Ashby et al. BMC Genomics 2013, 14(Suppl 2):S1
http://www.biomedcentral.com/1471-2164/14/S2/S1

BMC Genomics

RESEARCH Open Access

New enumeration algorithm for protein structure comparison and classification

Cody Ashby[1], Daniel Johnson[1], Karl Walker[2], Iyad A Kanj[3], Ge Xia[4*], Xiuzhen Huang[5*]

From ISCB-Asia 2012
Shenzhen, China. 17-19 December 2012

Abstract

Background: Protein structure comparison and classification is an effective method for exploring protein structure-function relations. This problem is computationally challenging. Many different computational approaches for protein structure comparison apply the secondary structure elements (SSEs) representation of protein structures.

Results: We study the complexity of the protein structure comparison problem based on a mixed-graph model with respect to different computational frameworks. We develop an effective approach for protein structure comparison based on a novel independent set enumeration algorithm. Our approach (named: ePC, **e**fficient **e**numeration-based **P**rotein structure **C**omparison) is tested for general purpose protein structure comparison as well as for specific protein examples. Compared with other graph-based approaches for protein structure comparison, the theoretical running-time $O(1.47^{rn}n^2)$ of our approach ePC is significantly better, where n is the smaller number of SSEs of the two proteins, r is a parameter of small value.

Conclusion: Through the enumeration algorithm, our approach can identify different substructures from a list of high-scoring solutions of biological interest. Our approach is flexible to conduct protein structure comparison with the SSEs in sequential and non-sequential order as well. Supplementary data of additional testing and the source of ePC will be available at http://bioinformatics.astate.edu/.

Background

Protein structure comparison is an effective method for exploring protein structure-function relations and for studying evolutionary relations of different species. It can also be applied to identify the active sites of carrier proteins, the binding sites of antibodies, the inhibition sites of enzymes, and the common structural motifs of proteins, which has significant applications in biological and biomedical research.

The computational methods for protein structure comparison usually represent a protein structure by atomic coordinates in the Euclidean space, as a distance matrix [1] whose entries represent the distances between two residues of the protein, or as a contact map [2], where a binary matrix is used to represent the distances between the residue pairs. A structure graph representation of a protein tertiary structure was first defined in [3] for protein structure prediction. In this current work, we adopt the structure graph representation in [3]. We develop a very efficient graph-based approach for protein structure comparison. Our approach transforms the comparison problem to an independent set problem in an auxiliary graph, and then applies a novel enumeration algorithm to identify the best out of a set of good comparison candidates.

We first show the problem of comparing a query structure to another structure is intractable with respect to several computational frameworks. For example, we show that the problem is $\mathbb{NP}-\text{hard}$ (even for very restricted instances), cannot be approximated to a ratio $n^{\frac{1}{2}-\varepsilon}$, for any $\varepsilon > 0$, unless $\mathbb{P} = \mathbb{NP}$, and is $W[1]$- complete with respect to the framework of parameterized complexity. We also show that a useful case of the problem is solvable

* Correspondence: xiag@lafayette.edu; xhuang@astate.edu
[4]Department of Computer Science, Lafayette College, Pennsylvania, USA
[5]Department of Computer Science, Arkansas State University, Arkansas, USA
Full list of author information is available at the end of the article

in polynomial time by reducing it to the 2-CNF-SATISFIABILITY problem.

Whereas the above results are negative hinting at the challenging nature of the problem, the graph-based approach we use allows us to model the problem as a maximum independent set problem, for which a repertoire of effective exact algorithms exist in the literature. We use an algorithm developed by (some of) the authors [4] to enumerate the top-K maximum independent sets in a graph in time $O(1.47^n n^2)$, where n is the number of vertices in the graph (Note that the algorithm in [4] enumerates the top-K minimum vertex covers in a graph, but obviously can be used to enumerate the top-K maximum independent sets in a graph using the standard reduction between vertex cover and independent set); this enumeration algorithm allows us to sift through the top SSE alignments for the protein structure comparison problem, looking for the best amongst them in terms of accuracy. Compared with other graph-base approaches, the theoretical running-time $O(1.47^{rn} n^2)$ of our approach ePC is the current best, where n is the smaller number of SSEs of the two proteins, r is an introduced parameter of small values.

Many different approaches for protein structure comparison apply the secondary structure elements (SSEs) representation and database searching, such as deconSTRUCT [5], SSM [6], GANGSTA [7], MASS [8,9], VAST [10], TOPS [11] and approaches in [12-19]. Our approach ePC utilizes the SSE-based representation of the protein structure, and takes into consideration the global 3D structural arrangements of the SSEs of the proteins. We compare our approach with two other SSE-based approaches: deconSTRUCT, an approach for general purpose protein structure comparison and database search, and SSM, a high-resolution structure comparison approach. Our approach has comparable performance as deconSTRUCT. With a more general and simplified representation and a unified graph enumeration algorithm, our approach could detect a substructure or motif structure in a set of large structures, more than one common substructure shared by a set of proteins. It is very flexible. Our approach could use a wide range of evaluation functions for protein structure comparison. It could be applied to handle sequential and non-sequential order of SSE alignment and be extended to handle challenging protein multiple structure alignment and protein subset alignment.

Methods

A mixed graph for a protein structure is constructed from the PDB file as follows: each vertex represents a core/secondary structure element (i.e., an alpha helix element, or, a beta strand element), each undirected edge represents the interaction between two cores, and each directed edge (arc) represents the loop between two consecutive cores (from the N-terminal to the C-terminal). A mixed graph representation is used for protein structure prediction in [3]. The DSSP program [20,21] was used for the assignments of secondary structure elements for the protein entries from the Protein Data Bank (PDB). Refer to the protein structure and the corresponding mixed graph representation in Figures 1 for protein with ID: 6ldh. Alpha helix elements are represented by circles and beta strand elements are represented by squares. Therefore, a mixed graph can be represented as a triple $G = (V(G), A(G), E(G))$, where $V(G)$ is the vertex-set of G, $E(G)$ is the set of undirected edges of G, and $A(G)$ is the set of directed edges of G, which induces a directed path spanning all vertices of G, thus defining a linear order among the vertices of $V(G)$. The aforementioned mixed graph representation incorporates the SSE type, the sequential order of the SSEs, and the interactions of the SSEs. When comparing two protein structures, the problem could now be reduced to finding the common subgraph of the two mixed graph.

