMF:4	170	1MBC:_	1CPC:A
3	1ATP:E	1YDR:E	59	1ARB:_	5PTP:_	115	1AIE:_	2FUA:_	171	1MBC:_	1LIA:A
4	1ATP:E	1CTP:E	60	2PIA:_	1FNB:_	116	1CPT:_	1FCT:_	172	1MBC:_	1CPC:B
5	1ATP:E	1PHK:_	61	3RUB:L	6XIA:_	117	1LBD:_	1PSM:_	173	1MBC:_	1QGW:C
6	1ATP:E	1KOA:_	62	2SAR:A	9RNT:_	118	4ICB:_	1CTD:A	174	1MBC:_	1LIA:B
7	1ATP:E	1KOB:A	63	3CD4:_	2RHE:_	119	2SEC:I	1EGP:A	175	1MBC:_	1COL:A
8	1ATP:E	1AD5:A	64	1AEP:_	256B:A	120	1SCE:A	1PUC:_	176	1MBC:_	2CP4:_
9	1ATP:E	1CKI:A	65	2MNR:_	4ENL:_	121	1BPI:_	1BUN:B	177	1MBC:_	1EUM:A
10	1ATP:E	1CSN:_	66	1LTS:D	1BOV:A	122	1BPI:_	5EBX:_	178	1MBC:_	1FPO:A
11	1ATP:E	1ERK:_	67	2GBP:_	2LIV:_	123	1WAJ:_	1NOY:A	179	1MBC:_	1OXA:_
12	1ATP:E	1FIN:A	68	1BBT:1	2PLV:1	124	1WAJ:_	1XWL:_	180	1MBC:_	1LE2:_
13	1ATP:E	1GOL:_	69	2MTA:C	1YCC:_	125	1ACX:_	1COB:B	181	1MBC:_	2FHA:_
14	1ATP:E	1JST:A	70	1TAH:A	1TCA:_	126	1ACX:_	1TMF:A	182	1MBC:_	1NFN:_
15	1ATP:E	1IRK:_	71	1RCB:_	2GMF:A	127	1PTS:A	1MUP:_	183	1MBC:_	1GRJ:_
16	1ATP:E	1FGK:A	72	1SAC:A	2AYH:_	128	2GBL:_	1UBQ:_	184	3TRX:_	4TRX:_
17	1ATP:E	1FMK:_	73	1DSB:A	2TRX:A	129	2GB1:_	4FXC:_	185	3TRX:_	1MDI:A
18	1ATP:E	1WFC:_	74	1STF:I	1MOL:A	130	1UBQ:_	4FXC:_	186	3TRX:_	1AIU:_
19	1ATP:E	1KNY:A	75	2AFN:A	1AOZ:A	131	1PLC:_	2RHE:_	187	3TRX:_	1ERV:_
20	1ATP:E	1TIG:_	76	1FXI:A	1UBQ:_	132	1PLC:_	1ACX:_	188	3TRX:_	1F9M:B
21	1MDC:_	1IFC:_	77	1BGE:B	2GMF:A	133	1ACX:_	1RBE:_	189	3TRX:_	1F9M:A
22	1NPX:_	3GRS:_	78	3HLA:B	2RHE:_	134	1ABA:_	1TRS:_	190	3TRX:_	1GH2:A
23	1ONC:_	7RSA:_	79	3CHY:_	2FOX:_	135	1ABA:_	1DSB:A	191	3TRX:_	1EP7:A
24	1OSA:_	4CPV:_	80	2AZA:A	1PAZ:_	136	1ABA:_	1PBF:_	192	3TRX:_	1EP7:B
25	1PFC:_	3HLA:B	81	1CEW:I	1MOL:A	137	1MJC:_	5TSS:A	193	3TRX:_	1FAA:A
26	2CMD:_	6LDH:_	82	1CID:_	2RHE:_	138	1PGB:_	5TSS:A	194	3TRX:_	1TOF:_
27	2PNA:_	1SHA:A	83	1CRL:_	1EDE:_	139	2TMV:P	256B:A	195	3TRX:_	2TIR:_
28	1BBH:A	2CCY:A	84	2SIM:_	1NSB:A	140	1TNF:A	1BMV:1	196	3TRX:_	1THX:_
29	1C2R:A	1YCC:_	85	1TEN:_	3HHR:B	141	1UBQ:_	1FRD:_	197	3TRX:_	1FB6:B
30	1CHR:A	2MNR:_	86	1TIE:_	4FGF:_	142	2RSL:C	3CHY:_	198	3TRX:_	1QUW:A
31	1DXT:B	1HBG:_	87	2SNV:_	5PTP:_	143	3CHY:_	1RCF:_	199	3TRX:_	1KTE:_
32	2FBJ:L	8FAB:B	88	1GPL:_	2TRX:A	144	1MBC:_	5MBN:_	200	3TRX:_	1JHB:_
33	1GKY:_	3ADK:_	89	1CPC:L	1COL:A	145	1MBC:_	1MBN:_	201	3TRX:_	3GRX:_
34	1HIP:_	2HIP:A	90	1KNY:A	1TIG:_	146	1MBC:_	1MYH:A	202	3TRX:_	1H75:A
35	2SAS:_	2SCP:A	91	1MAE:H	2BBK:J	147	1MBC:_	1HDS:B	203	3TRX:_	1EGO:_
36	1FCL:A	2FB4:H	92	2MHR:_	2BRD:_	148	1MBC:_	2DHB:A	204	3TRX:_	1ILO:A
37	2HPD:A	2CPP:_	93	1HCL:_	1JSU:A	149	1MBC:_	1MBA:_	205	3TRX:_	1ABA:_
38	1ABA:_	1EGO:_	94	2ASR:_	1OCC:C	150	1MBC:_	1DM1:A	206	3TRX:_	1FO5:A
39	1EAF:_	4CLA:_	95	2ASR:_	1MMO:D	151	1MBC:_	1HLM:_	207	3TRX:_	1MEK:_
40	2SGA:_	5PTP:_	96	2ASR:_	2BRD:_	152	1MBC:_	2LHB:_	208	3TRX:_	1A8Y:_
41	2HHM:A	1FBP:A	97	256B:A	1AEP:_	153	1MBC:_	2FAL:_	209	3TRX:_	1E2Y:A
42	1AAJ:_	1PAZ:_	98	256B:A	1CIY:_	154	1MBC:_	1HBG:_	210	3TRX:_	1E2Y:C
43	5FD1:_	1QZ:A	99	256B:A	1AGS:A	155	1MBC:_	1ITH:A	211	3TRX:_	1QMV:A
44	1ISU:A	2HIP:A	100	2ASR:_	1LKI:_	156	1MBC:_	1FLP:_	212	3TRX:_	1BJX:_
45	1GAL:_	3COX:_	101	2ASR:_	1FPS:_	157	1MBC:_	1ECA:_	213	3TRX:_	1GPI:B
46	1CAU:B	1CAU:A	102	1LIS:_	1CIY:_	158	1MBC:_	2HBG:_	214	3TRX:_	1QQ2:A
47	1HOM:_	1LFB:_	103	1CFP:A	4ICB:_	159	1MBC:_	1ASH:_	215	3TRX:_	1EZK:A
48	1TLK:_	2RHE:_	104	1RPA:_	1HIW:A	160	1MBC:_	1HBI:B	216	3TRX:_	1EWX:A
49	2OMF:_	2POR:_	105	1HYP:_	1MZM:_	161	1MBC:_	1GDI:_	217	3TRX:_	1QK8:A
50	1LGA:A	2CYP:_	106	1CLC:_	1HOE:_	162	1MBC:_	1HLB:_	218	3TRX:_	1FG4:A
51	1MIO:C	2MIN:B	107	1UTG:_	1NOX:_	163	1MBC:_	1LH2:_	219	3TRX:_	1A8L:_
52	4SBV:A	2TBV:A	108	1FAR:_	1PTQ:_	164	1MBC:_	1H97:A	220	3TRX:_	1FG4:B
53	8I1B:_	4FGF:_	109	1KUM:_	1TUL:_	165	1MBC:_	1DLY:A	221	3TRX:_	1FVK:A
54	1HRH:A	1RNH:_	110	1PYI:A	1PYC:_	166	1MBC:_	1IDR:A	222	3TRX:_	1F37:A
55	1MUP:_	1RBP:_	111	1VIH:_	1PYT:A	167	1MBC:_	1DLW:A	223	3TRX:_	1F37:B
56	1CPC:L	1COL:A	112	1LYP:_	1OLG:A	168	1MBC:_	1ALL:A	224	3TRX:_	1GHH:A

Table 1 in Ortiz et al., 2002), nos. 144–183 (see Table 1 in Ye et al., 2004). and nos. 184–224 (see Table 2 in Ye et al., 2004).

