BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 7 2011, pages 925–932
doi:10.1093/bioinformatics/btr044

Structural bioinformatics

Advance Access publication February 3, 2011

GOSSIP: a method for fast and accurate global alignment of protein structures

I. Kifer[1,*], R. Nussinov[2] and H. J. Wolfson[1,*]

[1]School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences and [2]Department of Human Molecular Genetics and Biochemistry, Sackler Institute of Molecular Medicine, Sackler Faculty of Medicine, Tel Aviv University, Tel Aviv 69978, Israel

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: The database of known protein structures (PDB) is increasing rapidly. This results in a growing need for methods that can cope with the vast amount of structural data. To analyze the accumulating data, it is important to have a fast tool for identifying similar structures and clustering them by structural resemblance. Several excellent tools have been developed for the comparison of protein structures. These usually address the task of local structure alignment, an important yet computationally intensive problem due to its complexity. It is difficult to use such tools for comparing a large number of structures to each other at a reasonable time.

Results: Here we present GOSSIP, a novel method for a global all-against-all alignment of any set of protein structures. The method detects similarities between structures down to a certain cutoff (a parameter of the program), hence allowing it to detect similar structures at a much higher speed than local structure alignment methods. GOSSIP compares many structures in times which are several orders of magnitude faster than well-known available structure alignment servers, and it is also faster than a database scanning method. We evaluate GOSSIP both on a dataset of short structural fragments and on two large sequence-diverse structural benchmarks. Our conclusions are that for a threshold of 0.6 and above, the speed of GOSSIP is obtained *with no compromise* of the accuracy of the alignments or of the number of detected global similarities.

Availability: A server, as well as an executable for download, are available at http://bioinfo3d.cs.tau.ac.il/gossip/.

Contact: wolfson@tau.ac.il; ilonak@post.tau.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on September 20, 2010; revised on December 4, 2010; accepted on January 21, 2011

1 INTRODUCTION

The space of solved protein structures is increasing rapidly. New protein structures are being solved on a daily basis, allowing for a more complete understanding of the protein universe. The number of possible protein conformations is thought to be limited (Friedberg *et al.*, 2007; Zhang and Skolnick, 2005a), which suggests the existence of a large structural redundancy in the protein structures submitted to the PDB. The two 'gold standards' for the structural classification of the PDB - SCOP (Andreeva *et al.*, 2008) and CATH (Greene *et al.*, 2007) provide an answer for this redundancy, yet they cannot keep up with the rate of addition of new structures to the PDB. One of the reasons is the relatively high computational time required by protein structure comparisons. Many excellent protein structure comparison methods have been devised over the years, for example Holm and Park (2000); Nussinov and Wolfson (1991); Shindyalov and Bourne (1998); Subbiah *et al.* (1993); Zhang and Skolnick (2005b). Often they employ heuristics that allow them to be relatively efficient yet still accurate. In order to diminish computational costs, most methods use a C_α representation of the protein (Csaba *et al.*, 2008; Shatsky *et al.*, 2004). Some methods use an even more compact representation of secondary structural elements (SSEs) (Can and Wang, 2003; Dror *et al.*, 2003; Krissinel and Henrick, 2004) to reduce running times. The purpose of most of these methods is to find evolutionary relationships between different proteins. In many cases, this does not require the entire protein structures to be similar and hence, typical structure alignment methods aim at detecting equivalent substructures in different proteins. While this in itself is extremely important, it slows down the speed of the computation tremendously because the number of potentially similar substructures is exponential in the length of the protein. The method we describe here, named GOSSIP (GlObal Structure SuperposItion of Proteins), is aimed at performing an all-against-all global alignment of any given set of protein structures. It does not search for partial structural matches between regions of two aligned proteins, and hence does not aim to provide an alternative for existing local structural alignment methods. Rather, we propose a method that provides fast, accurate, large-scale comparisons of protein structures. A significant advantage of our method is that it can be applied to any set of protein structures, represented by C_α coordinates, regardless of protein size. No preprocessing stage is necessary as in many other methods that are intended for fast one-against-all comparisons. Moreover, no knowledge of secondary structure or other features of the protein is required. This allows the method to be applicable for structural fragments, where the secondary structure content is limited or non-existent (e.g. loops and coils). The motivation for our work comes from a problem we encountered during our research that required the creation of sequence profiles based on structural information. We needed to cluster several hundred thousand structural fragments with varying lengths. These fragments were roughly divided into groups according to secondary structure, which still left several

*To whom correspondence should be addressed.

