Accelerating Screening of 3D Protein Data with a Graph Theoretical Approach *

Cornelius Frömmel[†], Christoph Gille[†]**, Andrean Goede[†],
Clemens Gröpl[‡], Stefan Hougardy[‡],
Till Nierhoff[‡], Robert Preißner[†], and Martin Thimm[‡]

June 10, 2003

** To whom correspondence should be addressed

*Clemens Gröpl supported by Deutsche Forschungsgemeinschaft (DFG), grant no. Pr 296/6-3, Stefan Hougardy and Martin Thimm supported by the DFG Research Center "Mathematics for key technologies", Andrean Goede and Robert Preißner supported by the Berlin Center for Genome Based Bioinformatics

[†]Institut für Biochemie, Charité, Monbijoustr. 2, Berlin, D-10117, Germany
[‡]Institut für Informatik, Humboldt-Universität zu Berlin, Berlin, D-10099, Germany

Abstract

Motivation. The Dictionary of Interfaces in Proteins (DIP) is a data base collecting the three-dimensional structure of interacting parts of proteins that are called patches. It serves as a repository, in which patches similar to given query patches can be found. The computation of the similarity of two patches is time consuming and traversing the entire DIP requires some hours. In this work we address the question how the patches similar to a given query, can be identified by scanning only a small part of DIP. The answer to this question requires the investigation of the distribution of the similarity of patches.

Results. The score values describing the similarity of two patches can roughly be divided into three ranges that correspond to different levels of spatial similarity. Interestingly, the two iso–score lines separating the three classes can be determined by two different approaches. Applying a concept of the theory of random graphs reveals significant structural properties of the data in DIP. These can be used to accelerate scanning the DIP for patches similar to a given query. Searches for very similar patches could be accelerated by a factor of more than 25. Patches with a medium similarity could be found ten times faster than by brute-force search.

Contact. `christoph.gille@charite.de`,
`hougardy@informatik.hu-berlin.de`

1 Introduction

1.1 *In-silico* screening

Structure-based *in-silico* screening (ISS) methods are widely used within the pharmaceutical industry lead discovery process (Good, 2001) as proven by numerous successful applications: farnesyltransferase inhibitors (Perola et al., 2000), erythromycin analogs, (Siani et al., 2000), microtubule-stabilizer, (Wu et al., 2001) morphine analogs (Poulain et al., 2001), Cdk4 inhibitors (Honma et al., 2001) and antiviral quinolones (Filipponi et al., 2002). Generally, two ISS-approaches can be distinguished. Similarity searching (Willett et al., 1998; Xue & Bajorath, 2000) and docking, (Gane & Dean, 2000). The high computational effort of ISS is usually reduced by a number of simplifications and restrictions. For instance flexibility of ligands and targets are at least partly restricted and solvent are often ignored (Stahl, 2000). The reduction of the search space decreases the computational effort as realized by the application of topological pharmacophore models (Schneider et al., 1999) or by random selection of compounds from a virtual combinatorial library (Beroza et al., 2000). In the present work we describe the reduction of structure comparisons using a graph theoretical approach.

1.2 The Dictionary of Interfaces in Proteins (DIP)

Peptides sample a huge conformational space due to the large number of rotatable bonds. In a previous work the Dictionary of Interfaces in Proteins (DIP) was developed to exploit the learning set of 10^7 molecular docking interfaces included in the 20,000 known protein structures (Preißner et al., 1998). Matching between protein substructures and low molecular weight compounds works well (Preißner et al., 2001), thus bridging the way to a new conformational space for ISS.