Based on this test set, we compare SLIPSA with DaliLite, CE, and SSM in terms of  $N_{mat}$  (the number of aligned positions) and  $(C_\alpha)$  *RMSD*. The detailed results are listed in Tables 4–9. In each test case, SLIPSA outputs an  $(C_\alpha)$  *RMSD* not greater than that of DaliLite, CE, or SSM. If  $N_{mat}$  of SLIPSA is larger than  $N_{mat}$  of DaliLite, CE, or SSM, we call it an  $N_{mat}$  increment. Similarly, if the  $(C_\alpha)$  *RMSD* of SLIPSA is smaller than the  $(C_\alpha)$  *RMSD* of DaliLite, CE, or SSM, we call it a  $(C_\alpha)$  *RMSD* decrement. For the convenience of illustration, the results are sorted in an ascending order of the  $N_{mat}$  increment rate. The  $N_{mat}$  increment rate is calculated by  $(N_{mat\_SLIPSA} - N_{mat\_X}) / N_{mat\_X}$  and the  $(C_\alpha)$  *RMSD* decrement rate is calculated by  $(RMSD_X - RMSD_{SLIPSA}) / RMSD_X$ , where  $X$  is DaliLite, CE or SSM. Figure 7 illustrates such increments and decrements in percentage. The following points should be mentioned: (1) no SSM comparison is performed in our earlier paper (Zhao et al., 2007); (2) from the time we completed this paper, it is possible to see result changes on any of the alignment websites, and we have observed minor changes on some of them; and (3) the SLIPSA experiments do not use any external alignment as input, although our algorithm is able to refine the alignment results retrieved from other web servers.

### 4.3. Discussion of the results

Table 2 shows some statistical data based on the results in Figure 7. For each case in which an alignment result from either DaliLite, CE, or SSM is missing, we were not able to compare it with SLIPSA. Also, since DaliLite, CE, and SSM may have different  $(C_\alpha)$  *RMSD* values for a given test case, they are not compared mutually. Common protein alignment scoring methods such as Z-score, Q-score, P-score, and geometric measures proposed in Kolodny et al. (2005) all take  $N_{mat}$  and  $(C_\alpha)$  *RMSD* into account. Due to the *RMSD* flexibility of SLIPSA, it is easy to compare SLIPSA with DaliLite, CE, and SSM on the basis of  $N_{mat}$  because in most cases SLIPSA outputs an equal  $(C_\alpha)$  *RMSD*. In our experiments, when compared with DaliLite, CE, and SSM, respectively, SLIPSA outputs a larger  $N_{mat}$  in 66.67%, 61.82%, and 86.70% of the cases. The maximum  $N_{mat}$  increment rate of SLIPSA is 65.33%, 64.58%, and 109.09%. On average, SLIPSA increases 4.56%, 4.13%, and 7.37% of the  $N_{mat}$ . In 26.67%, 29.09%, and 81.19% of the cases, SLIPSA outputs a smaller  $(C_\alpha)$  *RMSD* with the maximum  $(C_\alpha)$  *RMSD* decrement rate being 13.21%, 11.11%, and 16.56%. To sum up, in most cases, we see SLIPSA results with a larger or same  $N_{mat}$  and a same or smaller  $(C_\alpha)$  *RMSD*. In some cases where SLIPSA outputs a smaller  $N_{mat}$ , we also see a smaller  $(C_\alpha)$  *RMSD*.

We also attempt to compare SLIPSA with DaliLite, CE, and SSM in the cases of weak similarities. To simplify the comparison process, we tentatively define a weak similarity as a large  $(C_\alpha)$  *RMSD* between aligned chains. This definition may be incomplete; however, we have already observed some interesting results. In brief, SLIPSA obtains high average  $N_{mat}$  increment rates in the weak similarity cases, and the larger the  $(C_\alpha)$  *RMSD*, the higher the average  $N_{mat}$  increment rate (Table 3).

TABLE 2. STATISTICS ON THE EXPERIMENTAL RESULTS

<i>Number of valid cases</i>	<i>DaliLite 210</i>	<i>CE 220</i>	<i>SSM 218</i>
Cases with larger $N_{mat}$ by SLIPSA	149 (66.67%)	136 (61.82%)	189 (86.70%)
Cases with smaller $N_{mat}$ by SLIPSA	14 (6.67%)	26 (11.82%)	8 (3.67%)
Maximum $N_{mat}$ increment by SLIPSA	49	56	51
Maximum $N_{mat}$ decrement by SLIPSA	2	9	12
Maximum $N_{mat}$ increment rate by SLIPSA	65.33%	64.58%	109.09%
Maximum $N_{mat}$ decrement rate by SLIPSA	2.74%	6.45%	25.53%
Average $N_{mat}$ increment by SLIPSA	4.15	3.63	7.24
Average $N_{mat}$ increment rate by SLIPSA	4.56%	4.13%	7.37%
Cases with smaller <i>RMSD</i> by SLIPSA	56 (26.67%)	64 (29.09%)	177 (81.19%)
Maximum <i>RMSD</i> decrement by SLIPSA	0.7	0.4	0.52
Maximum <i>RMSD</i> decrement rate by SLIPSA	13.21%	11.11%	16.56%
Average <i>RMSD</i> decrement by SLIPSA	0.04	0.04	0.05
Average <i>RMSD</i> decrement rate by SLIPSA	1.55%	1.42%	2.07%

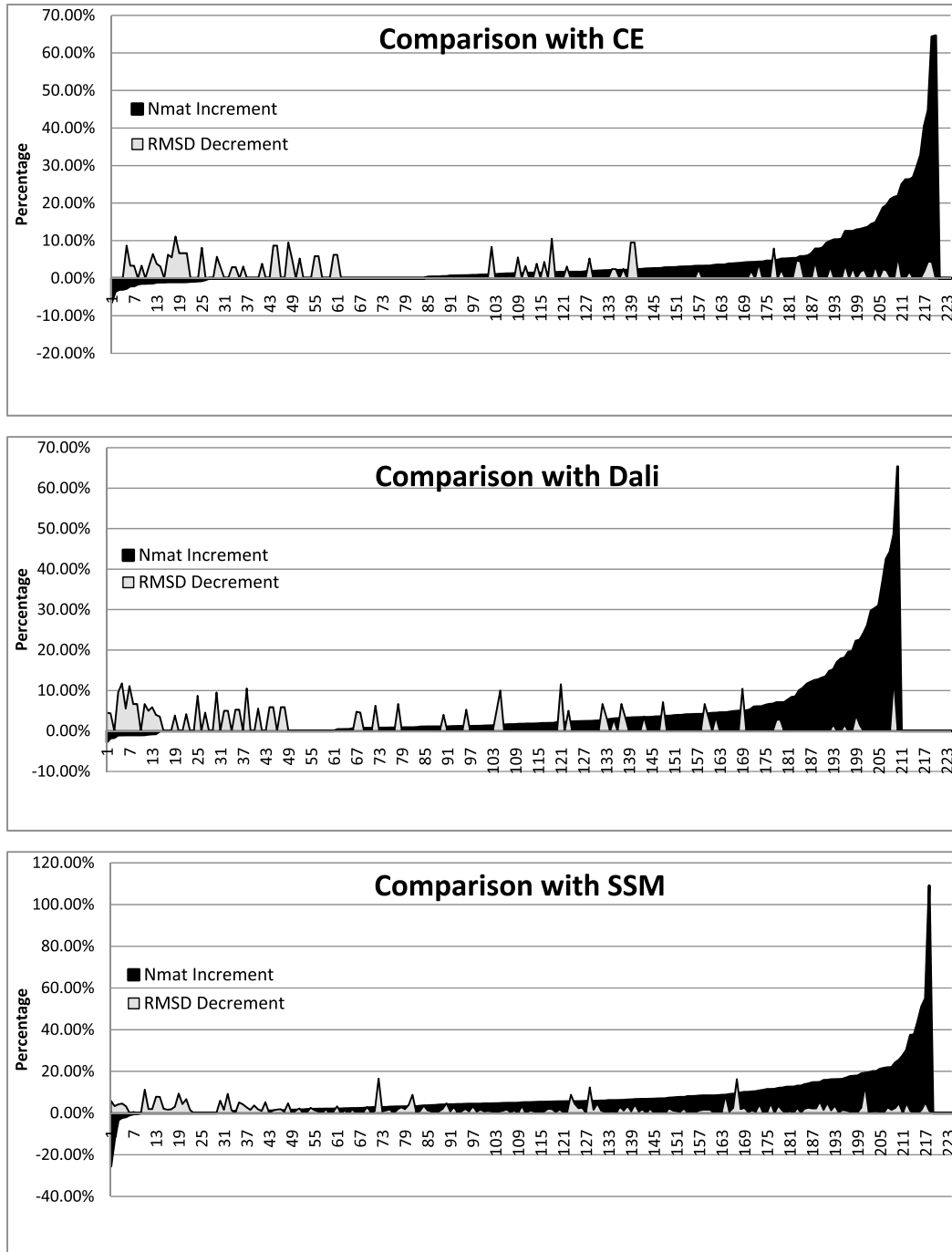


FIG. 7. Comparing SLIPSA with DaliLite, CE, and SSM.

