SEGA: Semi-Global Graph Alignment for Structure-based Protein Comparison

Marco Mernberger, Gerhard Klebe and Eyke Hüllermeier
Philipps-Universität Marburg

Draft of a paper to appear in
IEEE/ACM Transactions on Computational Biology and Bioinformatics

Abstract

Comparative analysis is a topic of utmost importance in structural bioinformatics. Recently, a structural counterpart to sequence alignment, called *multiple graph alignment*, was introduced as a tool for the comparison of protein structures in general and protein binding sites in particular. Using approximate graph matching techniques, this method enables the identification of approximately conserved patterns in functionally related structures. In this paper, we introduce a new method for computing graph alignments motivated by two problems of the original approach, a conceptual and a computational one. First, the existing approach is of limited usefulness for structures that only share common substructures. Second, the goal to find a globally optimal alignment leads to an optimization problem that is computationally intractable. To overcome these disadvantages, we propose a semi-global approach to graph alignment in analogy to semi-global sequence alignment that combines the advantages of local and global graph matching.

1 Introduction

Structural bioinformatics has gained increasing attention in the past ten years. With the steady improvement of structure prediction methods, the inference of protein function based on structure information becomes more and more important. Owing to the commonly accepted paradigm stating that similar protein function is mirrored by similar structure, the comparison of protein structures is a central task in this regard.

Many approaches to the functional analysis of proteins already exist, either on the sequence level [1, 2] or on the structural level [3, 4, 5, 6, 7, 8, 9]. However, these methods typically aim at the comparison of whole protein structures, some of them for example based on representations on a folding level, such as the well-known DALI approach [4].

While sequence-based methods have proven to be an invaluable tool for the detection of evolutionary similarities and the inference of protein function, prediction accuracy declines for proteins whose sequence identity falls under a certain percentage [10, 11]. Moreover, these methods can not be used to pinpoint the spatial location of functionally important residues, which is of great interest in pharmaceutical chemistry.

Fold-based methods, on the other hand, can retrieve evolutionary conservation on a more remote level. While often successful, similarity on the fold level does not always correspond to functional similarity, and cases are known where proteins with similar folds carry out different functions as well as cases where the same function is carried out by proteins with different fold geometries [12, 13, 14, 15].

Our approach uses a surface-based representation of protein binding sites [16]. Contrary to taking the whole protein structure and amino acid sequence into account, we focus on putative *protein binding sites*, clefts on the surface of proteins where certain biologically active ligands or cofactors can bind. We argue that these sites are the essential structural entities of a protein that convey

functional similarity, as they must be present to accommodate common substrates and thus are more relevant for functional analysis than the overall fold. ¿From an application point of view, the comparison of these binding pockets is especially relevant in the field of pharmaceutical chemistry in order to detect potential cross-reactivities.

For these reasons, we consider a model of the physico-chemical conditions within these binding sites to be more meaningful for a functional analysis of proteins. As we deliberately refrain from using whole sequence or fold information, our method is in principle capable of discovering functional similarities in distantly related proteins and even proteins that do not share a common fold, which distinguishes it from classical approaches based on whole protein comparison [4, 5, 6].

On a formal level, we model protein binding sites in terms of graph representations, which allows us to derive a structural alignment of binding pockets in the form of a *graph alignment* as introduced in [17].

In this paper, we propose a new method for computing graph alignments motivated by two problems of the original approach [17]. First, the existing approach is of limited usefulness for graphs that only share similar subgraphs as it tries to find an optimal alignment of the whole graphs. Second, finding a globally optimal graph alignment leads to an optimization problem which is computationally intractable. To overcome these problems, we propose a semi-global graph alignment approach, called SEGA (SEmi-global Graph Alignment).

2 Related Work

Most approaches to protein structure comparison can roughly be divided into fold-based, template-based and surface-based methods, each group focusing on a different aspect of similarity. Fold-based methods typically aim at a comparison of the overall structure of proteins in terms of fold geometry [18, 19]. Among these are the well-known DALI method [4], CE [5], MAMMOTH [6] or CATHEDRAL [20], which also employs graph theory.

Template-based methods on the other hand focus on identifying conserved spatial arrangements of functionally important residues, such as catalytic triads, which are often more conserved than the overall fold. Methods in this field employ geometric hashing [21, 22], dynamic programming [23], graph theory [24, 7] and other strategies [25, 26]. Most of these approaches scan user defined or automatically generated templates against a database to detect frequent patterns.

Surface-based methods usually seek to extract and compare protein binding sites, i.e., clefts on the surface of proteins where a certain ligand can be bound. These methods mainly differ in the way these surface pockets are modeled and extracted. SURFNET [27] identifies binding sites by fitting spheres of different sizes between protein residues, whereas FEATURE [28] uses statistical descriptions of the 3D environment. The pvSOAR approach [29] combines sequence information and spatial positioning of pocket-flanking residues, which was expanded later by the derivation of evolutionary substitution matrices [30]. SOIPPA represents functional sites by using a delaunay tesselation of C_α-atoms [31].

The CavBase approach, upon which our method is based, represents binding sites by a spatial arrangement of physicochemical properties. A similar representation is used by SiteEngine [32] and MultiBind [9], albeit with slightly different rules for the assignment of these properties. SiteEngine employs geometric hashing for the comparison of protein binding sites, whereas CavBase again utilizes graph theory.

Graph theory plays an important role in all three cases as it offers a convenient and versatile framework for the modeling of structures in a formal way, making them amenable to algorithmic methods. In bioinformatics, they have been used for the modeling of protein structure data (e.g. [3, 33, 7, 34, 17, 31, 35]), as well as biological networks [36, 37]. The existence of a variety of different approaches for graph comparison, specifically designed for different types of graphs, is hence hardly surprising.

In chemoinformatics, graph representations have been used for a long time. Classic approaches are

based on subgraph isomorphism [38], thus a lot of algorithms exist for the calculation of common subgraphs, utilizing clique detection [39, 40], back-tracking [41, 42] or optimization techniques [43, 44]. An excellent review is given by Raymond and Willet [44].

In the case of protein structure analysis, the approach of Artymiuk *et al.* [3] utilizes a subgraph isomorphism algorithm [45] to search for amino acid side chain patterns. The more recent approach of Xie and Bourne uses weighted subgraph isomorphism [31], while the SuMo approach of Jambon *et al.* employs heuristics to find correspondences [7].

In the field of kernel-based machine learning, a number of algorithms have been developed that measure structural similarity by using kernel functions as similarity measure. A kernel function defined on a set X is an $X \times X \to \mathbb{R}$ mapping satisfying certain formal properties, including symmetry and positive semi-definiteness. Such kernel functions can be viewed as similarity measures based on shared substructures of the graphs, e.g., random walks [46] or shortest paths [47]. In other words, these functions build upon local similarities.

Another prominent concept in graph analysis is the graph edit distance initially introduced by Sanfeliu and Fu [48]. Here, the similarity between two graphs is given by the minimal sequence of edit operations needed to transform one graph into the other. The set of allowed edit operations is predefined and typically includes insertions, deletions, and label/weight changes of nodes and edges. Originating from the field of pattern recognition, this concept has also been used for the comparison of protein binding sites [17, 35].

3 Global vs. local graph comparison

When analyzing experimentally determined structures, one has to deal with inaccuracies due to low resolution, measurement errors or simply experimental limitations. For example, a protein structure derived from crystallography might differ from the biological structure, especially with regard to side chain orientation of the amino acids, as they will interact differently with their physiological surroundings *in vivo*. Moreover, molecular structures are not static but flexible and thus subjected to conformational changes. Therefore, methods designed for the comparison of molecular structures have to allow for flexibility.

Additionally, functionally related structures may only share similar substructures, a notion that has been supported by recent studies [49]. This is also true for protein binding sites. In CavBase, protein binding sites are extracted as cavities on the surface using the LigSite algorithm [50]. A major problem of cleft-detection algorithms (e.g. based on alpha-shapes or grid scanning) is their inaccuracy in determining the borders of the cavity, leading to different binding site representations even for the same protein. Moreover, protein ligands usually occupy only a small portion of such a cavity, hence functionally related binding pockets do not necessarily share the same overall architecture.

The question is how to address these issues in the context of graph theory. As protein binding sites can be modeled as graphs, similar binding sites can be retrieved by graph comparison approaches, more precisely, graph alignment, a graph-based counterpart to sequence alignment. Essentially, a graph alignment establishes a one-to-one correspondence between single nodes of the graphs, which translates into correspondences between basic biological units (e.g. domains, residues, atoms).

In analogy to sequence alignment, graph comparison is done either on a *global* or a *local* scale. Global approaches seek to find a correspondence or matching between graphs as a whole, e.g., in the form of a global graph alignment. Local approaches reduce a comparison between complete graphs to the (multiple) comparison of (small) substructures. For comparing graphs derived from molecular structure data, both principles have different advantages and disadvantages.

Global approaches take the whole graph topology into account to derive mutual correspondences between components of the graphs, from which a correspondence between basic structural units of the modeled proteins or binding sites can be established. This usually comes at the price of a high computational complexity, as finding an optimal correspondence comes down to solving a hard (combinatorial) optimization problem that is typically approached by means of heuristic methods.