groups of tens of thousands of structures (mostly groups of shorter fragments). We performed an all-against-all alignment of each group using MultiProt (Shatsky *et al.*, 2004), an efficient and accurate method for local structural pairwise and multiple alignment. The comparison procedure was performed on a cluster of ∼100 Intel 2.66 GHz CPUs, and took several months. Although this clustering is not a frequent procedure, it often needs to be repeated several times, for example, in cases of tuning an alignment parameter or if replacing the entire fragment set. To avoid the significant delay in our work, we have developed an efficient tool that can quickly align a large number of structures to one another, and output the optimal superposition. We note, that the task we set out to solve is not as complex as the task of a standard structure alignment tool, since we are not interested in similar substructures. The question we wished to answer for each pair of proteins is, whether the size of the optimal correspondence list between them is larger than some threshold. In addition to our immediate task, the developed procedure has other important applications. First, any research that requires the global clustering of a large dataset of structural fragments can benefit from this procedure (Baeten *et al.*, 2008; Bryson *et al.*, 2005; Budowski-Tal *et al.*, 2010; Pandini *et al.*, 2010). Second, it can be used for removal of structural redundancy from a protein set. Although sequence alignment methods can be used for this task [e.g. Wang and Dunbrack (2003)], there are many cases in which sequentially different PDB structures exhibit high structural similarity. Moreover, many examples are known of different structural topologies that share high sequence similarity (Kosloff and Kolodny, 2008). Another important application is a swift database search for identification of proteins similar to a query structure. Scanning through the fast-growing PDB in search of structural neighbors is becoming a heavy computational task (Hasegawa and Holm, 2009). Currently, not many structure alignment methods can perform a search in real time through the entire PDB, providing a list of structural neighbors and corresponding structural superpositions. One such server is SSM (Krissinel and Henrick, 2004), which represents proteins compactly using secondary structure elements. Yet it is still not fast enough for a single-CPU PDB search and thus runs on a large cluster of computers simultaneously. A swift database alignment program could make structure manipulation more accessible to researchers who do not have access to extensive computational power. This necessity was lately recognized by many groups and a new category of fast filtering methods, also referred to as projection methods, has evolved (Budowski-Tal *et al.*, 2010; Carpentier *et al.*, 2005; Carugo and Pongor, 2002; Choi *et al.*, 2004; Lisewski and Lichtarge, 2006; Røgen and Fain, 2003; Zotenko *et al.*, 2006). These methods usually represent the protein as a 1D string or feature vector, thus reducing the comparison complexity to that of a sequence alignment procedure. Recently several database scanning methods have been reported to compare well with trusted, much more computationally intensive structure alignment servers (Budowski-Tal *et al.*, 2010; Carpentier *et al.*, 2005). However, filtering methods have their drawbacks: first, they do not compute a transformation between the aligned proteins and thus have difficulty producing an accurate alignment. Second, to obtain good results they often require the compared proteins to be of minimal length or that other knowledge be available (e.g. secondary structure). Third, they sometimes require a heavy preprocessing stage of the protein set to be compared. In this article, we propose a global structural alignment method that has the following properties: it can be applied to any set of protein structures with C_α coordinates, with variable protein lengths, it outputs a transformation and superposition, and for an alignment length of 60% and up it is as sensitive and as accurate as trusted structural aligners, yet several times faster. Also, it requires no preprocessing.

In this work, we use a linear representation of a protein using a set of geometric signatures in order to efficiently detect similar protein structures. The idea of using geometrical signatures in structural alignment has been used before (Alireza *et al.*, 2005; Can and Wang, 2003; Carpentier *et al.*, 2005; Kishon *et al.*, 1991). However, to the best of our knowledge it has mostly been used for applications of local structure comparison. When local substructures are sought, small signatures are required that span few residues. For example (Can and Wang, 2003), use signatures of triplets of amino acids. Such a representation results in many similar local signatures between non-related proteins, and necessitates some filtering stage or incorporation of other information such as secondary structure into the signature. A global alignment allows the use of signatures that span a larger portion of the structure. This in turn gives a better separation of different proteins with similar local substructures but different overall structures. Hence, the speed of the procedure is greatly increased.

In the following, we describe our method for global clustering of protein structures. We evaluate its performance with respect to two accurate methods for local structural alignment, MultiProt (Shatsky *et al.*, 2004) and CE (Shindyalov and Bourne, 1998), and also with respect to a database scanning procedure named YAKUSA (Carpentier *et al.*, 2005). For our evaluation, we use two sets of protein fragments and two sequence-diverse datasets of CATH and SCOP structures. We show that GOSSIP is able to detect global similarities with similar success to trusted structure aligners, yet faster by several orders of magnitude.

2 APPROACH

The input to our program is a structure list and two thresholds: the minimal alignment size between any two similar molecules (we denote by th_{sim}), and the maximal distance between two matching C_αs (denote th_{dist}).

Our method is based on constructing structural fingerprints, which characterize the protein shape (Fig. 1). These signatures are inserted into a lookup table, which is then queried to extract similar signatures to a queried molecule. The different signatures drawn from each extracted molecule are then aligned to the signatures of the query molecule using a dynamic programming algorithm in order to obtain an optimal superposition.

As always with such approaches, there is a tradeoff between the sensitivity of a structural signature, or its ability to detect other close signatures, and its specificity, i.e. its ability to distinguish between true and false similarities. In the case of very specific signatures, only very similar protein structures will be identified and sensitivity will be low. In the opposite case of a very general signature, the lookup table cannot aid in distinguishing between signatures of non-similar proteins. Hence, it was important to come up with a structural signature that characterizes the protein well, while also allowing for generalization.

Below we first describe the structural signature we devised. Next, we outline the procedure for insertion and querying the lookup table.

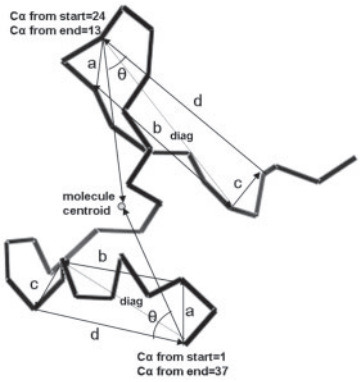

Fig. 1. An illustration of the signature used for each protein position. The letters a,b,c,d mark the lengths of the polygon edges. θ is the angle between the vector induced by the two most distant polygon points and the centroid of the molecule. Two more features which we use are the serial number of the first polygon C_α from the C and N terminus of the molecule.

Finally, we present the algorithm for aligning the signatures and deriving superpositions from such an alignment.