Goldman et al. [2] studied the protein comparison problem using the notion of *contact maps*. Contact maps are undirected graphs whose vertices are linearly ordered. Goldman et al. [2] formulated the protein comparison problem as a CONTACT MAP OVERLAP problem, in which we are given two contact maps and we need to identify a subset of vertices S in the first contact map, a subset of vertices S' in the second with $|S| = |S'|$, and an order preserving (w.r.t. linear ordering) bijection $f : S \to S'$, such that the number of edges in S (i.e., between the vertices in S) that correspond (under f) to edges in S' is maximized. In [2], the authors proved that the CONTACT MAP OVERLAP problem is MAXSNP-complete even when both contact maps have maximum degree one.

Song et al. [3] studied the problem of mixed-graph comparison, when each vertex v in the first mixed-graph is associated with a subset of vertices S_v in the second mixed-graph, and the bijection f is restricted to map v to a vertex in S_v. Song et al. [3] proved that this problem is NP-complete, even when the size of each subset S_v, referred to as the *map width* is at most 3. Our results in the following section refine and extend the results in [3] in several aspects. We first prove that the problem defined in [3] is intractable with respect to many computational frameworks. For example, we show that the problem: (1) is $\mathbb{NP}-\text{hard}$ (even for very restricted instances), (2) cannot be approximated to a ratio $n^{\frac{1}{2}-\varepsilon}$, for any $\varepsilon > 0$, unless $\mathbb{P} = \mathbb{NP}$, and (3) is W[1]-complete with respect to the framework of parameterized complexity. We also show that a useful case of the problem is solvable in polynomial time by reducing it to the 2-CNF-SATISFIABILITY problem.

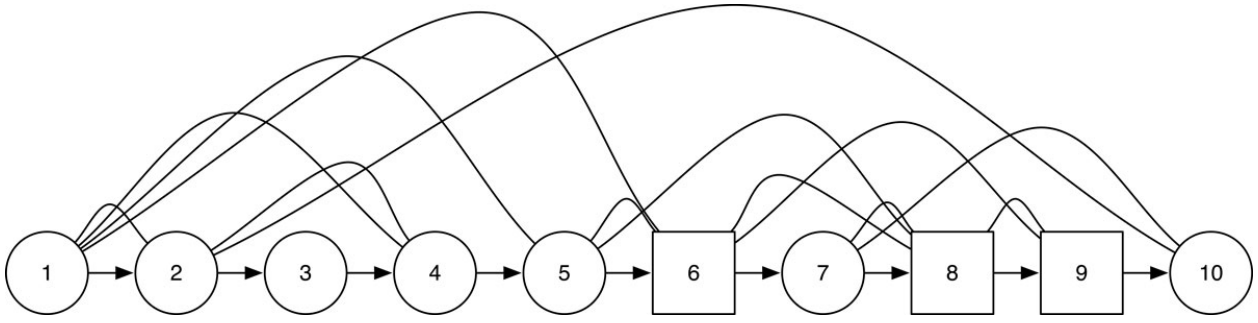

Figure 1 Structure graph for 6ldh. Alpha helix elements are represented by circles and beta strand elements are represented by squares.

The graph embedding problem and complexity results

In this section, we study the complexity of the mixed graph embedding problem, which corresponds to the problem of identifying the query protein structure (e.g., a motif structure) as a substructure in a larger protein structure.

We define the GRAPH EMBEDDING problem as follows:

GRAPH EMBEDDING

Given two mixed graphs $G = (V(G), A(G), E(G))$ and $H = (V(H), A(H), E(H))$, where H is referred to as the *host graph*, such that each vertex $v \in V(G)$ has a list $L(v) \subseteq V(H)$ of vertices in H that it can be mapped to, decide if there exists an injection $f: V(G) \to V(H)$ such that:

(i) $f(v) \in L(v)$ for every $v \in V(G)$;

(ii) for any two vertices $v, v' \in V(G)$, there is a directed path from v to v' in G if and only if there is a directed path from f(v) to $f(v')$ in H; and

(iii) for any two vertices $v, v' \in V(G)$, if $vv' \in E(G)$ then $f(v)f(v') \in E(H)$.

We shall call an injective embedding f satisfying properties (i)-(iii) above a *valid embedding*.

Informally speaking, the GRAPH EMBEDDING problem asks if we can embed G into H in such a way that the precedence order determined by the arcs of G is respected by this embedding, and the undirected edges of G are respected by this embedding.

We define the restriction of the GRAPH EMBEDDING problem, denoted r-GRAPH EMBEDDING, where r is positive integer, by restricting the cardinality of the set $L(v)$ to be at most r, for every $v \in V(G)$; that is, in the restrictions of the problems, a vertex in $V(G)$ can be mapped to at most r vertices in H.

If one cannot embed the whole graph G into H, it is natural to seek an embedding that embeds the maximum number of vertices in G into H, while respecting conditions (i)-(iii) above. Therefore, we define a version of GRAPH EMBEDDING, denoted GRAPH EMBEDDING$_{\geq}$, by introducing a nonnegative parameter k, and asking whether there exists a subset $S \subseteq V(G)$ with $|S| \geq k$, and an injection $f: S \to V(H)$ such that:

(i) $f(v) \in L(v)$ for every $v \in S$;

(ii) for any two vertices $v, v' \in S$, if there is a directed path from v to v' in G then there is a directed path from $f(v)$ to $f(v')$ in H; and

(iii) for any two vertices $v, v' \in S$, if $vv' \in E(G)$ then $f(v)f(v') \in E(H)$.

The optimization/maximization version of the GRAPH EMBEDDING$_{\geq}$ problem, denoted MAXIMUM GRAPH EMBEDDING, asks for a set S of maximum cardinality that satisfies conditions (i)-(iii) above. Similarly, we can define the problems r-GRAPH EMBEDDING$_{\geq}$ and MAXIMUM r-GRAPH EMBEDDING.

It was shown in [3] that a more general problem than r-GRAPH EMBEDDING, in which the set of edges $A(G)$ do not necessarily induce a path, is $\mathbb{NP}-$ complete for any $r \geq 3$. The same proof actually shows that the r-GRAPH EMBEDDING problem is $\mathbb{NP}-$ complete for any $r \geq 3$. We show next that the 2-GRAPH EMBEDDING is solvable in polynomial time.