TABLE 3. COMPARISON BASED ON WEAK SIMILARITY

	<i>DaliLite</i>		<i>CE</i>		<i>SSM</i>	
	<i>Valid cases</i>	<i>Avg. <math>N_{mat}</math> Inc.</i>	<i>Valid cases</i>	<i>Avg. <math>N_{mat}</math> Inc.</i>	<i>Valid cases</i>	<i>Avg. <math>N_{mat}</math> Inc.</i>
$RMSD \geq 5.0$	12	26.48%	14	21.62%	0	/
$RMSD \geq 4.0$	20	23.48%	41	14.75%	9	17.50%
$RMSD \geq 3.0$	77	10.09%	102	7.64%	51	12.15%

TABLE 4. COMPARING SLIPSA WITH CE

No.	CE, n/r/t	SLIPSA		No.	CE, n/r/t	SLIPSA	
		n/r/t	$n^+ / r^-$			n/r/t	$n^+ / r^-$
118	31/1.5/3.16	29/1.5/0.45	-6.45/0.00	119	33/0.6/3.13	33/0.6/0.55	0.00/0.00
17	256/2.8/18.97	247/2.8/83.78	-3.52/0.00	120	92/1.1/3.42	92/1.1/1.23	0.00/0.00
32	201/2.2/4.6	195/2.2/16.08	-2.99/0.00	121	55/1.5/3.27	55/1.5/0.34	0.00/0.00
41	237/3/10.3	230/3/149.39	-2.95/0.00	144	152/0.3/3.48	152/0.3/1.84	0.00/0.00
201	74/2.3/3.36	72/2.1/8.05	-2.70/8.70	145	152/0.5/3.41	152/0.5/1.42	0.00/0.00
215	97/3/3.94	95/2.9/20.69	-2.06/3.33	146	153/0.6/3.46	153/0.6/0.44	0.00/0.00
216	97/3/3.88	95/2.9/22.83	-2.06/3.33	148	141/1.6/3.57	141/1.5/11.92	0.00/6.25
109	64/3.5/3.66	63/3.5/13.64	-1.56/0.00	152	137/1.6/3.58	137/1.5/11.11	0.00/6.25
55	143/3/4.73	141/2.9/21.44	-1.40/3.33	155	138/1.6/3.63	138/1.6/21.7	0.00/0.00
222	73/2.6/3.48	72/2.6/9.23	-1.37/0.00	156	137/1.7/3.52	137/1.6/22.44	0.00/5.88
203	75/3/3.31	74/2.9/7.98	-1.33/3.33	157	136/1.7/3.52	136/1.6/17.16	0.00/5.88
38	77/3.1/3.39	76/2.9/6.95	-1.30/6.45	161	144/2.2/3.82	144/2.2/19.67	0.00/0.00
27	93/2.6/3.5	92/2.5/10.66	-1.08/3.85	164	141/2.2/3.86	141/2.2/33.03	0.00/0.00
220	96/3.2/3.74	95/3.1/23.77	-1.04/3.13	175	117/3.2/4.81	117/3.2/101.28	0.00/0.00
218	97/3/3.83	96/3/21.31	-1.03/0.00	184	105/0.4/3.23	105/0.4/0.33	0.00/0.00
197	102/1.6/3.3	101/1.5/3.86	-0.98/6.25	185	105/0.7/3.16	105/0.7/0.34	0.00/0.00
195	103/1.8/3.32	102/1.6/6.39	-0.97/11.11	186	105/1.3/3.27	105/1.3/0.69	0.00/0.00
196	103/1.8/3.33	102/1.7/5.63	-0.97/5.56	187	105/1.3/3.24	105/1.3/0.73	0.00/0.00
188	104/1.5/3.27	103/1.4/2.77	-0.96/6.67	190	105/1.5/3.3	105/1.5/0.84	0.00/0.00
189	104/1.5/3.19	103/1.4/2.66	-0.96/6.67	191	105/1.5/3.33	105/1.5/2.06	0.00/0.00
192	105/1.5/3.22	104/1.4/3.03	-0.95/6.67	193	104/1.6/3.3	104/1.6/2.84	0.00/0.00
54	117/1.9/3.63	116/1.9/7.09	-0.85/0.00	194	104/1.6/3.33	104/1.6/2.91	0.00/0.00
53	121/2.6/3.58	120/2.6/10.69	-0.83/0.00	198	101/2.1/3.35	101/1.9/5.78	0.00/9.52
93	266/2.3/10.42	264/2.3/14.64	-0.75/0.00	199	89/3.4/3.56	89/3.3/22.89	0.00/2.94
72	147/3.7/7.52	146/3.4/45.5	-0.68/8.11	212	96/2.4/3.44	96/2.4/9	0.00/0.00
16	253/3.4/15.03	252/3.4/67.88	-0.40/0.00	213	100/3.5/4.45	100/3.3/16.06	0.00/5.71
1	336/0.3/7.2	336/0.3/0.84	0.00/0.00	217	96/3.2/3.84	96/3.1/21.63	0.00/3.13
2	336/0.4/7.24	336/0.4/0.66	0.00/0.00	223	73/2.6/3.33	73/2.5/9.94	0.00/3.85
3	336/0.5/7.22	336/0.5/2.73	0.00/0.00	58	296/3/16.52	297/3/254.23	0.34/0.00
5	254/1.4/7.58	254/1.4/18.22	0.00/0.00	50	261/2.4/10.41	262/2.4/108.42	0.38/0.00
21	128/1.9/3.64	128/1.9/2.84	0.00/0.00	9	259/2.7/14.97	260/2.7/73.45	0.39/0.00
24	69/2.3/4.21	69/2.1/8.31	0.00/8.70	15	257/3.7/17.84	258/3.7/77.91	0.39/0.00
25	96/3.3/3.55	96/3.3/8.75	0.00/0.00	60	214/2.4/10.02	215/2.4/120.22	0.47/0.00
30	346/1.8/6.4	346/1.8/93.78	0.00/0.00	45	415/3.2/41.37	417/3.2/735	0.48/0.00
34	68/2/3.28	68/1.9/2.05	0.00/5.00	33	157/3.1/5.02	158/3.1/122.55	0.64/0.00
35	169/3.5/4.88	169/3.5/47.72	0.00/0.00	40	154/3/5.22	155/3/20.72	0.65/0.00
57	157/3.5/5.44	157/3.5/74.16	0.00/0.00	163	146/2.6/4.19	147/2.6/15.38	0.68/0.00
59	197/3.2/7.22	197/3.2/89.67	0.00/0.00	149	141/1.8/3.67	142/1.8/21.33	0.71/0.00
63	91/1.8/3.48	91/1.8/6.41	0.00/0.00	153	141/1.8/3.73	142/1.8/23.03	0.71/0.00
67	252/4.6/16.77	252/4.6/836.94	0.00/0.00	159	138/2/3.63	139/2/21.97	0.72/0.00
75	249/2.5/17.41	249/2.5/113.05	0.00/0.00	22	383/2.8/25.96	386/2.8/396.19	0.78/0.00
78	84/3.4/3.36	84/3.3/18.2	0.00/2.94	14	252/2.4/9.55	254/2.4/34.19	0.79/0.00
81	81/2.3/3.69	81/2.1/4.14	0.00/8.70	140	115/4.1/5.11	116/4.1/36.81	0.87/0.00
85	87/1.9/3.42	87/1.8/5.16	0.00/5.26	167	111/3.1/3.8	112/3.1/32.97	0.90/0.00
86	116/2.9/3.72	116/2.9/38.28	0.00/0.00	83	219/3.8/25.97	221/3.8/418.92	0.91/0.00
91	354/0.7/6.91	354/0.7/21.5	0.00/0.00	79	108/3.6/3.71	109/3.3/58.25	0.93/8.33
102	93/4.5/10.42	93/4.5/173.11	0.00/0.00	207	101/2.3/3.42	102/2.3/8.81	0.99/0.00
105	72/3.5/3.42	72/3.4/23.53	0.00/2.86	26	280/2.2/8.5	283/2.2/112.55	1.07/0.00
106	65/3/5.69	65/3/8.5	0.00/0.00	39	177/2.8/5.49	179/2.8/31.3	1.13/0.00
108	46/1.7/3.24	46/1.7/0.5	0.00/0.00	12	252/2.2/9.58	255/2.2/30.3	1.19/0.00
110	38/1.3/3.3	38/1.3/0.72	0.00/0.00	123	336/1.6/43.94	340/1.6/186.31	1.19/0.00
112	25/0.8/3.25	25/0.8/0.58	0.00/0.00	42	80/1.8/3.42	81/1.7/3.75	1.25/5.56
113	30/0.9/3.18	30/0.9/0.8	0.00/0.00	132	80/3.3/3.33	81/3.3/6.83	1.25/0.00
114	24/1/3.18	24/1/1.22	0.00/0.00	206	79/3.3/3.4	80/3.3/9.02	1.27/0.00
115	24/0.6/3.17	24/0.6/0.81	0.00/0.00	204	73/3.1/3.35	74/3/5.41	1.37/3.23
117	32/1.6/3.48	32/1.6/4.25	0.00/0.00	147	143/1.6/3.42	145/1.6/10.64	1.40/0.00

We list all the experimental results.

n, r, and t stand for  $N_{mat}$ ,  $(C_{\alpha})$  RMSD, and the execution time (s), respectively;  $n^+$  is the  $N_{mat}$  increment (%); and  $r^-$  is the  $(C_{\alpha})$  RMSD decrement (%).

The results are sorted in an ascending order of  $n^+$ .

