iSARST: an integrated SARST web server for rapid protein structural similarity searches

Wei-Cheng Lo¹, Che-Yu Lee¹, Chi-Ching Lee¹,² and Ping-Chiang Lyu¹,³,∗

¹Institute of Bioinformatics and Structural Biology, ²Institute of Information System Application and ³Department of Life Sciences, National Tsing Hua University, Hsinchu, Taiwan

INTRODUCTION

Protein structural data are increasing exponentially nowadays. This fact has made structural comparison indispensable for protein functional and evolutionary studies, the basic approach of which is to relate proteins according to their structural similarities. To achieve the requirements of high-throughput data analyses, which are especially common in structural genomics researches, fast and accurate tools are in a high demand to access structural similarity searches. Searching methods working on amino acid sequence data such as BLAST (1) and FASTA (2) are extremely rapid, though they have long been known insensitive to detect structural relationships among proteins sharing low sequence homology (3). Alignment algorithms which directly solve geometric problems in superimposing three-dimensional (3D) protein structures can be very accurate, but most of them are not fast enough to serve as the basis of instant protein similarity search web services (4).

To combine the speed advantages of sequence-based methods and the accuracy merits of using structural data, many linear encoding algorithms have been proposed, such as those by Levine et al. (5), Lesk (6) and those of TOPSCAN (7), YAKUSA (8), 3D-BLAST (4) and SARST (9). By transforming 3D protein structural data into one-dimensional (1D) text strings or numerical series, these algorithms convert complicated geometric problems of structural superimpositions to much easier sequence comparison problems, which can be solved rapidly by applying traditional sequence alignment techniques. Among recently proposed linear encoding methods, Ramachandran Sequential Transformation (RST) (9) has been shown suitable to develop efficient protein structural similarity search tools. For instance, SARST (Structural similarity search Aided by RST) can run over 240 000 times as rapid as Combinatorial Extension (CE) (10) with comparable precisions in database searching (9). Besides, RST has been demonstrated applicable to detecting circular permutations (CPs) in proteins (11). CP is an evolutionary event that causes the amino- and carboxyl-termini of the resulted protein variants to be located at different positions of the original protein (12–14), while the overall 3D structures and biological functions remain preserved (15,16), with sometimes increased stability, activity or functional diversity (17–19). CP has been applied in folding researches (20–22) and many bioengineering fields (17,23–26). In detecting CP,
CPSARST (CP Search Aided by RST) achieved a speed around 9000 times higher than SAMO (protein Structure Alignment tool based on Multiple Objective optimization) (27) with similar alignment qualities. In addition, it was proposed capable of serving as a functional assignment system for hypothetical proteins when co-linear similarity search methods failed to properly annotate them (11).

Although the average precision of SARST is close to that of CE, it is basically a search tool. We thus proposed that it can be combined with some highly accurate structural comparison tool, e.g. FAST (Fast Alignment and Search Tool) (28), into a good web service, in which SARST rapidly screens the target database and then the structural comparison tool refines (re-orders) the hit list (9). The advantage of this combination is that, because most dissimilar structures can be eliminated in the screening stage, there is no need to perform one-against-all structural alignments, which may cost the user even more than a day (4,9), to obtain a precisely ordered hit list. However, to re-order a hit list of 500 proteins, for instance, takes from minutes to over an hour when common alignment methods are applied (27–29), which is too long yet to make an efficient and convenient web-based tool. The situation of CPSARST is similar; even if the ‘double filter-and-refine’ strategy greatly enhances its performance, this 2 x 2 step strategy still takes >2 min to search the current PDB (11).

For developing a rapid, accurate and multi-functional protein structural similarity search service, we have integrated SARST and CPSARST along with several structural alignment methods, i.e. FAST (28), TM-align (29), SAMO (27) and SE (Seed Extension) (30), into a multi-processor and batch-processing system named iSARST (the integrated service of SARST). In this service, (i) the RST algorithm forms the basis of rapid database searching, (ii) refinement engines, FAST and TM-align, provide a high accuracy in the ordering of hits, (iii) CPSARST and SAMO make it versatile since they can do circularly permuted and order-independent structural alignment, respectively and (iv) the SE algorithm equips it a state-of-the-art method to produce accurate structure-based sequence alignments. The developmental principles of iSARST include (i) giving the user as quick responses as possible, (ii) providing a batch-processing environment and (iii) offering user-friendly interfaces. When assessed with the datasets in Refs (9,31), iSARST well preserved the high precisions of the refinement engines, while the calculation time was greatly reduced. Retrieving and superimposing 500 homologs from the current PDB only takes 7.8 s. If the input proteins had been queried previously, the cached results can be regained in a second. The result pages of iSARST are designed in a way that structural examinations, functional assignments and successive database searches can be carried out conveniently. Server side programs are modularized; new search methods and refinement tools can be integrated easily. Besides, its multi-processor implementation system is quite flexible, any computer equipped with Linux operating system, conventional C libraries and PHP language can join iSARST as a node upon request. We hope that this efficient, versatile and convenient web server can be a good assistant and collaboration platform for structural biologists in this post-genomics era.