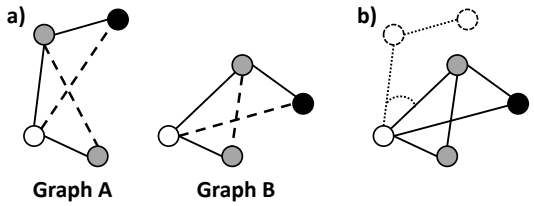

Figure 1: Two almost identical graphs (a), except for the variation of the angle at the white node (b), which influences the length of several edges (dashed).

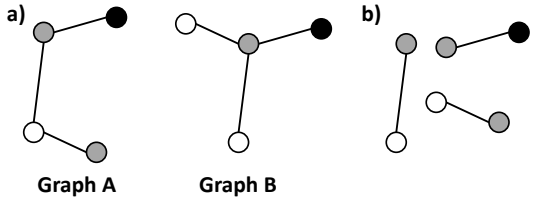

Figure 2: Two graphs that are quite different in terms of graph topology (a). Yet, their decomposition into subgraphs of size two yields the same set of components (b).

More importantly, when using graphs derived from molecular structures, a global graph comparison is more easily affected by topological differences due to conformational changes and inaccuracies of the modeled structures and thus might fail to detect similarities among related proteins in different conformations. While the issue of flexibility has been approached by several algorithms to protein comparison [51, 52], little has been done in the field of graph methods beyond the use of tolerance thresholds and label mismatches.

A simple illustration of this effect is shown in Fig. 1. The two "geometric" graphs depicted there are almost identical, except for the variation of the angle at the white node. Yet, this small modification already affects the whole graph topology, for example the length of the edges indicated by dashed lines.

Local methods, on the other hand, typically derive a similarity measure by comparing local graph features, e.g. subgraphs of a specific type. Main contributions to such similarity functions have recently been made in the field of kernel-based machine learning [53]. These kernel methods are less affected by topological variation and usually offer better runtime performance. However, they typically lack interpretability, as they do not retrieve an alignment from which functionally important features can be extracted.

Moreover, using such an approach inevitably carries the risk of producing a high similarity for graphs whose overall topology is different, due to the loss of information caused by decomposing the graph into substructures. In fact, decompositions of this type are typically not bijective, i.e., the complete graph cannot be recovered from the components. A simple illustration is shown in Fig. 2. The two graphs shown there are quite different in terms of their overall topology. Yet, they are decomposed into the same set of components (subgraphs of size two). Thus, a local method operating on these components will produce a high degree of similarity.

In order to avoid any terminological confusion, we like to point of out that, in this paper, the adjectives "local" and "global" always refer to the way in which graphs are compared, independently of whether a graph represents a whole molecular structure (sometimes called "global structure comparison") or only a part thereof ("local structure comparison").

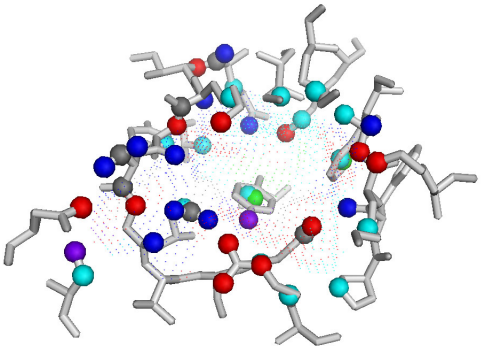

Figure 3: CAVBASE representation of a protein binding site. Amino acids are shown in light grey. Pseudocenters are depicted as spheres (donor = red, acceptor = blue, donor/acceptor = purple, pi = grey, aromatic = green, aliphatic = cyan). Dots represent a surface approximation.

4 The SEGA Method

Having recognized the limitations of purely global or purely local graph comparison, an obvious idea is to combine the advantages of both principles while avoiding their disadvantages. Such a *semi-global* approach will be proposed in this section.

4.1 Modeling Molecular Structure Data

In the case of protein binding sites, we build upon the CavBase representation, a database storing geometric information about protein binding sites [16] derived from the Protein Data Bank (PDB) [54]. As the consideration of each atom of each residue flanking a protein cavity would be computationally demanding, a reduction of the spatial information is necessary. This is typically done by using only C_α atom coordinates, which neglects the type of interaction an amino acid can participate in.

In CavBase, protein cavities are instead represented by a set of *pseudocenters*, spatial points representing the possible interactions that may occur in a certain area within the cavity, based on the bordering amino acids. A set of pseudocenters hence corresponds to an approximate description of a protein binding site in terms of its most important characteristics, namely its geometric structure and physico-chemical properties. An example is shown in Fig. 3. Note that this representation is independent of sequence order or fold information.

Currently, CavBase distinguishes between seven types of pseudocenters: hydrogen-bond donor, hydrogen-bond acceptor, mixed donor/acceptor, aromatic, aliphatic, metal groups and pi centers. In the respective graph model, the pseudocenters are represented as nodes, labeled with the corresponding pseudocenter type. Edges are undirected and weighted by the Euclidean distance between pseudocenters.

4.2 Graph Alignment

Similar to a sequence alignment, a graph alignment establishes a one-to-one correspondence between the nodes of different graphs. In the special case of two graphs, a corresponding pairwise graph alignment is a solution of a kind of bipartite graph matching problem.

As our graphs represent molecular structures that can differ in size, we have to allow for the possibility that nodes remain unmatched, which is realized by allowing gaps denoted by $\perp$. Formally, a pairwise graph alignment can be defined as follows.

Definition 1. *(*Pairwise graph alignment*) Let* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *be node-labeled and*

edge-weighted graphs (with a set of nodes V_i and a set of edges $E_i \subset V_i \times V_i$). Then,

$$A \subset (V_1 \cup \{\bot\}) \times (V_2 \cup \{\bot\})$$

is a pairwise graph alignment if the following properties hold: Each tuple in A contains at least one real node, i.e., $(\bot, \bot) \notin A$, and each node of each graph occurs exactly once in the alignment:

$$\forall\, v^{(1)} \in V_1 : \#\{(a_1, a_2) \in A \,|\, v^{(1)} = a_1\} = 1$$
$$\forall\, v^{(2)} \in V_2 : \#\{(a_1, a_2) \in A \,|\, v^{(2)} = a_2\} = 1$$

This definition of a pairwise graph alignment can be extended to the definition of a multiple graph alignment in a canonical way [17]. In this work, however, we focus on the construction of pairwise alignments to obtain a measure of similarity between two graphs.

As in the case of sequences, the quality of a graph alignment is assessed by an underlying scoring system that rewards "matches" and penalizes "mismatches" (both between nodes and edges) as well as gaps. The graph alignment problem then consists of finding an alignment with maximal score.

4.3 The SEGA Algorithm

SEGA essentially constructs a graph alignment for graph representations of putative protein binding sites. However, instead of optimizing a global alignment directly, it assembles it from local matches of subgraphs. The algorithm resorts to the complete graph topology only to resolve ambiguities that may arise.

Since we want to establish a correspondence between nodes of different graphs, we need a measure of similarity between these nodes. This can be obtained by simply comparing the node labels and their immediate surroundings, the neighborhood of the nodes, determined by the closest neighboring nodes and the edges connecting them.

For binding pockets, this corresponds to the comparison of pseudocenters and the spatial constellation of physicochemical properties in close proximity of these centers. Contrary to other approaches, we do not aim for the identification of completely matching substructures. Instead, we rather obtain an estimation of the geometric similarity of two neighborhoods by comparing triplets of pseudocenters that constitute such a neighborhood.

By deriving a mutual assignment of similar triangles and summing up the number of matches, we derive an intuitive measure of similarity that can be used to obtain a local distance matrix.

In a second step, this distance matrix can be used to construct an alignment between two graphs by deriving a mutual assignment of all nodes based on the distance matrix. This is done in an incremental way. We argue that, depending on the neighborhood size, the occurrence of a pair of nodes with identical or near-identical neighborhoods in two different binding sites will be rare, and therefore more meaningful than pairs with higher distances. Consequently, such occurrences should be considered first.

4.3.1 Neighborhood Distance Measure

Formally, the algorithm can be described as follows. A graph G is a tuple (V, E), where V is a set of nodes and $E \subseteq V \times V$ a set of edges. We denote by $\ell(v)$ the label of a node $v \in V$, and by $e(v, w)$ the weight of an edge $(v, w) \in E$. [1]

Given two input graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = n$ and $|V_2| = m$, we first establish the distance matrix

$$D = (d_{ij})_{1 \le i \le n, 1 \le j \le m} \tag{1}$$

[1] Since we are working with undirected edges, $(v, w) \in E$ implies $(w, v) \in E$, and a more correct syntax for an edge would be $\{v, w\}$; for convenience, we shall stick to the more commonly used tuple notation.

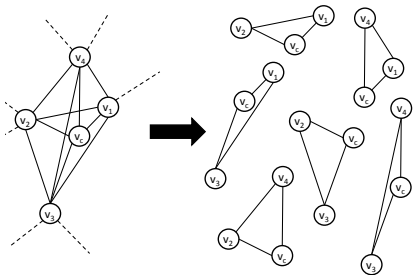

Figure 4: Decomposition of the neighborhood of node v_c with $n_{neigh} = 4$. The subgraph defined by the n_{neigh} nearest nodes is decomposed into triangles containing the center node v_c.

of dimensionality $n \times m$. The entry d_{ij} corresponds to the distance between the nodes $v_i^{(1)} \in V_1$ and $v_j^{(2)} \in V_2$ ($1 \leq i \leq n$, $1 \leq j \leq m$), and is inversely related to a corresponding similarity degree

$$s_{ij} = sim(v_i^{(1)}, v_j^{(2)}) \ . \tag{2}$$

The similarity (2) between two nodes $v_i^{(1)}$ and $v_j^{(2)}$ is defined in terms of the similarity of their respective neighborhoods. For a given (center) node $v_c \in V$, let $N(v_c, n_{neigh}) \subseteq V$ consist of the closest n_{neigh} nodes in V, i.e., those nodes having the smallest Euclidean distance from v_c. The neighborhood of v_c is then defined by the set $\mathbf{N}(v_c, n_{neigh})$ of all triangles $\{u, v_c, w\}$ with $u, w \in N(v_c, n_{neigh})$, $u \neq w$ (see Fig. 4).