1.3 Accelerating queries in DIP

Currently, searching DIP for conformations similar to a query molecular structure is performed by matching the query against all structures in DIP (linear search), which takes some hours. The issue of faster-than-linear search in databases has been examined extensively in the area of information retrieval from textual data collections, where the method of singular value decomposition is very useful as the data can be represented in vectorial form (Jiang et al., 1999). However, the application of such methods to accelerate search in DIP is not feasible, as the objects in this database have no obvious vectorial description. This is most evident by the fact that the similarities neither obey the triangle inequality nor are they transitive. Similarity values may be be calculated from score values by subtraction from 100%. If the relation of data objects is only given by pairwise similarity scores, as is the case in DIP, then a reduction of the number of comparisons is typically based on prior grouping of the objects into clusters of similar objects, which are not necessarily disjoint. The clustering problem has been considered mainly in the context of computer vision so far (Puzicha et al., 1999) and is now also widely applied in computational biology. The biological areas range from sequences of proteins (Hofmann & Buhmann, 1997) to gene expression data (Hartuv et al. 1999, Hartuv & Shamir, 2000, Matula, 1972, Wu & Leahy 1993) and even protein domains (Bolten et al., 2001).

Their success was hope and motivation to exploit the power of clustering for DIP.

2 Databases and Methods

2.1 Data

The data objects that we are dealing with are molecular surface patches (MSP) in DIP, (Preißner et al. 1998) as illustrated in Fig. **??**. An MSP is a set of atoms of a secondary structural element (SSE) that are in spatial proximity to atoms of another SSE, ligand or solvent.

For the design of efficient search strategies, we independently investigated five data sets containing about 1000 MSPs with 25-29, 40-49, 50-59, 60-69, and more than 100 atoms, which are representative for DIP.

This subdivision was necessary since superposition requires patches of similar size. All pairs of patches were superimposed using the superposition algorithm of DIP (Preißner et al., 1998) which is available at http://www.drug-redesign.de/superposition.html. As a one-dimensional scoring function for superpositions the following formula was used:

$$S = n_r \exp(-\mathrm{rmsd}/\text{Å}), \tag{1}$$

where n_r is the proportion of atoms of the smaller patch that are superimposed and *rmsd* is the root mean square distance. The values S range from 0 to 100[%].

2.2 Methods

2.2.1 Discrimination of three ranges of similarity

The 3D superposition algorithm finds an assignment of atoms in one patch to atoms in the other one, resulting in a list of pairs of atoms. Each atom is part of a certain amino acid of the protein and a list of pairs of corresponding amino acids can be derived. Usually, this list contains about 3 to 7 amino acids for patches with 30 to 70 atoms. In this list we can count assignments of identical amino acids to measure sequence similarity between both patches. In our analysis we distinguish three classes of patch pairs: pairs with *dissimilar sequences*, pairs with *similar sequences* having at least two identical amino acids, and *sequence identical* pairs, *i.e.* pairs where all assigned amino acids are identical. Sequence identical pairs originate in almost all cases from homologous proteins.

2.2.2 The evolution of the threshold graph

For each set of MSPs, the pairwise scores can be arranged in a similarity matrix, each of whose rows and columns corresponds to one MSP. This matrix can be considered as the (weighted) adjacency matrix of a graph that we call "similarity graph". For every given threshold score t this similarity graph can be turned into an (unweighted) graph ("threshold graph") in the following way: each vertex represents one patch and there is an edge between a pair of patches if the score of the pair exceeds t. We use the concept of the threshold graph to analyze the similarity data for hidden structure as follows: For a set of MSPs, 100 threshold graphs are computed, where the value of t ranges from 0.01 to 1 in steps of 0.01. This sequence of threshold graphs, which can be considered as an "evolution of the threshold graph" is examined for the value of several graph parameters. For instance, a pair of vertices in a graph is connected if there is a path of edges of the graph, along which one can walk from one to the other. A connected component is a maximal set of vertices that are pairwise connected. As a special case, an isolated vertex, which has no edges to any other vertex, is a connected component on its own. A largest connected component is one, which has the maximal number of vertices among all connected components (Fig. **??**). The parameters that we observed during the evolution were: the number of

components, the average size of components, the number of isolated vertices, and the size of the largest component. The goal was to identify parameters of the graph that exhibit some "unexpected behavior" when plotted *versus* the threshold value t.

