FAMCS: Finding All Maximal Common Substructures in Proteins

Zhen Yao*, Juan Xiao, Anthony K. H. Tung, and Wing Kin Sung

Department of Computer Science, National University of Singapore, Singapore 117543.

Finding the common substructures shared by two proteins is considered as one of the central issues in computational biology because of its usefulness in understanding the structure-function relationship and application in drug and vaccine design. In this paper, we propose a novel algorithm called FAMCS (Finding All Maximal Common Substructures) for the common substructure identification problem. Our method works initially at the protein secondary structural element (SSE) level and starts with the identification of all structurally similar SSE pairs. These SSE pairs are then merged into sets using a modified Apriori algorithm, which will test the similarity of various sets of SSE pairs incrementally until all the maximal sets of SSE pairs that deemed to be similar are found. The maximal common substructures of the two proteins will be formed from these maximal sets. A refinement algorithm is also proposed to fine tune the alignment from the SSE level to the residue level. Comparison of FAMCS with other methods on various proteins shows that FAMCS can address all four requirements and infer interesting biological discoveries.

Key words: protein structure, maximal common substructures, secondary structure element (SSE)

Introduction

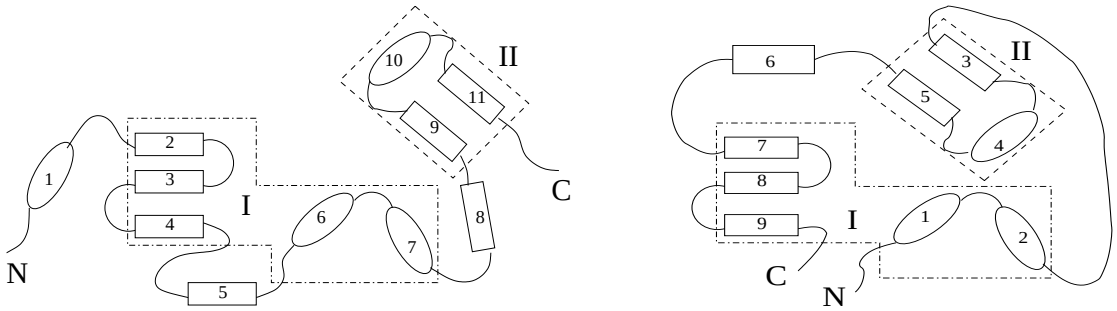
The rapid growth in the number of discovered protein structures and the interest in using such information for understanding the protein structure-function relationship have led to the great demand for protein structure analysis tools. One kind of important analysis is to identify the common substructures shared among proteins. Common substructures can be used to infer common evolutionary origins, to perform protein/motif classification, and to predict the structure-function relationship.

Strictly speaking, it is the maximal common substructures (MCSs) of two proteins that are desired to be discovered. By an MCS, we mean a set of structural elements that is common in both proteins, and it is impossible to include one more pair of elements (one from each protein) to make the enlarged set still common to both proteins. For example, in Figure 1, proteins P and Q share two MCSs, designated as regions I and II, respectively. Though there is a set of structurally common β-sheets in both proteins—2–4 in P and 7–9 in Q, it is not an MCS because the β-sheets can be combined with a pair of α-helices, and the resulting structure is still common. However, the combination of substructures I and II are not structurally the same in both proteins, hence, they remain as two MCSs. The formal definition of MCS is given in Methods.

Despite the large number of algorithms developed, there are still several issues that need to be addressed in the MCS identification problem:

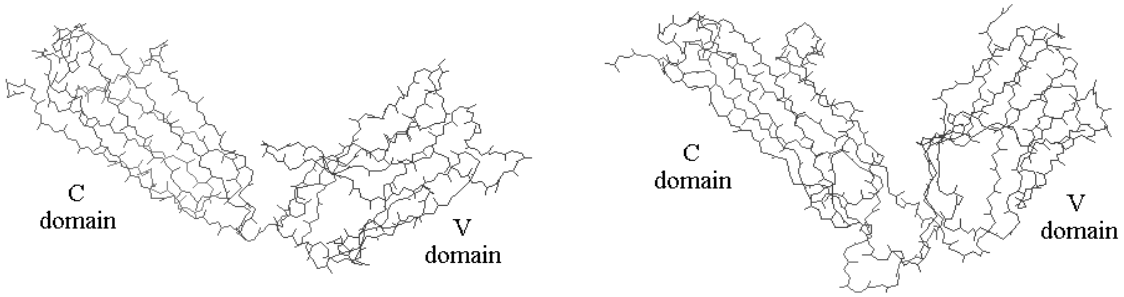
1. Finding *all* MCSs. Many proteins have multi-domains, where each domain has a particular functionality. Proteins might have several similar domains, especially if they belong to the same family, but the relative position of these domains could be different in different proteins. For example, as shown in Figure 2, both the immunoglobulin fab fragment (1MCP, chain L) and the murine T-cell antigen receptor (1TCR, chain B) have a constant (C) and a variable (V) domain. The angle between these two domains in 1MCPl is obtuse, while a significant bend results in a sharp angle in 1TCRb. Both domains and their different relative positions are interesting to biologists. Methods searching for a single good alignment of two proteins are however unable to obtain this answer since either only one of the common domains is aligned well or both of them are aligned with a large RMSD (Root Mean Square Deviation) while missing the different relative positions between them.

*** Corresponding author.**
E-mail: yaozhen@alumni.nus.edu.sg



A. Protein P **B**. Protein Q

Fig. 1 Simplified 3D structures of proteins P and Q. α-helix is represented by ellipse, while β-sheet is represented by rectangle.



A. 1MCP, chain L (1MCPl) **B**. 1TCR, chain B (1TCRb)

Fig. 2 The 3D structures of the backbone of the immunoglobulin fab fragment (1MCP, chain L) and the murine T-cell antigen receptor (1TCR, chain B). They have two domains in common: the constant (C) and the variable (V) domain, but their relative positions are different.

2. Discover MCSs in non-topological case. *Non-topological alignment* occurs when the alignment order is different from the backbone order. Structure I in Figure 1 is an example of the MCS in non-topological case. There are also many real examples where some of them are produced by sequence rearrangement (*1*, *2*) or by convergent evolution (*3*, *4*). The importance of addressing non-topological case was discussed in previous studies (*5*–*7*). However, most existing solutions cannot handle this issue.