Let

$$t^{(1)} = \{v_1^{(1)}, v_2^{(1)}, v_3^{(1)}\} \in \mathbf{N}(v_i^{(1)}, n_{neigh}),\ v_1^{(1)} = v_i^{(1)},$$
$$t^{(2)} = \{v_1^{(2)}, v_2^{(2)}, v_3^{(2)}\} \in \mathbf{N}(v_j^{(2)}, n_{neigh}),\ v_1^{(2)} = v_j^{(2)},$$

be two triangles from the neighborhoods of nodes $v_i^{(1)} \in V_1$ and $v_j^{(2)} \in V_2$, respectively. We say that the two triangles *match* if a mapping $\phi : t^{(1)} \to t^{(2)}$ exists with either

$$\phi(v_1^{(1)}) = v_1^{(2)}, \quad \phi(v_2^{(1)}) = v_2^{(2)}, \quad \phi(v_3^{(1)}) = v_3^{(2)},$$

or

$$\phi(v_1^{(1)}) = v_1^{(2)}, \quad \phi(v_2^{(1)}) = v_3^{(2)}, \quad \phi(v_3^{(1)}) = v_2^{(2)},$$

and for which the following conditions hold:

$$(i) \quad \ell(v_2^{(1)}) = \ell(\phi(v_2^{(1)})), \quad \ell(v_3^{(1)}) = \ell(\phi(v_3^{(1)})),$$

$$(ii) \quad \max \left\{ \begin{array}{l} |e(v_1^{(1)}, v_2^{(1)}) - e(\phi(v_1^{(1)}), \phi(v_2^{(1)}))|, \\ |e(v_2^{(1)}, v_3^{(1)}) - e(\phi(v_2^{(1)}), \phi(v_3^{(1)}))|, \\ |e(v_3^{(1)}, v_1^{(1)}) - e(\phi(v_3^{(1)}), \phi(v_1^{(1)}))| \end{array} \right\} \leq \epsilon$$

The parameter $\epsilon \geq 0$ is a tolerance threshold that determines the allowed deviation of edge lengths. Roughly speaking, two triangles form a match if there is a superposition that preserves node labels and edge lengths. The only exception concerns the two center nodes $v_i^{(1)}$ and $v_j^{(2)}$: These two nodes are necessarily assigned to each other, but their labels can be different. As will become clear later on, this enables the construction of *approximate* graph alignments that may contain mismatches (mutually assigned nodes with different label). This is important, since protein structure data is uncertain and noisy, e.g., due to measurement errors. Moreover, approximate matching techniques account for biological variation caused by mutations that alter the amino acid sequence of a molecule and, therefore, have an influence on the structure.

The similarity (2) between two nodes $v_i^{(1)} \in V_1$ and $v_j^{(2)} \in V_2$ is now defined by the squared maximal number of matching triangles from $\mathbf{N}(v_i^{(1)}, n_{neigh})$ and $\mathbf{N}(v_j^{(2)}, n_{neigh})$ that can be assigned in a mutually exclusive way. That is, each triangle from $\mathbf{N}(v_i^{(1)}, n_{neigh})$ can only be matched with at most one triangle from $\mathbf{N}(v_j^{(2)}, n_{neigh})$, and vice versa. Note that

$$0 \leq s_{ij} \leq s_{max} = \frac{n_{neigh}(n_{neigh}-1)}{2} \tag{3}$$

To determine s_{ij}, we solve an optimal assignment problem. More specifically, we apply the well-known Hungarian algorithm [55] to the $s_{max} \times s_{max}$ matrix whose entry at position (k, l) is 0 if the k-th triangle in $\mathbf{N}(v_i^{(1)}, n_{neigh})$ can be matched with the l-th triangle in $\mathbf{N}(v_j^{(2)}, n_{neigh})$; otherwise, the entry is 1. The Hungarian algorithm is a combinatorial optimization algorithm that solves the assignment problem based on a cost matrix as input. Since the Hungarian algorithm, whose time complexity is $O((s_{max})^3)$, computes the cost of a cost-minimal assignment, it returns $s_{max} - sim(v_i^{(1)}, v_j^{(2)})$, i.e., the number of triangles that could not be matched. This value defines the distance between $v_i^{(1)}$ and $v_j^{(2)}$, i.e.,

$$d_{ij} = s_{max} - sim(v_i^{(1)}, v_j^{(2)}) \ . \tag{4}$$

4.3.2 Deriving a Global Alignment

The result of the above computation is an $n \times m$ distance matrix (1). This matrix is used as an input for the second step of our algorithm, which seeks to find an optimal mutual assignment of nodes from V_1 and V_2, respectively. Since d_{ij} can be considered as the cost of assigning nodes $v_i^{(1)}$ and $v_j^{(2)}$ to each other, this problem can again be formulated as an optimal assignment problem, namely as the problem to find the assignment with the minimal sum of costs. In principle, the Hungarian algorithm could again be used to solve this problem.

However, a solution thus obtained, even if being optimal in the sense of minimizing the total cost of node assignments, will usually not provide a reasonable alignment with respect to the overall graph topology. This is because the algorithm does not take the spatial relationships between the nodes into consideration. Moreover, it will resolve ambiguities in an arbitrary way. In fact, due to the nature of the underlying distance matrix, whose entries are integers between 0 and s_{max}, it is likely that a cost-minimal solution is not unique. The Hungarian algorithm will simply pick one among the optimal solutions, which is not necessarily in agreement with the overall graph structures.

To avoid this problem, we construct an alignment in an incremental way and resort to global information from the graph topology to resolve such ambiguities. More specifically, we start by constructing a seed solution in the form of a partial assignment of nodes. To this end, we only look at the nodes $v_i^{(1)} \in V_1$ and $v_j^{(2)} \in V_2$ having a distance of 0 and, hence, being highly "affine". Then, we realize those assignments that are unambiguous and hence highly reliable. With

$$f_c(v_i^{(1)}) = \{\, v_j^{(2)} \in V_2 \,|\, d_{ij} \leq c \,\} \ ,$$
$$g_c(v_j^{(2)}) = \{\, v_i^{(1)} \in V_1 \,|\, d_{ij} \leq c \,\} \ ,$$

(where e.g. $f_c(v_i^{(1)})$ denotes the set of vertices in G_2 whose distance to $v_i^{(1)}$ is not greater than c) we assign pairs $v_i^{(1)}$ and $v_j^{(2)}$ satisfying $f_0(v_i^{(1)}) = \{v_j^{(2)}\}$ and $g_0(v_j^{(2)}) = \{v_i^{(1)}\}$, while nodes $v_i^{(1)}$ with $|f_0(v_i^{(1)})| > 1$ (and $v_j^{(2)}$ with $|g_0(v_j^{(2)})| > 1$) are not yet assigned, as for these several conflicting assignments are possible.

The seed solution thus obtained must satisfy the constraint that the set of mapped points for each graph contains a basis of $\mathbb{R}^3$ in order to determine the relative position of a new node in three-dimensional space in an unambiguous way. If this condition is not met, we collect a sufficient number

of candidate pairs by relaxing the condition on the distance, i.e., we allow a maximal distance $c > 0$. Let $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$ denote the nodes occurring in these candidates. We construct all possible candidate assignments

$$\Big((s_1^{(1)}, s_1^{(2)}),\, (s_2^{(1)}, s_2^{(2)}),\, (s_3^{(1)}, s_3^{(2)}),\, (s_4^{(1)}, s_4^{(2)}) \Big) \subseteq S_1 \times S_2$$

of size four that represent a unique three-dimensional geometry and are unambiguous in the sense that $s_i^{(2)} \in f_c(s_i^{(1)})$ and $s_j^{(2)} \notin f_c(s_i^{(1)})$ as well as $s_i^{(1)} \in g_c(s_i^{(2)})$ and $s_j^{(1)} \notin g_c(s_i^{(2)})$ for all $1 \leq i \neq j \leq 4$. As a seed solution, we then select the candidate minimizing the spatial deviation

$$\sum_{1<i<j<4} \Big| e(s_i^{(1)}, s_j^{(1)}) - e(s_i^{(2)}, s_j^{(2)}) \Big| \;\; ,$$

in order to match the candidates that are most similar in terms of geometry.