2.2.3 Random permutation of the edges of the similarity graph

To test for "unexpected behavior" of the graph parameter, we considered random permutations of the entries of the adjacency matrix. The evolution of the threshold graph of the permuted similarity graph is examined for the same graph parameters as the original one. If the behavior of a graph parameter in the permuted similarity graph deviates considerably from the original similarity graph, this indicates that the original similarity graph has an "unexpected structure". Note that this criterion has a high level of confidence for the following reason:

Permuting the original similarity graph conserves the distribution of score values and thus the number of edges of each threshold graph. Therefore, the evolution of the permuted similarity graph is like the classical evolution of the graph $G_{n,m}$, where n is the number of MSPs and m increases from 0 to $\frac{1}{2}n(n-1)$. This evolution has been extensively studied in the theory of random graphs (Bollobás 2001). According to this theory these graph parameters, including those investigated here, are usually very stable upon permutation for all thresholds t.

2.2.4 Accelerated screening by hierarchical subdivision

The first search procedure that we considered organizes the data in a binary tree as follows. A first MSP is chosen as the root of the tree. All other MSPs are compared with this root and ordered by their score values. Now a critical score value s is computed and the MSPs are partitioned into two sets A and B such that the set A contains all MSPs with a score value smaller than s and B contains all MSPs with a score value larger than s. The set A is taken as the left child and the set B as the right child of the root. The critical score value s is computed in such a way that the sets A and B have almost the same size while at the same time there are a fewest possible number (ideally zero) edges in the similarity graph that connect MSPs from the set A with MSPs from the set B. This procedure is repeated recursively until the sets A and B have size 1 or there is no critical value s that allows to almost half the current set.

To search for a given MSP, the binary tree is used as follows. Starting with the root the given MSP is compared to the current vertex in the tree. The tree is followed to the left branch, if the score value of the MSP compared with the current vertex is smaller than the critical score value s stored with the current vertex. Otherwise the right branch is taken. Whenever a comparison with a vertex in the binary tree yields a score value higher than the given threshold, the vertex is output. Note that the number of comparisons is only dependent on the depth of the tree. Therefore it is desirable to have a binary tree that is as balanced as possible. In the best case only $\log_2 n$ comparisons are needed if n MSPs are in the data structure.

2.2.5 Accelerated screening by dominating sets

A subset of vertices, called *representatives*, is chosen so that every other vertex in the graph is covered by at least one of these representatives. A vertex is *covered* by a representative, if there exists an edge in the threshold graph between the vertex and the representative. In graph theoretic terms this is called a dominating set. This dominating set is built up using a simple greedy approach. First the vertices of the threshold graph are ordered by decreasing average weight (in the similarity graph) of their neighbourhood. Proceeding in this order, vertices are added to the dominating set as long as there are vertices in the graph that are not covered. There may be vertices that are linked to several representatives.

The search for a given MSP is then performed as follows: We again use the above ordering of the representatives. First the given MSP is compared to the representative. If the score value is high enough, the query MSP is also compared to all the neighbours of the representative, otherwise these neighbours are skipped.

Note that this procedure is also applicable, if the similarity matrix is not fully known, as one can easily add new MSPs to this search structure and, if necessary, update the set of representatives.

The tradeoff between the reduction of comparisons and the number of false negatives can be controlled by additional parameters, *e.g.* the procedure can be configured to cover vertices by heavy edges, or by more than one representative.

The main reason for the missed patches is that during the superposition procedure not all atoms of the patches are superimposed. When the superposition of two patches A and B yields a high rmsd and superposition of patches B and C have a low rmsd the pair A and C does not necessarily superimposed with a high rmsd. This is particularly the case when the set of assigned atoms of B differs markedly between a superposition with A compared to a superposition with C.

Applying this considering to the search strategy the patch A may be a representative, B may be a query and C may be a missed patch. Because A and B are dissimilar the algorithm falsly concludes that B and C are dissimilar as well.

3 Results and Discussion

3.1 The distribution of the score value

The prerequisite for the acceleration of searches in DIP was the investigation of the distribution of the score values, because it significantly influences the behavior and success of any such approach. The density plot of the rmsd versus the relative number of superimposed atoms revealed three agglomerations, a very small one with very high scores, a medium sized one with high scores, and a larger one with low scores (Fig. **??**).