METHODS

The flowchart of iSARST can be found in Figure 1. After receiving the query structure, the master node will linearly encode it and perform database search. In the refinement stage, proteins in the hit list are scattered to all slave nodes and then superimposed to the query protein by using an accurate structural comparison tool specified by the user. The RMSD (root mean square distance) values, alignment sizes and structural similarity scores are gathered by the master node to re-order the hit list, which is output with superimpositions and functional information. Finally, the refined data are cached in several forms to ensure a quick response once the same proteins are queried again in the future.

Linear encoding of protein structures

The RST algorithm (9) is implemented in iSARST to linearly encode protein structures. Traditional Ramachandran plot was organized with a nearest-neighbor clustering approach into 22 regions represented by different symbols. In this way, a protein structure can be transformed into a structurally meaningful string residue-by-residue according to $\phi$ and $\psi$ angles along its backbone. These 1D structural strings are called Ramachandran (RM) strings.

Structural similarity searches

To perform rapid database searches, all proteins in the PDB (32) and SCOP (33) have been pre-transformed into several RM string databases of various identity cutoffs. SARST and CPSARST both recruit blastall program (1) as the search engine. SARST is developed for common (co-linear) structural homologs; the database search is a straightforward execution of blastall. CPSARST specifically finds circular homologs. In the screening stage, it performs two rounds of similarity searches, with normal length (nl) and duplicated length (dl) of the query structure, respectively. After comparing results of these two rounds, the hits showing improved alignment qualities in the dl alignment will be chosen as CP candidates. The criteria are as follows,

\[
\begin{align*}
\text{score}_{\text{nl}} &> 1 \\
\log_{10}\left(\frac{E\text{-value}_{\text{dl}}}{E\text{-value}_{\text{nl}}}ight) &> 0.5
\end{align*}
\]

where score is the bit score calculated by blastall using the standard SARST scoring matrix (9) to measure the similarity between two RM strings. $E$-value (expectation value) is an assessment of the significance of score. Given that a hit has a score $S$, $E$-value is the expected
Figure 1. Flowchart of iSARST. The query structure is first transformed into a structurally meaningful Ramachandran string and then used to screen target database by SARST or CPSARST. In refinement stage, the raw hit list is re-ordered according to the structural similarity scores calculated by accurate structure comparison method like FAST (28), TM-align (29) or SAMO (27). Final outputs of iSARST are tables listing co-linear homologs or circular permutants of the query protein. Structure superimpositions and related inspection tools are provided, too.
number of different alignments occurring by chance with scores $\geq S$ in this particular database search (1,11).

Refinement of searching results

After database searches, the ordering of retrieved structural homologs is refined by some accurate structural comparison tool. Currently, we utilize FAST (28), TM-align (29) and SAMO (27) as refinement engines. FAST and TM-align have been shown to exhibit high structural alignment qualities (28,29), in many cases even outperforming DALI (34). Among the published structural comparison methods, they have very outstanding running speeds, e.g. superimposing a pair of proteins in 0.2–0.5 s in average with a 1.2-GHz processor (28,29). The speed of SAMO is similar to that of DALI, which requires $\sim 10$ s for a pair-wise alignment (11,27); it is implemented in $i$SARST because of the excellent ability of order-independent structural alignment (27). Structurally similar proteins with different topologies can be identified by SAMO, which may help to reveal the evolutionary mechanisms of protein structure and function. Values of RMSD and alignment size calculated by refinement engines will be integrated into a single measure called structural diversity defined by Lu (35):

$$\text{structure diversity} = \frac{\text{RMSD}}{\left( \text{avg. size of } q, s \right)^{1.5}}$$

where $\text{avg.}(L_q, L_s)$ is the average length of the query and subject proteins. A lower structural diversity stands for a higher structural similarity. This measure is used to re-order the raw hit list.

When running CPSARST, the refinement process is more complicated since two rounds of alignments shall be done, with and without circularly permuting the PDB structure (11). Only those hits with improved structural similarities to the query protein with a circularly permuting manipulation of the PDB file will be output as final CP candidates.

Indexes like RMSD and alignment size may show the structural relationships between proteins; however, to understand their functional relationship properly, one may still need to examine the structure-based sequence alignment. We have implemented SE algorithm (30) to promote the quality of structure-based sequence alignments made by the refinement engines. Sequence identity and similarity values are provided by $i$SARST, too. Amino acids are considered to be similar if they have positive pairing scores in the BLOSUM62 matrix (36).