Theorem 0.1 *The* 2-GRAPH EMBEDDING *problem is solvable in polynomial time.*

PROOF. We reduce the problem to 2-CNF-SATISFIABILITY, which is solvable in polynomial time (for example, see [22]. Recall that in the 2-CNF-SATISFIABILITY problem we are given a Boolean formula in the *conjunctive normal form* (CNF) (i.e., the formula is the conjunction of clauses, and each clause is the disjunction of a literals, which are variables or negations of variables), in which each clause contains at most two literals, and we are asked to decide whether or not the formula is satisfiable. Let (G, H) be an instance of 2-GRAPH EMBEDDING satisfying $|L(v)| \leq 2$, for every $v \in$ V (G). We show how to construct in polynomial time an instance F of 2-CNF-SATISFIABILITY

such that G has a valid embedding into H if and only if F is satisfiable.

For every vertex $v \in G$: if $L(v) = \{v'\}$ we add a variable $x_{vv'}$ and add the clause $\{x_{vv'}\}$ to F; and if $L(v) = \{v', v''\}$ we add the two variables $x_{vv'}$, $x_{vv''}$and the two clauses $\{\overline{x}_{vv'}, \overline{x}_{vv''}\}$, $\{\overline{x}_{vv'}, \overline{x}_{vv''}\}$ to F. This ensures that every vertex v in G is mapped to one and only one vertex in H (i.e., the map is a well-defined function). (We assume that $|L(v)| \neq 0$; otherwise, the instance can be rejected.)

For every two vertices v and u in G such that there is a directed path from v to u in G (i.e., v appears before u in the directed path in G), and for very $v' \in L(v)$ and $u' \in L(u)$ such that $v' = u'$ or u'appears before u in the directed path in H, we add the clause $\{\overline{x}_{vv'}, \overline{x}_{uu'}\}$ to F. This ensures that the desired mapping is injective, and ensures that the mapping respects the precedence order among the vertices in G that is defined by the directed path in G (property (ii)).

For every two vertices v and u in G such that $vu \in E(G)$, and for very $v' \in L(v)$ and $u' \in L(u)$ such that $v'u' \notin E(H)$, we add the clause $\{\overline{x}_{vv'}, \overline{x}_{uu'}\}$ to F. This ensures that the desired mapping respects the undirected edges of G (property (iii)).

This completes the construction of F. Clearly, this construction can be carried out in polynomial time.

It is not difficult to verify that (G, H) is a yes-instance of 2-GRAPH EMBEDDING if and only if F is a yes-instance of 2-CNF-SATISFIABILITY. This implies that 2-GRAPH EMBEDDING is polynomial-time solvable. □

The above theorem, together with the result in [3], provides a complete characterization of the complexity (NP-hardness) of r-GRAPH EMBEDDING with respect to r.

If we consider the r-GRAPH EMBEDDING parameterized by r, the fact that the problem is NP-complete for $r \geq 3$ [3] implies that the problem is not solvable in time $O(n^r)$ unless $\mathbb{P} = \mathbb{NP}$, and hence, with respect to the parameterized complexity framework, the problem is not in the class $\mathbb{XP}$. Therefore, there is not much hope behind seeking parameterized algorithms (with respect to r) for the problem. Moreover, the NP-hardness proof for r-GRAPH EMBEDDING ($r \geq 3$) is via a reduction from 3-CNF-SATISFIABILITY (each clause contains at most three literals) that produces two graphs G and H, each of size linear in the number of clauses of the 3-CNF-SATISFIABILITY instance. Therefore, based on the results in [23], we can conclude that r-GRAPH EMBEDDING ($r \geq 3$) is not solvable in subexponential time unless the exponential-time hypothesis (ETH) fails [23].

We investigate next the complexity of the r- GRAPH EMBEDDING$_{\geq}$ problem.

Theorem 0.2 *The* r-GRAPH EMBEDDING$_{\geq}$ *problem is* $\mathbb{NP}$ *– complete, for any* $r \geq 1$.

PROOF. It suffices to prove the $\mathbb{NP}$ – completeness of the 1-GRAPH EMBEDDING$_{\geq}$ problem. We only prove the NP-hardness, as it is very easy to show the membership of the problem in $\mathbb{NP}$. The proof is via a reduction from the CLIQUE problem: Given a graph and a nonnegative integer k, determine if the graph has a clique (complete subgraph) of size k.

Let (G', k) be an instance of CLIQUE, where $V(G') = \{v'_1, ..., v'_n\}$. We construct the instance (G, H, k) of 1- GRAPH EMBEDDING$_{\geq}$ as follows. The set of vertices $V(G) = \{v_1, ... ,v_n\}$ and $V(H) = \{u_1, ... ,u_n\}$ are copies of $V(G')$. We connect the vertices v_1 ,..., v_n in G by a directed path, and $u_1, ... ,u_n$ in H by a directed path, and define $L(v_i) = \{u_i\}$, for $i = 1, ... ,n$. Finally, the undirected edges of G form a clique, and the undirected edges of H are those of G'; that is, $v_iv_j \in E(G)$ for every $1 \leq i \neq j \leq$ n, and $u_iu_j \in E(H)$ if and only if $v'_iv'_j \in E(G')$. This completes the reduction, which is obviously computable in polynomial time.

It is not difficult to verify that (G', k) is a yes-instance of CLIQUE if and only if (G, H, k) is a yes-instance of 1-GRAPH EMBEDDING$_{\geq}$. This completes the proof. □

The reduction in the above theorem is an fpt-reduction, from the CLIQUE problem to 1- GRAPH EMBEDDING$_{\geq}$, where the parameter is the size of the subgraph sought k. Since CLIQUE is known to be W [1]-hard in the parameterized complexity hierarchy, we obtain:

Theorem 0.3 The r- GRAPH EMBEDDING$_{\geq}$ *problem is* W [1]*-complete, for any* $r \geq 1$. *(Note that membership in* W [1] *follows from the results in the next section.*)

Finally, we observe that the same reduction in Theorem 0.2 provides an L-reduction [24] (i.e., approximation-preserving reduction) from MAXIMUM CLIQUE (the problem of computing a clique of maximum cardinality in a graph) to MAXIMUM 1-GRAPH EMBEDDING. It is well known that, unless $\mathbb{P} = \mathbb{NP}$, MAXIMUM CLIQUE cannot be approximated to a ratio $n^{\frac{1}{2}-\varepsilon}$ for any $\varepsilon > 0$ [25]. It follows that:

Theorem 0.4 *Unless* $\mathbb{P} = \mathbb{NP}$, *the* MAXIMUM r-GRAPH EMBEDDING *problem cannot be approximated to a ratio* $n^{\frac{1}{2}-\varepsilon}$ *for any* $\varepsilon > 0$.