A statistical analysis of the average score values in the presence and absence of sequence similarity (Table **??**) supports the following interpretation of the data produced by the superposition algorithm (Fig. **??**). The large agglomeration to the top left corresponds to sequence-dissimilar MSP pairs while the part to the lower right results mainly from sequence-identical MSP pairs.

The iso-score line separating the top left and the middle parts corresponds to the score value, that is attained by as many sequence-dissimilar MSPs as sequence-similar MSPs. Therefore, pairs that have an accidental similarity, *i.e.* a similarity that cannot be explained from the sequence of the protein, are expected in the middle part of the distribution, which corresponds to the second level of spatial similarity.

3.2 Bends in the plot of the largest graph component

Comparing the evolution of the threshold graph derived from the original similarity graph with the threshold graph of the randomly permuted similarity graph we found that the largest graph component was the only graph parameter that exhibited "unexpected behavior" (Fig. **??**). The respective plot of the size of the largest graph component *versus* the threshold t shows two remarkable bends, which are absent in the permuted data (Fig. **??**). Surprisingly, the score values at, which these two bends occur (Fig. **??**) coincide with the score values that separate the three ranges of spatial similarity (Fig. **??**). This is a constant feature for all analyzed sets of patches extracted from DIP irrespectively of the patch size. Each set contains only patches with sizes within a certain range. However, the exact values vary with the size of the patches. The score values of the first bend depend only on the mean size of patches and are well approximated by the empirical formula

$$15 + \frac{700}{\min(n_1, n_2)} \tag{2}$$

where n_1 and n_2 denote the sizes of the two MSPs. This formula estimates the score value that separates the first and the second level of spatial similarity as shown for MSP sizes between 16 and 89 (Tab. **??**).

3.3 Acceleration of DIP searches

There is a strong need for approaches that accelerate the process of identification of structural neighbors to a given query patch.

We applied two techniques from graph theory to order molecular objects according to their three-dimensional shape. Both methods are generally applicable but each is advantageous in different situations.

Arranging the patches in a binary tree yielded almost balanced search trees for scores in the third level of spatial similarity. As a consequence, this rapid approach is only suitable to find very similar patches. In the examined data, all close structural neighbors were identified with only 1/25 of the comparisons needed for linear search.

Screening DIP using dominating sets proved to be efficient in the second level of spatial similarity (Fig. **??**) achieving a ten fold acceleration (Tab. **??**). This is paid for by a moderate proportion of false negatives, usually 15%, which is tolerable for an application of DIP as a repository of potential building blocks for ligands. The speed-up factor and the sensitivity rise with increasing amount of data and can be influenced by setting the least score value required for a patch to be represented by another one. Since small compounds are in many aspects similar to MSPs and most entries in the Cambridge database (Hall et al., 1991) consist of less than 100 atoms the techniques presented in this paper may also be applied to accelerate the identification of ligands for proteins.

4 References

Bajorath, J. (2001) Selected concepts and investigations in compound classification, molecular descriptor analysis and virtual screening, *J. Chem. Inf. Comput. Sci.*, **41**, 233-245.

Beroza, P., Bradley, E.K., Eksterowicz, J.E., Feinstein, R., Greene J., Grootenhuis, P.D., Henne, R.M., Mount J., Shirley, W.A., Smellie, A., Stanton, R.V., Spellmeyer, D.C. (2000) Applications of random sampling to virtual screening of combinatorial libraries, *J. Mol. Graph Model*, **18**, 335-342.

Bollobás, B. (2001) Random graphs, 2nd ed., Cambridge University press.

Bolten E., Schliep A., Schneckener S., Schomburg D., Schrader R. (2001) Clustering protein sequences–structure prediction by transitive homology, *Bioinformatics*, **17**, 935-941.