3 METHODS

3.1 Structural signature

To obtain a global alignment of structures, we represent each position in a protein with signatures that are rotation and translation invariant. Figure 1 demonstrates the components of a structural signature. For the i-th C_α position in a molecule ($C\alpha_i$), we define the quadrilateral $[C\alpha_i, C\alpha_{i+2}, C\alpha_{i+\text{quadSize}-2}, C\alpha_{i+\text{quadSize}}]$ where quadSize is a length that is determined according to the molecule size. The smaller the molecule, the smaller is the size of the quadrilateral that determines each signature. We then build a feature vector characterizing this quadrilateral, which is composed of the following eight items:

(1) The four side lengths of the quadrilateral.

(2) The length of the diagonal between $C\alpha_i$ and $C\alpha_{i+\text{quadSize}-2}$.

(3) The angle θ between the vector defined by $(C\alpha_i, C\alpha_{i+\text{quadSize}})$, and the vector defined by $C\alpha_i$ and the molecule centroid.

(4) The indices i and $(N_{\text{length}} - i + 1)$ where N_{length} is the protein length.

At the end of this preprocessing, each molecule is defined by $(N - \text{quadSize} + 1)$ 8-tuple structural signatures.

3.2 Storing in a lookup table

To store the structural information of the signatures, we use a lookup table that is based on the Geometric Hashing technique (Lamdan and Wolfson, 1988; Nussinov and Wolfson, 1991). Our lookup table is implemented as an 8D grid, in which the granularity of each dimension is determined separately. This enables us to store our feature vectors in a data structure that automatically classifies them into bins by signature similarity. Determining the granularity of the grid can be tricky. A grid with a high resolution will be too specific to catch similar signatures, while lowering its resolution will cause a reduction in its discriminating power and heavily increase the running time of the querying procedure. In our algorithm, the resolution of each dimension is set as follows:

(1) The side lengths of the relevant quadrilateral should be quite similar for close structural signatures, hence we set the resolution of these four dimensions to 1Å.

(2) Due to the triangle inequality, the resolution of the diagonal was set to twice the resolution of the side length.

(3) We observed that the angle θ distinguishes well between similar quadrilaterals located in different parts of the molecule. This is especially true for similar patterns of α helix. However, even small backbone differences can cause a change in this angle, especially for small values of quadSize. Hence, we experimentally set this value to 25° for small molecules or fragments up to a size of 50°, 20° for molecules up to a size of 100° and 15° for larger molecules.

(4) For two similar molecules, we expect the indices i and $N_{\text{length}} - i + 1$ to be similar. However, insertions or deletions will cause shifts in these indices. Hence, we set the resolution of these two dimensions to be the maximal size of insertion or deletion that will still guarantee the minimal percentage of similarity th_{sim}. This size is equal to:

$$th_{\text{maxIndel}} = (1.0 - th_{\text{sim}}) * \max(len(P_1), len(P_2)) \quad (1)$$

Where P_1 and P_2 are the two aligned proteins, and $len(P_i)$ is the length in amino acids of protein P_i.

In global alignment, there is no need to compare structures of non-similar lengths. Hence, we divide the sizes into mutually exclusive size ranges, and create a separate lookup table for every range of molecule sizes. The signatures of a molecule with size S will be entered into a lookup table with size ranges $[lb, ub]$ only if:

$$th_{\text{sim}} \times S \geq lb \text{ and } S \div th_{\text{sim}} \leq ub \quad (2)$$

For each lookup table, the θ angle resolution and C_α index resolution are set separately. The parameter quadSize is also specific per table. The values were experimentally determined to be 4, 10, 12 and 15 for molecules of maximal length of 50, 100, 200 and > 200 accordingly.

3.3 Querying the lookup table

Equation 2 implies that a molecule can be preprocessed into several separate lookup tables. However, each molecule only needs to be queried once at most to find its neighbor structures. Hence, a molecule with size S is only queried inside a unique lookup table with size ranges $[lb, ub]$ s.t. $lb \leq S \leq ub$.

Recall, that each C_α in a specific molecule induces a single feature vector. Hence, the process of querying a molecule M_q is as follows:

(1) Go over the list of M_q's feature vectors. Use each to query the lookup table and extract similar signatures from other molecules.

(2) For each molecule M_r with matching signatures, count the number of distinct C_α positions in M_q that are covered by signatures of M_r.

(3) Discard all molecules in which the number of covered query positions is below a certain threshold (determined according to th_{sim}).

(4) For every molecule M_{ri} that passed the threshold, map each position in M_q to its matching quadrilaterals of M_{ri}.

This procedure obtains a list of mappings between M_q's quadrilaterals and those of each molecule that passed the threshold. The procedure is repeated for every queried molecule in each lookup table.