Graph embedding to independent set

In this section we show that the MAXIMUM r-GRAPH EMBEDDING problem can be modeled as an MAXIMUM INDEPENDENT SET problem. Recall that an *independent set* in a graph is set of vertices such that no two of them are adjacent, and the MAXIMUM INDEPENDENT SET problem asks for an independent set of maximum cardinality in a graph.

Let (G, H) be an instance of MAXIMUM r-GRAPH EMBEDDING. Suppose that $V(G) = \{g_1, g_2, ... ,g_n\}$ with directed edges from g_i to g_{i+1}, for $1 \leq i \leq n - 1$, and suppose that $V(H) = \{h_1, h_2, ... ,h_m\}$, $m \geq n$, and with directed edges

from h_i to h_{i+1}, for $1 \le i \le m - 1$. Suppose that each vertex of G can be mapped to one of at most r vertices in H.

Theorem 0.5 *If* MAXIMUM INDEPENDENT SET *is solvable in time* 2^{cn}, *then* MAXIMUM r-GRAPH EMBEDDING *is solvable in* 2^{crn} *time.*

PROOF. Create an auxiliary graph X as follows. For each possible choice mapping g_i to h_j, create a vertex x_{ij}. For any two vertices x_{ij} and x_{kl}, add an edge between them if and only if one of the following conditions are true:

1. $i = k$ or $j = l$.
2. $i < k$ and $j > l$, or $i > k$ and $j < l$.
3. There is an undirected edge between g_i and g_k in G, while there is no undirected edge between h_j and h_l in H.

Note that Condition 2 could be removed when the order of the mapped vertices are not required to be the same for the two graphs.

It is clear that any independent set of X corresponds to a common subgraph of G and H of the same size. So the problem of finding a maximum common subgraph of G and H is reduced to the problem of finding a maximum independent set of X, which has rn vertices. In particular, to find if G is a subgraph of H it suffices to find an independent set of size n. Therefore if MAXIMUM INDEPENDENT SET is solvable in time 2^{cn}, then MAXIMUM r-GRAPH EMBEDDING is solvable in 2^{crn} time. □

If we use the current-best exact algorithm for MAXIMUM INDEPENDENT SET by Robson [26] that runs in time $O(2^{n/4})$, we conclude that:

Theorem 0.6 *The* MAXIMUM r-GRAPH EMBEDDING *problem is solvable in time* $O(2^{rn/4})$, *where n is the number of vertices in graph G.*

Algorithm for structure comparison

The problem of protein structure comparison could be modeled as finding an independent set problem of an auxiliary graph. When aligning two protein structures, the auxiliary graph X is created as is in the proof of Theorem 2.5. Note that when aligning three and more protein structures, the auxiliary graph X could be created similarly.

Refer to the following for the outline of the algorithm for protein structure comparison.

1 *(Preprocessing)*. Generate the two structure graphs for the two proteins, based on both their secondary structure information (local structure) and tertiary structure (global structure) information.
2 *(Auxiliary graph)*. Build the auxiliary graph from the two structure graphs;
3 *(Top K independent sets)*. Generate the top K maximum independent sets of the auxiliary graph by applying the enumeration algorithm developed in [4].
4 *(Matched SSEs)*. Evaluate the generated top K maximum independent sets and generate the SSE pairs with the best score of the two proteins.

We analyze the time complexity of the algorithm:

Step 1: The algorithm processes the two proteins to generate the corresponding two structure graphs, where each vertex of a graph represents an SSE of the corresponding protein. Suppose the number of the vertices of each structure graph is bounded by n.

Step 2: We introduce a parameter r as the maximum number of pairs associated with each vertex of the structure graphs. The number of vertices of the auxiliary graph is bounded rn.

Step 3: Through calling the enumeration algorithm develop in [4], it takes time $O(1.47^{rn})$ to generate the top K independent sets of the auxiliary graph.

Step 4: It takes time $O(1.47^{rn}n^2)$ to evaluate the generated independent sets and identify the independent set, which corresponding to the SSE pairs with the best score of the two proteins.

Refer to [27] for a discussion of the theoretical running times of several other graph-based approaches for protein structure comparison, which are of $O((mn)^n)$ or $O(m^{n+1})n)$, where m and n demote the size of the structure graphs. The theoretical running-time $O(1.47^{rn}n^2)$ of our approach ePC is the current best, where n is the smaller number of SSEs of the two proteins, r is a parameter of small values.

Testing results

Our approach ePC is designed for general-purpose protein structure comparison. In this section we test our approach for this purpose using SABmark-sup and SABmark-twi [28], and specific novel folds studied in the literature. Our approach is implemented using C++. The testing is mainly performed on a regular Macbook (8GB Mem). The running-time testing is conducted on a Dell server (PowerEdge 2950III, 32GB Mem). Due to the space limit, some testing results are not presented.

Given two proteins, A and B, the score of the a SSE pair is the sum of the L_{ij} of the residues for the SSE pair. L_{ij} is defined in [29] denoting the similarity between a segment centered around residue i of one protein and a segment centered around residue i of the other protein, where $L_{ij} = \min\{D(d^A_{i-2,i+2}, d^B_{j-2,j+2}), D(d^A_{i-2,i+1}, d^B_{j-2,j+1}), D(d^A_{i-1,i+2}, d^B_{j-1,i+2})\}$, where $D(d_1, d_2) = 0.1 - |d_1 - d_2|/(d_1 + d_2)$.

Let S be the sum of the scores of all the aligned SSEs. The normalized score $S_n = S/\sqrt{(l_A * l_B)}$, where l_A and l_B are the lengths of the two proteins. A_c is the number of SSEs in A, B_c is the number of SSEs in B and MCS_n is the size of the common subgraph of the two protein

structure graphs, the CORE-COV is a percentage defined by: MCS_n / min(A_c, B_c).

Testing different parameter values

There are two important parameters of our algorithm r and K, where r is the maximum number of SSE pairs associated with each SSE of the structure graphs, and K is the number of enumerated independent sets. Note that the score L_{ij} of the SSE pairs is the criteria for identifying the associated r SSEs. We test the impacts of the two parameter values on the running time and scoring for the protein comparison.