Filipponi, E., Cruciani, G., Tabarrini, O., Cecchetti, V., Fravolini, A. (2002) QSAR study and VolSurf characterization of anti-HIV quinolone library. *J. Comput. Aided. Mol. Des.*, **15**, 203-217.

Gane, P.J., Dean, P.M. (2000) Recent advances in structure-based rational drug design. *Curr. Opin. Struct. Biol.*, **10**, 401-404.

Good, A. (2001) Structure-based virtual screening protocols, *Curr. Opinion. Drug. Disc. Dev.*, **4**, 301-307.

Hall S.R., Allen F.H., Brown I.D. (1991) The Crystallographic Information File: a New Standard Archive File for Crystallography *Acta Cryst.*, **47**, 655-685

Hartuv, E., Schmitt, A., Lange, J., Meier-Ewert, S., Lehrach, H., Shamir, R. (1999) An algorithm for clustering cDNAs for gene expression analysis using short oligonucleotide fingerprints, *RECOMB 1999*, 188-197.

Hartuv, E., Shamir, R. (2000) A clustering algorithm based on graph connectivity, *Information Processing Letters*, **76**, 175-181.

Hofmann T., Buhmann, J. (1997) Pairwise Data Clustering by Deterministic Annealing, *IEEE Trans. Pattern Analysis Machine Intelligence*, **19**, 1-17.

Honma, T., Hayashi, K., Aoyama, T., Hashimoto, N., Machida, T., Fukasawa, K., Iwama, T., Ikeura, C., Ikuta, M., Suzuki-Takahashi, I., Iwasawa, Y., Hayama, T., Nishimura, S., Morishima, H. (2001) Structure-based generation of a new class of potent Cdk4 inhibitors: new de novo design strategy and library design. *J. Med. Chem.* , **44**, 4615-4627.

Jiang, F., Kannan R., Littman M., Vempala S., Efficient Singular Value Decomposition via Improved Document Sampling, Technical Report CS-1999-5, Duke University.

Matula D.W. (1972) k-Components, clusters and slicings in graphs, *SIAM J. Appl. Math.*, **22**, 459-480

Perola, E., Xu, K., Kollmeyer, T.M., Kaufmann, S.H., Prendergast, F.G., Pang, Y.P. (2000) Successful virtual screening of a chemical database for farnesyltransferase inhibitor leads. *J. Med. Chem.*, **43**, 401-408.

Poulain, R., Horvath, D., Bonnet, B., Eckhoff, C., Chapelain, B., Bodinier, M.C., Deprez, B. (2001) From hit to lead. Analyzing structure-profile relationships, *J. Med. Chem.*, **44**, 3391-3401.

Poulain, R., Horvath, D., Bonnet, B., Eckhoff, C., Chapelain, B., Bodinier, M.C., Deprez, B. (2001) From hit to lead. Combining two complementary methods for focused library design. Application to mu opiate ligands, *J. Med. Chem.*, **44**, 3378-3390.

Preißner, R., Goede, A., Frömmel, C. (1998) Dictionary of interfaces in proteins (DIP). Data bank of complementary molecular surface patches, *J. Mol. Biol.*, **280**, 535-550.

Preißner, R., Goede, A., Rother, K., Osterkamp, F., Koert, U., Frömmel, C. (2001) Matching organic libraries with protein-substructures, *J Comput. Aided Mol. Des.*, **15**, 811-817.

Puzicha, J., Hofmann, T., Buhmann, J. (1999) A Theory of Proximity Based Clustering: Structure Detection by Optimization, *Pattern Recognition*, **33**, 617-634.

Schneider, G., Neidhart, W., Giller, T., Schmid, G. (1999) "Scaffold hopping" by topologicical pharmacophore search: a contribution to virtual screening, *Angew. Chem. Int. Ed. Engl.*, **38**, 2894-2896.

Siani, M.A., Skillman, A.G., Carreras, C.W., Ashley, G., Kuntz, I.D., Santi, D.V. (2000) Development and screening of a polyketide virtual library for drug leads against a motilide pharmacophore *J. Mol. Graph Model.*, **18**, 497-511,539-540.

