Superposition and Alignment of Labeled Point Clouds

Thomas Fober, Serghei Glinca, Gerhard Klebe and Eyke Hüllermeier
Philipps-Universität Marburg
35032 Marburg, Germany

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

Abstract

Geometric objects are often represented approximately in terms of a finite set of points in three-dimensional Euclidean space. In this paper, we extend this representation to what we call *labeled point clouds*. A labeled point cloud is a finite set of points, where each point is not only associated with a position in three-dimensional space, but also with a discrete class label that represents a specific property. This type of model is especially suitable for modeling biomolecules such as proteins and protein binding sites, where a label may represent an atom type or a physico-chemical property. Proceeding from this representation, we address the question of how to compare two labeled points clouds in terms of their similarity. Using fuzzy modeling techniques, we develop a suitable similarity measure as well as an efficient evolutionary algorithm to compute it. Moreover, we consider the problem of establishing an *alignment* of the structures in the sense of a one-to-one correspondence between their basic constituents. From a biological point of view, alignments of this kind are of great interest, since mutually corresponding molecular constituents offer important information about evolution and heredity, and can also serve as a means to explain a degree of similarity. In this paper, we therefore develop a method for computing pairwise or multiple alignments of labeled point clouds. To this end, we proceed from an optimal superposition of the corresponding point clouds and construct an alignment which is as much as possible in agreement with the neighborhood structure established by this superposition. We apply our methods to the structural analysis of protein binding sites.

1 Introduction

Geometric objects are often represented in terms of a set of points in three-dimensional Euclidean space. This type of representation is finite and hence approximate (even though the number of points can become very large, as for example in laser range scanning), focusing on the most important characteristics of the object while ignoring less important details. A well-known example of a representation of this kind is the *Molfile* format [16], where molecules are described in terms of the spatial coordinates of all atoms. However, since not only the position but also the type of an atom is of interest, this representation is not a simple point cloud. Likewise, other biomolecular structures, such as proteins and protein binding sites, are not only characterized by their geometry but also by additional features, such as physico-chemical properties. In this paper, we therefore introduce the concept of a *labeled point cloud*. A labeled point cloud is a finite set of points, where each point is not only associated with a position in three-dimensional space, but also with a discrete class label that represents a specific property. Formally, a labeled point cloud P is a set of points $\{p_1, p_2, \ldots, p_n\}$ with two associated functions: $c : P \to \mathbb{R}^3$ maps points to coordinates in the Euclidean space, and $\ell : P \to \mathcal{L}$ assigns a label to each point.

Since theory formation in the biological sciences is largely founded on similarity-based and analogical reasoning principles, the comparison of two (or more) objects with each other is a fundamental problem in bioinformatics. In this paper we propose the method of *labeled point cloud superposition* (LPCS), from which we then derive a proper similarity measure for labeled points clouds. Our measure proceeds from the idea of equivalence (inclusion) of point clouds in a set-theoretic sense while being tolerant toward exceptions (on the level of label information) and geometric deformations.

Despite producing a degree of similarity, an LPCS does not establish a unique alignment, that is, a one-to-one correspondence between the basic constituents of the structures. From a biological point of view, alignments of this kind are of great interest, as they offer important information about evolution and heredity. In sequence analysis, for example, one is typically not only interested in the degree of similarity between two DNA or amino acid sequences, but also in finding those parts of the two sequences that match each other and, therefore, give an explanation for the similarity score. Therefore, we additionally develop an approach to structure alignment, called *multiple point cloud alignment*. To this end, we proceed from an optimal superposition of the corresponding point clouds and construct an alignment which is as much as possible in agreement with the neighborhood structure established by this superposition.

The remainder of the paper is organized as follows. Subsequent to an overview of related work in Section 2 and a brief introduction to protein binding sites and their formal representation in Section 3, we introduce the concept of LPCS in Section 4. The problem of computing an LPCS is then addressed in Section 5, where an evolution strategy is proposed for this purpose. Section 6 introduces the concepts of pairwise and multiple point cloud alignment. Section 7 is devoted to the experimental validation of the approach, and Section 8 concludes the paper.

2 Related Work

The problems of structural similarity computation and structural alignment have been addresses in many research fields including, amongst others, pattern recognition, data mining and machine learning, databases, structural bio- and chemoinformatics. Two major directions can be distinguished, namely geometrical and graph-based approaches. Subsequently, we give a brief overview of related work in these two fields, focusing on those approaches most relevant for our method.