Now, suppose a current seed in the form of a partial alignment to be given. We may still have the problem that some nodes could not be assigned unambiguously. In order to solve these ambiguities in a reasonable way, we can again formulate an optimal assignment problem, this time augmented by drawing upon global information. In the k-th iteration, we assign nodes having a distance of at most c_k, where c_k is the k-th smallest cost value in the matrix D. More specifically, let $W_1 \subset V_1$ ($W_2 \subset V_2$) denote the set of nodes from V_1 (V_2) that have already been assigned in a previous iteration. Moreover, let

$$U_1^k = \{ v_i^{(1)} \in V_1 \,|\, f_{c_k}(v_i^{(1)}) \neq \emptyset \} \setminus W_1 \;\; ,$$
$$U_2^k = \{ v_j^{(2)} \in V_2 \,|\, g_{c_k}(v_j^{(2)}) \neq \emptyset \} \setminus W_2 \;\; .$$

We then derive a (partial) assignment of nodes in U_1^k and U_2^k by applying the Hungarian algorithm to a cost matrix defined as follows. The matrix contains an entry for each pair of nodes $v_i^{(1)} \in U_1^k$ and $v_j^{(2)} \in U_2^k$. If $v_j^{(2)} \notin f_{c_k}(v_i^{(1)})$, the corresponding cost value is set to a sufficiently high constant C (indicating that these two nodes should not be assigned). Otherwise, the cost value is determined by resorting to information from the (global) graph structure, namely by comparing the position of $v_i^{(1)}$ relative to the current seed nodes W_1 with the position of $v_j^{(2)}$ relative to W_2. More precisely, the cost is defined by

$$\sum_{q=1,2\ldots|W_1|} \Big| \, |v_i^{(1)} - w_q^{(1)}| - |v_j^{(2)} - w_q^{(2)}| \, \Big| \;\; ,$$

where $w_q^{(1)}$ and $w_q^{(2)}$ denote, respectively, the q-th node in W_1 and W_2 (which are mutually assigned), and $|v_i^{(1)} - w_q^{(1)}|$ is the Euclidean distance between $v_i^{(1)}$ and $w_q^{(1)}$.

Applying the Hungarian algorithm yields a cost-minimal assignment. If $v_i^{(1)}$ and $v_j^{(2)}$ participate in this assignment, i.e., have been assigned to each other, we add $v_i^{(1)}$ to W_1 and $v_j^{(2)}$ to W_2 if $v_j^{(2)} \in f_{c_k}(v_i^{(1)})$, i.e., if the corresponding cost value is smaller than C.

This procedure iterates until all nodes of one graph are assigned, or until a predefined upper cost value c_{max} has been reached (remaining nodes are not assigned). Setting such an upper limit (below the maximal score in D) allows to compute a partial alignment. This is often more reasonable than enforcing an alignment of the complete structures, e.g. if parts of the structures do obviously not match.

4.3.3 Defining a Distance Measure

Our algorithm produces a global graph alignment, deriving an assignment between all constituents of a binding site. As binding sites might share several common subparts that are not necessarily

arranged in the same manner, quality measures based on root mean squared deviation (RMSD) that are usually applied in e.g. template-based approaches [26, 23], are not suitable in our case, unless the binding sites are globally similar.

To define a more general, size-independent measures of the quality of the constructed alignment A, which can be interpreted as a distance between the two structures G_1 and G_2, we proceed from a measure that can be seen as a degree of inclusion of G_1 in G_2:

$$\delta(G_1, G_2) = \frac{\sum_{(v_i^{(1)}, v_j^{(2)}) \in A} d_{ij} + c_p \cdot (|A| - |G_1|)}{|G_1|} \ . \tag{5}$$

The constant c_p is a penalty that accounts for unmatched nodes which can simply be set to the highest obtainable distance in the case where no triangles can be matched. A degree of inclusion of G_2 in G_1 is defined analogously.

Based on (5), we define two measures of distance between G_1 and G_2, a "conjunctive" and a "disjunctive" one:

$$\Delta_{max}(G_1, G_2) \quad = \max\{\delta(G_1, G_2), \delta(G_2, G_1)\} \tag{6}$$
$$\Delta_{min}(G_1, G_2) \quad = \min\{\delta(G_1, G_2), \delta(G_2, G_1)\} \tag{7}$$

The measure (6) can be seen as a relaxed equality and is based on the expression of set equality ($A = B$) in terms of two-sided inclusion ($A \subset B$ and $B \subset A$). Thus, it requires that, to be similar, G_1 and G_2 must be approximately equal in the sense of a mutual inclusion: G_1 is (approximately) included in G_2 and likewise G_2 in G_1. As opposed to this conjunctive combination of the two degrees of inclusion, the disjunctive combination (7) only requires a one-sided inclusion: either G_1 is included in G_2 or G_2 in G_1. Obviously,

$$\Delta_{min}(G_1, G_2) \ \leq \ \Delta_{max}(G_1, G_2) \ .$$

The question which of these two measures, the conjunctive or the disjunctive one, yields more suitable similarity degrees cannot be answered in general and instead depends on the application at hand, in particular on the purpose for which the similarity is used (e.g., function prediction) and the way in which protein binding sites are extracted and modeled (e.g., whether or not the model may include parts of the protein not belonging to the binding site itself). Therefore, we define our ultimate distance measure as a (linear) combination of the two above measures (6) and (7):

$$\begin{aligned} \Delta(G_1, G_2) \ &= \alpha \cdot \Delta_{max}(G_1, G_2) + \\ &\quad + (1 - \alpha) \cdot \Delta_{min}(G_1, G_2) \ . \end{aligned} \tag{8}$$

As a side note, we remark that, formally, (8) is a special case of a so-called OWA (ordered weighted average) aggregation of the two degrees of inclusion, G_1 and G_2, and the parameter $\alpha \in [0, 1]$ controls the trade-off between the two extreme aggregation modes: The closer α is to 1, the closer the aggregation is to the minimum, i.e., the more demanding it becomes.[2]

In principle, choosing a high value of α favors the detection of largely similar binding sites, for example belonging to the same protein family or fold. A low value of α would be beneficial for the detection of more remote similarities. (8) is inversely related to a similarity score.

4.3.4 Statistical significance

Any score used to compare proteins must be judged against the likelihood that a given score could arise by chance. To obtain a measure of significance for the induced similarity scores, we chose an empirical approach and calculated ten thousand pairwise alignments of randomly drawn cavities from the CavBase.

[2] The value α corresponds to the "degree of andness" of the aggregation (8), i.e., the degree to which this aggregation behaves like a conjunctive combination [56]; likewise, $1 - \alpha$ corresponds to the "degree of orness".

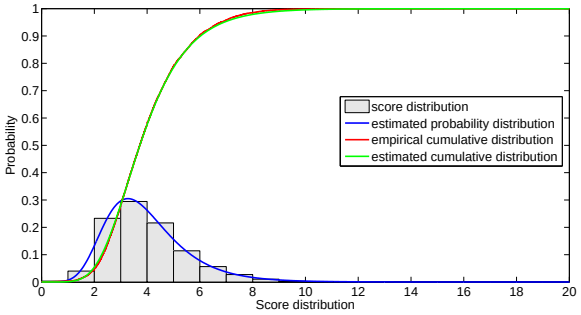

Figure 5: Estimation of a probability density function based on ten thousand randomly drawn comparisons for $\alpha = 0$.

One possinility to assess significance for database searches is to use extreme value distributions (EVD) [26]. In terms of database searches, finding the most similar item to a query usually involves maximizing over all similarity scores. Interpreting these scores as random variables, the maximum score can be viewed as an extreme value based on score distribution. The shape of the obtained score distribution was indeed characteristic of an EVD. Since the EVD type was not know in advance, we used the scores to estimate a generalized EVD by maximum likelihood estimation, yielding type II (Frechét) EVDs. We used the corresponding cumulative distribution to calculate p-values for our alignment scores. As can be seen, the fit is extremely accurate.

Since the score we obtain by using (8) does obviously depend on the parameter α, we fitted EVDs individually for different values of α. Fig. 5 shows a histogram of the scores obtained, together with the estimated type II EVD for $\alpha = 0$. To convey an idea of the goodness of fit, we also plotted the empirical and the estimated cumulative density function.

5 Experimental Results

First of all, we were interested in an assessment of the robustness of our approach in the presence of noise and structural variation. This point is addressed in Section 5.3, where we compare the performance of SEGA to several other graph-based approaches when confronted with different degrees of structural variation.

In Section 5.4, we evaluate our approach in a retrieval experiment on a real-world benchmark dataset. To this end, we used a dataset containing structurally diverse proteins that have also been used for the evaluation of SiteEngine, which operates on a similar concept of protein binding pockets [32]. Thus, we were able to compare our results directly to a different surface-based approach to protein structure comparison.

In Section 5.5, we investigate the performance of SEGA when confronted with the complete CavBase.

Finally, Section 5.6 presents a study to assess the capability of SEGA to classify protein binding sites with respect to the ligands they bind, especially for proteins with only remote structural similarity, as this is a main application for our approach.

5.1 Data

As a real-world benchmark set, we used a set of representative proteins constructed for the evaluation of SiteEngine, which operates on a similar concept of binding pockets. This dataset contains several structurally different classes of proteins, among them proteins binding fatty acids, serine proteases, adenine-containing ligands and others (for a detailed description see [32]).

In our classification study we used two additional datasets. The first one was initially constructed in a previous study to assess the classification accuracy of global graph matching methods [35]. It

contained 143 adenosine-5'-triphosphate (ATP) and 214 nicotine amide dinucleotide (NADH) binding pockets whose ligands are bound in similar conformation (though not necessarily orientation; for a detailed description see [35]). The idea was that binding sites that bind the ligand in similar conformation are more likely to share a global architecture.