3. Identify MCSs involving multi-chains. A functional group may span on several polypeptide chains in a multi-chain protein. For instance, in the *met* repressor-operator complex, the DNA-binding site consists of two β-strands, one from each chain (*8*). Many existing methods, especially those for structural alignment, can only work on a single chain from each protein.

4. Rank and select the results. The total number of MCSs of two proteins can be very large, sometimes thousands, depending on the protein sizes and how they are similar to each other. With such a huge result set, it is impractical for biologists to dig out useful information. Therefore, it is important to sort the MCSs. Moreover, it is not true that each MCS corresponds to a structural domain. Rather, many MCSs have intersecting regions (the same alignment portions) or conflicting regions (the same elements of one protein are aligned with different elements of the other protein). Such MCSs cannot co-exist in proteins. Though subtle structural incompatibility can be mined out from these MCSs, biologists are more interested in co-present MCSs where each might be a domain. Thus, besides the ability to discover all MCSs, it is also desired to have the means to select a subset that includes most significant MCSs, which are neither intersecting nor conflicting, though only one of them can be aligned well at a time.

There are currently two different approaches for solving the problem. The first approach is to deduce the answer from structural alignment problem, where the 3D structures of two proteins, or more often two polypeptide chains, are to be superimposed so that a similarity score function is optimized. Usually, a better score corresponds to an alignment with smaller RMSD and more aligned structural elements (usually

residues). All of the aligned structural elements form an MCS. The general problem of structural alignment has been proven to be NP-hard (*9*). Thus, during the recent decade, many heuristic methods have been proposed using various techniques: Monte Carlo optimization [DALI (*5*, *10*)], Dynamic programming [STRUCTAL (*11*), LOCK (*12*)], Graph theory [VAST (*13*, *14*), SARF2 (*15*)], Combinatorial extension of alignment path [CE (*16*)], Geometric hashing [MASS (*17*)], Hidden Markov models [SCALI (*6*)], Genetic algorithm [Ref. *18*, *19*; K2 (*7*)], Clustering-based method [FAST (*20*)], and so on. The review can be found in the work by Signh and Brutalg (http://cmgm.stanford.edu/∼brutlag/Papers/singh00.pdf).

The second approach includes various methods (*21*–*27*). However, except Grindley's method (*22*), the answers of most of them are essentially the aligned parts, as if these two proteins are subjected to structural alignments but not ALL common substructures. This is supposed to be the critical difference between the problem of structural alignment identification and that of common substructure identification. The techniques they used are similar as those used in structural alignment methods.

As far as we know, in the first approach, only MASS (*17*) can find more than one common substructure. Since it targets at multiple proteins, to reduce time complexity, heuristics is incorporated into geometric hashing, which makes the result not complete. Griendley's method (*22*) in the second approach is the only one we know that can discover all MCSs. It achieves so by finding all maximal cliques in a correspondence graph. However, they do not have a ranking scheme, nor their method has the ability to select a co-present subset. And their answer stops at the SSE (secondary structural element) level while it is usually desired to know the residue correspondence.

In this paper, we propose an algorithm called FAMCS (Finding All Maximal Common Substructures) for identifying common substructures between proteins. It can address all the above four issues. In order to achieve efficiency, FAMCS works firstly on the secondary structure level to prevent processing a large number of residues, and then employs an orientation-invariant representation to avoid the expensive cost performing rotation and transformation to obtain optimal orientation for the two proteins under investigation.

FAMCS works by first identifying all structurally similar SSE pairs, which are then merged into substructures containing multiple SSE pairs using a modified Apriori algorithm (*28*). The algorithm deduces the answer level by level. At the i^{th} level, candidate substructures containing i pairs of SSEs are generated from common substructures with $i-1$ SSE pairs found at the $(i-1)^{\text{th}}$ level. If a candidate is proven to be still common to both proteins, it will then be used to generate candidates for the next level of search. Eventually, all maximal sets of SSE pairs that are deemed to be similar will be found, which represent the MCSs. They are then ranked according to the size and the similarity score. An optional step is provided to select a co-present subset that contains most significant MCSs. As it could be desirable to know the exact residue correspondence, FAMCS also provides a simple heuristic algorithm to refine the answer to the residue level. This is necessary only if the users are interested to know more details after they look at the result at the SSE level.

Results

Implementation and settings

We conducted experiments on various protein pairs to access the robustness of FAMCS on SUN's E450 running solaris, in which FAMCS has been implemented in C++. The protein structures were taken from the Protein Data Bank (*29*). Structural information of secondary structures was recovered by a modified version of WebMol (*30*). Basically, its process is: (1) use the DSSP algorithm (*31*) to define the secondary structures of the input protein, and (2) calculate the dihedral angle and the closest distance among all SSEs to fill up the AD matrix, which is a part of our protein representation.

Parameter tuning

There are a couple of parameters in FAMCS: thresholds for length difference (T_l), angle difference (T_a), distance difference (T_d), similarity threshold (T_{sim}), weight for angle similarity (W_a), and weight for distance similarity (W_d).

In order to tune the parameters, we selected ten protein pairs (Table 1) that are either having known common substructures with important biological function (the first five pairs), or from Chew's paper (Ref. *27*; the next three pairs), or randomly selected from different families in the SCOP database (Ref. *32*;

Table 1 Parameter Tuning for T_l, T_a, and T_d

T_l, T_a, T_d	Protein Pair									
	1MCPl	1GGGa	1F4N	1B0U	1MJP	2CRO	1A1Ea	3HSC	1HYWa	1GMI
	1TCRb	1WDNa	256Ba	1AM1	1ECR	2WRPr	2ABL	2YHX	3UBPa	1HYWa
7, 30, 2	*		*		*	*	*		*	*
7, 30, 3	*		*		*	*	*		*	*
7, 30, 4	*		*	*	*	*	*	*	*	*
7, 45, 2	*		*		*	*	*		*	*
7, 45, 3	*	*	*	*	*	*	*	*	*	*
7, 45, 4	*	*	*		*	*	*		*	*
7, 60, 2	*	*	*		*	*	*		*	*
7, 60, 3	*	*	*	*	*	*	*	*	*	*
7, 60, 4	*	*	*		*	*	*		*	*

* The one that produces the best result. (7, 45, 3) and (7, 60, 3) generated the optimal results.