2.1 Geometrical Approaches

A common approach in this branch is to apply methods from computational geometry to objects represented in terms of point clouds. Roughly, such methods can be divided into the following categories: exact point matching, one-to-one matching, approximate point matching, and partial point matching. The first two categories are essentially dealing with the question of isomorphism [1], that is, equality instead of similarity. For applications in bioinformatics, where inexact structures of different size need to be compared, this approach is obviously not flexible enough and therefore inappropriate.

Approaches that neither require exact matches nor a one-to-one correspondence are often based on the *Hausdorff* distance between sets [20, 19, 17, 29]. These approaches have different drawbacks, however. In particular, many of them are restricted to two-dimensional objects, for which the computational complexity can already become high, or they are not invariant toward rotation. Besides, these methods do not take point labels into consideration.

Another interesting approach, namely geometric hashing, allows for calculating a partial alignment between point clouds [34, 3, 26, 25, 42, 27]. First, k-tuples are drawn from a point cloud and stored in a hash table. Then, in the recognition phase, k-tuples of another point cloud are drawn and looked up in the hash table. If two matching k-tuples are found, one from the first and one from the second point cloud, they can be superimposed and thus define a *transformation* that can be used to derive an alignment.

More specialized methods can be used if additional information is available. For example, the authors in [33] make use of the C-α atoms to align a set of (labeled) point clouds in a very efficient way. However, due to the fact that similar binding site patterns may occur in proteins with different folds, such approaches are limited. A classical example for protein binding sites for which such approaches will fail is (Chymo)Trypsin and Subtilisin [32], other examples can be found in [39, 38]. These binding sites are similar but do not share similar folds.

2.2 Graph-Based Approaches

A second important direction is to use graph-based methods: Geometric objects are first mapped to a graph representation, capturing geometric information in terms of edge weights and label information in terms of node labels, and are then compared by means of graph-theoretic methods.

The authors in [2] especially advocate the use of *graph kernels* as similarity measures [7, 15]. Roughly speaking, a graph kernel is a real function κ that, to compute a degree of similarity $\kappa(G, G')$ between two graphs G and G', first decomposes each graph into a set of simple components, then compares all components with each other, and finally adds the similarity degrees thus obtained. In our experimental study, we use the well-known shortest path and random walk kernels. In the former case, the components are given by all shortest paths between any pair of nodes in a graph [8]. In the latter case, the (infinite) set of components is defined by the set of all walks in a graph [15]. In both cases, the components are hence one-dimensional structures that can easily be compared by similarity measures on strings. The runtime complexity is $O(M^4)$ for shortest path and $O(M^6)$ for random walk kernels, where M is the number of nodes in the graphs.

The concepts of maximum common subgraph [10] and minimum common supergraph [11] have been widely used for the comparison of chemical compounds [31]. Like other problems related to subgraph isomorphism, they are computationally complex. While being acceptable for small molecular structures, complexity indeed becomes prohibitive for large structures such as proteins. Moreover, subgraph isomorphism is not tolerant toward noise. Relaxations and approximate methods have been proposed [12, 35, 43, 37], but these increase computational complexity even further.

In [40], the authors introduce the concept of a *multiple graph alignment* as a structural counterpart to sequence alignment, and use it for the comparison of protein binding sites. Roughly speaking, a multiple graph alignment is an alignment of several protein structures, each of which is represented in the form of a graph. An alignment of that kind is produced using a two-step procedure: First, a seed solution is obtained by means of exact graph matching techniques. Then, this seed is successively expanded by means of a greedy optimization technique.

2.3 Graph-Based vs. Geometric Modeling

The LPCS method introduced in this paper can be seen as a combination of geometric and graph-based approaches: Using point clouds, it captures geometric information in an explicit way, but at the same time, it allows for adding label information as typically used in graph representations. Consequently, LPCS inherits properties, and hence advantages and disadvantages, from both directions, with the advantages arguably being more important than the disadvantages.