3.4 Finding an optimal superposition

For a molecule M_{ri} that has not been discarded by the lookup table querying stage, our task is reduced to the alignment of two sequences representing the query and matched structure. Since the mapping produced by the querying stage is one-to-many, we employ a dynamic programming technique (Needleman and Wunsch, 1970). We create a 2D matrix, with its number of rows and columns determined by the size of M_q and M_{ri}. Each entry in the matrix is a C_α position of the query and target molecules. A match between positions (j, k) is defined as a matching quadrilateral identified by the lookup table at position k of M_{ri} to position j of M_q. A gap is defined if a position in the query does not have a matching quadrilateral in the target

molecule. There are no mismatches in this implementation. A match score is defined as:

$$\text{match}(j,k) = 3.0 - w_1 \times d_{\text{edge}} - w_2 \times d_{\text{angle}} \quad (3)$$

Where d_{edge} is the absolute difference between the side length ($C\alpha_j, C\alpha_j +$ quadSize) and ($C\alpha_k, C\alpha_k +$quadSize), and d_{angle} is the absolute difference between the θ values of the two quadrilaterals. The values w_1 and w_2 are set experimentally to 2 and 0.1 accordingly. There is an open gap penalty of -5.0 and an extension gap penalty of -2.0. An entry in the DP matrix is as follows:

$$\text{score}(j,k) = \max \begin{cases} \text{score}(j-1,k) + P_{\text{gap}} \\ \text{score}(j,k-1) + P_{\text{gap}} \\ \delta(j,k) \times (\text{score}(j-1,k-1) + \text{match}(j,k)) \end{cases} \quad (4)$$

$\delta(j,k)$ indicates whether the quadrilateral at position k of M_{ri} was found by the lookup table to match the quadrilateral at position j of M_q, and P_{gap} can either be the open gap penalty or the extension gap penalty, according to whether this is a beginning of a gap or not.

Thus, we fill the dynamic programming matrix. Since our alignment is global, we do not need to fill the entire matrix. We only fill a cell with indices (j,k) in the matrix if it obeys:

$$\begin{cases} k \geq j - th_{\text{maxIndel}}, \\ k \leq j + th_{\text{maxIndel}} \end{cases} \quad (5)$$

Where th_{maxIndel} is the size of the maximal insertion or deletion possible under th_{sim} as defined by Equation (1). This technique was introduced in (Landau *et al.*, 1985), and enables us to further increase the efficiency of our algorithm.

When the matrix is complete, we perform a back-tracing procedure to get the highest scoring path through the DP matrix. To find a starting position for the back trace, we search the last column and the last row for the cell with the maximal score.

The procedure results in an assignment of C_α positions between query and target structures. This induces a transformation that minimizes the *rmsd* between the matched C_α coordinates. We perform an iterative refinement procedure to improve the size of the match:

(1) Given a transformation, use dynamic programming to calculate the optimal matching C_α pairs. The parameter th_{dist} determines the maximal distance between two matching C_αs.

(2) Use the given match list to calculate a new transformation.

This procedure is repeated for at most three iterations, and stops if the transformation does not change in two consecutive iterations. If after the refinement the size of the match is smaller than the predefined threshold induced by th_{sim}, the target is discarded.

4 RESULTS AND DISCUSSION

As described in Section 1, the idea for this algorithm has come from the need for a fast and accurate alignment procedure that can handle structural protein fragments of various sizes. Hence in the following we evaluate the performance of our alignment procedure on several datasets. We look at both the quality of the clustering in terms of the number of similarities above th_{sim} which it detects, and also examine the algorithm running times. All experiments described below were run on a single Intel Xeon CPU 2.33 GHz with 16 G RAM. Despite the large memory availability, at no point during the described experiments did the RAM requirements of GOSSIP exceed 350 M.

We perform two different evaluations. First, we evaluate the ability of GOSSIP to achieve the purpose for which it was devised, i.e. to obtain accurate alignments of large sets of structural fragments of various sizes in a short period of time. For this purpose, we compare ourselves to two methods from different categories: MultiProt (Shatsky *et al.*, 2004), and YAKUSA (Carpentier *et al.*, 2005). MultiProt is a method for local multiple structure alignment and can be used for pairwise structure alignment as well. Multiprot, similar to GOSSIP, relies only on the C_α positions of the compared molecules. Also, like in GOSSIP, one of the parameters to the program is the accuracy of the alignment, similarly to th_{dist}. This parameter is necessary for comparing different alignment sizes given by different algorithms. YAKUSA is a swift database scanning method that can scan the entire database of SCOP structures in $\sim$1 min on our Intel Xeon 2.33 GHz CPU. The method is based on representing the protein backbone using dihedral angles calculated from four consecutive C_α atoms. Proteins are described as strings of angles, hence allowing a 1D comparison between proteins. We chose to compare to this method due to several reasons. First, it is very fast. Second, it is shown to compare well to many trusted structural aligners (Carpentier *et al.*, 2005). Third, similarly to a structural aligner, it provides a correspondence list between superposed residues. And fourth, a stand-alone program is available for download, allowing us to perform a thorough comparison to our method.

Our second evaluation is intended to asses the performance of GOSSIP on a set of proteins for which sequence alignment is not sensitive enough to detect similarities. For this, we use two benchmarks: a dataset described in Kolodny *et al.* (2005) that consists of 2930 sequence-diverse structures from version 2.4 of the CATH database (Orengo *et al.*, 1997) and the Astral 1.55 40% non-redundant set consisting of 3539 proteins. We examine the performance of GOSSIP on several similarity thresholds, and compare it to two structural aligners, MultiProt and CE (Shindyalov and Bourne, 1998), as well as to YAKUSA and to BLAST sequence alignment.

We note again that we do not aim to outperform MultiProt, or other local structure alignment and filtering methods, in their own domain. Our purpose is different—to provide a fast, accurate, global alignment method. However, we compare ourselves to local alignment methods because we could not find a stand-alone executable or server that is explicitly meant for global structural alignment.