Stahl, M. (2000) Modifications of the scoring function in FlexX for virtual screening, *Perspect. Drug. Disc. Des.*, **20**, 83-98.

Willett, P., Barnard, J.M., Downs, G.M. (1998) Chemical similarity searching, *J. Chem. Inf. Comput. Sci.*, **38**, 983-996.

Wu, J.H., Batist, G., Zamir, L.O. (2001) Identification of a novel steroid derivative, NSC12983, as a paclitaxel-like tubulin assembly promoter by 3-D virtual screening. *Anticancer Drug Des.*, **16**, 129-133.

Wu Z., Leahy R., (1993) An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, *IEEE Trans. Pattern Analysis Machine Intelligence*, **15**, 1101-1113

Xue, L., Bajorath, J. (2000) Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Comb. Chem. High. Throughput Screen*, **3**, 363-372.

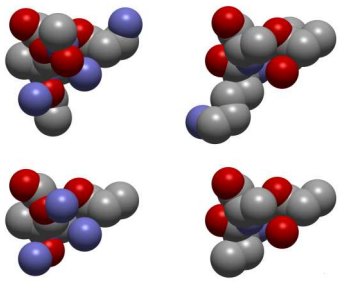

Figure 1: Two patches (top) and their superimposed atoms (below).

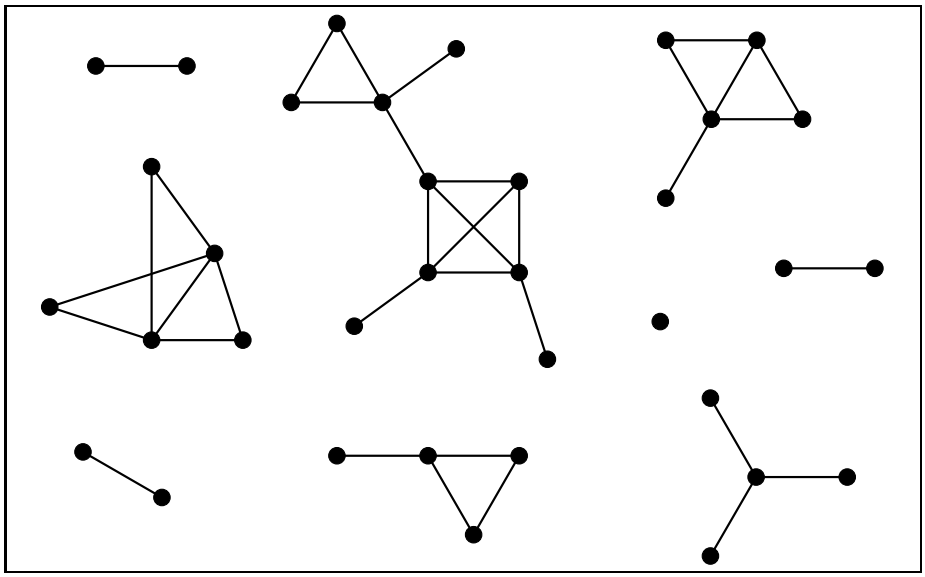

Figure 2: Part of the threshold graph ($t = 53$) for 1153 patches of size $\geq$100, including a component of size 10. All edges in the graph correspond to scores higher than 53.