TABLE 5. COMPARING SLIPSA WITH CE

No.	CE, n/r/t	SLIPSA		No.	CE, n/r/t	SLIPSE	
		n/r/t	$n^+ / r^-$			n/r/t	$n^+ / r^-$
202	71/2.6/3.32	72/2.5/6.78	1.41/3.85	208	97/2.4/4.23	101/2.4/13.48	4.12/0.00
205	71/2.7/3.28	72/2.7/10.72	1.41/0.00	171	119/3.5/4.59	124/3.5/82.08	4.20/0.00
150	141/1.9/3.77	143/1.9/28.14	1.42/0.00	101	118/5/8.89	123/4.9/393.2	4.24/2.00
66	68/2.3/3.39	69/2.2/1.17	1.47/4.35	168	117/3.3/4.28	122/3.3/72.02	4.27/0.00
49	266/3/9.83	270/3/224.11	1.50/0.00	211	92/2.1/3.63	96/2.1/8.45	4.35/0.00
160	133/1.9/3.66	135/1.7/24.03	1.50/10.53	214	92/2.5/3.8	96/2.4/12.75	4.35/4.00
7	259/2.7/14	263/2.7/49.09	1.54/0.00	48	87/1.9/3.44	91/1.9/3.98	4.60/0.00
94	124/2.7/4.92	126/2.7/69.13	1.61/0.00	173	130/4.9/5.06	136/4.9/129.34	4.62/0.00
18	244/3/14.67	248/3/72.45	1.64/0.00	76	64/3.8/3.16	67/3.5/3.08	4.69/7.89
28	122/1.9/3.47	124/1.9/10.03	1.64/0.00	183	62/4.4/5.52	65/4.4/39.23	4.84/0.00
43	61/3/3.5	62/3/5.75	1.64/0.00	96	117/4.3/6.19	123/4.2/126.28	5.13/2.33
111	61/3/3.38	62/3.1/8.23	1.64/3.13	99	97/5.4/4.64	102/5.4/108.45	5.15/0.00
137	61/2.7/3.58	62/2.7/3.5	1.64/0.00	65	268/3.1/12.05	282/3.1/336.36	5.22/0.00
172	122/3.8/4.7	124/3.8/104.3	1.64/0.00	4	302/1.5/7.66	318/1.5/17.58	5.30/0.00
165	112/3/3.94	114/3/66.5	1.79/0.00	98	94/4.3/9.27	99/4.1/99.56	5.32/4.65
44	54/1.9/3.2	55/1.8/1.14	1.85/5.26	219	86/2.2/3.81	91/2.1/6.39	5.81/4.55
46	159/2/4.03	162/2/9.19	1.89/0.00	84	275/3/14.53	291/3/317.33	5.82/0.00
71	104/4.3/3.98	106/4.3/58.97	1.92/0.00	169	117/3.5/4.05	124/3.5/91.53	5.98/0.00
47	50/1.1/3.33	51/1.1/1.84	2.00/0.00	69	76/2/3.72	81/2/7.41	6.58/0.00
23	98/2.2/3.58	100/2.2/5.56	2.04/0.00	130	64/4/3.08	69/4/4.64	7.81/0.00
29	97/1.9/3.48	99/1.9/4.09	2.06/0.00	141	64/4.4/3.53	69/4.2/4.69	7.81/4.55
64	94/4.1/3.98	96/4/82.73	2.13/2.44	19	111/4.2/10.78	120/4.2/256.16	8.11/0.00
97	94/4.1/3.77	96/4/77.3	2.13/2.44	103	64/2.6/3.53	70/2.6/7.69	9.38/0.00
70	187/2.4/10.16	191/2.4/50.64	2.14/0.00	104	71/3.2/6.46	78/3.1/46.8	9.86/3.13
125	92/4/3.88	94/3.9/20.7	2.17/2.50	200	78/3.2/3.47	86/3.2/33.42	10.26/0.00
209	92/2/3.63	94/2/7.86	2.17/0.00	92	107/4.2/5.48	118/4.2/77.84	10.28/0.00
154	137/2.1/3.66	140/1.9/22.22	2.19/9.52	36	48/4.3/3.3	53/4.3/6.88	10.42/0.00
158	137/2.1/3.68	140/1.9/22.11	2.19/9.52	129	48/3.6/3.14	54/3.5/3.53	12.50/2.78
6	257/2.7/16.56	263/2.7/62.72	2.33/0.00	142	80/4.1/3.34	90/4.1/53.98	12.50/0.00
80	84/2.9/3.5	86/2.9/11.05	2.38/0.00	224	56/5.5/3.28	63/5.3/18.69	12.50/3.64
122	40/5.3/3.18	41/5.3/1.7	2.50/0.00	177	62/3.8/4.89	70/3.8/152.3	12.90/0.00
8	236/2.5/13.73	242/2.5/78.69	2.54/0.00	95	115/5.8/12.84	130/5.7/653.22	13.04/1.72
56	115/3.2/5.63	118/3.2/97.81	2.61/0.00	178	75/4.4/5	85/4.3/105.83	13.33/2.27
89	115/3.2/5.45	118/3.2/97.98	2.61/0.00	176	88/4.3/4.89	100/4.3/275.34	13.64/0.00
166	114/3.8/4.16	117/3.8/87.33	2.63/0.00	100	97/5.6/5.05	111/5.6/160.03	14.43/0.00
77	107/3.9/4.27	110/3.9/131.34	2.80/0.00	107	54/3/3.83	62/2.9/17.06	14.81/3.33
134	70/2.7/3.67	72/2.7/8.36	2.86/0.00	138	48/2.9/3.41	56/2.9/2.88	16.67/0.00
151	140/2.1/3.83	144/2.1/20.91	2.86/0.00	127	80/4.1/3.28	95/4/24.56	18.75/2.44
162	139/2.3/3.55	143/2.3/20.81	2.88/0.00	131	72/5/3.17	86/4.9/18.23	19.44/2.00
31	133/1.8/3.72	137/1.8/18.14	3.01/0.00	181	62/3.9/5.38	75/3.9/126.09	20.97/0.00
51	394/3.1/40.63	406/3.1/652.8	3.05/0.00	179	88/4.5/4.95	107/4.5/299.27	21.59/0.00
87	130/3.1/4.16	134/3.1/25.64	3.08/0.00	90	64/7/3.34	78/6.6/77.94	21.88/5.71
82	97/2.9/3.8	100/2.9/13.55	3.09/0.00	180	80/6/3.81	100/6/390.19	25.00/0.00
170	125/4/4.48	129/4/83.06	3.20/0.00	88	80/5.4/3.8	101/5.3/63.84	26.25/1.85
10	248/2.4/10.42	256/2.4/59.03	3.23/0.00	182	80/6/3.75	101/6/303	26.25/0.00
174	124/4.2/4.66	128/4.1/111.48	3.23/2.38	133	56/7.3/3.27	71/7.3/23.3	26.79/0.00
210	92/2/3.54	95/2/10.5	3.26/0.00	20	54/4/6.22	70/4/27.47	29.63/0.00
37	366/3.4/24.1	378/3.4/729.19	3.28/0.00	61	171/3.9/35.27	227/3.9/863.27	32.75/0.00
143	116/3.9/4.06	120/3.9/103.66	3.45/0.00	221	64/5.3/3.41	90/5.2/60.84	40.63/1.89
13	253/2.6/13.14	262/2.6/80	3.56/0.00	135	56/4.5/3.25	81/4.3/46.39	44.64/4.44
62	84/4.5/3.48	87/4.5/7.31	3.57/0.00	73	56/4.6/3.64	92/4.4/36.38	64.29/4.35
139	84/3.5/3.88	87/3.5/31.16	3.57/0.00	136	48/5.2/3.64	79/5.2/26.2	64.58/0.00
52	156/1.9/5.36	162/1.9/9.64	3.85/0.00	68	n/a	/	/
74	78/1.7/3.28	81/1.7/3.3	3.85/0.00	124	n/a	/	/
11	253/2.6/13.2	263/2.6/121.42	3.95/0.00	126	n/a	/	/
116	25/1.3/3.54	26/1.3/2.63	4.00/0.00	128	n/a	/	/

See Table 4 for abbreviations.