Table 2 Parameter Tuning for W_d and W_a

W_d, W_a	Protein Pair									
	1MCPl	1GGGa	1F4N	1B0U	1MJP	2CRO	1A1Ea	3HSC	1HYWa	1GMI
	1TCRb	1WDNa	256Ba	1AM1	1ECR	2WRPr	2ABL	2YHX	3UBPa	1HYWa
0.3, 0.7			*		*		*	*	*	*
0.5, 0.5	*	*	*	*	*	*	*	*	*	*
0.7, 0.3			*		*	*	*		*	*

* The one that produces the best result.

the rest). The parameter settings were evaluated by the results' conformity with the known common substructures, or their RMSD measured at the residue level.

Since SSE length does not play an important role in the 3D structure determination (though it affects the residue level alignment), we would like to loosen this threshold. Five and seven amino acids were tested in the tuning process, respectively. From the study of the distribution of SSE angle and distance (*33*), angle values evenly distribute over the entire range, while distance values skew at 8Å to 16Å. We decided our trial values centered at 1/4 of the popular ranges, and varied by 15° and 1Å respectively. Namely, 30°, 45°, and 60° for T_a, while 2Å, 3Å, and 4Å for T_d. The tuning results are shown in Table 1 (only those for $T_l = 7$ are shown since $T_l = 5$ performed worse than or equal to $T_l = 7$ for all protein pairs). Though both $T_l = 7$, $T_a = 60$, $T_d = 3$ and $T_l = 7$, $T_a = 45$, $T_d = 3$ generated the optimal results, the former took much longer time. Thus, the later was chosen as the default setting, and $T_{sim} = 0$.

To set the weight wisely, three different sets of weight were tried in order to compare the results under three cases: more weight on distance, more weight on angle, and equal weight. The weight values were evaluated in the same way as threshold tuning. We observed that the common substructures found by different weight do not differ much, and equal weight for angle and distance tends to perform well in most cases (Table 2), which is reasonable since both angle and distance are important in structure determination. Therefore, we chose $W_a = 0.5$ and $W_d = 0.5$.

Experiment results

The protein pairs used in method evaluation are the same ones in parameter tuning. We understand that this might degrade its generality and the number of proteins is limited. However, in order to assess the ability to address the four issues outlined in the Introduction, we need to analyze the result in detail, which is impractical on too many protein pairs, and it is hard to find protein pairs with known interesting common substructures. Whereas, FAMCS is able to handle the four issues by its logic. The experiment is to show some real examples. We have tried our best to include proteins of various types and sizes.

We compared our method with both common substructure identification methods and structural alignment algorithms on the ability to handle the four issues. Chew's work (*27*) is a recent method purposely searching for common substructures. DALI and VAST are structural alignment tools that perform the best (http://cmgm.stanford.edu/∼brutlag/Papers/singh00.pdf). The alignment at the SSE level, the RMSD value, and the number of residues aligned for FAMCS, DALI, VAST, and Chew's work for the ten protein pairs are listed in Table 3. The top co-present MCS is shown for FAMCS except for 1MCPl/1TCRb and 1GGGa/1WDNa where the top two are shown.

Table 3 Comparison of FAMCS with DALI, VAST, and Chew's Work*

Protein	Method	SSE alignment	RMSD	Residue No.
1MCPl (all β)	FAMCS	(I) (V domain) **1-4:1-4**, **5-8:6-10**	1.3	71
1TCRb (all β)		(II) (C domain) 9:11, 10:13, **12:15**, **13-15:17-19**	2.6	86
	DALI	**1-4:1-4**, NS:5, **5-8:6-10**, NS+10:NS, **12:15**, **13-15:17-19**	7.3	149
	VAST	**1-4:1-4**, **5-8:6-10**	1.9	95
1GGGa (α/β)	FAMCS	(I) (Middle) **7-9:8-10**, 10:11, **11:12**, **12-15:13-16**	0.5	74
1WDNa (α/β)		(II) (Head+tail) **1-5:1-5**, **16-17:17-18**	0.5	100
	DALI	1:1, **1-5:1-5**, 6:6-7, **7-9:8-10**, **12-15:13-16**, **16-17:17-18**	4.2	174
	VAST	1:1, **1-5:1-5**, NS+11:11-12, **13-14:14-15**	3.4	172
1F4N (all α) multi-chain	FAMCS	1:3, **2:4**, 3:1, 4:2	9.1	90
256Ba (all α)	DALI	1:1, **2:4**, 3:2, 4:3	14.4	91
1B0U (α/β)	FAMCS	5:10, 11:12, 15:3, 19:8, 20:9	4.0	45
1AM1 (α/β)	DALI	No similarity detected	N/A	N/A
1MJP (all α) multi-chain	FAMCS	1:11, 5:14 (1 and 5 are β-strands on 2 chains), 2:12	3.2	31
1ECRa ($\alpha+\beta$)	DALI	No similarity detected	N/A	N/A
2CRO (all α)	FAMCS	**2-3:4-5** (C_α alignment 14-37:65-88)	0.8	24
2WRPr (all α)	DALI	1(tail):3(tail), **2-3:4-5**, 5:6(tail)+NS	4.7	38
	Chew's	**2-3:4-5** (C_α alignment 16-39:66-89)	3.9	24
1A1Ea ($\alpha+\beta$)	FAMCS	**1-5:6-10** (SH2 domain perfectly matched)	0.82	70
2ABL (all β)	DALI	**1-5:6-10**	1.8	95
	VAST	**1-5:6-10**	1.06	88
	Chew's	NS:NS, **4-5:9-10**	1.29	60
3HSC (α/β)	FAMCS	**1-3:4-6**, **12:8**, **14-17:9-12**, 24:19	2.7	73
2YHX (α/β)	DALI	**1-3:4-6**, 4+NS:NS, 11:7, **12:8**, 13:NS, **14-19:9-12**, 18(tail)-23:NS+15-24+NS, 24:25, 25+NS+26:26-28	5.7	265
	Chew's	16-18:4-6	3.9	28
1HYWa ($\alpha+\beta$)	FAMCS	1:2, 4:1	2.7	34
3UBPa ($\alpha+\beta$)	DALI	1:1, 4:2, NS:5	3.1	39
1GMI (all β)	FAMCS	5:4, 8:3	6.43	17
1HYWa ($\alpha+\beta$)	DALI	No similarity detected	N/A	N/A