As our approach was built with the intention to recover more distant similarities, we constructed a second classification dataset of protein binding sites, again for the ligands ATP and NADH, for which each protein belongs to a different SCOP fold [18]. Proteins were sampled from CavBase and, in case of multiple proteins belonging to the same fold, one was randomly selected, yielding a total of 50 NADH-binding pockets and 77 ATP-binding pockets.

5.2 Methods

In the following experiments, we compared our SEGA algorithm to several existing methods in terms of performance:

- Bron-Kerbosch algorithm (BK) [39], a clique-detection algorithm commonly used in protein structure comparison [34, 57] and currently still the standard procedure in CavBase.
- GAVEO [35], an evolutionary algorithm for the calculation of graph alignments.
- GH, a greedy heuristic based on clique detection [17].
- SH, a variant of SEGA, where the Hungarian algorithm [55] was used to construct a global alignment from the distance matrix D.

All these algorithms were parametrized with a tolerance threshold $\epsilon = 0.2$Å for the comparison of edge lengths. The parameter n_{neigh} was set to $n_{neigh} = 10$ for SEGA and the Hungarian variant, as these values showed the best performance in preliminary studies investigating the influence of n_{neigh} on the alignment quality (data not shown). For the other algorithms, standard parametrization was used as specified in the original publications.

For the classification and retrieval experiments, we further included two kernel methods that have been proposed for the comparison of protein structures, the random walk kernel (RW) [46] and the shortest path kernel (SP) [47].

5.3 Robustness Toward Structural Variation

In a preliminary study, we wanted to compare the performance of SEGA compared to other graph-based approaches in the presence of noise and mutations (since the type of pseudocenters itself can change due to mutations). To do this in a systematic way, we needed protein binding sites for which we could vary the degree of structural variation and mutation while still knowing the correct alignment in advance. Since this is hardly possible for real-world datasets, we opted for a semi-synthetic approach.

We randomly selected 100 protein binding pockets from CavBase that were co-crystallized with a ligand. These pockets have the benefit of being relatively large which is useful in order to generate non-trivial deformations.

For each binding pocket we generated 10 structurally diverse synthetic pockets by subjecting the original pocket to structural noise, mutation or both. Structural noise was introduced by translating each center by a randomly directed vector with a normally distributed length controlled by a deviation parameter p_{dev} (standard deviation). To introduce mutations we subjected all pseudocenters to label mutation controlled by a mutation parameter p_{mut} (mutation probability).

We then calculated pairwise alignments of the binding pockets and determined the percentage of correctly matched pseudocenters of the core pocket for different values of p_{dev} and p_{mut}. Fig. 6 shows the mean percentage of correctly mapped centers for different algorithms[3] under structural

[3]The kernel methods were excluded, as these methods can only calculate a distance measure between graphs.

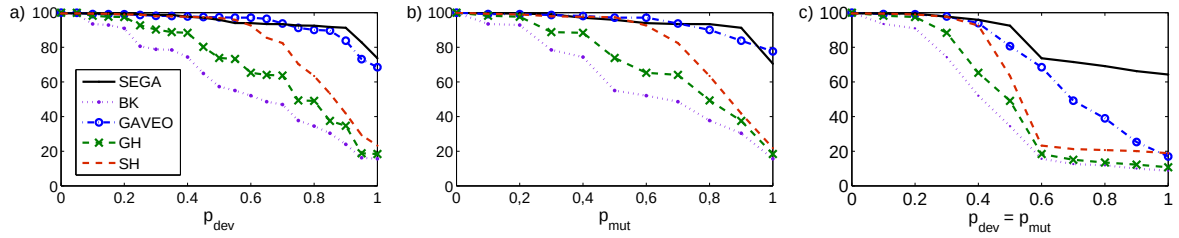

Figure 6: Average percentage of correctly mapped pseudocenters (y-axis) for different levels of: a) distortion, b) mutation c) both.

Table 1: Mean μ and standard deviation σ for the runtime performance of the different algorithms based on 1000 comparisons in seconds $\times 10^2$.

	BK	GAVEO	GH	SH	SEGA
μ	0.0134	5.8451	0.2811	0.0423	0.0185
σ	0.0497	21.9902	0.1288	0.0553	0.0227

noise, mutation and a combination of both. Runtime requirements of the approaches are summarized in Table 1.

In the synthetic experiment, SEGA shows the most stable performance with respect to the other graph-based approaches, rivaled only by GAVEO, although the runtime requirements of GAVEO are much higher. When combining both sources of variation, the evolutionary optimization fails to attain similar strong results.

In terms of runtime requirements, BK and SEGA show the best runtime behavior. GAVEO has by far the highest runtime, which is not surprising, since evolutionary optimization is known to be expensive. However, the clique-based BK and GH have the drawback of a space complexity of $\mathcal{O}(n^4)$ (n = number of nodes), which is problematic for calculating alignments for large graphs on current machines. The other approaches only have a space complexity of $\mathcal{O}(n^2)$.

5.4 Analysis on Benchmark Data

The goal of this study was to test our approach on a real benchmark dataset to assess whether SEGA is capable of recovering similar binding sites. As a benchmark set, we used a dataset that was compiled at the Nussinov-Wolfson group for the evaluation of SiteEngine [32]. We selected the same query proteins that were used in the SiteEngine retrieval experiments and provided with ranked results.

Initially, we used a fatty acid binding protein, the adipocyte lipid-binding protein from *M. musculus* (1lib), as query structure and ranked the results according to (8), shown in Table 2. To give an idea of the similarity of the pockets, we also included the sequence identity of the cavity flanking residues and the RMSD value based on the calculated alignments.

Shulman-Peleg *et al.* reported five protein binding sites to be structurally very similar to the query protein [32]. In fact, two cavities are derived from different crystal structures of the same protein. These six proteins are the highest in the ranking, followed by other fatty acid binding proteins. This differs from the SiteEngine results, which retrieved proteins from other classes among the top ten (e.g ketosteroid isomerase, HIV protease and others). These were not present in our top ranking results and achieved a non-significant score.

SEGA was able to retrieve 13 of the 15 fatty acid-binding proteins present in the dataset exclusively on the top ranks with significant p-values, although some of them exhibit a different binding pattern than the query protein (1kqw, 1mdc, 1cbs) [32]. This is in contrast to the SiteEngine experiment, in which some of these structures did not receive a high rank. As SiteEngine uses a more rigid

Table 2: Ranking of comparisons between the fatty acid binding protein 1lib and the SiteEngine benchmark set ($\alpha = 1$).

Rank	PDB	Function	Score	P-value	Seq.id.	Cavity Seq.id.	RMSD ($\mathring{A}^2$)
1	1lib	Adipocyte lipid-binding protein	90.00	$< 10^{-10}$	1.00	1.00	0.00
2	1lie	Adipocyte lipid-binding protein	41.04	$< 10^{-10}$	1.00	0.88	3.12
3	1lid	Adipocyte lipid-binding protein	40.16	$< 10^{-10}$	1.00	0.97	2.74
4	1hms	Heart muscle fatty acid-binding protein	20.72	$3.96 \cdot 10^{-10}$	0.64	0.62	5.22
5	1b56	Epidermal fatty acid-binding protein	10.14	$2.79 \cdot 10^{-5}$	0.53	0.54	5.78
6	1pmp	Myelin P2	10.09	$2.94 \cdot 10^{-5}$	0.66	0.76	5.18
7	1ftp	Locus muscle fatty acid-binding protein	9.13	$8.88 \cdot 10^{-5}$	0.44	0.43	8.31
8	1opb	Cellular retinol binding protein II	8.61	$21.61 \cdot 10^{-4}$	0.38	0.35	8.22
9	1cbs	Cellular retinoic acid binding protein II	8.19	$2.61 \cdot 10^{-4}$	0.37	0.39	8.12
10	2cbr	Cellular retinoic acid binding protein II	7.30	$7.17 \cdot 10^{-4}$	0.38	0.39	8.01
11	1opa	Cellular retinol binding protein II	7.00	$1.06 \cdot 10^{-3}$	0.38	0.35	7.92
12	1kqw	Cellular retinol binding protein	6.60	$1.70 \cdot 10^{-3}$	0.38	0.35	8.48
13	1mdc	Insect fatty acid binding protein	5.97	$3.65 \cdot 10^{-3}$	0.35	0.40	10.71
14	1fnj	Chorismate Mutase	5.95	$3.71 \cdot 10^{-3}$	0.11	0.40	8.42

RMSD-based criterion, it is likely that proteins with a different binding pattern are not considered similar while the more tolerant measure of SEGA still manages to recover similarities among these proteins.

We also retrieved a chorismate mutase (1fnj) with a significantly high score. Upon visual inspection, we found that both binding pockets contain sub-pockets showing a similar spatial positioning of pseudocenters, and hence similar neighborhoods which lead to a high score. We can only interpret this as a false positive result.

Additionally, we performed retrieval experiments for the remaining queries 1atp, 1mjh and 1lhu and repeated these experiments with the competitor algorithms. The results are summarized by 11-point precision-recall curves (Fig. 7).

In all cases except one, SEGA outperformed the other approaches, with the exception of 1lhu. In this case, none of the algorithms yielded a good result, indicating that there might be no recognizable structural resemblance between the query structure and other estradiol-binding pockets in the dataset. This difference from the results of [32] might be due to the differences in pocket extraction and representation between CavBase and SiteEngine.