TABLE 6. COMPARING SLIPSA WITH DALILITE

No.	DaliLite, n/r/t	SLIPSE		No.	DaliLite, n/r/t	SLIPSA	
		n/r/t	$n^+ / r^-$			n/r/t	$n^+ / r^-$
201	73/2.2/14.08	71/2.1/6.58	-2.74/4.55	194	105/1.7/9.2	105/1.7/2.06	0.00/0.00
44	58/2.3/8.44	57/2.2/1.22	-1.72/4.35	203	72/2.8/14.88	72/2.7/7.11	0.00/3.57
136	62/3.4/13.95	61/3.4/15.67	-1.61/0.00	207	102/2.3/15.04	102/2.3/8.77	0.00/0.00
197	102/1.7/8.75	101/1.5/3.98	-0.98/11.76	211	97/2.4/9.08	97/2.3/9.13	0.00/4.17
198	102/2.1/19.21	101/1.9/5.75	-0.98/9.52	219	91/2.1/9.95	91/2.1/5.98	0.00/0.00
195	103/1.8/9	102/1.6/6.41	-0.97/11.11	14	261/2.9/23.14	262/2.9/30.56	0.38/0.00
196	103/1.8/14.03	102/1.7/5.59	-0.97/5.56	50	261/2.4/19.28	262/2.4/112.16	0.38/0.00
188	104/1.5/8.83	103/1.4/2.5	-0.96/6.67	16	246/3.1/28.06	247/3.1/67.97	0.41/0.00
189	104/1.5/8.82	103/1.4/2.58	-0.96/6.67	17	246/2.8/31.31	247/2.8/85.34	0.41/0.00
190	104/1.3/8.77	103/1.3/0.88	-0.96/0.00	3	334/0.5/23.08	336/0.5/2.69	0.60/0.00
192	105/1.5/10.24	104/1.4/3.03	-0.95/6.67	52	162/2.1/13.68	163/2/10.44	0.62/4.76
28	125/2/8.54	124/1.9/11.45	-0.80/5.00	151	144/2.2/13.61	145/2.1/20.67	0.69/4.55
155	139/1.7/18.55	138/1.6/22.63	-0.72/5.88	26	286/2.5/28.3	288/2.5/201.66	0.70/0.00
162	145/2.5/33.33	144/2.4/19.42	-0.69/4.00	149	142/1.9/18.94	143/1.9/22.39	0.70/0.00
1	336/0.3/26.83	336/0.3/0.63	0.00/0.00	55	140/2.9/8.88	141/2.9/22.02	0.71/0.00
2	336/0.4/23.27	336/0.4/0.64	0.00/0.00	148	140/1.6/20.36	141/1.5/11.94	0.71/6.25
4	323/1.7/23.27	323/1.7/16.08	0.00/0.00	152	135/1.4/8.72	136/1.4/10.69	0.74/0.00
24	67/1.4/11.54	67/1.4/6.72	0.00/0.00	11	265/2.9/28.2	267/2.9/56.59	0.75/0.00
27	92/2.6/8.81	92/2.5/10.97	0.00/3.85	7	263/2.8/23.39	265/2.8/59.31	0.76/0.00
30	347/1.9/35.91	347/1.9/62.39	0.00/0.00	5	255/1.6/18.16	257/1.6/23.42	0.78/0.00
34	67/1.8/11.89	67/1.7/1.72	0.00/5.56	15	254/3.6/26.06	256/3.6/77.13	0.79/0.00
38	72/2.2/8.77	72/2.1/3.22	0.00/4.55	94	126/3/14.83	127/2.8/67.3	0.79/6.67
42	80/1.7/8.3	80/1.6/3.69	0.00/5.88	8	243/2.6/29.36	245/2.6/62.44	0.82/0.00
47	56/1.9/8.97	56/1.8/2.64	0.00/5.26	171	120/3.3/23.28	121/3.3/87.58	0.83/0.00
60	216/2.5/22.92	216/2.5/124.41	0.00/0.00	169	119/3.3/23.64	120/3.3/60.06	0.84/0.00
66	67/1.9/8.58	67/1.8/2.41	0.00/5.26	53	118/2.5/8.97	119/2.5/12.23	0.85/0.00
74	85/1.9/8.33	85/1.9/3.64	0.00/0.00	41	224/2.8/23.22	226/2.8/90.05	0.89/0.00
76	60/2.6/9.36	60/2.6/3.3	0.00/0.00	23	97/1.9/8.73	98/1.9/3.72	1.03/0.00
81	81/2.3/9.42	81/2.1/4.23	0.00/8.70	32	194/2.3/16.45	196/2.3/16.03	1.03/0.00
90	46/2.4/8.53	46/2.4/13.89	0.00/0.00	29	96/1.6/8.47	97/1.6/4.52	1.04/0.00
108	46/1.7/9.39	46/1.7/0.52	0.00/0.00	209	96/2.3/10.17	97/2.3/9.86	1.04/0.00
113	31/1.1/9.4	31/1.1/0.63	0.00/0.00	210	96/2.3/9.41	97/2.3/10.48	1.04/0.00
120	97/1.9/9.23	97/1.9/1.09	0.00/0.00	213	95/2.5/17.77	96/2.5/11.97	1.05/0.00
121	58/1.8/8.94	58/1.8/0.31	0.00/0.00	214	95/2.5/14.69	96/2.4/12.69	1.05/4.00
137	62/2.7/13.66	62/2.7/3.48	0.00/0.00	212	94/2.2/9.86	95/2.2/7.8	1.06/0.00
144	153/0.5/13.84	153/0.5/1.84	0.00/0.00	215	90/2.5/9.95	91/2.5/17.08	1.11/0.00
145	153/0.6/13.02	153/0.6/1.39	0.00/0.00	13	266/3/28.14	269/3/58.67	1.13/0.00
146	153/0.6/18.63	153/0.6/0.45	0.00/0.00	6	261/2.8/32.77	264/2.8/58.47	1.15/0.00
147	145/1.7/18.86	145/1.6/10.52	0.00/5.88	39	174/2.6/13.47	176/2.6/31.84	1.15/0.00
150	143/2/17.42	143/1.9/30.41	0.00/5.00	85	86/1.9/11.66	87/1.8/5.25	1.16/5.26
153	143/2/8.85	143/1.9/24.89	0.00/5.00	12	256/2.4/23.3	259/2.4/27.7	1.17/0.00
154	140/2.1/13.74	140/1.9/22.11	0.00/9.52	35	168/3.6/16.65	170/3.6/45.03	1.19/0.00
156	137/1.7/8.84	137/1.6/22.42	0.00/5.88	68	168/2.6/14.16	170/2.6/20.11	1.19/0.00
157	136/1.7/20.34	136/1.6/17.14	0.00/5.88	51	412/3.6/77.19	417/3.6/560.52	1.21/0.00
159	139/2/8.88	139/2/21.86	0.00/0.00	206	78/3.2/11.33	79/3.2/6.34	1.28/0.00
160	135/1.9/8.83	135/1.7/23.84	0.00/10.53	33	154/3/14.06	156/3/54.19	1.30/0.00
161	147/2.4/12.13	147/2.4/19.5	0.00/0.00	37	374/3.5/48.26	379/3.5/682.97	1.34/0.00
163	147/2.6/18.78	147/2.6/15.33	0.00/0.00	158	138/2/19.23	140/1.9/21.92	1.45/5.00
164	141/2.2/11.94	141/2.2/32.8	0.00/0.00	31	135/2/8.47	137/1.8/21.45	1.48/10.00
165	111/2.7/13.73	111/2.7/36.61	0.00/0.00	22	395/3.5/57.27	401/3.5/338.8	1.52/0.00
184	105/0.4/14.84	105/0.4/0.33	0.00/0.00	9	253/2.5/24.27	257/2.5/73.16	1.58/0.00
185	105/0.7/9.02	105/0.7/0.33	0.00/0.00	10	253/2.5/23.48	257/2.5/72.05	1.58/0.00
186	105/1.3/8.83	105/1.3/0.7	0.00/0.00	75	248/2.6/43.77	252/2.6/124.45	1.61/0.00
187	105/1.3/8.8	105/1.3/0.73	0.00/0.00	170	117/3.2/11.75	119/3.2/94.17	1.71/0.00
191	105/1.5/8.81	105/1.5/2.03	0.00/0.00	58	292/3/38.2	297/3/260.61	1.71/0.00
193	104/1.6/8.58	104/1.6/2.81	0.00/0.00	54	114/2/11.64	116/2/7.19	1.75/0.00

See Table 4 for abbreviations.