* Only the SSE alignments are shown. An aligned segment is presented in the form of "$i:j$" (which means the i^{th} element of the first protein is aligned with the j^{th} element of the second protein), or "i–j:k–l" (which means the i^{th} to the j^{th} element of the first protein are aligned with the k^{th} to the l^{th} element of the second protein). Common alignment pattern found by different methods are highlighted. "NS" represents the structure that is neither α-helix nor β-strand. "i+NS" means the i^{th} SSE followed by a non-SSE part. "i(head)" or "i(tail)" means only the very head or tail portion of the i^{th} SSE participates in the alignment. For FAMCS, SSE alignment of one row is one MCS. Only the top co-present MCSs from FAMCS are shown, except for 1MCPl/1TCRb and 1GGGa/1WDNa, where the top two are displayed. Answers of DALI and VAST are from their web servers (VAST only provides alignment for structural neighbors). Results of Chew's work are taken from its paper (*27*), hence, many data is unavailable. Wherever the data is unavailable, an "N/A" is put in the table.

Though the main goal of our method is effectiveness rather than efficiency, it is still interesting to have an idea of the speed and output size. In Table 4, we show the total number of MCSs found, the number of co-present MCSs, total and break-down time, together with the protein size and the L_2 size (*i.e.* the total number of similar SSE pairs from Step 1 of our algorithm).

Table 4 Result Sizes and Execution Time vs. Protein Sizes*

Protein	Size			MCS No.		Time (second)		
pair	SSE	Residue	L_2	All	Co-present	Total	Step 1	Step 2
1MCPl	15	220	2,030	2,545	2	2,078	1	2,077
1TCRb	19	247						
1GGGa	18	220	915	709	2	53	0	53
1WDNa	18	223						
1F4N	4	60	8	8	1	0	0	0
256Ba	4	106						
1B0U	22	258	558	485	3	13	0	13
1AM1	12	213						
1MJP	4(a)+4(b)	208	32	37	2	0	0	0
1ECRa	19	305						
2CRO	5	64	5	7	1	0	0	0
2WRPr	6	104						
1A1Ea	5	104	56	29	1	0	0	0
2ABL	10	163						
3HSC	26	382	1,072	1,021	5	104	1	103
2YHX	22	457						
1HYWa	4	58	1	3	1	0	0	0
3UBPa	5	100						
1GMI	10	136	3	3	1	0	0	0
1HYWa	4	58						

* The notation L_2 is taken from our algorithm (see Methods). L_2 size essentially is the total number of similar SSE pairs. Step 1 time refers to the time to find all similar SSE pairs; Step 2 time refers to the time to merge common substructures level by level to get all MCSs; total time includes Step 1 time, Step 2 time, and the time to select co-present MCSs.

Discussion

Discover all MCSs

FAMCS can find all MCSs. Different MCSs of the same protein pair can infer interesting structural differences. For the 1MCPl and 1TCRb example in the Introduction, FAMCS successfully identified the V and C domains as two MCSs, which correspond to the first and the second co-present answer respectively. Thus, the user is not only informed of the similarity between the two immune system proteins, but also aware of the different spatial arrangements of domains. However, DALI aligned both domains together. This produces a worse RMSD value, and conceals the different domain spatial relationship. Though VAST achieved a small RMSD value, it only identified the V domain while missing the C domain.

Another interesting example is the conformational change upon the ligand binding of the glutamine-binding protein. The ligand-free form (1GGGa) and the glutamine-bound complex (1WDNa) were compared in FAMCS. The top MCS found correspondence to the middle part of the protein, while the second top MCS comprises the head and tail. Therefore, we can deduce that there are significant changes in the backbone before and after the middle part. It accords well with the data of conformational change: 41.1° in the ϕ angle of Gly89 [in FAMCS's answer, the middle part MCS starts from the 89^{th} C_α (after refining to the residue level, the middle part is C_α alignment 89-96:89-96, 111-146:111-146, 148-176:148-176, the head

and tail part is C_α alignment 5-58:5-58, 183-221:183-221)] and 34.3 ° in the ψ angle of Glu181 (in FAMCS's answer, the tail part starts from the 183$^{\text{th}}$ C_α) (*34*). DALI and VAST aligned all the MCSs as one, which not only results in much worse RMSD values, but also is unable to deduce the interesting structural changes.

Non-topological case

DALI is also able to deal with the non-topological case (*5*). One example is the ROP dimer (1F4N) and the chain A of cytochrome b56 (256Ba). Both DALI and FAMCS detected non-topological structural similarity but in different patterns. FAMCS managed to align almost the same number of residues and achieved a much better RMSD value. It also discovered a sheet of five β-strands of different topology at the ATP-binding site of the histidine permease from *Salmonella typhimurium* (1B0U) and the Hsp90 molecular chaperone (1AM1). However, DALI didn't detect any similarity between these two proteins.

Compare multi-chain proteins as a whole

FAMCS can compare two entire proteins, no matter how many polypeptide chains each of them has. This property is important, as illustrated by the *met* repressor-operator complex (1MJP) and the *Escherichia coli* replication-terminator protein (1ECR). In these two proteins, a double-stranded antiparallel β-ribbon is inserted into the major groove of the DNA. In 1ECR, the β-ribbon consists of two non-neighboring SSEs: the 11$^{\text{th}}$ and 14$^{\text{th}}$ SSEs. 1MJP is a dimer where two β-strands, one from each subunit (1$^{\text{st}}$ and 5$^{\text{th}}$ SSEs, on chains A and B, respectively), together form the β-ribbon. Since DALI server only aligns two single chains, this important common substructure was not detected. Their overall structures are quite different, hence, they were not aligned in VAST's server either. Besides the β-ribbon, FAMCS also found an α-helix between the two β-strands, probably inferring some folding preference or constraints.