Multiprocessor implementations

$i$SARST is now running on an IBM BladeCenter system plus several linux machines (Supplementary Table S1). The cluster environment was established with Rocks operation system. Programs, structure source files and cached data stored on the master node were shared with slave nodes through Network File System (NFS). The user interface and most server-side programs are written in PHP language in a modularized way. The search engine, blastall v.2.2.13, is an intra-machine parallel program. We discovered that when the number of paralleling threads was set as twice the number of processors contained in a machine, it showed the highest speed. Here, we do not use mpiBLAST (37) because the time cost of distributing calculation works to other nodes is relatively high, i.e. several seconds in our preliminary tests. In the refinement stage, aligning one subject protein to the query structure is treated as an individual task. To deal with as many tasks in parallel as possible, each node server is set to run a number of threads according to the number of processors it possesses. Tasks are distributed to slave nodes by programs written in MPI C and PHP. To ensure a quick response to the user, the assignment principles are as follows. (i) Nodes responding faster are assigned with more tasks. (ii) Tasks arriving at similar time have the same priority to be carried out. (iii) There is at least one thread in each node coping with the tasks in a random order, and thus even those users who submit queries much later than others will still get quick responses from $i$SARST.

EXPERIMENTS

As a searching service, $i$SARST has been evaluated with information retrieval experiments using the same dataset

<table>
<thead>
<tr>
<th>Hit list size</th>
<th>Avg. recall (%)</th>
<th>Avg. running time with different refinement engines (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td>100</td>
<td>75.4</td>
<td>3.11</td>
</tr>
<tr>
<td>250</td>
<td>82.9</td>
<td>4.88</td>
</tr>
<tr>
<td>500</td>
<td>85.1</td>
<td>7.78</td>
</tr>
<tr>
<td>1000</td>
<td>87.3</td>
<td>13.38</td>
</tr>
<tr>
<td>2500</td>
<td>91.0</td>
<td>29.69</td>
</tr>
<tr>
<td>5000</td>
<td>93.9</td>
<td>61.31</td>
</tr>
<tr>
<td>10000</td>
<td>96.8</td>
<td>102.46</td>
</tr>
<tr>
<td>25000</td>
<td>99.6</td>
<td>242.38</td>
</tr>
<tr>
<td>34055</td>
<td>100.0</td>
<td>320.89</td>
</tr>
</tbody>
</table>

Query and target databases used in these information retrieval experiments are the same as those in (31) and (9). The target database contains 34 055 protein domains collected from SCOP. Eighty processors were recruited to share the calculations. Without this multi-processor system, the running time on a single machine can be approximately 60 times longer. For instance, at 100% recall level, when FAST was applied to align one query to all target proteins, it took 19 003 s in average.
as Aung and Tan (31) and Lo et al. (9). We first found that iSARST exactly preserves the high average precisions of its refinement engines at any recall level. For instance, at an 85.0% average recall, when FAST is used as the refinement engine, the average precision of iSARST is 85.2%, the same as that of FAST evaluated in (9). As shown in Table 1, to reach this level of average recall, iSARST only has to retrieve 500 hits from this 34,055 polypeptide database, and superimposing these 500 protein pairs by using FAST takes only 7.8 s when 80 processors are recruited.

To know the performance of iSARST when the number of coexisting users is large, we used a number of client programs to execute it simultaneously. The results (Supplementary Figure S2) indicated that, the time cost in database searching and the responding time of

![Figure 2](https://academic.oup.com/nar/article-abstract/37/suppl_2/W545/1127839)

**Figure 2.** Final output of iSARST. (a) Hit list. This list can be re-ordered according to various indexes and protein functions by clicking column titles. Functions of the top 5 hits are summarized and highlighted in red. Any protein listed here can be re-submitted to perform a new round of search simply by clicking the searching icon. Several filtering and operational parameters are adjustable in this page. (b) Structure inspection tools and a circularly permuted structural alignment. PDB entries 1dglA (the fifth letter is the chain ID) and 1gs9A are lectins from *Dioclea grandiflora* (40) and protein ERGIC-53 from *Rattus norvegicus* (41), respectively; they are carbohydrate binding proteins, a large family in which many CP cases have been identified. The natural CP relation between these two proteins can be detected by iSARST, even if their sequence identity is merely 10%. Aligned residue pairs are listed in the right frame. The original structure-based sequence alignment made by the refinement engine, e.g. TM align (29) in this case, and the alignment improved by SE (30) are shown in the lower region. The circularized sequence alignment graph in the center is useful to identify CP. In this example, these proteins can be well aligned only when the 127 amino terminal residues of 1DGL are permuted to its carboxyl terminus. The dot matrix plot is drawn in a way that the darkness of a residue pair is in proportion to its score defined in BLOSUM62 (36). In addition, residues aligned by the refinement engine are colored green. When there is a CP relationship, two parallel green lines can be observed. (c) Results of a co-linear structural alignment. To confirm the existence of a CP, one can compare the results made by co-linear and circularly permuted alignments. As shown in this case, these two circular permutants can only be partially aligned in the co-linear mode. The alignment size is much smaller than that in (b). Besides, there are more unaligned buds in the circularized graph and only one green line can be seen in the dot matrix plot.
refinement engine rise only linearly as the number of simultaneous submissions \( (n) \) increases. To the end, \( iSARST \) has a time complexity of \( O(n) \).