TABLE 7. COMPARING SLIPSA WITH DALILITE

No.	DaliLite, n/r/t	SLIPSA		No.	DaliLite, n/r/t	SLIPSA	
		n/r/t	$n^+ / r^-$			n/r/t	$n^+ / r^-$
65	285/3.4/48.03	290/3.4/500.44	1.75/0.00	67	260/6.7/28.1	273/6/1084.95	5.00/10.45
123	340/2/67.59	346/2/192.13	1.76/0.00	178	80/4.1/9.67	84/4.1/114.92	5.00/0.00
138	55/2.9/8.99	56/2.9/2.86	1.82/0.00	43	57/2.6/8.43	60/2.6/5.06	5.26/0.00
166	106/2.7/13.66	108/2.7/36.13	1.89/0.00	103	67/2.8/10.84	71/2.8/10.34	5.97/0.00
167	106/2.7/8.89	108/2.7/33.78	1.89/0.00	72	133/3/13.78	141/3/31.47	6.02/0.00
93	262/2.6/23.55	267/2.6/15.75	1.91/0.00	199	83/3.2/8.85	88/3.2/28.22	6.02/0.00
49	261/2.7/43.21	266/2.7/204.52	1.92/0.00	129	47/2.8/9.86	50/2.8/2.98	6.38/0.00
59	189/2.9/18.13	193/2.9/97.23	2.12/0.00	101	106/4.3/13.58	113/4.3/285.97	6.60/0.00
63	94/2.6/8.38	96/2.3/8.45	2.13/11.54	78	75/3/8.66	80/3/15.23	6.67/0.00
70	188/2.5/13.92	192/2.5/49.75	2.13/0.00	56	114/3.6/14.02	122/3.5/117.25	7.02/2.78
48	89/2/8.44	91/1.9/4.3	2.25/5.00	89	114/3.6/13.63	122/3.5/117.34	7.02/2.78
87	131/3.1/13.27	134/3.1/26.02	2.29/0.00	62	71/3.2/8.77	76/3.2/6.34	7.04/0.00
200	86/3.4/9.36	88/3.4/26.72	2.33/0.00	105	65/3.1/8.83	70/3.1/23.09	7.69/0.00
127	85/2.9/10.17	87/2.9/12.42	2.35/0.00	135	72/3.9/24.28	78/3.9/62.7	8.33/0.00
83	211/3.5/28.71	216/3.5/497.88	2.37/0.00	182	71/3.7/38.27	77/3.7/113.52	8.45/0.00
18	250/3.4/23.03	256/3.4/54.63	2.40/0.00	224	50/3.8/9.7	55/3.8/9.66	10.00/0.00
84	291/3.3/62.72	298/3.3/355.72	2.41/0.00	77	94/3.3/8.52	104/3.3/81.64	10.64/0.00
168	121/3.5/30.89	124/3.5/87.14	2.48/0.00	132	69/2.8/9.05	77/2.8/10.5	11.59/0.00
69	80/2.1/8.5	82/2.1/8.63	2.50/0.00	96	116/5.3/13.34	130/5.3/181.38	12.07/0.00
119	36/3/10.63	37/2.8/0.7	2.78/6.67	98	64/2.2/13.92	72/2.2/43.89	12.50/0.00
222	71/2.8/9.52	73/2.7/23.52	2.82/3.57	142	71/3.2/18.77	80/3.2/19.98	12.68/0.00
202	69/2.4/8.73	71/2.4/6.45	2.90/0.00	61	206/4.1/32.95	233/4.1/1182.63	13.11/0.00
106	66/3.4/9.08	68/3.3/7.59	3.03/2.94	131	67/3.3/22.08	76/3.3/8.3	13.43/0.00
107	65/3.7/8.59	67/3.7/16.09	3.08/0.00	104	81/5.2/14.11	93/5.2/108.55	14.81/0.00
141	63/3/19.34	65/2.8/3.61	3.17/6.67	183	66/6.3/11.77	76/6.2/89.92	15.15/1.59
130	62/2.9/14.44	64/2.8/2.98	3.23/3.45	109	53/3.2/9.66	62/3.2/10.36	16.98/0.00
45	401/3.1/52.25	414/3.1/521.05	3.24/0.00	177	84/7/9.48	99/7/700.17	17.86/0.00
174	121/3.9/13.66	125/3.9/77.64	3.31/0.00	181	83/6.7/13.75	98/6.6/649.63	18.07/1.49
172	120/3.8/24.03	124/3.8/104.23	3.33/0.00	71	82/3.3/8.42	98/3.3/38.05	19.51/0.00
57	149/3/16.75	154/3/95.56	3.36/0.00	20	56/3.4/8.67	67/3.4/15.17	19.64/0.00
217	89/2.8/14.98	92/2.7/19.17	3.37/3.57	118	27/2.5/8.97	33/2.4/0.42	22.22/4.00
111	59/3.1/9.36	61/3.1/7.27	3.39/0.00	100	89/6.6/8.56	109/6.5/221.02	22.47/1.52
40	147/2.7/13.34	152/2.7/21.13	3.40/0.00	176	83/4.4/33.33	103/4.4/242.14	24.10/0.00
86	114/3.1/8.85	118/3.1/29.55	3.51/0.00	124	180/5.2/89.42	227/5.2/4821.3	26.11/0.00
175	113/3.2/19.06	117/3.2/100.83	3.54/0.00	95	101/5.9/23.8	131/5.9/659.64	29.70/0.00
73	82/2.8/8.83	85/2.6/12.73	3.66/7.14	19	99/4.9/13.27	129/4.9/314.61	30.30/0.00
112	27/2.3/8.67	28/2.3/0.88	3.70/0.00	180	71/4.9/17.13	93/4.9/171.09	30.99/0.00
125	81/2.8/21.94	84/2.8/15.56	3.70/0.00	179	79/4.6/14.47	108/4.6/312.5	36.71/0.00
143	104/3.2/20.03	108/3.2/62.34	3.85/0.00	133	47/6.7/9.24	67/6.7/19.44	42.55/0.00
79	103/3/8.77	107/3/77.69	3.88/0.00	88	61/3.8/8.55	88/3.8/45.11	44.26/0.00
221	77/2.7/24.17	80/2.7/14.05	3.90/0.00	36	37/5.3/9.78	55/4.6/9.77	48.65/13.21
64	74/1.8/8.51	77/1.8/18.58	4.05/0.00	102	75/10.3/13.86	124/10.3/1903.98	65.33/0.00
97	74/1.8/8.39	77/1.8/18.5	4.05/0.00	21	n/a	/	/
223	73/3/9.27	76/3/13.47	4.11/0.00	46	n/a	/	/
82	97/3.2/11.18	101/3.2/13.97	4.12/0.00	80	n/a	/	/
208	97/2.4/13.91	101/2.4/13.44	4.12/0.00	91	n/a	/	/
134	70/3/14.22	73/2.8/8.69	4.29/6.67	99	n/a	/	/
205	70/2.9/14.1	73/2.8/10.03	4.29/3.45	110	n/a	/	/
204	68/2.7/9.54	71/2.7/12.2	4.41/0.00	114	n/a	/	/
216	90/2.8/12.86	94/2.7/26.19	4.44/3.57	115	n/a	/	/
25	88/2.8/8.57	92/2.8/10.06	4.55/0.00	116	n/a	/	/
220	88/2.7/9.56	92/2.7/24.88	4.55/0.00	117	n/a	/	/
218	87/2.5/10.89	91/2.5/19.34	4.60/0.00	122	n/a	/	/
92	105/3.1/11.81	110/3.1/82.5	4.76/0.00	126	n/a	/	/
139	82/3.3/30.33	86/3.3/35.14	4.88/0.00	128	n/a	/	/
173	121/4/13.53	127/4/72.64	4.96/0.00	140	n/a	/	/

See Table 4 for abbreviations.



