Published online 6 May 2009 *Nucleic Acids Research, 2009, Vol. 37, Web Server issue* ***W545–W551***
doi:10.1093/nar/gkp291

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

Wei-Cheng Lo[1], Che-Yu Lee[1], Chi-Ching Lee[1,2] and Ping-Chiang Lyu[1,3,*]

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

Received February 22, 2009; Revised and Accepted April 14, 2009

ABSTRACT

***i*SARST is a web server for efficient protein structural similarity searches. It is a multi-processor, batch-processing and integrated implementation of several structural comparison tools and two database searching methods: SARST for common structural homologs and CPSARST for homologs with circular permutations. *i*SARST allows users submitting multiple PDB/SCOP entry IDs or an archive file containing many structures. After scanning the target database using SARST/CPSARST, the ordering of hits are refined with conventional structure alignment tools such as FAST, TM-align and SAMO, which are run in a PC cluster. In this way, *i*SARST achieves a high running speed while preserving the high precision of refinement engines. The final outputs include tables listing co-linear or circularly permuted homologs of the query proteins and a functional summary of the best hits. Superimposed structures can be examined through an interactive and informative visualization tool. *i*SARST provides the first batch mode structural comparison web service for both co-linear homologs and circular permutants. It can serve as a rapid annotation system for functionally unknown or hypothetical proteins, which are increasing rapidly in this post-genomics era. The server can be accessed at http://sarst.life.nthu.edu.tw/iSARST/.**

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,

*To whom correspondence should be addressed. Tel: +886 3 5742762; Fax: +886 3 5715934; Email: lslpc@life.nthu.edu.tw

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 × 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 *i*SARST (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 *i*SARST 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), *i*SARST 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 *i*SARST are designed in a way that structural examinations, functional assignments and successive database searches can be carried out conveniently. Server side programs are modulized; 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 *i*SARST 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 *i*SARST 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 *i*SARST 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 ϕ and ψ angles along its backbone. These 1D structural strings are called Ramachandran (RM) strings.