General comparison with other methods

If we look at the results for all protein pairs of different structural classes according to SCOP in Table 3, we will have the following observations:

1. Almost all pure SSE-SSE alignments in DALI and VAST's results are detected by FAMCS. When two proteins mainly consist of SSEs and especially when their common substructures contain mostly SSEs, FAMCS's result is very similar to that of DALI or VAST. For example, for 1MCPl and 1TCRb, almost every SSE alignment without "NS" in DALI and VAST's answers has its counterpart in FAMCS's answer. Another good instance is 1A1Ea and 2ABL. The entire chain of 1A1Ea is an SH2 domain (made up of two α-helices and three β-strands), where 2ABL consists of an SH3 domain and an SH2 domain. FAMCS exactly matches the SH2 domains in the two proteins, so do DALI and VAST.

2. In DALI and VAST's results, there are sometimes cases of SSE segments aligned with non-SSE segments, or non-SSE segments aligned with non-SSE segments. In these cases, FAMCS is unable to detect the similarity. Take 3HSC and 2YHX as an example. These are two large proteins where the SSEs in the head-half structure are quite similar while those in the tail-half are not. As shown in DALI's answer, the SSEs in the tail-half of 3HSC can actually be aligned well with non-SSE parts in 2YHX. However, because FAMCS's alignment largely relies on structural similarity of SSEs, FAMCS only detects a short common segment of seven residues in the tail-half.

3. Though sometimes SSE segments align with non-SSE segments, these cases usually create worse RMSD values. For instance, for protein pair 2CRO and 2WRPr, DALI aligned the 5$^{\text{th}}$ SSE of 2CRO with the tail part of the 6$^{\text{th}}$ SSE of 2WRPr and a non-SSE segment after it. The RMSD value would be only 1.62 if this part is excluded from the DALI's alignment, instead of the current value of 4.66. Therefore, it is reasonable for FAMCS starts from SSE alignments.

4. After refining to the residue level, FAMCS tends to produce better RMSD values, sometimes aligning comparable number of residues as DALI and VAST do, but sometimes less. Many of the common segments missed by FAMCS are not secondary structures. FAMCS outperforms Chew's work in all the protein pairs whose data is available in the paper (*27*).

Output size and efficiency

From Table 4, we can see that the number of all MCSs may be very large, while the number of co-present MCSs is small enough for users to analyze every one in detail—here comes the need to eliminate intersecting and conflicting MCSs.

Neither the number of all MCSs nor the execution time is necessarily larger if the proteins are bigger. The protein pairs 1MCPl/1TCRb and 3HSC/2YHX illustrate this point. Rather, from all the data, the number and time seem closely related to the total number of similar SSE pairs (L_2 size). This is as expected since both the number of levels to merge common substructures and the number of common substructure candidates highly depend on the way how two proteins share similar elements, which is captured in L_2. Though larger proteins will take longer to generate the L_2 set, the Step 1 time is almost negligible. Therefore, it is hard to predict either the execution time or the size of the final answer without any prior knowledge about the proteins.

Methods

Representation of a protein

A good way of protein structural modeling is important to any study on protein structure since the information included may affect the effectiveness while the complexity of the representation may affect the efficiency of the solution a lot. Our representation is orientation-invariant [orientation-invariant property is very helpful since it avoids searching for the best relative orientation between two proteins, which can take as much as $O(n^6)$ time to compute] and contains reasonably small number of elements with information of significant properties that determine the protein structure.

In our project, the 3D structure of a protein is described as the conformation of the protein's SSEs, and each SSE is viewed as a vector in the 3D space. The dihedral angle and the closest approach distance between all SSE pairs define an orientation-invariant 3D structure. Figure 3 illustrates how to calculate the angle and distance between two SSEs.

When come to examining structural similarity, the type and the length of SSEs are also important. Thus, our protein representation consists of the *type sequence* (TP) and an *angle-distance* (AD) matrix. TP is a string of the two major SSE types: α and β. AD stores the lengths (in diagonal), angles (in the upper triangle part), and distances (in the lower triangle part). Mathematically, AD is defined as:

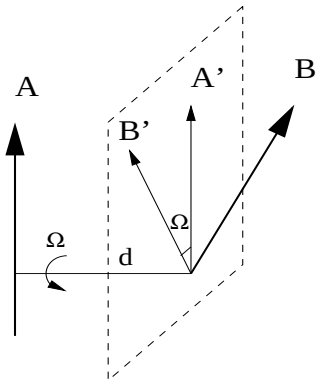


Fig. 3 Each SSE is represented by a vector with length and direction obtained from the N- and C-terminal C_α atoms. Let A and B be two vectors corresponding to two SSEs, d is the closest distance between A and B. A' and B' are the projected vectors onto the plane that is normal to d. The dihedral angle (Ω) is the angle between A' and B' measured along the plane.

$$AD_{i,j} = \begin{cases} d_{i,j} & \text{if } i > j, \\ l_i & \text{if } i = j, \\ \Omega_{i,j} & \text{if } i < j. \end{cases} \quad (1)$$

where $1 \leq i,j \leq$n, l_i is the length of the i^{th} SSE, $d_{i,j}$ and $\Omega_{i,j}$ are the distance and the angle between the i^{th} and j^{th} SSEs in the protein, respectively.

If the protein is a multi-chain protein, all its SSEs are linked to one sequence according to the chain order.

Problem definition

Definitions and notations

A common substructure of two proteins is usually made up of several disjoint regions of the backbone (*5*). As we are working on the level of secondary structures, a **Common Substructure** (CS) of proteins P and Q is a set of SSE pairs $\mathcal{S} = \{(P_x, Q_{x'}), (P_y, Q_{y'}), \ldots, (P_z, Q_{z'})\}$, where U_v represents the v^{th} SSE of protein U, and for all $(P_i, Q_{i'}), (P_j, Q_{j'}) \in \mathcal{S}$, where $i \neq j$ and $i' \neq j'$, they must be similar SSE pairs, that is, $Sim_{pair}\big((P_i, Q_{i'}), (P_j, Q_{j'})\big) > T_{sim}$, where Sim_{pair} is defined in Equation 2 and T_{sim} is a similarity threshold set by users.