4.1 Fragment clustering

To test the ability of our program to swiftly detect similarities in sets of protein fragments, we use two subsets of protein building blocks. The first, denoted Sub_{2SH} contains 1500 fragments with two secondary structure elements, a strand and a helix. The length of all fragments in the set is in the range [16–141], with an average length of 27.9. Such fragments cannot be well compared when secondary structure content is required. The second, denoted Sub_{8SH}, contains 1589 fragments with 8 secondary structure elements, starting with a strand and ending with a helix. In these fragments, the number of residues is in the range [56–235], with an average length of 115.4. We took relatively small molecule subsets so that we could compare them in feasible times to an all-against-all using MultiProt alignment.

We performed the following runs for all three methods: GOSSIP all-against-all was run with parameters values $th_{\text{sim}} = 0.8$ (i.e. a match of at least 80% of the larger of the two molecules is required) and $th_{\text{dist}} = 2.0$ Å. MultiProt was run with $th_{\text{dist}} = 2.0$ Å. To achieve

Table 1. Method comparison on the two fragment sets

Fragment subset method	Sub_{2SH}				Sub_{8SH}			
	Gold standard	MultiProt	YAKUSA	GOSSIP	Gold standard	MultiProt	YAKUSA	GOSSIP
Running time	–	197 min	5.1 min	8.3 min	–	3087 min	15.7 min	8.21 min
Num similarities	3840	2567	1852	3106	3238	3159	2902	3170
Num unique similarities	–	289	733	812	–	19	45	24

Comparison of the three methods on an all-against-all procedure that was run on the two subsets Sub_{2SH} and Sub_{8SH}. Running times are given on a single Intel Xeon CPU, 2.33 GHz.

$th_{\text{sim}} = 0.8$ for MultiProt, we ran a pairwise all-against-all, and then filtered out the results with a match size of less than 80% of the larger of the two molecules. For YAKUSA, we ran a database scan against the relevant protein subset for every molecule in that subset. In YAKUSA, we could not control accuracy parameters, since it is hard to compare angstroms to angle degrees. We did, however, make the following adaptations:

- For Sub_{2SH}, we changed the minimal seed length to 4 (our minimal quadrilateral length) instead of the default value of 6.
- We only considered as hit alignments that covered more than 80% of the longer structure.
- We discarded alignments for which the best RMSD was above 3.0Å.
- Since YAKUSA's backbone representation does not allow the alignment of the first and last two residues, we altered our fragment datasets so that each fragment was cut to be larger by three residues (one at the beginning and two at the end), except for fragments at the edges of a structure. Alignment ratio was calculated relative to the original fragment length.
- We set YAKUSA to output 500 top scoring results instead of the default 50, in order not to miss similarities that were preceded by false positives.

To compare the methods we devised a *gold standard*, similarly to Budowski-Tal *et al.* (2010). Our gold standard considers a hit to be any similarity above 80% that was detected by either one of the methods. Since false positives are rare at high values of similarity, we can consider all similarities detected by either method as part of the gold standard. We then counted how many of these similarities were detected by each method, and how many unique similarities each method introduced.

Table 1 presents the results of running the methods on the two described subsets. Clearly, the running times of MultiProt are by far longer than those of the other two methods, which use a linear representation of the protein structure, especially on the larger fragment dataset due to a larger number of possible local similarities. GOSSIP and YAKUSA show similar running times, but YAKUSA is able to identify less similarities above 80%. This is especially evident for the shorter subset where it detects less than half the gold standard. To summarize, Table 1 suggests that GOSSIP is able to detect fragment similarities at least as successfully as MultiProt, and in times which are comparable to those of YAKUSA. An interesting observation is the high number of unique similarities detected by YAKUSA. Many of them are due to differences in the measures of alignment accuracy—distance between C_α atoms versus dihedral angle differences.

It is interesting to characterize the cases where each method prevails over the others. Due to shortage of space, this analysis is presented as Supplementary Material.

4.2 Alignment of sequence-diverse benchmarks

In this section, we analyze the performance of GOSSIP on two benchmarks. The benchmark described in (Kolodny *et al.*, 2005) consists of 2930 protein structures from CATH, with an average length of 144.7 residues. Each sequence in the dataset has a FASTA E-value (Pearson and Lipman, 1988) of above 10^{-4} to every other sequence in the benchmark.[1] The Astral 1.55 40% non-redundant dataset (Brenner *et al.*, 2000) consists of 3613 protein structures with an average length of 173.2 residues, for which any pair of structures has a sequence similarity of less than 40%. Version 1.55 is dated July 2001[2] and hence several structures in this benchmark were declared obsolete. These were removed from our dataset, leaving us with 3539 protein structures. For both of the described benchmarks, finding structural similarities is not trivial.

For all runs in this test, we use an accuracy threshold (th_{dist}) of 3.5 Å. All GOSSIP runs were performed on a single Intel Xeon CPU 2.33 GHz. We compare our performance to two other structural aligners—MultiProt and CE (Shindyalov and Bourne, 1998). For both methods, we ran an all-against-all alignment procedure on the two benchmarks. For MultiProt, we defined an accuracy level of 3.5 Å, similar to ours. We also present a comparison of GOSSIP to BLAST (Altschul *et al.*, 1990) and to YAKUSA (Carpentier *et al.*, 2005).

We ran GOSSIP with eight different similarity thresholds, between 0.6 and 0.95, in order to assess its performance for a varying degree of global similarity in protein structures. We focus on these similarity thresholds because lower values of th_{sim} require detection of similar substructures which is a problem that we have not set out to solve. We expect GOSSIP to have difficulty in performing local structure alignment because (i) it uses descriptors that span substantial portions of a structure and will thus miss many local structural similarities and (ii) it aligns the signatures globally when searching for an initial correspondence and may thus often fail to identify an optimal local alignment. In addition, decreasing the similarity threshold increases running time considerably.