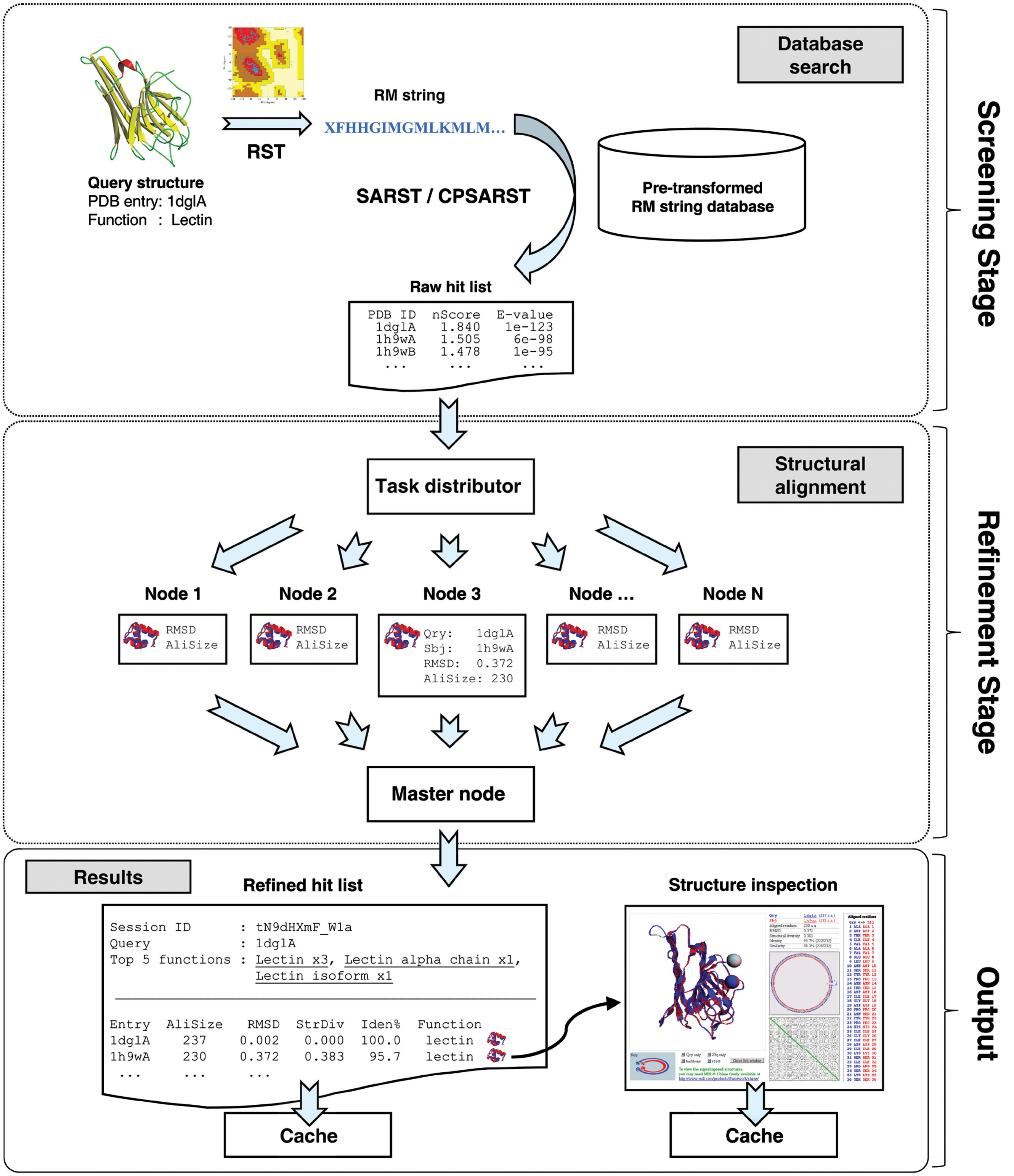
Structural similarity searches

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 cut-offs. 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 permutants. 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,

$$\frac{score_{dl}}{score_{nl}} > 1 \qquad \mathbf{1}$$

$$\log_{10}\left(\frac{E\text{−}value_{dl}}{E\text{−}value_{nl}}\right) > 0.5 \qquad \mathbf{2}$$

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



```
PDB ID   nScore   E-value
 1dglA    1.840    1e-123
 1h9wA    1.505     6e-98
 1h9wB    1.478     1e-95
  ...      ...       ...
```

```
Qry:     1dglA
Sbj:     1h9wA
RMSD:    0.372
AliSize: 230
```

```
Session ID      : tN9dHXmF_W1a
Query           : 1dglA
Top 5 functions : Lectin x3, Lectin alpha chain x1,
                  Lectin isoform x1

Entry  AliSize   RMSD   StrDiv  Iden%  Function
1dglA    237     0.002   0.000  100.0   lectin
1h9wA    230     0.372   0.383   95.7   lectin
 ...     ...      ...
```

Figure 1. Flowchart of *i*SARST. 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 *i*SARST 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 ~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):

$$\text{structure diversity} = \frac{\text{RMSD}}{\left(\frac{\text{alignment size}}{\text{avg}(L_q, L_s)}\right)^{1.5}} \qquad \mathbf{3}$$

where 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 modulized 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