Probably the most important difference between geometric and graph-based approaches concerns their flexibility toward variations and conformational changes of the underlying structures. Since geometric approaches, especially those based on point clouds, are rigid, graph-based methods can have an advantage in situations where much flexibility is needed. On the other hand, this flexibility comes at a certain price and can also have disadvantages, especially when it is not needed. First, graph-based methods can be too flexible in the sense of producing solutions that are geometrically infeasible. In fact, many types of geometrical constraints cannot be verified on the graph level. Second, graph-based methods drop large parts of the (global) geometric information and represent the rest only in an implicit way. This information must hence be reconstructed from the graph representation whenever needed. Due to the discrete nature of graphs, this normally leads to combinatorial optimization problems that are hard to solve. As a result, graph-based methods are often computationally complex.

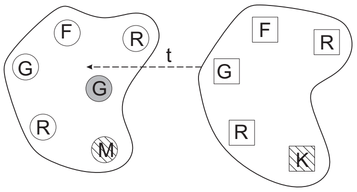


Figure 1: Two point clouds A (left, points as circle) and B (right, points as squares): The intra-point distances are the same in both point clouds, except for the additional gray point in A. Labels are depicted as letters within the circles and boxes, respectively.

3 Modeling Protein Binding Sites

In this paper, our special interest concerns the modeling of protein binding sites. More specifically, our work builds upon CavBase [32], a database for the automated detection, extraction, and storing of protein cavities (hypothetical binding sites) from experimentally determined protein structures (available through the PDB). In CavBase, a set of points is used as a first approximation to describe a binding pocket. The database currently contains 248,686 hypothetical binding sites that have been extracted from 61,516 publicly available protein structures using the LIGSITE algorithm [18].

The geometrical arrangement of the pocket and its physico-chemical properties are first represented by predefined *pseudocenters* – spatial points that represent the geometric center of a particular property. The type and the spatial position of the centers depend on the amino acids that border the binding pocket and expose their functional groups. They are derived from the protein structure using a set of predefined rules [32]. As possible types for pseudocenters, hydrogen-bond donor, acceptor, mixed donor/acceptor, hydrophobic aliphatic, metal ion, pi (accounts for the ability to form π–π interactions) and aromatic properties are considered.

Pseudocenters can be regarded as a compressed representation of areas on the cavity surface where certain protein-ligand interactions are experienced. Consequently, a set of pseudocenters is an approximate representation of a spatial distribution of physico-chemical properties. Obviously, just like in the case of Molfile, this representation is already in the form of a labeled point cloud: Points correspond to pseudocenters and are labeled with a physico-chemical property and their spatial coordinates.

4 Labeled Point Cloud Superposition

Intuitively, two labeled point clouds are similar if they can be spatially superimposed, at least approximately. That is, by fixing the first and “moving” the second one (as a whole, i.e., without changing the internal arrangement of points) in a proper way, an approximate superposition of the two structures is obtained. More specifically, we will say that two point clouds are well superimposed if, for each point in one of the structures, there exists a point in the other cloud that is spatially close and has the same label. As an illustration, the example in Figure 1 shows two point clouds A and B, for simplicity only in two dimensions. By moving B to the left (or A to the right), a superposition can be found so that, except for the hatched and gray nodes, all points in A spatially coincide with a corresponding point in B having the same label, and vice versa. So, A and B can be considered as being similar, at least to some extent.

4.1 Similarity as Fuzzy Equivalence

More formally, let

$$A = \{(x_1, \ell(x_1)), \ldots, (x_m, \ell(x_m))\}$$

be a point cloud consisting of m points $x_i = (x_{i1}, x_{i2}, x_{i3}) \in \mathbb{R}^3$ with associated label $\ell(x_i) \in \mathcal{L}$, where $\mathcal{L}$ is a discrete set of labels (in the context of modeling protein binding sites, as discussed in the previous section, $\mathcal{L}$ is given by the seven types of pseudocenters). Moreover, let

$$B = \{(y_1, \ell(y_1)), \ldots, (y_n, \ell(y_n))\}$$

be a second point cloud to be compared with A. In the following, we define a function $\mathrm{SIM}(\cdot, \cdot)$ that returns a degree of similarity between two such structures A and B.