In the above definition, P_i and $Q_{i'}$ are called **aligned SSEs**, and they are the **counterparts** of each other. The **size of a Common Substructure** is the number of SSE pairs it contains, denoted by

$|CS|$. The **combination** of several CSs is the union of their corresponding sets.

Furthermore, if no superset of $\mathcal{S}$ is a CS, $\mathcal{S}$ is said to be an MCS. Two proteins can have many MCSs. In Figure 1, P and Q have two MCSs: $\mathcal{S}_1 = \{(P_2, Q_7), (P_3, Q_8), (P_4, Q_9), (P_6, Q_1), (P_7, Q_2)\}$ (I) and $\mathcal{S}_2 = \{(P_9, Q_3), (P_{10}, Q_4), (P_{11}, Q_5)\}$ (II). They cannot be combined into one larger CS because at least (P_2, Q_7) and (P_{10}, Q_4) are not similar SSE pairs. Thus, the **Maximal Common Substructure Identification Problem** is: given two proteins P and Q, to identify all their MCSs.

If there are two different MCSs, $\mathcal{S}_1$ and $\mathcal{S}_2$, where there is an SSE pair $(P_i, Q_{i'})$ that both in $\mathcal{S}_1$ and $\mathcal{S}_2$, we say $\mathcal{S}_1$ and $\mathcal{S}_2$ are **intersecting**. If there exists SSE pairs (P_x, Q_y) in $\mathcal{S}_1$ and (P_x, Q_z) in $\mathcal{S}_2$ where $Q_y \neq Q_z$, then $\mathcal{S}_1$ and $\mathcal{S}_2$ are said to be **conflicting**.

Similarity function

Two similar SSE pairs are expected to have the same type (it is nonsence to align two different types of SSEs because α-helix and β-sheet have very different physical and biochemical properties), similar length for aligned SSEs, and similar spatial relationship between them. Let P_i denote the i^{th} SSE of protein P, the similarity of two SSE pairs $(P_x, Q_{x'})$ and $(P_y, Q_{y'})$ is defined as

$$\begin{aligned} Sim_{pair}\big((P_x, Q_{x'}), (P_y, Q_{y'})\big) \\ = Sim_{type} + Sim_{length} \\ + W_a \cdot Sim_{angle} + W_d \cdot Sim_{dist} \end{aligned} \quad (2)$$

where $Sim_{type}, Sim_{length}, Sim_{angle}$, and Sim_{dist} are the similarity measurements for type, length, angle, and distance of the SSE pairs $(P_x, Q_{x'})$ and $(P_y, Q_{y'})$. The larger the value of $Sim_{pair}\big((P_x, Q_{x'}), (P_y, Q_{y'})\big)$ is, the more similar the SSE pair $(P_x, Q_{x'})$ is with the SSE pair $(P_y, Q_{y'})$.

Since we require SSE counterparts to have exactly the same type and very similar length, the type similarity and the length similarity are defined as:

$$Sim_{type} = \begin{cases} -\infty & \text{if } type(P_x) \neq type(Q_{x'}) \\ & \text{or } type(P_y) \neq type(Q_{y'}), \\ 0 & \text{otherwise} \end{cases}$$

where $type(P_i)$ returns the type of the i^{th} SSE of protein P. If TP_P is the type sequence of protein P, $type(P_i) = TP_P[i]$.

$$Sim_{length} = \begin{cases} -\infty & \text{if } |len(P_x) - len(Q_{x'})| > T_l \\ & \text{or } |len(P_y) - len(Q_{y'})| > T_l, \\ 0 & \text{otherwise} \end{cases}$$

where $len(P_i)$ returns the length of the i^{th} SSE of protein P. If $AD(P)$ is the AD matrix for protein P, $len(P_i) = AD(P)_{i,i}$.

The angle similarity and the distance similarity are defined as below:

$$Sim_{angle} = max\big(0, 1 - \frac{|angle(P_x, P_y) - angle(Q_{x'}, Q_{y'})|}{T_a}\big),$$

$$Sim_{dist} = max\big(0, 1 - \frac{|dist(P_x, P_y) - dist(Q_{x'}, Q_{y'})|}{T_d}\big).$$

Let $AD(P)$ be the AD matrix for protein P, then

$$angle(P_i, P_j) = \begin{cases} AD(P)_{i,j} & \text{if } i < j, \\ AD(P)_{j,i} & \text{otherwise} \end{cases}$$

$$dist(P_i, P_j) = \begin{cases} AD(P)_{j,i} & \text{if } i < j, \\ AD(P)_{i,j} & \text{otherwise} \end{cases}$$

T_a and T_d are the thresholds for the difference in angle and distance, respectively. W_a and W_d are the weight for angle and distance to control the extent that they affect the similarity score. They are fractions between 0 and 1.

Algorithm of FAMCS

Since a common substructure consists of similar SSE pairs, our algorithm tries firstly to identify all similar SSE pairs, and then to combine similar SSE pairs into larger common substructures. Then the MCSs found are sorted according to their sizes and average similarity scores. An optional step is provided to select a co-present subset from all the MCSs discovered. The final step is to refine the answers to the residue level.

Step 1: Find all similar SSE pairs

In order to make our result optimal rather than heuristic, we compute the similarity Sim_{pair} between every possible SSE pair in one protein with those in the other. Then we report all SSE pairs with score higher than T_{sim}. Although the time complexity would be $O(n^4)$, it is still quite efficient practically since: (1) the number of SSEs in a typical globular protein is only around 15 according to Dror *et al* (*17*), and (2) many SSE pairs could be filtered quickly by simply checking their types and lengths.

Step 2: Combine to discover MCSs

From Step 1, we get all similar SSE pairs. To discover MCSs made up of many SSE pairs, a straightforward method is to enumerate all possible combinations of similar SSE pairs, and then check whether each combination is an MCS. But obviously, this method is too inefficient for both time and space since many combinations are not CSs, especially for those containing many SSEs. Fortunately, Theorem 1 shows that, if a set of similar SSE pairs is found to be not a CS, there is no need to generate its supersets since they cannot be CSs.

Theorem 1: The CS has the Apriori property, namely, any nonempty subset of a CS must also be a CS.