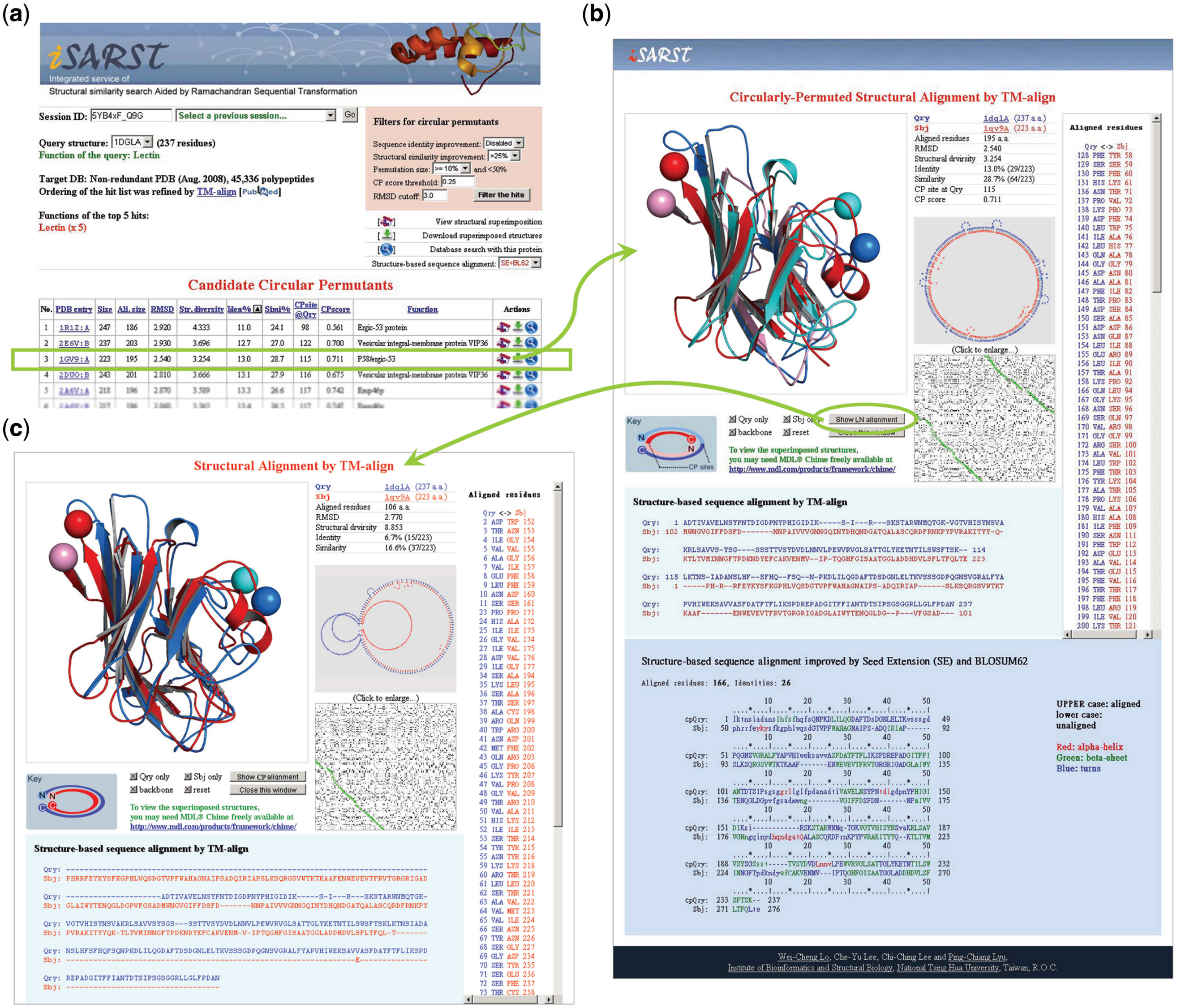
Table 1. Average recall and running time of *i*SARST over various sizes of hit list

Hit list size	Avg. recall (%)	Avg. running time with different refinement engines (s)		
		FAST	TM-align	SAMO
100	75.4	3.11	4.03	19.94
250	82.9	4.88	6.07	30.45
500	85.1	7.78	9.41	47.43
1000	87.3	13.38	15.46	77.89
2500	91.0	29.69	32.47	167.15
5000	93.9	61.31	66.47	295.33
10 000	96.8	102.46	130.45	506.21
25 000	99.6	242.38	273.15	1184.95
34 055	100.0	320.89	364.91	1574.95

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 *i*SARST exactly preserves the high average precisions of its refinement engines at any recall level. For instance, at a 85.0% average recall, when FAST is used as the refinement engine, the average precision of *i*SARST is 85.2%, the same as that of FAST evaluated in (9). As shown in Table 1, to reach this level of average recall, *i*SARST 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 *i*SARST 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



(a)

iSARST
Integrated service of
Structural similarity search Aided by Ramachandran Sequential Transformation

Session ID: 5YB4xF_Q9G Select a previous session... Go

Query structure: 1DGLA (237 residues)
Function of the query: Lectin

Target DB: Non-redundant PDB (Aug. 2008), 45,336 polypeptides
Ordering of the hit list was refined by TM-align [PubMed]

Functions of the top 5 hits:
Lectin (x 5)

Filters for circular permutants

Sequence identity improvement: Disabled
Structural similarity improvement: >25%
Permutation size: >= 10% and <50%
CP score threshold: 0.25
RMSD cutoff: 3.0 Filter the hits

View structural superimposition
Download superimposed structures
Database search with this protein
Structure-based sequence alignment: SE+BL62