To assess whether it is possible to retrieve a meaningful ranking for the class of estradiol-binding pockets at all, we therefore included two more query structures (1ere, 1qkt) from this class in our experiments. For these queries we were able to retrieve similar pockets from the dataset and again SEGA performed best.

5.5 Retrieval of Similar Binding Sites from CavBase

A higher level of approximation always carries the risk of detecting a large percentage of false positives. In this experiment we wanted to address the question whether SEGA can retrieve meaningful results on the top ranks even when confronted with a whole database of structures. To this end, we queried the CavBase database with two different query structures that have also been used in the evaluation of the pvSOAR algorithm [29].

As these structures where already used in a retrieval experiment spanning the whole Protein Data Bank, we knew in advance which protein classes were likely to be retrieved, which was useful to determine false positives in the rankings. As SEGA was designed to discover also remotely similar sites, judging retrieved results is is not trivial, especially for the SiteEngine queries. Results are shown in Tables 3 and 4.

In the first experiment, we used the main pocket of a member of the acetylcholinesterase family (2ack) as a query structure (volume 1192.4 $\mathring{A}^3$). The active center contains a catalytic triad consisting of a serine, histidine and glutamic acid (S200, H440, E327).

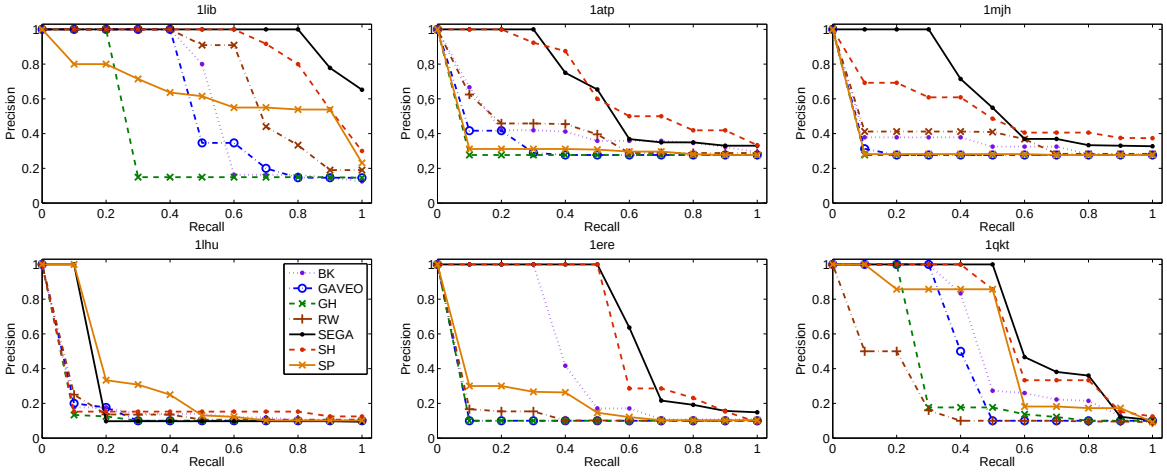

Figure 7: 11-pt precision-recall curves for the retrieval of different proteins on the SiteEngine dataset.

Table 3: Results of querying the CavBase with an acetylcholinesterase structure (2ack)($\alpha = 1$). Omitted ranks contained exclusively acetylcholinesterases.

Rank	PDB	Protein	Score	P-value	Seq.id.	Cavity Seq.id.	RMSD ($\mathring{A}^2$)
1	2ack	Acetylcholinesterase	90	$< 10^{-16}$	1.00	1.00	0.00
2	1ax9	Acetylcholinesterase	36.19	3.3×10^{-16}	1.00	0.97	1.83
3	2wg1	Acetylcholinesterase	31.79	1.4×10^{-14}	1.00	0.97	2.80
...	...	...	...	...	...	...	...
113	1w76	Acetylcholinesterase	7.57	5.4×10^{-4}	1.00	0.54	9.60
114	1c2b	Acetylcholinesterase	7.18	8.6×10^{-4}	0.59	0.73	6.35
115	2wil	Butyrylcholinesteras	6.07	3.2×10^{-3}	0.52	0.44	8.80
116	1qo9	Acetylcholinesterase	5.24	8.8×10^{-3}	0.40	0.47	7.76
117	1fss	Acetylcholinesterase	5,04	1.1×10^{-2}	1.00	0.62	10.29
118	1p0q	Butyrylcholinesterase	4,70	1.7×10^{-2}	0.53	0.29	9.62
119	1jxj	α-Amylase	4,68	1.7×10^{-2}	0.14	0.06	10.21
120	1c7i	PNB esterase	4,57	1.9×10^{-2}	0,33	0.44	11.26
121	1qe3	PNB esterase	4,53	2.1×10^{-2}	0,34	0.29	9.81

Table 4: Results of querying the CavBase with an alpha amylase structure (1bag)($\alpha = 1$).

Rank	PDB	Protein	Score	P-value	Seq.id.	Seq.id. Cavity	RMSD ($\mathring{A}^2$)
1	1bag	α-amylase	90.00	$< 10^{-10}$	1.00	1.00	0.00
2	1ua7	α-amylase	15.16	2.3×10^{-7}	1.00	0.88	5.24
3	3dc0	α-amylase	14.49	5.2×10^{-7}	0.94	0.91	5.95
4	1g94	α-amylase	11.51	1.8×10^{-5}	0.26	0.30	7.46
5	1aqh	α-amylase	10.04	1.0×10^{-4}	0.24	0.33	7.80
6	1u33	α-amylase, pancreatic	9.72	1.5×10^{-4}	0.28	0.46	9.07
7	1l0p	α-amylase	9.66	1.6×10^{-4}	0.26	0.35	8.45
8	1vah	α-amylase, pancreatic	9.64	1.7×10^{-4}	0.26	0.44	5.65
9	1b2y	α-amylase	9.49	2.0×10^{-4}	0.28	0.37	8.43
10	1xd0	α-amylase	9.44	2.1×10^{-4}	0.28	0.28	9.26
11	1kck	Cyclodextrin glucanotransferase	9.28	2.6×10^{-4}	0.25	0.41	8.59
...	...	...	...	...	...	...	...
78	1qho	α-amylase	5.81	1.7×10^{-2}	0.24	0.27	8.34
79	2d3n	Glucan 1,4-alpha-maltohexaosidase	5.76	1.8×10^{-2}	0.26	0.35	7.44
80	1ud5	α-amylase	5.72	1.9×10^{-2}	0.26	0.50	7.17
81	1w9x	α amylase	5.69	2.0×10^{-2}	0.28	0.36	8.23
82	1hvx	α-amylase	5.56	2.3×10^{-2}	0.26	0.62	7.28
83	5cgt	Cyclodextrin glucanotransferase	5.55	2.4×10^{-2}	0.25	0.28	7.89
84	3dhu	α-amylase	5.54	2.4×10^{-2}	0.23	0.17	9.69
85	3bc9	α amylase, catalytic region	5.54	2.4×10^{-2}	0.27	0.42	7.89
86	1e40	α-amylase	5.50	2.5×10^{-2}	0.26	0,36	6.82
87	1cgv	Cyclodextrin glucanotransferase	5.47	2.6×10^{-2}	0.25	0.32	9.73
88	1cgt	Cyclodextrin glucanotransferase	5.40	2.8×10^{-2}	0.25	0.36	9.11

Binkowski *et al.* reported that all significant results obtained by searching the CASTp database [58] with 2ack as query structure belonged to the same family, as the binding site geometry represents a unique pattern typical of this protein family. Our results are in good agreement with this, although we did not use additional sequence information.

All top ranking results with a p-value < 0.05 belonged almost exclusively to the acetylcholinesterase family (including those structures retrieved by Binkowski *et al.*), demonstrating that our algorithm is capable of retrieving similar binding sites from CavBase correctly with high specificity. Additionally, we retrieved more distantly related proteins, namely butyrylcholinesterases and para-nitrobenzyl esterases, which both belong to the acetylcholinesterase-like proteins catalyzing similar reactions on different substrates.

Of the catalytic triad, only two residues, S200 and H440, are actually located within the pocket of the structure. The glutamic acid is usually not present in the main pockets of members belonging to this family, as it is too buried. Nevertheless, as an indicator of the accuracy of the calculated alignments, we checked whether the remaining two residues, respectively their pseudocenters are matched correctly, which was the case in all comparisons with one exception, where none of the catalytic residues were present in the pocket.

We subsequently tested SEGA with a second query, the enzymatic pocket of a bacterial α-amylase (1bag). The largest cleft of α-amylase from *B. subtilis* (volume 1273.4 $\mathring{A}^3$) forms the substrate binding site for polysacharides and is located on a TIM-barrel domain. This structure was also used by Binkowski *et al.*, who recovered similar proteins from different families by a database search.

Significant results with a p-value < 0.01 consisted of other members of the α-amylase, partly orthologous proteins from different organisms, partly related proteins with different functions. One example was the α-amylase of *B. stearothermophilus* (1qho) which catalyzes the production of α-maltose from glucan. Moreover, we found members of the cyclodextrin/cyclomaltodextrin glycosyltransferases (E.C.2.4.1.19), which catalyze the degradation of starch to cyclodextrins. These proteins belong to the same glucosidase superfamily as α-amylases. The results are again in accordance with the findings of [29].

The sequence identity between 1bag and many of the retrieved proteins is about 25%, which is well below the range where functional inference becomes difficult [59]. Based on sequence information alone, we would have missed many of these molecules, including the above-mentioned 1qho, whose sequence identity to 1bag is 22%.