Table 2 presents the running times measured by performing all-against-all comparisons on the two sequence-diverse datasets. For GOSSIP, YAKUSA and BLAST we present the running times of the

[1] E-values were calculated by querying each sequence against the entire set of CATH version 2.4 domains.

[2] This early version was used due to its smaller size, for the purpose of simplifying the performance of our large-scale analysis procedure.

Table 2. Comparison of running times for CATH and Astral benchmarks

Benchmark	MultiProt	CE	GOSSIP				YAKUSA	BLAST
Similarity threshold	–	–	**0.6**	**0.7**	**0.8**	**0.9**	–	–
CATH benchmark	448.8 h*	925.4 h*	9 h	2.5 h	26 min	5.1 min	2.6 h	∼2 min
Astral benchmark	911.7 h*	2480.5 h*	17.3 h	4.7 h	43.6 min	9.1 min	4 h	∼4 min

Numbers succeeded by an asterisk indicate estimated running times, as explained in the text.

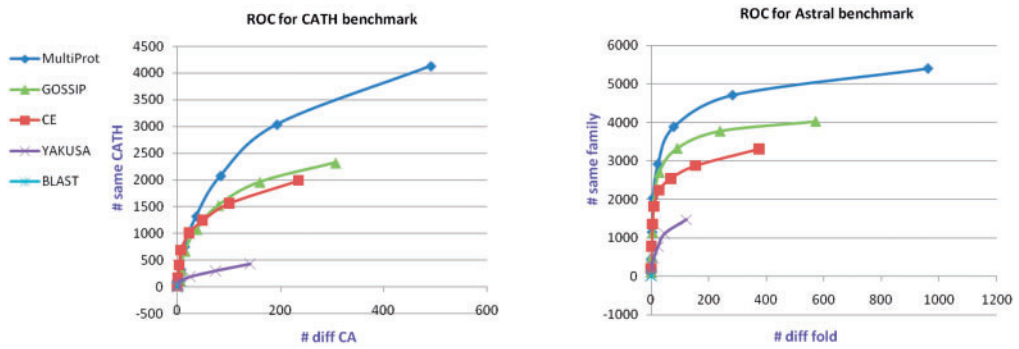

Fig. 2. ROC curve analysis for the sequence-diverse benchmarks. The *Y*-axis presents the number of true similarities detected by each method as defined by the appropriate classification (CATH and SCOP). The *X*-axis shows the number of structural similarities found with different classifications. True and false positives were calculated for eight different thresholds of structural similarity, between 0.6 and 0.95.

procedure on the dataset. MultiProt and CE runs were performed in parallel on a cluster of CPUs, hence we provide only educated estimates of their expected running times. For both the CATH and Astral benchmarks, we chose a subset consisting of 267 and 348 proteins, respectively. The proteins in each subset were chosen such that their average length and standard deviation are similar to the average protein length and standard deviation of the entire benchmark dataset (for the CATH benchmark $\mu = 144.7$ and $\sigma = 87.47$, for the Astral benchmark $\mu = 173.2$ and $\sigma = 120.23$). We ran an all-against-all comparison on each of these subsets on a single Intel Xeon 2.33 GHz CPU, and measured the time it took. We then calculated the expected running time on the entire benchmark by multiplying the observed time by the ratio of the number of aligned pairs of the entire dataset and that of the appropriate subset. This procedure was performed both for MultiProt and CE, on each dataset separately. The results presented in Table 2 indicate that the running times of GOSSIP are several folds below those estimated for the two compared structural aligners, and resemble the running times achieved by the 1D aligner YAKUSA. In addition, GOSSIP running times decrease rapidly with increasing similarity thresholds. We conclude that in cases where similarities below a certain similarity threshold are not of interest, GOSSIP can provide results in a significantly shorter time than standard structural aligners, provided that its ability to detect alignments above a given similarity threshold is comparable to that of standard structural aligners.

To show that GOSSIP's performance does not fall from that of other structural aligners, we perform the following analysis. We first define a measure for evaluating the performance of a structural aligner. For this, we use the corresponding classifications as our labels of truth: for the benchmark by (Kolodny *et al.*, 2005), we define a positive label to be two structures from the same CATH classification (homologous structures). In our CATH benchmark, there are 11 465 such similarities. For the Astral benchmark, we define a positive label for two structures that belong to the same SCOP family. There are 8573 such similarities in our Astral benchmark. For each value of th_{sim} , we measure the number of true similarities each method is able to detect above that threshold. We also calculate the number of possibly erroneous similarities by counting the number of structural similarities detected from different CA classifications and different fold classification for the CATH and Astral benchmarks, respectively. We note that similarities from different classifications are not necessarily erroneous since different classification systems have different criteria for placing two structures into the same class. However, this number can help to identify whether a method introduces an unreasonable number of false positives.

The number of similarities is measured as follows. For MultiProt and GOSSIP, we count the number of similarities above each similarity threshold, under accuracy threshold (th_{dist}) of 3.5. YAKUSA and CE do not require an accuracy threshold in their calculations, and using the RMSD value instead of the accuracy threshold is not equivalent. Hence, for each threshold th_{sim} we calculate the maximal RMSD (denoted $rmsd_{\text{max}}$) reached by MultiProt for all the similarities it found. We then count the number of hits for YAKUSA and CE with an alignment of length above th_{sim} and rmsd at most $rmsd_{\text{max}}$.