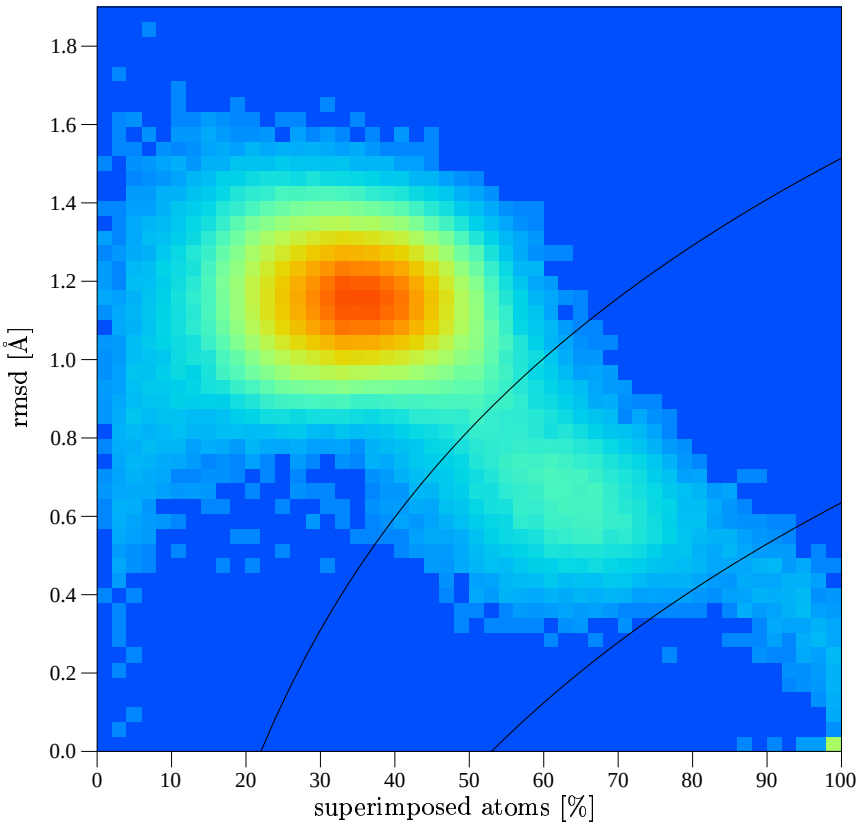

Figure 3: Distributional plot of the pairwise superpositions of 1153 patches of size $\geq$100. Two iso-score lines at values 23% and 53%. The colors of the points in the pane code the numbers of patch comparisons with the particular ratio of superimposed atoms (X-axis) and rmsd (Y-axis).

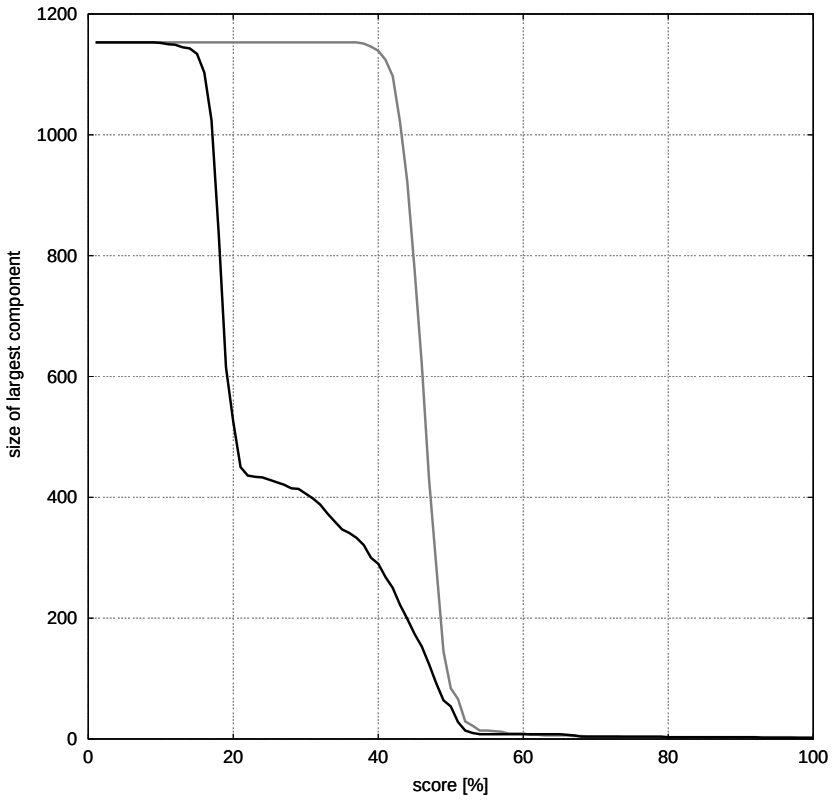

Figure 4: Evolution of the largest component in the original (black) and the randomly permuted (grey) similarity graph for the pairwise superposition of 1153 patches of size ≥ 100. Dependence of the number of patches contained in the largest graph component on the threshold score t.