Interpreting retrieval results for the SiteEngine queries is less easily done, since we deliberately used a surface-based approach to include also remotely similar proteins from other folds and families. Nevertheless, to preserve a common standard throughout the experiments, we included these structures as well in a comparative retrieval experiment. Thus, we added the SiteEngine queries to the set of pvSOAR query structures and repeated the retrieval experiment against CavBase with SEGA and the competitor algorithms[4]. Retrieved proteins were judged relevant if one of the following criterions were met:

- Protein belongs to the same SCOP superfamily.
- Protein is annotated with the same EC number.
- Protein is known to bind the same ligand.

Table 5 shows the results. Since the number of relevant hits in CavBase were unknown, we resorted to the *average precision at k* to evaluate the results. For each query SEGA showed the best results.

5.6 Classification of Protein Binding Sites

As we specifically designed SEGA for the classification of proteins with respect to their ligands, we finally compared SEGA to other graph-based methods in a classification scenario. This gives us

[4]The high runtime requirements of GAVEO renders a retrieval of all queries against CavBase infeasible, hence we had to exclude it from this experiment

Table 5: Average precision at top k ranks for different approaches.

k	BK	GH	RW	SEGA	SH	SP
10	0.675	0.350	0.150	0.900	0.875	0.100
20	0.700	0.188	0.125	0.763	0.750	0.050
30	0.608	0.133	0.108	0.683	0.692	0.042
40	0.544	0.113	0.088	0.619	0.619	0.038
50	0.505	0.090	0.090	0.570	0.555	0.035

Table 6: Results of k-nearest neighbor classification (percentage of correct predictions) with leave-one-out cross-validation of the original ATP/NADH dataset ($\alpha = 1$).

k	BK	GAVEO	GH	RW	SA	SEGA	SH	SP
1	76.1	78.9	76.6	59.7	89.8	91.6	89.2	60.6
3	76.3	76.6	71.8	59.7	86.8	92.4	87.9	60.6
5	77.4	78.0	72.4	59.7	86.5	91.3	85.9	63.4
7	75.7	78.6	71.8	60.8	83.7	91.6	86.8	62.5

another *indirect* way to evaluate our approach in terms of classification accuracy. The idea is that, the better a similarity measure is, the better the performance of a similarity-based classifier using this measure should be. We realized this idea using the simple k-nearest neighbor classifier on a binary classification problem: ATP versus NADH binding pockets.

Table 6 shows the results of a leave-one-out cross validation on the first ATP/NADH classification problem based on different graph-based comparison methods. As a baseline we also included a classification based on sequence alignments (SA) using Smith-Waterman.

This dataset was previously used for the evaluation of the GAVEO approach [35] to test the capability of global graph alignments to predict the ligand correctly for cavities that bind a ligand in similar conformation.

While the nature of the dataset should favor global methods, SEGA still outperforms both global (GH and GAVEO) and local graph matching techniques (BK, RW and SP) which illustrates the benefit of a combined semi-global approach. As a bonus, our approach is even slightly better than the classification based on sequence alignment.

Since we developed SEGA in order to detect more remote similarities, too, globally similar binding sites or ligands bound in similar conformations should not be mandatory. To test SEGA on a more difficult classification problem, we constructed another ATP/NADH dataset, this time containing only one structure per fold according to the SCOP classification. Results are represented in Table 7. Expectedly, as the structures are only remotely similar, a low α performed best.

In both experiments, SEGA was able to achieve high classification accuracies, on datasets containing closely related and remotely related proteins, respectively.

While SEGA showed strong performance on binary classification problems, we finally wanted to test the approach on a multiple classification problem. To this end, we resorted again to the SiteEngine

Table 7: Results of k-nearest neighbor classification (percentage of correct predictions) with leave-one-out cross-validation for the one-fold ATP/NADH dataset ($\alpha = 0.1$).

k	BK	GAVEO	GH	RW	SA	SEGA	SH	SP
1	70.8	54.3	33.9	50.4	70.0	91.3	88.2	56.7
3	64.5	57.5	51.9	56.7	67.7	89.0	88.2	55.9
5	62.2	63.8	39.4	66.1	70.0	87.4	85.0	56.7
7	62.9	63.8	39.4	63.0	68.5	83.5	82.7	57.9

Table 8: Results of k-nearest neighbor classification (percentage of correct predictions) with leave-one-out cross-validation for the SiteEngine dataset ($\alpha = 1$).

k	BK	GAVEO	GH	RW	SA	SEGA	SH	SP
1	61.7	59.1	45.4	18.0	72.2	79.8	77.0	8.7
3	54.1	55.2	33.3	25.7	68.3	76.0	72.7	20.2
5	49.2	53.6	31.7	25.1	65.9	70.5	69.4	16.4
7	44.8	54.1	30.6	24.6	65.1	68.3	69.4	13.7

benchmark set, this time performing an n-to-n comparison of all binding sites. Table 8 summarizes the results for the multiclass problem. Again SEGA outperformed the other approaches.

6 Conclusion

In this paper, we have presented a new method for the computation of a graph alignment as a means for comparing proteins on a structural level. Our method, called SEGA, is a semi-global strategy in the sense that it shares properties with both local and global graph matching, similar to semi-global sequence alignments.

More precisely, it establishes a correspondence between the basic building blocks of the model (such as atoms or pseudocenters) from comparisons on the level of local substructures, resorting to global structure information only when necessary, which is more robust towards structural deviations and inaccuracies.

Yet, it generates a complete alignment of pseudocenters. From a biological point of view, this kind of information is often desirable, and perhaps even more important than a related (numerical) degree of similarity.

In our experiments, SEGA could outperform both purely global graph alignment (GH, GAVEO) as well as purely local approaches (BK, RW, SP), supporting our assumption that a semi-global strategy is better suited to deal with the specific problems that arise when analyzing protein binding pockets derived from crystal structures (e.g. structural deviations, varying cavity sizes and noisy data).

The fact that SEGA also outperformed the baseline sequence alignment in classification experiments is another example for the benefit of using structure-based analysis in addition to the well-established sequence methods. Both concepts should be viewed as complementary for determining the function of unknown proteins.

Compared to hitherto existing methods for graph comparison, SEGA achieves a remarkably good alignment quality paired with a high computational efficiency. Moreover, the experiments indicated that the less rigid SEGA approach is capable of detecting more remote similarities among proteins. In the classification experiments, SEGA outperformed the existing methods by a wide margin, showing that this strategy is indeed beneficial for the analysis of protein structures.

Most methods and algorithms for analyzing graphs are quite specialized and not universally applicable. Given the existence of many types of graphs (directed vs. undirected, labeled vs. unlabeled, etc.), it is clear that a method suitable for one problem class might not be useful for (and perhaps not even applicable to) another one. This is of course also true for our SEGA algorithm. Yet, we believe that the basic ideas underlying this algorithm are of more general interest, and that its adaptation to other types of problems, both within the biological sciences and even beyond, is promising enough to be addressed in future work.

Acknowledgments

The authors gratefully acknowledge financial support by the German Research Foundation (DFG) and the LOEWE Research Center for Synthetic Microbiology, Marburg.

References

[1] L. Jensen, R. Gupta, H. Staerfeldt, and S. Brunak, "Prediction of human protein function according to Gene Ontology categories," *Bioinformatics*, vol. 19, no. 5, pp. 635–642, 2003.

[2] K. Sjolander, "Phylogenomic inference of protein molecular function: advances and challenges," *Bioinformatics*, vol. 20, no. 2, pp. 170–179, 2004.

[3] P. Artymiuk, A. Poirrette, H. Grindley, D. Rice, and P. Willett, "A Graph-theoretic Approach to the Identification of Three-dimensional Patterns of Amino Acid Side-chains in Protein Structures," *Journal of Molecular Biology*, vol. 243, no. 2, pp. 327–344, 1994.

[4] L. Holm and J. Park, "DaliLite Workbench for Protein Structure Comparison," *Bioinformatics*, vol. 16, no. 6, pp. 566–567, 2000.

[5] I. Shindyalov and P. Bourne, "A Database and Tools for 3-D Protein Structure Comparison and Alignment using the Combinatorial Extension (CE) Algorithm," *Nucleic Acids Research*, vol. 29, no. 1, pp. 228–229, 2001.

[6] A. Ortiz, C. Strauss, and O. Olmea, "MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison," *Protein Science*, vol. 11, no. 11, pp. 2606–2621, 2002.

[7] M. Jambon, A. Imberty, G. Deleage, and C. Geourjon, "A New Bioinformatic Approach to Detect Common 3 D Sites in Protein Structures," *Proteins Structure Function and Genetics*, vol. 52, no. 2, pp. 137–145, 2003.

[8] O. Dror, H. Benyamini, R. Nussinov, and H. Wolfson, "MASS: Multiple Structural Alignment by Secondary Structures." *Bioinformatics*, vol. 19, no. 1, pp. i95–104, 2003.

[9] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H. Wolfson, "The Multiple Common Point Set Problem and Its Application to Molecule Binding Pattern Detection," *Journal of Computational Biology*, vol. 13, no. 2, pp. 407–428, 2006.

[10] B. Rost, "Enzyme function less conserved than anticipated," *Journal of Molecular Biology*, vol. 318, no. 2, pp. 595–608, 2002.

[11] W. Tian and J. Skolnick, "How well is enzyme function conserved as a function of pairwise sequence identity?" *Journal of Molecular Biology*, vol. 333, no. 4, pp. 863–882, 2003.