We present our testing results for accuracy (using the score S as a criteria) and running-time of our approach with different parameter r values. We have conducted the testing of 200 protein pairs from SABmark-sup database with different parameter r values, where each SSE from one protein is matched with the top r SSEs from the other protein. Refer to Figure 2 for the average scores of 200 protein pairs from Sup database, when testing our approach with different parameter r values, r = 2, 3, 4, 5, 6, 7, 8, 9. Our testing results indicate that when the parameter r value increases, the score has increased. Refer to Figure 3 for the average running times of 200 protein pairs from Sup database, when testing our approach with different parameter r values. When r increases, the running time of our approach increases in general. However note that the running times when r = 5, 6, 7, 8, 9 are very similar; this is because trimming has been applied to reduce the sizes of the auxiliary graphs before the enumeration of the independent sets, and also because the impact of the parameter K on the running time. Especially the running time when r = 2 is significantly lower than the other cases, which matches our theoretical result that for r = 2 the r-GRAPH EMBEDDING problem is in P.

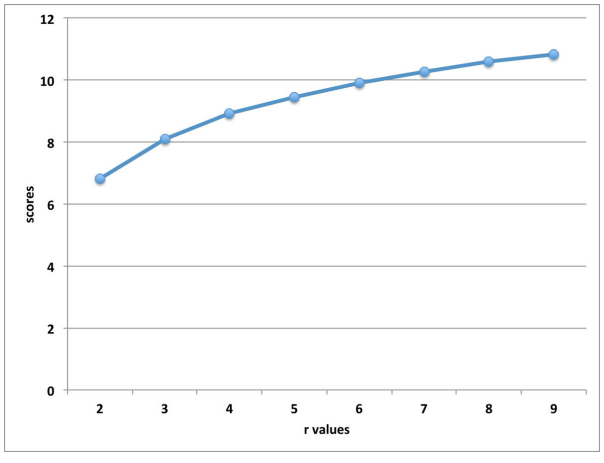

Figure 2 The running times for different r values. Note for all these testing, our approach use the same parameter K = 1000.

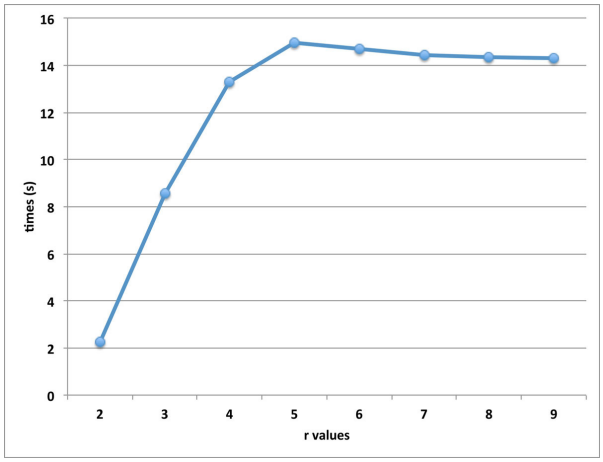

Figure 3 The scores for different r values. Note for all these testing, our approach use the same parameter K = 1000.

For the enumeration of independent sets, we have introduced a parameter K, which is the bound of the number of enumerated independent sets. Here we present our testing results for accuracy and running-time of our approach with different parameter K values (See Table 1). Similar as the testing for the parameter r, we have conducted the testing of 200 protein pairs from SABmark-sup database with different parameter K values, K = 125, 250, 500, 1000. Our testing results indicate that when the parameter K value increases, the score has increased and the running time also increases.

Performing structure comparison

Self-querying in a large database of structures. As pointed in [5], a necessary condition for a approach to be of practical value for structure comparison and classification, it should be able to find the query itself in a database of protein structures. To test this property of our approach, 1000 protein structures from the SABmark-sup database. Our approach with the normalized score function can identify the query structure with ranking No. 1 with 100% accuracy.

Detecting a substructure in a set of larger structures. Our approach can detect a smaller query structure (or, a motif structure) within a larger target structure.

Table 1 The running times and scores for different *K* values

K =	125	250	500	1000
time	1.90	3.57	6.76	12.73
score	8.89	9.08	9.17	9.28

Note for all these testing our approach use the same parameter r = 6.

We use the test set from the previous test and required for each domain to be matched to the target domain embedded in the original full-protein structure. Our approach with the normalized score can identify the substructure with ranking No. 1 with 100% accuracy.

Protein family classification. We compare the performance of our approach for protein family classification with deconSTRUCT, which is also an SSE-based method and designed for protein structure database filtering. We have tested 1000 proteins pairs of the SABmark [28]. Due to the space limit, we only discuss some of the representative testing result. We align protein d1a6m (core size:7; AAs: 151; from SABmark [28]) to proteins from 10 different families of the twi database with each family 10 proteins. Of the proteins in the top 10 ranking, 7 proteins identified through our approach are the proteins from the same family as protein d1a6m. For deconSTRUCT, 7 proteins of the identified 10 proteins (without ranking) are the proteins from the same family as protein d1a6m. Form the testing results, our approach has comparable performance with deconSTRUCT for the general purpose protein structure comparison and structure classification. The mixed graph representation of our approach ePC is much simpler compared with deconSTRUCT. Our approach ePC is more flexible than deconSTRUCT in that ePC can handle SSE alignments with and without respect to the order of SSEs, which will be discussed in the next section for specific examples.

Specific examples

We test our approach on specific examples for common substructures and novel folds which share common substructures with non-sequential SSEs.