Proof: Assume $\mathcal{S}$ is not a CS since there exists $(P_x, Q_{x'})$, where $(P_x, Q_{x'}) \in \mathcal{S}$, so that $Sim_{pair}\big((P_x, Q_{x'}), (P_y, Q_{y'})\big) \leq T_{sim}$. Then, any superset of $\mathcal{S}$ could not be a CS due to the same reason by the definition of CS. Hence, the contraposition of the theorem holds, so does the theorem.

Our algorithm is similar to the Apriori algorithm. Following the notation in the Apriori algorithm, let L_k be the set of CSs of size k. We start from L_2, namely, the set of CSs containing two similar SSE pairs. L_k is generated from L_{k-1} as follows. If $\mathcal{S}_i, \mathcal{S}_j \in L_{k-1}$, where $\mathcal{S}_i = \mathcal{S}_{com} \cup \{(P_u, Q_{u'})\}$ and $\mathcal{S}_j = \mathcal{S}_{com} \cup \{(P_v, Q_{v'})\}$, and $\mathcal{S}_{com} = \{(P_x, Q_{x'}), \ldots, (P_y, Q_{y'})\}$, then $\mathcal{S} = \mathcal{S}_{com} \cup \{(P_u, Q_{u'}), (P_v, Q_{v'})\}$ is a candidate CS of size k. To determine whether $\mathcal{S}$ is a CS, we need to check whether every two similar SSE pairs pass the similarity test. Since $\mathcal{S}_i$ and $\mathcal{S}_j$ are in L_{k-1}, every two similar SSE pairs within them are ensured to be similar. Therefore, the only thing we need to check is whether $(P_u, Q_{u'})$ and $(P_v, Q_{v'})$ is a similar SSE pair, *i.e.*, whether $\{(P_u, Q_{u'}), (P_v, Q_{v'})\} \in L_2$. If so, $\mathcal{S}$ is a CS and is put into L_k; otherwise, and if no set in L_k is a superset of $\mathcal{S}_i$, then $\mathcal{S}_i$ is an MCS. The algorithm terminates when:

1. $|L_k| < 2$ for some $k < min(m, n)$ (the case when there is none or only one CS of size k so there is no room for growth to CS of size $k+1$), or

2. k reaches $min(m, n)$, where m and n is the number of SSEs in P and Q, respectively (the case when the entire protein of smaller size has been recognized as a substructure in the larger protein).

Since a larger MCS implies more statistical significance, the MCSs found are ranked according to the size first, then by similarity score. The similarity of an MCS, Sim_{MCS}, is defined as the average of the similarity of all the SSE pairs in it, that is,

$$Sim_{MCS}(\mathcal{S}) = \frac{\sum_{i=1}^{|\mathcal{S}|} \sum_{j=1 \wedge j \neq i}^{|\mathcal{S}|} Sim_{pair}(pair_i, pair_j)}{|\mathcal{S}| * (|\mathcal{S}| - 1)/2}, \quad \text{where } pair_i, pair_j \in \mathcal{S}. \tag{3}$$

Note that the output is a complete set of all MCSs shared by two proteins. The pseudocode of the algorithm is shown in Figure 4.

Step 3: Select significant co-present MCSs

As mentioned in the Introduction, many MCSs have intersecting or conflicting regions. For example, MCSs $\{(1,2),(2,4)\}$ and $\{(1,3),(2,4)\}$ have intersecting region $(2,4)$, and $(1,2)$ conflicts with $(1,3)$. They cannot be two co-existing domains. In order to select a subset of significant co-present MCSs, we decide to only retain the MCS that ranks the highest among all intersecting or conflicting MCSs. The resulting MCSs may represent different domains, or may be able to infer interesting structural properties, as shown in the Discussion. The algorithm for this step is outlined in Figure 5. This step is in fact optional. Users can still get all MCSs if they want.

Step 4: Refine to the residue level

If a user is interested in a particular MCS, it is usually desirable to know the exact residue correspondence. Refining an MCS to the residue level is not a straightforward process since the lengths of aligned SSE pairs are usually different, and residues of non-SSE regions (regions that are neither α-helix nor β-strand) may also be a part of the optimal alignment. VAST, a successful structural alignment algorithm, solves this problem by a Gibbs sampling technique. We believe that its technique can be well adopted in our algorithm. Moreover, we also propose a simple refinement algorithm here.

After studying some real examples, we have observed that: (1) when two SSEs of different lengths are counterparts from two proteins, the shorter SSE usually aligns with a part inside the longer SSE, but the relative positions differ from case to case, and (2) each consecutive segment of a common substructure is usually an SSE flanked with a few residues on both sides. Inspired by the above two observations, our refinement method consists of the following three steps:

To generate MCSs from similar SSE pairs
Input: A set of similar SSE pairs *input*
Output: A list *result* of all MCSs
1. $L_2 = input$
2. for $(k = 3;\ k < min(m, n)$ and $|L_{k-1}| \geq 2;\ k++)$
// Generate L_k from L_{k-1}
3. for all i^{th} element of L_{k-1}, say $\mathcal{S}_i$
4. for all j^{th} element of L_{k-1} $(j \neq i)$, say $\mathcal{S}_j$
5. if ($\mathcal{S}_i = \mathcal{S}_{com} \cup \{(P_u, Q_{u'})\}$ and $\mathcal{S}_j = \mathcal{S}_{com} \cup \{(P_v, Q_{v'})\}$)
6. $\mathcal{S} = \mathcal{S}_{com} \cup \{(P_u, Q_{u'}), (P_v, Q_{v'})\}$
7. if $(\{(P_u, Q_{u'}), (P_v, Q_{v'})\} \in L_2)$
8. $L_k = L_k \cup \{\mathcal{S}\}$
// Identify MCSs
9. if $\mathcal{S}_i$ is not combined with any $\mathcal{S}_j$
10. $result = result \cup \{\mathcal{S}_i\}$
11. Sort *result* by MCS size and similarity Sim_{MCS}
12. Return *result*
End algorithm

Fig. 4 Algorithm to generate all MCSs from similar SSE pairs.

To select significant co-present MCSs
Input: A sorted list of all MCSs *input*
Output: A list *result* of co-present MCSs
1. Add *input*[0] into *result*
2. for $(i = 1;\ i < input.size;\ i++)$
3. if *input*[*i*] does not intersect with any MCS already in *result*
4. Add *input*[*i*] into *result*
5. Return *result*
End algorithm

Fig. 5 Algorithm to select significant co-present MCSs.