Figure 2 presents a comparison of the five methods on the two benchmarks using a ROC curve analysis. For each method, we

measure the number of true structural similarities it is able to detect (Y-axis) above each similarity threshold, starting with 0.95 and going down to 0.6, and plot it against the number of structural similarities detected that belong to different CA and fold classifications (X-axis). For BLAST sequence similarity percentage was used, since it is unclear how to correctly relate structural similarity levels to matching E-values. Notice that the ROC curves are not complete because the lowest similarity threshold we consider is 0.6 and thus similarities detected below this threshold are not entered into the statistics. The data from which these curves were constructed is provided as Supplementary Tables 1 and 2.

Figure 2 presents two interesting observations. First, it demonstrates that 1D alignment methods (BLAST and YAKUSA) are not sensitive enough on our sequence-diverse datasets. For BLAST, which is barely visible as a single spot on each plot, this observation merely provides additional evidence for the choice of the sequence-diverse datasets. For YAKUSA, it is interesting to ask why the results here are inferior to those achieved on the fragment subsets with respect to GOSSIP and MultiProt. The reason for this is that the fragment datasets contain more sequential redundancy, implying that structural accuracy is also higher and easier to detect. This is also supported by the fact that for the sequence-diverse benchmarks we use a higher accuracy threshold (3.5 Å, as opposed to 2.0 Å) and are able to locate only around half the pairs classified as similar, providing an additional indication that similarities in these datasets are harder to identify. Figure 2 also demonstrates that the number of similarities detected by all three structural aligners are on a different scale than the 1D aligners. Their comparative performance changes between the similarity thresholds: when $th_{\text{sim}} = 0.6$, a relatively local similarity threshold, MultiProt detects significantly more similarities than CE and GOSSIP. MultiProt's superiority becomes less significant as the similarity threshold increases and alignment becomes global. The results indicate that CE is slightly better in identifying similarities for high similarity thresholds, while MultiProt is better for lower similarity thresholds. GOSSIP is in between the two aligners for most thresholds, but is clearly in the same sensitivity and accuracy range as the two structural aligners. Note that GOSSIP's performance here may seem slightly inferior to the one presented on the fragment datasets with respect to MultiProt on a similar threshold of 0.8. This is explained by noticing that GOSSIP's advantage is mainly visible on the shorter fragment dataset, while the CATH and Astral datasets consist mostly of longer structures. In addition, the fragment datasets contain more structural redundancy and hence structural similarity is easier to detect. To summarize, by combining the observations presented in Figure 2 with those presented in Table 2, we can conclude that GOSSIP is able to perform global structure identification comparably to trusted structural aligners, while requiring significantly less computational power to do so.

5 CONCLUSIONS

Here we present GOSSIP, a very fast and accurate method for performing a global all-against-all alignment of the protein space. As opposed to most existing methods, we do not aim to identify local similar substructures between two or more proteins. Our aim is to give a binary answer of whether two proteins are similar above a given threshold, and to output a superposition between similar structures. With this tool, we hope to provide a solution for several important tasks that are currently difficult to perform: first, GOSSIP is already being used for clustering of large fragment datasets; second, it can also be used for fast and accurate structural redundancy removal from any set of proteins or protein fragments; and third, it can be used to perform swift database scans for identifying the structural neighbors of a given query protein. The main advantages of GOSSIP are as follows: first, it is very efficient. We show that for high similarity thresholds it is even more efficient than a swift procedure for database scanning. Second, it can be applied to any set of proteins, of any length, without requiring additional information such as secondary structure content. Third, it is not a filtering method, outputs a superposition and thus does not introduce false positives into its results. And fourth, it does not require any preprocessing as do many methods for structure comparison. GOSSIP's main drawback is that for low similarity thresholds it may not find the best superposition because it searches for a global alignment rather than for similar substructures.

We have evaluated the performance of GOSSIP on several datasets. We tested how it handles protein fragments of various sizes. We compared ourselves to MultiProt, a trusted method for protein structure alignment, and to YAKUSA, a well-performing database scanning method. We showed that our running times are by several folds shorter than MultiProt and that for longer fragments they are shorter than YAKUSA as well. Our ability to detect global similarities above a certain threshold is superior to the two compared methods, especially for shorter fragment datasets. We then went on to evaluate GOSSIP on two sequence-diverse datasets of CATH and SCOP structures and found that for a similarity threshold of 0.6 and up, its sensitivity and accuracy are comparable to two trusted structural aligners, yet its speed is faster several times.

We note that the time feasibility of our program can further be improved, since our method can easily be parallelized to run on several computers. Recall that we use different lookup tables for different molecule sizes, hence each lookup table is a separate entity and performs its clustering independently of the other lookups. At the end, the results of all clustering procedures can be merged.

This work is based on creating specific structural signatures to characterize parts of the protein. The key here is to find a balance between the specificity of the signature and its generalization power. Our experimental results indicate that we have found a good set of parameters for managing the lookup table, that balances between a granularity that is too small to detect similarities, and huge bucket sizes that cannot provide a good separation between different structural signatures.

We hope that GOSSIP will provide a useful means for performing fast and accurate analysis of large datasets of protein structures.

ACKNOWLEDGEMENT

We thank Dr Rachel Kolodny for providing us with the CATH benchmark that was used in this study.

Funding: IK was supported in part by a fellowship from the Edmond J. Safra Bioinformatics Program at Tel-Aviv university. The research of HJW has been supported in part by the Israel Science Foundation (grant no. 1403/09) and the TAU Minerva-Minkowski Geometry center. This project has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institute of Health, under contract number (HHSN261200800001E). The

content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products or organizations imply endorsement by the US Government. This research was supported (in part) by the Intramural Research Program of the NIH, National Cancer Institute, Center for Cancer Research.