Candidate Circular Permutants

No.	PDB entry	Size	Ali. size	RMSD	Str. diversity	Iden%	Simi%	CPsite @Qry	CPscore	Function	Actions
1	1R1Z:A	247	186	2.920	4.333	11.0	24.1	98	0.561	Ergic-53 protein	
2	2E6V:B	237	203	2.930	3.696	12.7	27.0	122	0.700	Vesicular integral-membrane protein VIP36	
3	1GV9:A	223	195	2.540	3.254	13.0	28.7	115	0.711	P58/ergic-53	
4	2DUO:B	243	201	2.810	3.666	13.1	27.9	116	0.675	Vesicular integral-membrane protein VIP36	
5	2A6V:A	218	196	2.870	3.589	13.3	26.6	117	0.742	Emp46p	
6	2A6V:B	217	196	2.860	3.565	13.4	26.3	117	0.747	Emp46p	

(b)

iSARST

Circularly-Permuted Structural Alignment by TM-align

Qry	1dqlA (237 a.a.)
Sbj	1qv9A (223 a.a.)
Aligned residues	195 a.a.
RMSD	2.540
Structural drvirsity	3.254
Identity	13.0% (29/223)
Similarity	28.7% (64/223)
CP site at Qry	115
CP score	0.711

(Click to enlarge...)

Aligned residues

```
Qry <-> Sbj
128 PHE TYR 58
129 SER SER 59
130 PHE PHE 60
131 HIS LYS 61
136 ASN THR 71
137 PRO VAL 72
138 LYS PRO 73
139 ASP PHE 74
140 LEU TRP 75
141 ILE ALA 76
142 LEU HIS 77
143 GLN ALA 78
144 GLY GLY 79
145 ASP ASN 80
146 ALA ALA 81
147 PHE ILE 82
148 THR PRO 83
149 ASP SER 84
150 SER ALA 85
151 ASP ASP 86
153 ASN GLN 87
154 LEU ILE 88
155 GLU ARG 89
156 LEU ILE 90
157 THR ALA 91
158 LYS PRO 92
166 GLN LEU 94
167 GLY LYS 95
168 ASN SER 96
169 SER GLN 97
170 VAL ARG 98
171 GLY GLY 99
172 ARG SER 100
173 ALA VAL 101
174 LEU TRP 102
175 PHE THR 103
176 TYR LYS 104
177 ALA THR 105
178 PRO LYS 106
179 VAL ALA 107
180 HIS ALA 108
181 ILE PHE 109
190 SER ASN 111
191 PHE TRP 112
192 ASP GLU 113
193 ALA VAL 114
194 THR GLU 115
195 PHE VAL 116
196 THR THR 117
197 PHE PHE 118
198 LEU ARG 119
199 ILE VAL 120
200 LYS THR 121
```

Key
N C C N CP sites

☒ Qry only ☒ Sbj only ☒ backbone ☒ reset Show LN alignment Close this window

To view the superimposed structures,
you may need MDL® Chime freely available at
http://www.mdl.com/products/framework/chime/

Structure-based sequence alignment by TM-align

```
Qry:   1 ADTIVAVELNSYPNTDIGDPNYPHIGIDIK-----S-I---R---SKSTARWNMQTGK-VGTVHISYNSVA
Sbj: 102 MWNGVGIFFDSFD-------NNPAIVVVGNNGQINYDHQNDGATQALASCQRDFRNKPYPVRAKITYY-Q-

Qry:     KRLSAVVS-YSG----SSSTTVSYDVDLNNVLPEWVRVGLSATTGLYKETNTILSWSFTSK-- 114
Sbj:     KTLTVMINNGFTPDKNDYEFCAKVENMV--IP-TQGHFGISAATGGLADDHDVLSFLTFQLTE 223

Qry: 115 LKTNS-IADANSLHF--SFHQ--FSQ--N-PKDLILQGDAFTDSDGNLELTKVSSSGDPQGNSVGRALFYA
Sbj:   1 -----PH-R--RFEYKYSFKGPHLVQSDGTVPFWAHAGNAIPS-ADQIRIAP------SLKSQRGSVWTKT

Qry:     PVHIWEKSAVVASFDATFTFLIKSPDREPADGITFFIANTDTSIPSGSGGRLLGLFPDAN 237
Sbj:     KAAF-------ENWEVEVTFRVTGRGRIGADGLAIWYTENQGLDG--P---VFGSAD--- 101
```