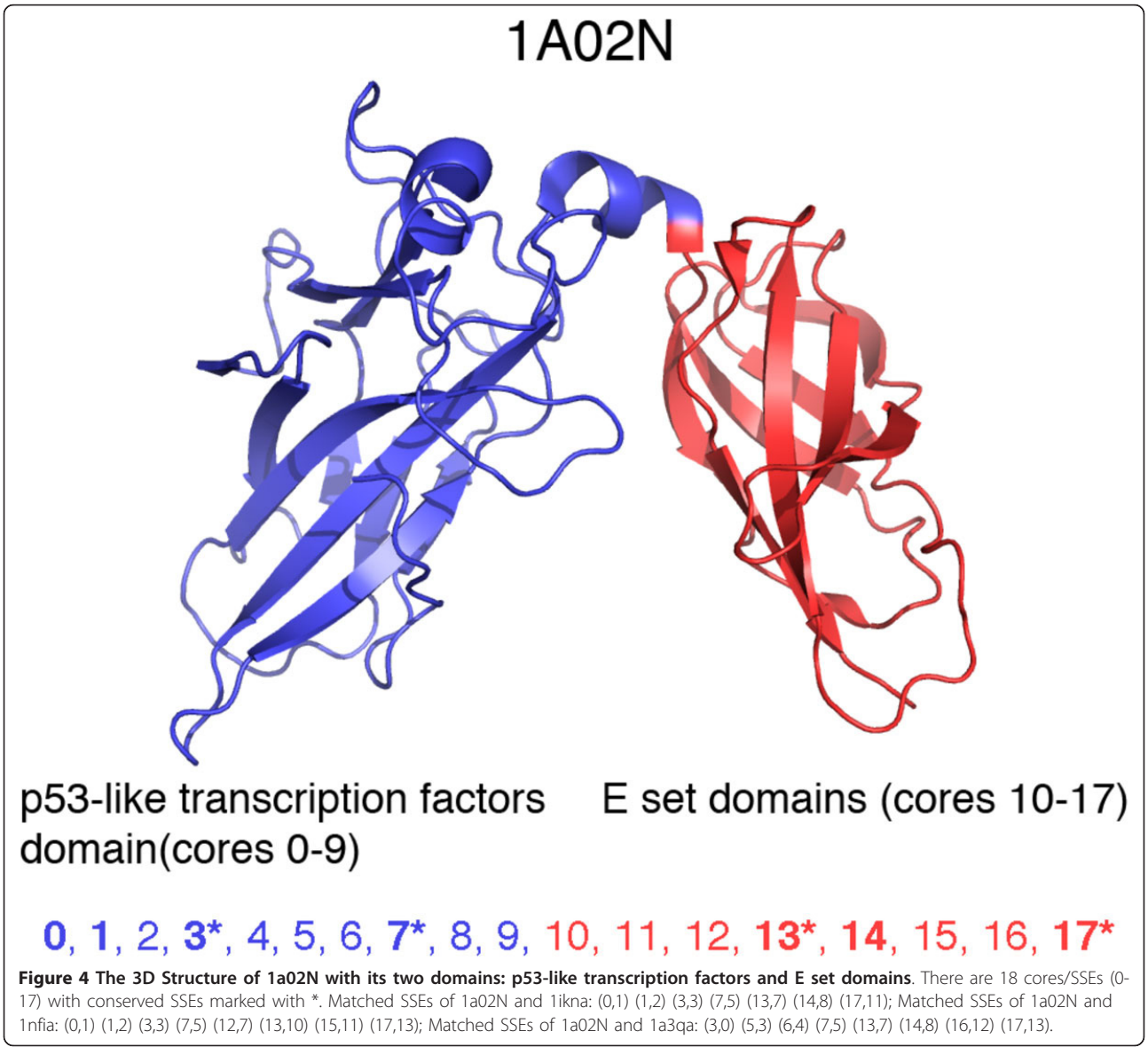

Figure 4 The 3D Structure of 1a02N with its two domains: p53-like transcription factors and E set domains. There are 18 cores/SSEs (0-17) with conserved SSEs marked with *. Matched SSEs of 1a02N and 1ikna: (0,1) (1,2) (3,3) (7,5) (13,7) (14,8) (17,11); Matched SSEs of 1a02N and 1nfia: (0,1) (1,2) (3,3) (7,5) (12,7) (13,10) (15,11) (17,13); Matched SSEs of 1a02N and 1a3qa: (3,0) (5,3) (6,4) (7,5) (13,7) (14,8) (16,12) (17,13).

Detection of several different common substructures. We test our approach ePC using the four protein structures (PDB codes: 1a02N, 1iknA, 1nfiA, and 1a3qA) studied in [8,9]. The proteins share two common domains: "p53-like transcription factors" and "E set domains". In [8,9] two different common substructures were detected, one for each domain. The first common substructure is part of the "p53-like transcription factors" domain. It consists of 114 residues, and it forms a sandwich of nine *beta*-strands. The second common substructure is part of the "E set domains" domain. It consists of 87 residues, and it forms a sandwich of seven *beta*-strands.

Please refer to the following testing results of our approaches, when 1a02N is compared with: 1iknA, 1nfiA, and 1a3qA. Our testing results match the results in [8,9]. Especially for the second common substructure that is part of the "E set domains" domain with conserved matched SSEs: 12, 13, 14, 15, 16, 17 of 1a02N. Please refer to Figure 4 for its 3D structure and the two domains.

Three novel folds. The three novel folds were discussed in [7] to study the unique feature of GANGSTA + to conduct non-sequential SSE alignment. Note that the protein structures that are structurally similar to the listed three new folds were detected through scanning the ASTRAL40 database by GANGSTA+. The detected similar protein structures have non-sequential SSE alignments with the three novel folds respectively. Please refer to our testing result in Table 2 Figure 5 and 6.

Discussion

We use an SSE-based graph model for general purpose protein structure comparison. We presented the computational complexity results related to the protein structure comparison problem. An effective algorithm is developed integrating a novel enumeration of independent sets and parameterized computation for the problem. Our approach is tested for protein structure comparison using benchmark testing sets. Compared with other SSE-based approaches, our approach has comparable performance for the general purpose protein structure comparison. We also demonstrate that our approach could be applied to identify common substructure with non-sequential SSEs and proteins sharing more than one common substructure.

Table 2 Structure search and comparison of the three novel folds with the structural analogs

New fold	Detected analog	DaliLite	TM-align	GANGSTA+	deconSTRUCT	SSM	ePC
2JMK/7/57	1GO4H/4/93	11.0/0/75	4.0/1/67	1.8/7/61	0/0	1/14	4/100%/8.3
2AJE/7/44	1J7NB/40/738	3.9/3/45	3.4/3/45	2.1/4/53	3/31	3/61	7/100%/10.9
2ES9/5/58	1SXJH/15/267	2.5/4/57	4.0/5/65	1.8/5/69	3/36	2/38	5/100%/9.9

The results for DaliLite, TM-align and GANGSTA+ are from [7]. The format of protein entries: PDB ID/number of SSEs/number of residues in these SSEs. The format of the testing result for DaliLite, TM-align and GANGSTA+: RMSD/number of aligned SSEs/number of aligned residues. The format of the testing results for deconSTRUCT and SSM: no. of matched SSEs/no. of aligned AAs, and for our approach ePC: no. of matched SSEs/CORE-COV/score).