[12] J. Thornton, C. Orengo, A. Todd, and F. Pearl, "Protein folds, functions and evolution," *Journal of Molecular Biology*, vol. 293, no. 2, pp. 333–342, 1999.

[13] S. Copley, W. Novak, and P. Babbitt, "Divergence of Function in the Thioredoxin Fold Suprafamily: Evidence for Evolution of Peroxiredoxins from a Thioredoxin-like Ancestor," *Biochemistry*, vol. 43, no. 44, pp. 13 981–13 995, 2004.

[14] K. Wang and R. Samudrala, "FSSA: a novel method for identifying functional signatures from structural alignments," *Bioinformatics*, vol. 21, no. 13, pp. 2969–2977, 2005.

[15] B. Polacco and P. Babbitt, "Automated discovery of 3D motifs for protein function annotation," *Bioinformatics*, vol. 22, no. 6, pp. 723–730, 2006.

[16] S. Schmitt, D. Kuhn, and G. Klebe, "A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology," *Journal of Molecular Biology*, vol. 323, no. 2, pp. 387–406, 2002.

[17] N. Weskamp, E. Hüllermeier, D. Kuhn, and G. Klebe, "Multiple graph alignment for the structural analysis of protein active sites," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 310–320, 2007.

[18] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, no. 4, pp. 536–540, 1995.

[19] C. Orengo, A. Michie, S. Jones, D. Jones, M. Swindells, and J. Thornton, "CATH-A Hierarchic Classification of Protein Domain Structures," *Structure*, vol. 5, no. 8, pp. 1093–1108, 1997.

[20] O. Redfern, A. Harrison, T. Dallman, F. Pearl, and C. Orengo, "CATHEDRAL: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures," *PLoS Computational Biology*, vol. 3, no. 11, pp. 2334–2347, 2007.

[21] A. Wallace, N. Borkakoti, and J. Thornton, "TESS: A Geometric Hashing Algorithm for Deriving 3D Coordinate Templates for Searching Structural Databases. Application to Enzyme Active Sites," *Protein Science*, vol. 6, no. 11, pp. 2308–2323, 1997.

[22] N. Leibowitz, R. Nussinov, and H. Wolfson, "MUSTA-A General, Efficient, Automated Method for Multiple Structure Alignment and Detection of Common Motifs: Application to Proteins," *Journal of Computational Biology*, vol. 8, no. 2, pp. 93–121, 2001.

[23] J. Barker and J. Thornton, "An algorithm for constraint-based structural template matching: application to 3D templates with statistical analysis," *Bioinformatics*, vol. 19, no. 13, pp. 1644–1649, 2003.

[24] P. Wangikar, A. Tendulkar, S. Ramya, D. Mali, and S. Sarawagi, "Functional Sites in Protein Families Uncovered via an Objective and Automated Graph Theoretic Approach," *Journal of Molecular Biology*, vol. 326, no. 3, pp. 955–978, 2003.

[25] G. Kleywegt, "Recognition of spatial motifs in protein structures," *Journal of Molecular Biology*, vol. 285, no. 4, pp. 1887–1897, 1999.

[26] A. Stark and R. Russell, "Annotation in three dimensions. PINTS: Patterns in Non-homologous Tertiary Structures," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3341–3344, 2003.

[27] F. Glaser, R. Morris, R. Najmanovich, R. Laskowski, and J. Thornton, "A method for localizing ligand binding pockets in protein structures," *Proteins*, vol. 62, pp. 479–488, 2006.

[28] S. Bagley and R. Altman, "Characterizing the microenvironment surrounding protein sites." *Protein Science*, vol. 4, no. 4, p. 622635, 1995.

[29] T. Binkowski, L. Adamian, and J. Liang, "Inferring functional relationships of proteins from local sequence and spatial surface patterns," *Journal of Molecular Biology*, vol. 332, no. 2, pp. 505–526, 2003.

[30] Y. Tseng, J. Dundas, and J. Liang, "Predicting protein function and binding profile via matching of local evolutionary and geometric surface patterns," *Journal of Molecular Biology*, vol. 387, no. 2, pp. 451–464, 2009.

[31] L. Xie and P. Bourne, "Detecting evolutionary relationships across existing fold space, using sequence order-independent profile–profile alignments," *Proceedings of the National Academy of Sciences*, vol. 105, no. 14, pp. 5441–5446, 2008.

[32] A. Shulman-Peleg, R. Nussinov, and H. Wolfson, "Recognition of Functional Sites in Protein Structures," *Journal of Molecular Biology*, vol. 339, no. 3, pp. 607–633, 2004.

[33] R. Spriggs, P. Artymiuk, and P. Willett, "Searching for Patterns of Amino Acids in 3D Protein Structures," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 2, pp. 412–421, 2003.

[34] K. Kinoshita and H. Nakamura, "Identification of the Ligand Binding Sites on the Molecular Surface of Proteins," *Protein Science*, vol. 14, no. 3, pp. 711–718, 2005.

[35] T. Fober, M. Mernberger, G. Klebe, and E. Hullermeier, "Evolutionary Construction of Multiple Graph Alignments for the Structural Analysis of Biomolecules," *Bioinformatics*, vol. Advanced Access, p. btp144, 2009. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btp144v1

[36] J. Berg and M. Lässig, "Local Graph Alignment and Motif Search in Biological Networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 41, pp. 14 689–14 694, 2004.

[37] I. Xenarios, L. Salwinski, X. Duan, P. Higney, S. Kim, and D. Eisenberg, "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions," *Nucleic Acids Research*, vol. 30, no. 1, pp. 303–305, 2002.

[38] H. Bunke and X. Jiang, *Graph Matching and Similarity*, D. M. Horia-Nicolai Teodorescu, Ed. Kluwer Academic Publishers Norwell, MA, USA, 2000.

[39] C. Bron and J. Kerbosch, "Algorithm 457: Finding All Cliques of an Undirected Graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[40] M. Pelillo, "A Unifying Framework for Relational Structure Matching," in *Proceedings. Fourteenth International Conference on Pattern Recognition*, vol. 2, 1998.

[41] J. McGregor, "Backtrack Search Algorithms and the Maximal Common Subgraph Problem," *Software - Practice and Experience*, vol. 12, no. 1, pp. 23–34, 1982.

[42] D. Schmidt and L. Druffel, "A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices," *Journal of the ACM*, vol. 23, no. 3, pp. 433–445, 1976.

[43] M. Wagener and J. Gasteiger, "The Determination of Maximum Common Substructures by a Genetic Algorithm: Application in Synthesis Design and for the Structural Analysis of Biological Activity," *Angewandte Chemie International Edition in English*, vol. 33, no. 11, pp. 1189–1192, 1994.

[44] J. Raymond, E. Gardiner, and P. Willett, "Heuristics for Similarity Searching of Chemical Graphs Using a Maximum Common Edge Subgraph Algorithm," *Jorunal of Chemical Information and Computer Sciences*, vol. 42, no. 2, pp. 305–316, 2002.

[45] J. Ullmann, "An Algorithm for Subgraph Isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, 1976.

[46] T. Gärtner, "A survey of kernels for structured data," *SIGKKD Explorations*, vol. 5, no. 1, pp. 49–58, 2003.

[47] K. Borgwardt, C. Ong, S. Schonauer, S. Vishwanathan, A. Smola, and H. Kriegel, "Protein Function Prediction via Graph Kernels," *Bioinformatics*, vol. 21, no. 1, pp. i47–i56, 2005.

[48] A. Sanfeliu and K. Fu, "A Distance Measure Between Attributed Relational Graphs for Pattern Recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 3, pp. 353–362, 1983.

[49] R. Najmanovich, N. Kurbatova, and J. Thornton, "Detection of 3D atomic similarities and their use in the discrimination of small molecule protein-binding sites." *Bioinformatics*, vol. 24, no. 16, p. i105, 2008.

[50] M. Hendlich, F. Rippmann, and G. Barnickel, "LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-binding Sites in Proteins," *Journal of Molecular Graphics and Modelling*, vol. 15, no. 6, pp. 359–363, 1997.

[51] M. Shatsky, R. Nussinov, H. Wolfson, T. Tatge, and S. Bioinformatics, "Flexible protein alignment and hinge detection," *Journal of Computational Biology*, vol. 11, pp. 8–106, 2004.

[52] G. Verbitsky, R. Nussinov, and H. Wolfson, "Flexible structural comparison allowing hinge-bending, swiveling motions," *Proteins Structure Function and Genetics*, vol. 34, no. 2, pp. 232–254, 1999.

[53] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press New York, NY, USA, 2004.

[54] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[55] H. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics*, vol. 52, no. 1, pp. 7–21, 2005.

[56] J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1994.

[57] S. Schmitt, M. Hendlich, and G. Klebe, "From Structure to Function: A New Approach to Detect Functional Similarity among Proteins Independent from Sequence and Fold Homology," *Angewandte Chemie International Edition*, vol. 40, no. 17, pp. 3141–3146, 2001.

[58] T. Binkowski, S. Naghibzadeh, and J. Liang, "CASTp: computed atlas of surface topography of proteins," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3352–3355, 2003.

[59] C. Orengo, A. Todd, and J. Thornton, "From protein structure to function," *Current Opinion in Structural Biology*, vol. 9, no. 3, pp. 374–382, 1999.