WEB SERVER DESCRIPTION

Input and the searching page

The query interface of \( iSARST \) accepts several different types of input, inclusive of (i) one or more PDB/SCOP entry IDs, (ii) a single PDB file or (iii) an archive file consisting of many protein structures in PDB format. After users submit the query data, a temporary searching page will appear to show the session ID and raw hit list. As the refinement process goes on, users can simultaneously see the progression and structural superimpositions; instead, they may close the browser and later on retrieve the results by (iv) specifying session IDs in the query interface. \( iSARST \) will also automatically make a list of previous sessions when they return, provided that cookies are enabled in their browsers.

Output: hit list

Primary outputs of \( iSARST \) are tables listing co-linear or circularly permuted structural homologs of the query proteins (Figure 2a). In the hit list page, there are two selection menus helping users switch to other previous queries. The list can be re-ordered according to RMSD, alignment sizes, structural diversities, sequence identities, functions, etc. Functions of the five hits with the highest structural similarity scores are summarized and highlighted to assist those who want to make a quick functional assignment. Any protein in the list can be re-submitted as a new query by a simple click, which makes successive database searches very easy. If the search engine is CPSARST, some extra filtering parameters will appear here. Users can adjust them based on their requirements or the property of query proteins. Definitions and suggestions to the use of these parameters can be found in (11).

Output: structure inspection page

Structure superimpositions can be downloaded through the hit list page or examined in an interactive inspection tool (Figure 2b and c). The structure inspection page provides a graphical display of the superimposition, which can be rotated, re-sized and shown in several modes such as cartoon, space-filled or ball-and-stick. When there is a CP relationship detected, C-\( \alpha \) atoms of terminal residues are drawn as balls so that their different locations can be easily recognized. Besides, two proteins are colored very differently; boundaries between the lighter and darker colors are the locations of CP site. Structure-based sequence alignment is shown as (i) a plain text representing unaligned regions as gaps and (ii) a graph of circularized text in which unaligned regions are drawn as budding loops. A smaller number or size of the loops stands for a larger number of residues that can be well-aligned. This circularized alignment is helpful to identify CP relationships, especially when the difference between co-linear and circularly permuted alignments is obvious. If some kind of structural rearrangement, inclusive of CP, had occurred between the aligned proteins, more than one colored segments can be seen in the dot matrix plot embedded here. SE algorithm (30) is implemented in this page to provide an improved structure-based sequence alignment, in which corresponding functional residues can be better aligned (30) and this may help users more correctly derive the functional relatedness between proteins.

APPLICATIONS AND FUTURE WORKS

As a rapid, accurate and versatile protein structural similarity search web server, \( iSARST \) provides user-friendly interfaces and informative outputs for scientists to examine protein structures and do functional annotations. Its modulized design permits follow-up integrations of new searching and refinement methods and thus \( iSARST \) is supposed to be a good platform for bioinformatics researchers to test new algorithms. In the near future, we will broaden the capabilities of \( iSARST \) by adding new modules that can specifically detect other interesting protein structural relationships such as 3D domain swapping (38) and non-CPs (39).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank our colleague Chun-Ting Yeh who set up the IBM BladeCenter system; Ling-Yun Wu, Yong Wang and Prof. Luonan Chen (Shanghai University, China) for the application of SAMO; Chin-Hsien Tai and Dr Byungkook Lee (National Cancer Institute, National Institutes of Health, USA) for providing us helpful instructions in the implementation of SE; and to the authors of BLAST, FAST and TM-align, which are extensively used in this work.

FUNDING

National Science Council, Taiwan, R.O.C. (grant numbers 96-3112-B-007-006, 97-2752-B-007-003-PAE and 97-3112-B-007-007). Funding for open access charge: National Science Council, Taiwan, R.O.C. (grant number 97-3112-B-007-007).

Conflict of interest statement. None declared.

REFERENCES