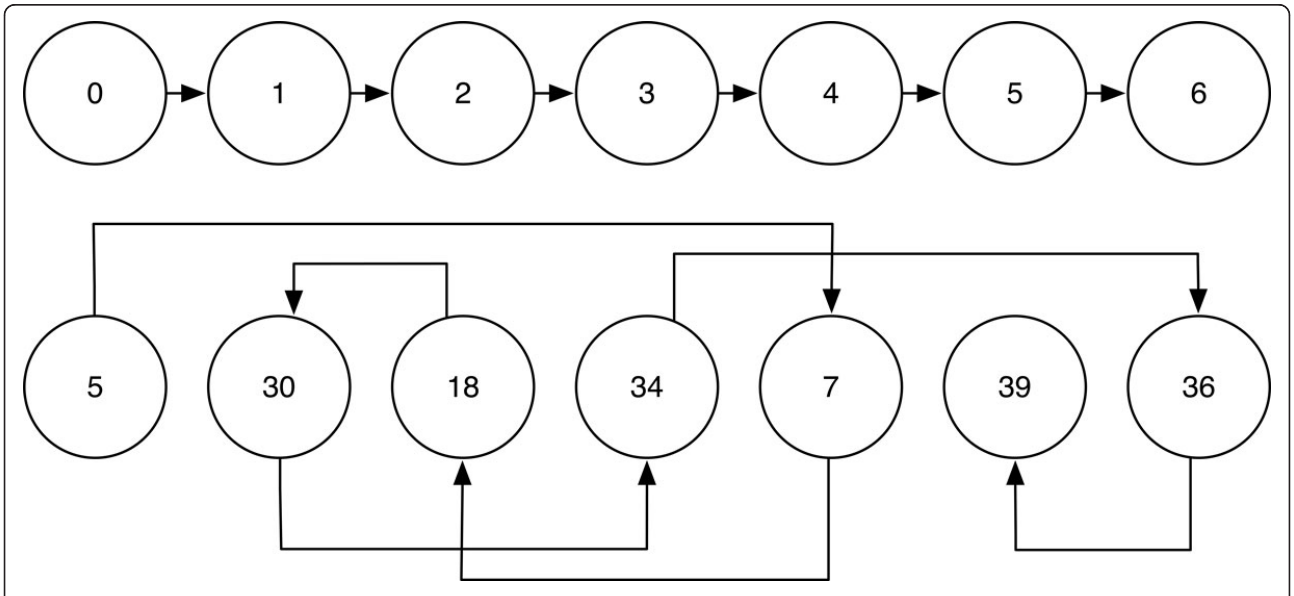

Figure 5 Structure alignment of PDB:2AJE and PDB:1J7NB. Structure alignment of the new fold PDB:2AJE and the structural analog PDB:1J7NB, showing nonsequential order of aligned SSEs.

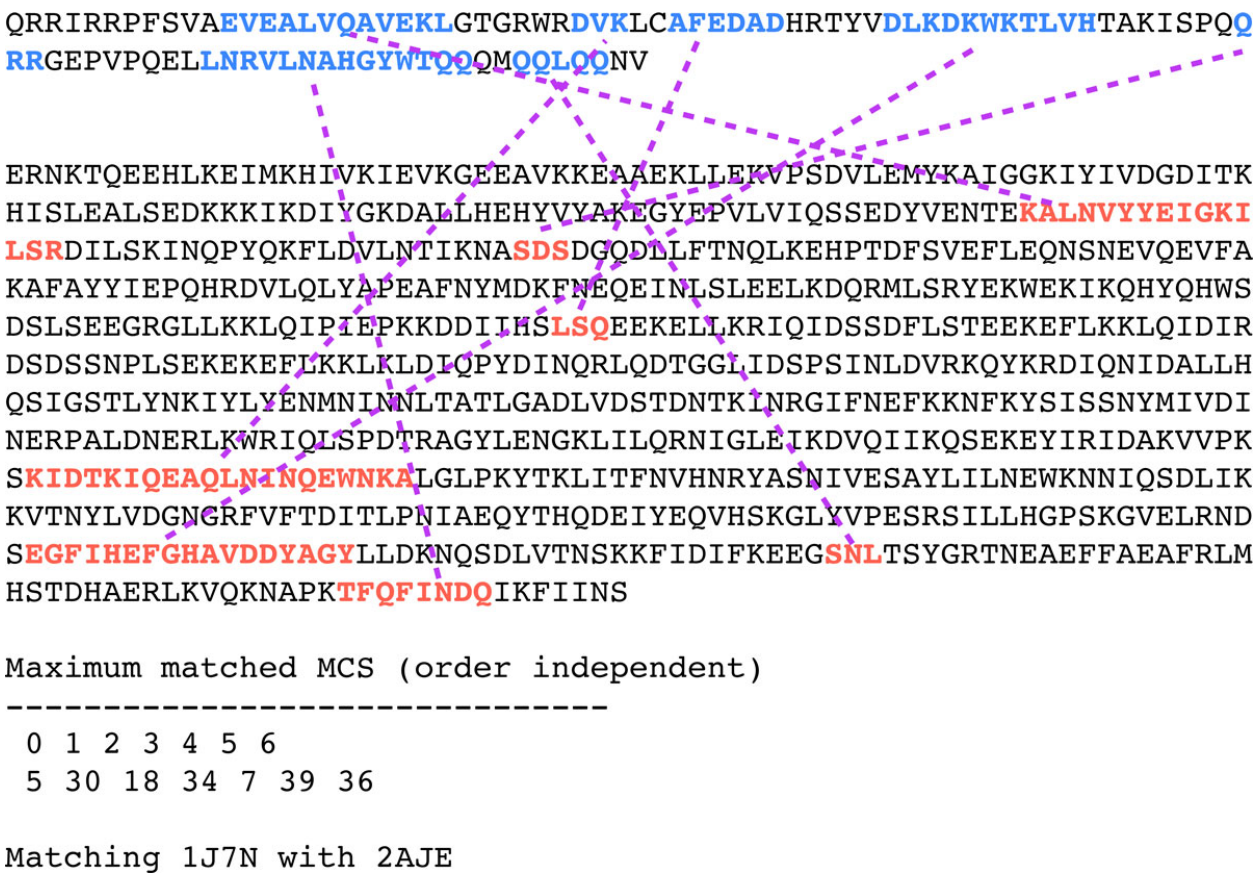

Figure 6 Aligned SSEs of PDB:2AJE and PDB:1J7NB. The amino acid sequences of the new fold PDB:2AJE and the structural analog PDB:1J7NB, showing the non-sequential order of aligned SSEs of the two protein sequences.