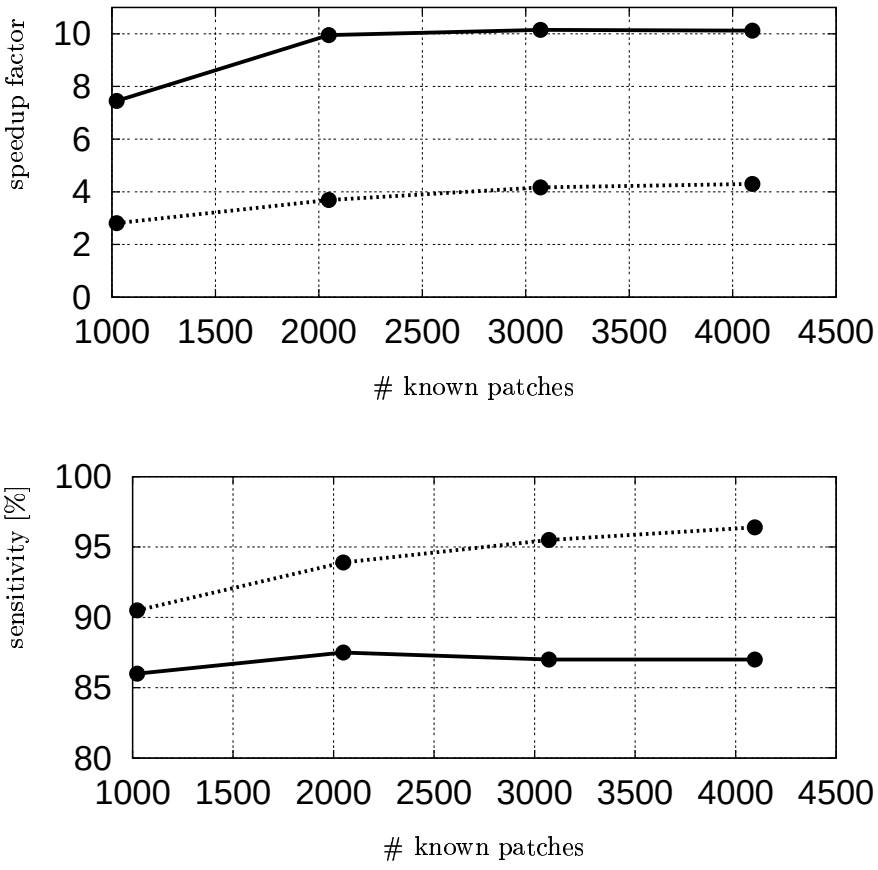

Figure 5: Comparison of the accelerated search with dominating sets for the identification of patches with a brute-force search where the query is subsequently compared to all patches of the database. Speedup and reduction of identified structural neighbors of the query (sensitivity) depending on the size of the database. Results for two parameter settings, one focusing on speed gain (solid), the other on a low proportion of missed patches (dashed).

Table 1: Average score value and variance for patch pairs exhibiting sequence identity, similarity and dissimilarity. Score value where the frequencies of comparisons of sequence similar patches and of comparisons of dissimilar patches are identical and interpolated values from formula (2). Three data sets differing in the number of atoms in the patches.

Size of the MSPs	40–49	50–59	60–69
Pairs with sequence dissimilarity			
Mean	15.2	13.9	12.8
variance	20.0	15.8	13.3
Pairs with sequence similarity			
Mean	17.3	16.0	15.4
variance	96.9	82.4	92.5
Pairs with sequence identity			
Mean	76.7	74.5	75.5
variance	158.3	165.6	150.3
Score value where frequencies are identical			
Statistically inferred	31.0	27.9	25.5
Interpolated	30.9	28.0	25.9