TABLE 8. COMPARING SLIPSA WITH SSM

No.	SSM, $n/r/t$	SLIPSA		No.	SSM, $n/r/t$	SLIPSA	
		$n/r/t$	$n^+/r^-$			$n/r/t$	$n^+/r^-$
224	47/3.16/8.05	35/2.97/9.3	-25.53/6.01	185	103/0.73/7.16	105/0.73/0.42	1.94/0.00
118	32/1.49/11.91	28/1.44/0.5	-12.50/3.36	52	153/1.56/9.27	156/1.53/9.86	1.96/1.92
122	31/2.85/6.31	30/2.73/1.16	-3.23/4.21	191	102/1.35/7.6	104/1.35/3.92	1.96/0.00
23	93/1.31/7.41	91/1.25/3.44	-2.15/4.58	193	101/1.44/11.73	103/1.43/4.28	1.98/0.69
121	56/1.5/12.49	55/1.45/0.42	-1.79/3.33	144	149/0.31/7.39	152/0.3/2.64	2.01/3.23
2	336/0.36/11.93	333/0.36/0.67	-0.89/0.00	13	244/2.2/7.55	249/2.2/81.25	2.05/0.00
14	236/1.94/18.28	235/1.93/79.36	-0.42/0.52	29	93/1.4/7.19	95/1.38/3.08	2.15/1.43
4	311/1.38/7.21	310/1.38/16.25	-0.32/0.00	148	138/1.53/11.69	141/1.53/14.95	2.17/0.00
1	336/0.33/7.83	336/0.33/0.61	0.00/0.00	108	45/1.77/5.69	46/1.74/0.64	2.22/1.69
3	336/0.46/7.27	336/0.46/2.67	0.00/0.00	26	269/2.03/10.49	275/2.02/165.38	2.23/0.49
21	127/1.83/7.3	127/1.8/2.89	0.00/1.64	55	131/2.31/11.52	134/2.31/16.77	2.29/0.00
28	124/2.04/11.61	124/1.88/11.64	0.00/7.84	100	85/3.33/10.17	87/3.31/91.92	2.35/0.60
34	63/1.38/7.17	63/1.32/1.38	0.00/4.35	78	78/3.08/6.34	80/2.99/19.84	2.56/2.92
47	53/1.39/12.03	53/1.26/1.97	0.00/9.35	175	113/3.14/11.77	116/3.13/125.08	2.65/0.32
69	81/2.08/7.36	81/2.04/8.59	0.00/1.92	32	184/1.9/7.91	189/1.88/31.75	2.72/1.05
74	85/2.14/7.58	85/1.9/5.28	0.00/11.21	201	73/3.14/11.11	75/2.62/17.3	2.74/16.56
110	38/1.27/11.27	38/1.25/0.92	0.00/1.57	70	182/2.28/7.66	187/2.26/70.36	2.75/0.88
113	31/1.11/6.26	31/1.11/0.69	0.00/0.00	54	109/1.56/11.59	112/1.54/6.86	2.75/1.28
114	24/1/14.02	24/1/1.27	0.00/0.00	163	138/2.21/8.5	142/2.17/24.16	2.90/1.81
115	29/1.93/9.65	29/1.78/0.95	0.00/7.77	156	134/1.63/7.25	138/1.63/32.05	2.99/0.00
119	33/0.61/7.54	33/0.61/1.03	0.00/0.00	30	333/1.75/7.43	343/1.72/65.5	3.00/1.71
129	49/2.8/5.75	49/2.8/2.64	0.00/0.00	31	132/1.75/7.36	136/1.7/20.11	3.03/2.86
145	152/0.47/11.47	152/0.47/1.78	0.00/0.00	195	98/1.5/10.35	101/1.44/5.11	3.06/4.00
146	153/0.56/7.25	153/0.56/0.63	0.00/0.00	196	98/1.59/7.64	101/1.56/6.2	3.06/1.89
153	143/1.91/7.55	143/1.87/32.06	0.00/2.09	66	65/1.94/7.77	67/1.77/2.45	3.08/8.76
157	136/1.66/7.14	136/1.63/23.78	0.00/1.81	16	225/2.61/7.38	232/2.6/89.39	3.11/0.38
192	102/1.35/10.33	102/1.26/4.73	0.00/6.67	168	120/3.65/7.52	124/3.61/102.97	3.33/1.10
194	103/1.62/8.8	103/1.57/3.95	0.00/3.09	137	57/2.24/7.34	59/2.16/4.41	3.51/3.57
219	90/2.06/7.94	90/2.02/9.48	0.00/1.94	40	141/2.26/7.28	146/2.23/20.55	3.55/1.33
161	141/2.19/7.7	142/2.06/28.34	0.71/5.94	49	246/2.28/7.56	255/2.26/103.47	3.66/0.88
154	137/1.83/7.59	138/1.8/29.59	0.73/1.64	5	241/1.29/11.55	250/1.29/19.88	3.73/0.00
155	137/1.74/7.48	138/1.58/34.03	0.73/9.20	42	76/1.5/5.72	79/1.47/3.13	3.95/2.00
152	134/1.37/12.2	135/1.35/13.67	0.75/1.46	206	76/3.26/13.83	79/3.23/9.13	3.95/0.92
53	110/1.88/7.17	111/1.88/12.06	0.91/0.00	112	25/1.27/9.3	26/1.21/0.88	4.00/4.72
184	103/0.39/10.02	104/0.37/0.48	0.97/5.13	140	98/3.24/8.98	102/3.21/22.28	4.08/0.93
186	102/1.09/7.22	103/1.05/0.89	0.98/3.67	207	98/2.38/10.85	102/2.31/11.91	4.08/2.94
187	102/1.07/15.32	103/1.05/0.95	0.98/1.87	11	240/2.2/10.64	250/2.2/86.77	4.17/0.00
188	102/1.42/7.49	103/1.36/3.16	0.98/4.23	9	238/2.13/7.7	248/2.11/41.92	4.20/0.94
189	102/1.42/7.55	103/1.38/3.17	0.98/2.82	62	70/2.95/5.97	73/2.89/7.53	4.29/2.03
190	102/1.3/7.54	103/1.28/1.09	0.98/1.54	169	115/3.38/14.08	120/3.27/94.8	4.35/3.25
197	98/1.34/7.55	99/1.27/4.92	1.02/5.22	172	115/3.45/9.48	120/3.42/96.53	4.35/0.87
198	98/1.88/8.38	99/1.86/6.59	1.02/1.06	222	69/2.43/11.95	72/2.41/16.16	4.35/0.82
212	92/1.98/14.97	93/1.98/8.28	1.09/0.00	223	69/2.43/7.48	72/2.39/22.92	4.35/1.65
27	89/2.36/8.19	90/2.33/9.47	1.12/1.27	123	311/1.13/9.57	325/1.12/225.61	4.50/0.88
39	169/2.39/7.3	171/2.35/29.41	1.18/1.67	151	133/1.85/8.08	139/1.83/24.94	4.51/1.08
125	82/2.68/7.03	83/2.63/13.41	1.22/1.87	68	155/2.27/11.61	162/2.25/23.77	4.52/0.88
93	241/1.45/10.39	244/1.44/18.23	1.24/0.69	120	88/1.12/7.81	92/1.12/1.53	4.55/0.00
81	79/2.12/7.14	80/2.02/3.95	1.27/4.72	106	65/3.33/6.3	68/3.3/10.13	4.62/0.90
158	136/1.82/7.18	138/1.82/30.92	1.47/0.00	48	85/1.71/11.47	89/1.69/4.11	4.71/1.17
159	134/1.79/7.23	136/1.77/33.52	1.49/1.12	7	231/2/19.33	242/1.99/65.95	4.76/0.50
24	65/1.35/7.13	66/1.32/6.52	1.54/2.22	79	105/3.4/7.44	110/3.34/104.19	4.76/1.76
12	246/2.04/13.78	250/2.04/36.33	1.63/0.00	75	230/2.24/9.74	241/2.21/159.78	4.78/1.34
86	114/2.85/7.16	116/2.84/50.16	1.75/0.35	25	83/2.35/7.23	87/2.35/15.66	4.82/0.00
138	54/2.75/10.13	55/2.68/4.86	1.85/2.55	76	60/2.87/7.56	63/2.77/7.59	5.00/3.48
173	107/2.91/7.89	109/2.87/143.53	1.87/1.37	60	198/2.05/10.56	208/2.05/102.94	5.05/0.00
41	209/2.36/7.58	213/2.34/94.55	1.91/0.85	80	79/2.41/11.66	83/2.38/12.05	5.06/1.24

See Table 4 for abbreviations.