Authors' contributions

XH, IK and GX carried out the study on the complexity and the design of the approach for the protein structure comparison problem, and drafted the manuscript. CA, DJ and KW participated in the implementation and the testing of the algorithm. All authors have approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This research is supported by the National Institute of Health grants from the National Center for Research Resources (5P20RR016460-11) and the National Institute of General Medical Sciences (8P20GM103429-11).

Declarations

The publication costs for this article were funded by the corresponding author's institution.
This article has been published as part of *BMC Genomics* Volume 14 Supplement 2, 2013: Selected articles from ISCB-Asia 2012. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcgenomics/supplements/14/S2.

Author details

[1]Molecular Bioscience Graduate Program, Arkansas State University, Arkansas, USA. [2]Bioinformatics Graduate Program, University of Arkansas at Little Rock, Arkansas, USA. [3]School of Computing, DePaul University, Illinois, USA. [4]Department of Computer Science, Lafayette College, Pennsylvania, USA. [5]Department of Computer Science, Arkansas State University, Arkansas, USA.

Published: 15 February 2013

References

1. Holm L, Sander C: **Protein structure comparison by alignment of distance matrices.** *J of Molecular Biology* 1993, **233**:123-138.
2. Goldman D, Istrail S, Papadimitriou CH: **Algorithmic Aspects of Protein Structure Similarity.** *FOCS* 1999, 512-522.
3. Song Y, Liu C, Huang X, Malmberg RL, Xu Y, Cai L: **Efficient parameterized algorithms for biopolymer structuresequence alignment.** *IEEE/ACM Trans Comput Biology Bioinform* 2006, **3(4)**:423-432.
4. Chen J, Kanj I, Meng J, Xia G, Zhang F: **On the effective enumerability of NP problems.** *Proceedings of the 2nd InternationalWorkshop on Parameterized and Exact Computation, volume 4169 of Lecture Notes in Computer Science* 2006, 215-226.
5. Zhang ZH, Bharatham K, Sherman WA, Mihalek I: **deconSTRUCT: general purpose protein database search on the substructure level.** *Nucleic Acids Research* 2010, **38(Web Server)**:W590-W594.
6. Krissinel E, Henrick K: **Secondary-structure matching (PDBeFold), a new tool for fast protein structure alignment in three dimensions.** *Acta Cryst D60* 2004, 2256-2268.
7. Guerler , Knapp : **Novel Folds and their Nonsequential Structural Analogs.** *Protein Science* 2008, **17:8**:1374-1382.
8. Dror O, Benyamini H, Nussinov R, Wolfson H: **MASS: Multiple structural alignment by secondary structures.** *Bioinformatics* 2003, **19(Suppl 1)**:i95-i104.
9. Dror O, Benyamini H, Nussinov R, Wolfson H: **Multiple structural alignment by secondary structures: algorithm and applications.** *Protein Science* 2003, **12**:2492-2507.
10. Gibrat JF, Madej T, Bryant SH: **Surprising similarities in structure comparison.** *Curr Opin Struct Biol* 1996, **6(3)**:377-385.
11. Michalopoulos I, Torrance GM, Gilbert DR, Westhead DR: **TOPS: an enhanced database of protein structural topology.** *Nucleic Acids Research* 2004, **32**:251-254.

12. Alesker V, Nussinov R, Wolfson H: **Detection of non-topological motifs in protein structures.** *Protein Eng* 1996, **9**:1103-1119.
13. Alexandrov N, Fischer D: **Analysis of topological and nontopological structural similarities in the PDB: New examples with old structures.** *Proteins* 1996, **25**:354-365.
14. Grindley H, Artymiuk P, Rice D, Willett P: **Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm.** *J Mol Biol* 1993, **229**:707-721.
15. Holm L, Sander C: **3-D lookup: Fast protein structure database searches at 90% reliability.** *The Third International Conference on Intelligent Systems for Molecular Biology* 1995, 179-187.
16. Koch I, Lengauer T, Wanke E: **An algorithm for finding maximal common subtopologies in a set of proteins.** *J Comp Biol* 1996, **3**:289-306.
17. Lu G: **TOP: A new method for protein structure comparisons and similarity searches.** *J Appl Crystallogr* 2000, **33**:176-183.
18. Mitchel E, Artymiuk P, Rice D, Willet P: **Use of techniques derived from graph theory to compare secondary structure motifs in proteins.** *J Mol Biol* 1990, **212**:151-166.
19. Yang AS, Honig B: **An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance.** *J Mol Biol* 2000, **301**:65-678.
20. Joosten RP, Te Beek TAH, Krieger E, Hekkelman ML, Hooft RWW, Schneider R, Sander C, Vriend G: **A series of PDB related databases for everyday needs.** *NAR* 2010, doi: 10.1093/nar/gkq1105.
21. Kabsch W, Sander C: **Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.** *Biopolymers* 1983, **22**:2577-2637.
22. Papadimitriou CH: *Computational Complexity* Addison-Wesley; 1994.
23. Impagliazzo R, Paturi R, Zane F: **Which problems have strongly exponential complexity?** *Journal of Computer and System Sciences* 2001, **63(4)**:512-530.
24. Papadimitriou CH, Yannakakis M: **Optimization, approximation, and complexity classes.** *J Comput Syst Sci* 1991, **43(3)**:425-440.
25. Håstad Johan: **Clique is Hard to Approximate Within $n^{1-epsilon}$.** *Proceedings of the 37th Annual Symposium on Foundations of Computer Science* 1996, 627-636.
26. Robson JM: **Finding a maximum independent set in time $O(2^n/^4)$, 2001.** *Technical Report* LaBRI, Universite Bordeaux I;1251-01.
27. Krissinel E, Henrick K: **Protein structure comparison service Fold at European Bioinformatics Institute.** [http://www.ebi.ac.uk/msd-srv/ssm].
28. Van Walle I, *et al*: **SABmark: a benchmark for sequence alignment that covers the entire known fold space.** *Bioinformatics* 2005, **21**:1267-1268.
29. Zhu J, Weng Z: **FAST: a novel protein structure alignment algorithm.** *Proteins* 2005, **58(3)**:618-627.

doi:10.1186/1471-2164-14-S2-S1
Cite this article as: Ashby *et al.*: New enumeration algorithm for protein structure comparison and classification. *BMC Genomics* 2013 **14**(Suppl 2):S1.