1. For each SSE pair in an MCS, try various shifts to search for an optimal alignment just for that SSE pair. For example, if two SSEs in a pair are P_x and $Q_{x'}$ of length m and n $(m < n)$ respectively, let $P_x[i..i+l] : Q_{x'}[j..j+l]$ denote an alignment of length $l + 1$ in which the i^{th} element of P_x is aligned with the j^{th} element in $Q_{x'}$, and so on until the $(i + l)^{\text{th}}$ element of P_x is aligned with the $(j + l)^{\text{th}}$ element of $Q_{x'}$, where the i^{th} element of an SSE refers to its i^{th} residue. Then, the optimal alignment of P_x and $Q_{x'}$ is defined as:

$$\arg\min_{P_x[1..m]:Q_{x'}[k..k+m]} \big(RMSD(P_x[1..m] : Q_{x'}[k..k+m]) \text{ for } \forall\, k \in [1, n-m+1]\big).$$

2. After getting the optimal alignment for all SSE pairs, to combine all of them usually produces worse RMSD values because the translations and orientations of the protein structure that resulting in these optimal alignments are different in most times. Thus, in this step, we perform more shifts within SSE pairs to obtain the best alignment with the smallest global RMSD.

3. According to the second observation above, we extend the alignment onto non-SSE parts flanking aligned SSEs on both sides. Note that the more residues included in the alignment, the worse the RMSD value tends to be. In order to balance the size and the RMSD, the extension terminates when RMSD/size drops or it meets a neighbor region.

References

1. Janowski, R., *et al.* 2001. Human cystatin C, an amyloidogenic protein, dimerizes through three-dimensional domain swapping. *Nat. Struct. Biol.* 8: 316-320.
2. Bennett, M.J., *et al.* 1994. Domain swapping: entangling alliances between proteins. *Proc. Natl. Acad. Sci. USA* 91: 3127-3131.
3. Rost, B. 1997. Protein structures sustain evolutionary drift. *Fold. Des.* 2: S19-24.
4. Milik, M., *et al.* 2003. Common structural cliques: a tool for protein structure and function analysis. *Protein Eng.* 16: 543-552.
5. Holm, L. and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233: 123-138.
6. Yuan, X. and Bystroff, C. 2005. Non-sequential structure-based alignments reveal topology-independent core packing arrangements in proteins. *Bioinformatics* 21: 1010-1019.
7. Szustakowski, J.D. and Weng, Z. 2000. Protein structure alignment using a genetic algorithm. *Proteins* 38: 428-440.
8. Somers, W.S. and Phillips, S.E. 1992. Crystal structure of the *met* repressor-operator complex at 2.8 Å resolution reveals DNA recognition by beta-strands. *Nature* 359: 387-393.
9. Lathrop, R.H. 1994. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.* 7: 1059-1068.
10. Holm, L. and Sander, C. 1996. Mapping the protein universe. *Science* 273: 595-603.
11. Gerstein, M. and Levitt, M. 1996. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 4: 59-67.
12. Singh, A.P. and Brutlag, D.L. 1997. Hierarchical protein structure superposition using both secondary structure and atomic representations. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 5: 284-293.
13. Gibrat, J.F., *et al.* 1996. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* 6: 377-385.
14. Falicov, A. and Cohen, F.E. 1996. A surface of minimum area metric for the structural comparison of proteins. *J. Mol. Biol.* 258: 871-892.
15. Alexandrov, N.N. 1996. SARFing the PDB. *Protein Eng.* 9: 727-732.
16. 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.
17. Dror, O., *et al.* 2003. Multiple structural alignment by secondary structures: algorithm and applications. *Protein Sci.* 12: 2492-2507.
18. May, A.C. and Johnson, M.S. 1995. Improved genetic algorithm-based protein structure comparisons: pairwise and multiple superpositions. *Protein Eng.* 8: 873-882.
19. Lehtonen, J.V., *et al.* 1999. Finding local structural similarities among families of unrelated protein structures: a generic non-linear alignment algorithm. *Proteins* 34: 341-355.
20. Zhu, J. and Weng, Z. 2005. FAST: a novel protein structure alignment algorithm. *Proteins* 58: 618-627.
21. Vriend, G. and Sander, C. 1991. Detection of common three-dimensional substructures in proteins. *Proteins* 11: 52-58.
22. Griendley, H.M., *et al.* 1993. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* 229: 707-721.
23. Koch, I., *et al.* 1996. An algorithm for finding maximal common subtopologies in a set of protein structures. *J. Comput. Biol.* 3: 289-306.
24. Fischer, D., *et al.* 1992. An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. *J. Biomol. Struct. Dyn.* 9: 769-789.
25. Chakraborty, S. and Biswas, S. 1999. Approximation algorithms for 3-D common substructure identification in drug and protein molecules. In *Proceedings of the Sixth International Workshop on Algorithms and Data Structures*, pp. 253-264, Vancouver, Canada.
26. Pennec, X. and Ayache, N. 1994. An O(n2) algorithm for 3D substructure matching of proteins. In *Shape and Pattern Matching in Computational Biology* (eds. Califano, A., *et al.*). Plenum Publishing, New York, USA.
27. Chew, L.P., *et al.* 1999. Fast detection of common

geometric substructure in proteins. *J. Comput. Biol.* 6: 313-325.

28. Agrawal, R., *et al.* 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington DC, USA.
29. Berman, H.M., *et al.* 2000. The protein data bank. *Nucleic Acids Res.* 28: 235-242.
30. Walther, D. 1997. WebMol—a Java-based PDB viewer. *Trends Biochem. Sci.* 22: 274-275.
31. Kabsch, W. and Sander, C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22: 2577-2637.
32. Murzin, A.G., *et al.* 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247: 536-540.
33. Chionh, C.H., *et al.* 2005. Towards SCALEable protein structure comparison and database search. *Int. J. Artif. Intell. Tools.* 14, in press.
34. Sun, Y.J., *et al.* 1998. The structure of glutamine-binding protein complexed with glutamine at 1.94 Å resolution: comparisons with other amino acid binding proteins. *J. Mol. Biol.* 278: 219-229.