TABLE 9. COMPARING SLIPSA WITH SSM

No.	SSM, $n/r/t$	SLIPSA		No.	SSM, $n/r/t$	SLIPSA	
		$n/r/t$	$n^+/r^-$			$n/r/t$	$n^+/r^-$
150	136/1.92/7.44	143/1.9/38.11	5.15/1.04	97	70/1.85/11.75	77/1.84/19.17	10.00/0.54
58	267/2.45/7.58	281/2.44/319.89	5.24/0.41	220	80/2.39/7.57	88/2.33/27.28	10.00/2.51
6	227/1.95/7.56	239/1.94/86.83	5.29/0.51	133	49/3.74/5.74	54/3.69/10.75	10.20/1.34
164	132/2.12/7.39	139/2.1/49.83	5.30/0.94	130	58/2.88/7.11	64/2.86/6.16	10.34/0.69
174	113/3.39/9.53	119/3.32/92.55	5.31/2.06	213	84/2.14/10.47	93/2.05/19.61	10.71/4.21
142	74/3.16/7.43	78/3.1/35.72	5.41/1.90	33	137/2.74/7.48	152/2.74/47.27	10.95/0.00
208	92/2.12/9.05	97/2.11/18.27	5.43/0.47	61	202/4.05/10.72	225/4.04/813.22	11.39/0.25
132	73/2.88/5.69	77/2.83/8.8	5.48/1.74	199	79/3.36/7.47	88/3.2/27.52	11.39/4.76
10	236/2.12/10.67	249/2.11/64.05	5.51/0.47	50	227/2.01/7.69	253/2.01/87.25	11.45/0.00
84	271/2.84/16.17	286/2.83/428.7	5.54/0.35	205	67/3.36/7.53	75/3.24/15.67	11.94/3.57
38	72/3.19/13.58	76/2.91/8.22	5.56/8.78	19	108/4.36/11.63	121/4.3/330.28	12.04/1.38
43	54/2.25/7.16	57/2.15/6.17	5.56/4.44	22	329/2.47/13.32	370/2.47/523.75	12.46/0.00
209	90/2.11/9.83	95/2.06/10.86	5.56/2.37	202	64/2.51/11.82	72/2.48/10.19	12.50/1.20
82	89/2.33/7.33	94/2.27/10.19	5.62/2.58	218	80/2.39/17.89	90/2.39/27.78	12.50/0.00
214	88/2.17/7.64	93/2.17/12.88	5.68/0.00	217	77/2.43/7.56	87/2.37/29.69	12.99/2.47
117	35/3.57/7.77	37/3.13/12.08	5.71/12.32	37	323/2.99/7.67	365/2.98/712.13	13.00/0.33
211	87/1.88/18.94	92/1.85/12.31	5.75/1.60	124	146/3.28/11	166/3.21/2676.58	13.70/2.13
136	52/2.39/13.16	55/2.28/14.63	5.77/4.60	139	71/2.74/7.32	81/2.67/32.8	14.08/2.55
147	137/1.64/7.58	145/1.62/13.66	5.84/1.22	95	89/3.13/12.67	102/3.06/258.66	14.61/2.24
35	153/3.15/7.29	162/3.13/41	5.88/0.63	94	109/2.67/18	125/2.61/84.16	14.68/2.25
105	66/3.12/7.19	70/3.11/36.42	6.06/0.32	20	61/4.14/7.27	70/3.91/18.38	14.75/5.56
8	214/1.98/9	227/1.96/71.78	6.07/1.01	71	70/1.98/7.38	81/1.95/15.67	15.71/1.52
170	115/3.45/9.31	122/3.45/117.59	6.09/0.00	109	57/4.29/10.73	66/4.08/18.52	15.79/4.90
127	82/2.97/7.22	87/2.88/14.48	6.10/3.03	57	131/2.99/7.39	152/2.95/69.23	16.03/1.34
166	97/2.46/7.33	103/2.43/38.45	6.19/1.22	107	56/3.67/7.27	65/3.54/20.75	16.07/3.54
160	126/1.75/7.23	134/1.69/35.91	6.35/3.43	92	93/2.97/7.34	108/2.96/80.75	16.13/0.34
17	219/2.29/11.67	233/2.27/91.05	6.39/0.87	36	43/4/8.13	50/3.94/11.34	16.28/1.50
141	61/2.94/7.08	65/2.82/4.47	6.56/4.08	176	83/3.98/9.03	97/3.93/224.13	16.87/1.26
87	121/2.69/7.48	129/2.67/56.39	6.61/0.74	83	188/3.81/7.58	221/3.78/595.67	17.55/0.79
90	45/2.6/10.41	48/2.55/20.91	6.67/1.92	203	62/2.86/8.19	73/2.86/9.8	17.74/0.00
210	90/2.11/15.58	96/2.11/11.59	6.67/0.00	101	101/4.75/7.33	119/4.73/331.2	17.82/0.42
149	134/1.91/10.19	143/1.87/28.72	6.72/2.09	73	69/2.28/13.66	82/2.22/9.97	18.84/2.63
45	371/2.69/7.63	396/2.68/542.58	6.74/0.37	85	73/2.1/7.41	87/1.82/7.97	19.18/13.33
98	73/2.93/7.44	78/2.93/70.36	6.85/0.00	182	67/3.88/7.33	80/3.88/129.41	19.40/0.00
111	58/3.25/9.35	62/3.23/10.09	6.90/0.62	135	60/3.27/7.35	72/3.24/62.58	20.00/0.92
72	130/2.94/7.57	139/2.94/36.45	6.92/0.00	177	65/4.41/9.55	78/4.37/169.52	20.00/0.91
64	69/1.69/7.19	74/1.65/16.47	7.25/2.37	167	90/2.81/7.23	109/2.79/64.45	21.11/0.71
143	109/3.93/7.19	117/3.88/80.41	7.34/1.27	88	65/3.26/7.41	79/3.25/60.22	21.54/0.31
51	375/3.07/11.72	403/3.04/549.48	7.47/0.98	181	55/3.2/12.02	67/3.11/100.33	21.82/2.81
162	132/2.27/11.63	142/2.22/29.97	7.58/2.20	179	87/4.6/11.95	106/4.53/377.38	21.84/1.52
204	66/2.73/8.05	71/2.71/17.03	7.58/0.73	103	58/2.85/9.13	72/2.78/17.91	24.14/2.46
171	113/3.39/9.58	122/3.39/191.53	7.96/0.00	134	59/3.23/7.28	74/3.08/17.8	25.42/4.64
65	251/2.85/8.91	271/2.84/312.78	7.97/0.35	67	185/4.02/13.11	236/4.02/1101.73	27.57/0.00
165	99/2.43/7.28	107/2.42/52.7	8.08/0.41	183	33/1.66/7.44	43/1.58/15.91	30.30/4.82
59	170/2.36/7.45	184/2.33/273.2	8.24/1.27	96	75/2.92/10.13	103/2.89/119.3	37.33/1.03
18	204/2.19/7.3	221/2.19/128.86	8.33/0.00	180	56/3.82/7.66	77/3.8/159.19	37.50/0.52
56	108/3.11/11.92	117/3.06/88.72	8.33/1.61	178	50/3.03/8.88	72/3.03/89.19	44.00/0.00
89	108/3.11/7.55	117/3.06/112.03	8.33/1.61	102	45/3.08/7.55	68/3.03/206.2	51.11/1.62
216	84/2.52/7.69	91/2.48/19.88	8.33/1.59	116	20/2.15/7.86	31/2.04/3.7	55.00/5.12
215	83/2.39/7.72	90/2.39/36.94	8.43/0.00	77	44/2.49/10.66	92/2.46/63.05	109.09/1.20
104	70/2.97/11.98	76/2.96/87.03	8.57/0.34	44	n/a	/	/
200	81/3.71/14.19	88/3.39/30.55	8.64/8.63	46	n/a	/	/
15	212/2.7/12.91	231/2.68/101.19	8.96/0.74	91	n/a	/	/
131	66/2.94/7.57	72/2.87/8.28	9.09/2.38	99	n/a	/	/
63	87/2.4/5.72	95/2.01/8.17	9.20/16.25	126	n/a	/	/
221	72/2.59/11.95	79/2.54/17.33	9.72/1.93	128	n/a	/	/

See Table 4 for abbreviations.

The execution time of each case by each algorithm is also recorded. A DaliLite, CE, or SSM time is measured between when we submit the alignment request and when we receive the result page. The SLIPSA execution time is measured by our MATLAB program. Note that different web tools may use machines with different computation power, and different software implementation would result in speed changes. Therefore, the time data only give a rough idea about how fast each web tool could be. The average execution time of DaliLite, CE, and SSM is 16.86 s, 6.14 s, and 9.15 s, respectively. When compared with them, the average execution time of SLIPSA is 105.97 s, 69.89 s, and 81.43 s, respectively. In about 50% of the cases, the SLIPSA time is below the DaliLite average, and the corresponding numbers for CE and SSM are about 25% and 28%, respectively. The speed of our algorithm is relatively slow, and occasionally we see an execution time of thousands of seconds, especially for those long chain pairs of weak similarity. It is possible to improve the speed by applying different hardware configuration and software optimization methods as well as parallel and distributed implementation.

### DISCLOSURE STATEMENT

No competing financial interests exist.

### REFERENCES

- Chew, L.P., Kedem, K., Huttenlocher, D.P., et al. 1999. Fast detection of geometric substructure in proteins. *J. Comput. Biol.* 6, 313–325.
- Eggert, D., Lorusso, A., and Fisher, R. 1995. A comparison of four algorithms for estimating 3-D rigid transformations. *Br. Mach. Vision Conf.* 237–246.
- Falicov, A., and Cohen, F.E. 1996. A surface of minimum area metric for the structural comparison of protein. *J. Mol. Biol.* 258, 871–892.
- Fischer, D., Elofsson, A., Rice, D., et al. 1996. Assessing the performance of fold recognition methods by means of a comprehensive benchmark. *Proc. 1st Pacif. Symp. Biocomput.* 300–318.
- Fischer, D., Nussinov, R., and Wolfson, H. 1992. 3D substructure matching in protein molecules. *Lect. Notes Comput. Sci.* 644, 136–150.
- Godzik, A. 1996. The structural alignment between two proteins: is there a unique answer? *Prot. Sci.* 5, 1325–1338.
- Holm, L., and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233, 123–138.
- Ilyin, V.A., Abyzov, A., and Leslin, C.M. 2004. Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point. *Protein Sci.* 13, 1865–1874.
- Kolodny, R., Koehl, P., and Levitt, M. 2005. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J. Mol. Biol.* 346, 1173–1188.
- Kolodny, R., Linial, N., and Levitt, M. 2004. Approximate protein structural alignment in polynomial time. *Proc. Natl. Acad. Sci. USA* 101, 12201–12206.
- Krissinel, E., and Henrick, K. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Cryst.* D60, 2256–2268.
- Lathrop, R. H. 1994. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.* 7, 1059–1068.
- Lessel, U., and Schomburg, D. 1994. Similarities between protein 3-D structures. *Protein Eng.* 7, 1175–1187.
- Madej, T., Gibrat, J.F., and Bryant, S.H. 1995. Threading a database of protein cores. *Proteins* 23, 356–369.
- Ortiz, A., Strauss, C., and Olmea, O. 2002. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.* 11, 2606–2621.
- Shindyalov, I.N., and Bourne, P.E. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* 11, 739–747.
- Singh, A.P., and Brutlag, D.L. 1997. Hierarchical protein superposition using both secondary structure and atomic representation. *Proc. Intell. Syst. Mol. Biol.* 284–293.
- Taylor, W.R. 1999. Protein structure comparison using iterated double dynamic programming. *Protein Sci.* 9, 654–665.
- Taylor, W.R., and Orengo, C. 1989. Protein structure alignment. *J. Mol. Biol.* 208, 1–22.
- Umeyama, S. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 376–380.
- Ye, Y., and Godzik, A. 2004. Database searching by flexible protein structure alignment. *Protein Sci.* 13, 1841–1850.

- Ye, J., Janardan, R., and Liu, S. 2005. Pairwise protein structure alignment based on an orientation-independent backbone representation. *J. Bioinform. Comput. Biol.* 4, 699–717.
- Yona, G., and Kedem, K. 2005. The URMS-RMS hybrid algorithm for fast and sensitive local protein structure alignment. *J. Comput. Biol.* 12, 12–32.
- Zhang, Y., and Skolnick, J. 2005. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* 33, 2302–2309.
- Zhao, Z., and Fu, B. 2007. A flexible algorithm for pairwise protein structure alignment. *2007 Int. Conf. Bioinform. Comput. Biol.* 16–22.

Address reprint requests to:

*Dr. Bin Fu*

*Department of Computer Science*

*University of Texas–Pan American*

*Edinburg, TX 78539*

*E-mail: binfo@cs.panam.edu*