Conflict of Interest: none declared.

REFERENCES

Alireza,S. *et al.* (2005) Pads: protein structure alignment using directional shape signatures. *Lect. Notes Comput. Sci.*, **3453**.

Altschul,S. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Andreeva,A. *et al.* (2008) Data growth and its impact on the scop database: new developments. *Nucleic Acids Res.*, **36**, D419–D425.

Baeten,L. *et al.* (2008) Reconstruction of protein backbones from the brix collection of canonical protein fragments. *PLos Comput. Biol.*, **4**.

Brenner,S. *et al.* (2000) The astral compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, **28**, 254–256.

Bryson,K. *et al.* (2005) Prediction of novel and analogous folds using fragment assembly and fold recognition. *Proteins*, **61**, 143–151.

Budowski-Tal,I. *et al.* (2010) Fragbag, an accurate representation of protein structure, retrieves structural neighbors from the entire pdb quickly and accurately. *Proc. Natl Acad. Sci. USA*, **107**, 3481–3486.

Can,T. and Wang,Y. (2003) Ctss: a robust and efficient method for protein structure alignment based on local geometrical and biological features. *IEEE Computer Society Bioinformatics Conf.*, **2**, 169–179.

Carpentier,M. *et al.* (2005) Yakusa: a fast structural database scanning method. *Proteins*, **61**, 137–151.

Carugo,O. and Pongor,S. (2002) Protein fold similarity estimated by a probabilistic approach based on c(alpha)-c(alpha) distance comparison. *J. Mol. Biol.*, **315**, 887–898.

Choi,I. *et al.* (2004) Local feature frequency profile: a method to measure structural similarity in proteins. *Proc. Natl Acad. Sci. USA*, **101**, 3797–3802.

Csaba,G. *et al.* (2008) Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, **24**, i98–i104.

Dror,O. *et al.* (2003) MASS: multiple structural alignment by secondary structures. *Bioinformatics*, **19** (Suppl. 1), i95–i104.

Friedberg,I. *et al.* (2007) Using an alignment of fragment strings for comparing protein structures. *Bioinformatics*, **23**, e219–e224.

Greene,L. *et al.* (2007) The cath domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res.*, **35**, D291–D297.

Hasegawa,H. and Holm,L. (2009) Advances and pitfalls of protein structural alignment. *Curr. Opin. Struct. Biol.*, **19**, 341–348.

Holm,L. and Park,J. (2000) Dalilite workbench for protein structure comparison. *Bioinformatics*, **16**, 566–567.

Kishon,E. *et al.* (1991) 3-d curve matching using splines. *J. Robot. Syst.*, **8**, 723–743.

Kolodny,R. *et al.* (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J. Mol. Biol.*, **346**, 1173–1188.

Kosloff,M. and Kolodny,R. (2008) Sequence-similar, structure-dissimilar protein pairs in the pdb. *Proteins*, **71**, 891–902.

Krissinel,E. and Henrick,K. (2004) Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta crystallographica*, **60**, 2256–2268.

Lamdan,Y. and Wolfson,H. (1988) Geomtric hashing: a general and efficient model-based recognition scheme. *Proceedings of the 2nd International Conference on Computer Vision (ICCV)*. pp. 238–249.

Landau,G. *et al.* (1985) An efficient string matching algorithm with k differences for nucleotide and amino acid sequences. *NAR*, **14**, 31–46.

Lisewski,A. and Lichtarge,O. (2006) Rapid detection of similarity in protein structure and function through contact metric distances. *Nucleic Acids Res.*, **34**, e152.

Needleman,S. and Wunsch,C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Nussinov,R. and Wolfson,H. (1991). Efficient detection of three-dimensional motifs in biological macromolecules by computer vision techniques. *Proc. Natl Acad. Sci. USA*, **88**, 10495–10499.

Orengo,C. *et al.* (1997) CATH - a hierarchic classification of protein domain structure. *Structure*, **5**, 1093–1108.

Pandini,A. *et al.* (2010) Structural alphabets derived from attractors in conformational space. *BMC Bioinformatics*, **11**, 97.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence analysis. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Røgen,P. and Fain,B. (2003) Automatic classification of protein structure by using gauss integrals. *Proc. Natl Acad. Sci. USA*, **100**, 119–124.

Shatsky,M. *et al.* (2004) A method for simultaneous alignment of multiple protein structures. *Proteins*, **56**, 143–156.

Shindyalov,I. and Bourne,P. (1998) Protein structure alignment by incremental combinatorical extension (ce) of the optimal path. *Protein Eng.*, **11**, 739–747.

Subbiah,S. *et al.* (1993) Structural similarity of dna-binding domains of bacteriophage repressors and the globin core. *Curr. Biol.*, **11**, 141–148.

Wang,G. and Dunbrack, R.L. (2003) Pisces: a protein sequence culling server. *Bioinformatics*, **19**, 1589–1591.

Zhang,Y. and Skolnick,J. (2005a) The protein structure prediction problem could be solved using the current pdb library. *Proc. Natl Acad. Sci. USA*, **102**, 1029–1034.

Zhang,Y. and Skolnick,J. (2005b) Tm-align: a protein structure alignment algorithm based on tm-score. *Nucleic Acids Res.*, **33**, 2302–2309.

Zotenko,E. *et al.* (2006) Secondary structure spatial conformation footprint: a novel method for fast protein structure comparison and classification. *BMC Struct. Biol.*, **6**.