Structure-based sequence alignment improved by Seed Extension (SE) and BLOSUM62

Aligned residues: **166**, Identities: **26**

```
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:    1 lktnsiadanslhfsfhqfsQNPKDLILQGDAFTDsDGNLELTKvsssgd    49
  Sbj:   50 phrrfeykysfkgphlvqsdGTVPFWAHAGNAIPS-ADQIRIAP------    92
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:   51 PQGNSVGRALFYAPVHIweksavvASFDATFTFLIKSPDREPADGITFFI   100
  Sbj:   93 SLKSQRGSVWTKTKAAF-------ENWEVEVTFRVTGRGRIGADGLAIWY   135
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:  101 ANTDTSIPsgsggrllglfpdanadtiVAVELNSYPNtdigdpnYPHIGI   150
  Sbj:  136 TENQGLDGpvfgsadmwng--------VGIFFDSFDN-------NPAIVV   175
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:  151 DIKsi-----------RSKSTARWNMq-TGKVGTVHISYNSvaKRLSAV   187
  Sbj:  176 VGNngqinydhqndgatQALASCQRDFrnKPYPVRAKITYYQ--KTLTVM   223
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:  188 VSYSGSsst-----TVSYDVDLnnvLPEWVRVGLSATTGLYKETNTILSW   232
  Sbj:  224 INNGFTpdkndyefCAKVENMV---IPTQGHFGISAATGGLADDHDVLSF   270
                  10        20        30        40        50
              ....*....|....*....|....*....|....*....|....*....|
cpQry:  233 SFTSK--  237
  Sbj:  271 LTFQLte  276
```

UPPER case: aligned
lower case: unaligned

Red: alpha-helix
Green: beta-sheet
Blue: turns

Wei-Cheng Lo, Che-Yu Lee, Chi-Ching Lee and Ping-Chiang Lyu,
Institute of Bioinformatics and Structural Biology, National Tsing Hua University, Taiwan, R.O.C.

(c)

Structural Alignment by TM-align

Qry	1dqlA (237 a.a.)
Sbj	1qv9A (223 a.a.)
Aligned residues	106 a.a.
RMSD	2.770
Structural drvirsity	8.853
Identity	6.7% (15/223)
Similarity	16.6% (37/223)

(Click to enlarge...)

Aligned residues

```
Qry <-> Sbj
2 ASP TRP 152
3 THR ASN 153
4 ILE GLY 154
5 VAL VAL 155
6 ALA GLY 156
7 VAL ILE 157
8 GLU PHE 158
9 LEU PHE 159
10 ASN ASP 160
11 SER SER 161
23 PRO PRO 171
24 HIS ALA 172
25 ILE ILE 173
26 GLY VAL 174
27 ILE VAL 175
28 ASP VAL 176
29 ILE GLY 177
34 SER ALA 194
35 LYS LEU 195
36 SER ALA 196
37 THR SER 197
38 ALA CYS 198
39 ARG GLN 199
40 TRP ARG 200
41 ASN ASP 201
42 MET PHE 202
43 GLN ARG 203
45 GLY PRO 206
46 LYS TYR 207
47 VAL PRO 208
48 GLY VAL 209
49 THR ARG 210
50 VAL ALA 211
51 HIS LYS 212
52 ILE ILE 213
53 SER THR 214
54 TYR TYR 215
55 ASN TYR 216
59 LYS LYS 218
60 ARG THR 219
61 LEU LEU 220
62 SER THR 221
63 ALA VAL 222
64 VAL MET 223
65 VAL ILE 224
66 SER ASN 225
67 TYR ASN 226
68 SER GLY 227
69 GLY ASP 234
70 SER TYR 235
71 SER GLU 236
72 SER PHE 237
73 THR CYS 238
```

Key
N N C C

☒ Qry only ☒ Sbj only ☒ backbone ☒ reset Show CP alignment Close this window

To view the superimposed structures,
you may need MDL® Chime freely available at
http://www.mdl.com/products/framework/chime/

Structure-based sequence alignment by TM-align

```
Qry: ----------------------------------------------------------------------
Sbj: PHRRFEYKYSFKGPHLVQSDGTVPFWAHAGNAIPSADQIRIAPSLKSQRGSVWTKTKAAFENWEVEVTFRVTGRGRIGAD

Qry: ---------------------ADTIVAVELNSYPNTDIGDPNYPHIGIDIK-----S-I---R---SKSTARWNMQTGK-
Sbj: GLAIWYTENQGLDGPVFGSADMWNGVGIFFDSFD-------NNPAIVVVGNNGQINYDHQNDGATQALASCQRDFRNKPY

Qry: VGTVHISYNSVAKRLSAVVSYSGS---SSTTVSYDVDLNNVLPEWVRVGLSATTGLYKETNTILSWSFTSKLKTNSIADA
Sbj: PVRAKITYYQK-TLTVMINNGFTPDKNDYEFCAKVENM-V-IPTQGHFGISAATGGLADDHDVLSFLTFQL-T-------

Qry: NSLHFSFHQFSQNPKDLILQGDAFTDSDGNLELTKVSSSGDPQGNSVGRALFYAPVHIWEKSAVVASFDATFTFLIKSPD
Sbj: ------------------------------------------------------------------E--------------

Qry: REPADGITFFIANTDTSIPSGSGGRLLGLFPDAN
Sbj: ----------------------------------
```

Figure 2. Final output of *i*SARST. (**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 1gv9A 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 *i*SARST, 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, *i*SARST has a time complexity of O(*n*).

WEB SERVER DESCRIPTION

Input and the searching page

The query interface of *i*SARST 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. *i*SARST 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 *i*SARST 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-α 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, *i*SARST 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 *i*SARST is supposed to be a good platform for bioinformatics researchers to test new algorithms. In the near future, we will broaden the capabilities of *i*SARST 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

1. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
2. Pearson,W.R. (2000) Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.*, **132**, 185–219.
3. Sauder,J.M., Arthur,J.W. and Dunbrack,R.L. Jr. (2000) Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, **40**, 6–22.

4. Yang,J.M. and Tung,C.H. (2006) Protein structure database search and evolutionary classification. *Nucleic Acids Res.*, **34**, 3646–3659.
5. Levine,M., Stuart,D. and Williams,J. (1984) A method for the systematic comparison of the three-dimensional structures of proteins and some results. *Acta Crystallogr*, **A40**, 600–610.
6. Lesk,A.M. (1998) *Proceedings of Prague Stringology Club Workshop '98* Prague, pp. 95–100.
7. Martin,A.C. (2000) The ups and downs of protein topology; rapid comparison of protein structure. *Protein Eng.*, **13**, 829–837.
8. Carpentier,M., Brouillet,S. and Pothier,J. (2005) YAKUSA: a fast structural database scanning method. *Proteins*, **61**, 137–151.
9. Lo,W.C., Huang,P.J., Chang,C.H. and Lyu,P.C. (2007) Protein structural similarity search by Ramachandran codes. *BMC Bioinformatics*, **8**, 307.
10. 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.
11. Lo,W.C. and Lyu,P.C. (2008) CPSARST: an efficient circular permutation search tool applied to the detection of novel protein structural relationships. *Genome Biol.*, **9**, R11.
12. Jeltsch,A. (1999) Circular permutations in the molecular evolution of DNA methyltransferases. *J. Mol. Evol.*, **49**, 161–164.
13. Weiner,J 3rd, Thomas,G. and Bornberg-Bauer,E. (2005) Rapid motif-based prediction of circular permutations in multi-domain proteins. *Bioinformatics*, **21**, 932–937.
14. Tsai,L.C., Shyur,L.F., Lee,S.H., Lin,S.S. and Yuan,H.S. (2003) Crystal structure of a natural circularly permuted jellyroll protein: 1,3-1,4-beta-D-glucanase from Fibrobacter succinogenes. *J. Mol. Biol.*, **330**, 607–620.
15. Lindqvist,Y. and Schneider,G. (1997) Circular permutations of natural protein sequences: structural evidence. *Curr. Opin. Struct. Biol.*, **7**, 422–427.
16. Vogel,C. and Morea,V. (2006) Duplication, divergence and formation of novel protein topologies. *Bioessays*, **28**, 973–978.
17. Qian,Z. and Lutz,S. (2005) Improving the catalytic activity of Candida antarctica lipase B by circular permutation. *J. Am. Chem. Soc.*, **127**, 13466–13467.
18. Anantharaman,V., Koonin,E.V. and Aravind,L. (2001) Regulatory potential, phyletic distribution and evolution of ancient, intracellular small-molecule-binding domains. *J. Mol. Biol.*, **307**, 1271–1292.
19. Todd,A.E., Orengo,C.A. and Thornton,J.M. (2002) Plasticity of enzyme active sites. *Trends Biochem. Sci.*, **27**, 419–426.
20. Li,L. and Shakhnovich,E.I. (2001) Different circular permutations produced different folding nuclei in proteins: a computational study. *J. Mol. Biol.*, **306**, 121–132.
21. Chen,J., Wang,J. and Wang,W. (2004) Transition states for folding of circular-permuted proteins. *Proteins*, **57**, 153–171.
22. Bulaj,G., Koehn,R.E. and Goldenberg,D.P. (2004) Alteration of the disulfide-coupled folding pathway of BPTI by circular permutation. *Protein Sci.*, **13**, 1182–1196.
23. Kojima,M., Ayabe,K. and Ueda,H. (2005) Importance of terminal residues on circularly permutated Escherichia coli alkaline phosphatase with high specific activity. *J. Biosci. Bioeng.*, **100**, 197–202.
24. Ostermeier,M. (2005) Engineering allosteric protein switches by domain insertion. *Protein Eng. Des. Sel.*, **18**, 359–364.
25. Galarneau,A., Primeau,M., Trudeau,L.E. and Michnick,S.W. (2002) Beta-lactamase protein fragment complementation assays as in vivo and in vitro sensors of protein protein interactions. *Nat. Biotechnol.*, **20**, 619–622.
26. Baird,G.S., Zacharias,D.A. and Tsien,R.Y. (1999) Circular permutation and receptor insertion within green fluorescent proteins. *Proc. Natl Acad. Sci. USA*, **96**, 11241–11246.
27. Chen,L., Wu,L.Y., Wang,Y., Zhang,S. and Zhang,X.S. (2006) Revealing divergent evolution, identifying circular permutations and detecting active-sites by protein structure comparison. *BMC Struct. Biol.*, **6**, 18.
28. Zhu,J. and Weng,Z. (2005) FAST: a novel protein structure alignment algorithm. *Proteins*, **58**, 618–627.
29. Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.
30. Tai,C.H., Vincent,J.J., Kim,C. and Lee,B. (2009) SE: an algorithm for deriving sequence alignment from a pair of superimposed structures. *BMC Bioinformatics*, **10 (Suppl. 1)**, S4.
31. Aung,Z. and Tan,K.L. (2004) Rapid 3D protein structure database searching using information retrieval techniques. *Bioinformatics*, **20**, 1045–1052.
32. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
33. Chandonia,J.M., Hon,G., Walker,N.S., Lo Conte,L., Koehl,P., Levitt,M. and Brenner,S.E. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res.*, **32**, D189–D192.
34. Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.
35. Lu,G. (2000) Top: a new method for protein structure comparisons and similarity searches. *J. Appl. Cryst.*, **33**, 176–183.
36. Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.
37. Lin,H., Ma,X., Chandramohan,P., Geist,A. and Samatova,N. (2005) *IEEE International Parallel & Distributed Processing Symposium*. Denver, CO.
38. Liu,Y. and Eisenberg,D. (2002) 3D domain swapping: as domains continue to swap. *Protein Sci.*, **11**, 1285–1299.
39. Bujnicki,J.M. (2002) Sequence permutations in the molecular evolution of DNA methyltransferases. *BMC Evol. Biol.*, **2**, 3.
40. Rozwarski,D.A., Swami,B.M., Brewer,C.F. and Sacchettini,J.C. (1998) Crystal structure of the lectin from Dioclea grandiflora complexed with core trimannoside of asparagine-linked carbohydrates. *J. Biol. Chem.*, **273**, 32818–32825.
41. Velloso,L.M., Svensson,K., Schneider,G., Pettersson,R.F. and Lindqvist,Y. (2002) Crystal structure of the carbohydrate recognition domain of p58/ERGIC-53, a protein involved in glycoprotein export from the endoplasmic reticulum. *J. Biol. Chem.*, **277**, 15979–15984.