Roughly speaking, we consider similarity as a generalized (fuzzy) equivalence. Moreover, motivated by the fact that, for ordinary sets A and B, $(A = B) \Leftrightarrow (A \subset B) \wedge (B \subset A)$, we reduce this equivalence to two inclusion relations, namely the inclusion of A in B and, vice versa, of B in A. Thus, we are first of all interested in whether each point $y \in B$ is also present in A (and each point $x \in A$ also present in B). More specifically, we are interested in the degree to which $y \in B$ is at least "fuzzily" present in A. Recall that a fuzzy subset F of a reference set U is characterized by its *membership function*, which is a $U \rightarrow [0, 1]$ mapping $\mu_F(\cdot)$ that generalizes the characteristic function of a set [45]. For each $u \in U$, $\mu_F(u)$ is the degree of membership of u in the fuzzy set F.

For a fixed $y \in B$, we define the membership degree of this point in A, that is, the degree to which this point is also present in A, by

$$\mu_A(y) \; = \; \exp\left(-\gamma \cdot d(y, A)\right) \quad , \tag{1}$$

where

$$d(y, A) = \min_{\substack{x \in A \\ \ell(x) = \ell(y)}} \quad \|y - x\|_1$$

is the distance between a point $y \in B$ and the closest point $x \in A$ having the same label ($d(y, A) = \infty$ and hence $\mu_A(y) = 0$ if no such point exists); for $x \in A$, $\mu_B(x)$ and $d(x, B)$ are defined analogously.

For a pair of sets A and B, the inclusion $B \subset A$ is equivalent to the equality $B = B \cap A$. Consequently, one possibility to relax the inclusion relation on sets to a "fuzzy inclusion" is to make use of a corresponding fuzzy equivalence like, for example, the Jaccard measure $|A \cap B|/|A \cup B|$. In the fuzzy case, set intersection and union are accomplished through t-norm and t-conorm operators $\top$ and $\perp$ [23], respectively, and set cardinality through summing membership degrees. Noting that $\mu_B(y) = 1$ for all $y \in B$ (and that 1 is the neutral element of a t-norm $\top$), this eventually yields

$$\begin{aligned} inc(B, A) &= \frac{|B \cap (B \cap A)|}{|B \cup (B \cap A)|} = \frac{|B \cap A|}{|B|} \\ &= \frac{\sum_{y \in B} \top(\mu_A(y), \mu_B(y))}{\sum_{y \in B} \mu_B(y)} = \frac{1}{|B|} \sum_{y \in B} \mu_A(y). \end{aligned}$$

4.2 Optimizing Similarity

As mentioned above, the idea of our approach is to define the similarity between two labeled point clouds in terms of the best superposition of these two clouds. Therefore, let $\mathrm{TF}(\cdot, t)$ be a function that moves a point cloud via rotation and translation, as specified by the six-dimensional vector $t = (\theta_1, \theta_2, \theta_3, \delta_1, \delta_2, \delta_3) \in [0, 2\pi]^3 \times \mathbb{R}^3$. Thus,

$$B^* = \mathrm{TF}(B, t) = \{(y_1^*, \ell(y_1^*)), \ldots, (y_n^*, \ell(y_n^*))\}$$

is the point cloud obtained by translating the point cloud B by $\delta = (\delta_1, \delta_2, \delta_3)$ (which means adding δ to each point $y \in B$) and rotating the result thus obtained by the angles θ_1, θ_2, and θ_3. Note that this operation leaves the label information unchanged (i.e., $\ell(y_i) = \ell(y_i^*)$). The position-invariant degree of inclusion of B in A is then given by

$$\mathrm{INC}(B, A) \; = \; \max_{t \in [0, 2\pi]^3 \times \mathbb{R}^3} inc(\mathrm{TF}(B, t), A) \quad , \tag{2}$$

and $\mathrm{INC}(A, B)$ is defined analogously.

Based on these degrees, the similarity between A and B, in the sense of a generalized equivalence, can be defined as

$$\mathrm{SIM}(A, B) \;=\; \min\{\,\mathrm{INC}(A, B),\; \mathrm{INC}(B, A)\,\} \;\; . \tag{3}$$

4.3 Generalizing Similarity

It is worth mentioning, however, that (3) is not always appropriate, especially if A and B greatly differ in size. In some applications, it makes sense to have a high similarity degree even if A is only a substructure of B, for example if A is a subpocket of B containing the most important catalytic residues (while the rest of the binding site B is functionally less important). Obviously, this is not guaranteed by (3). An interesting generalization, therefore, is to let

$$\begin{aligned}\mathrm{SIM}(A, B) =& \alpha \cdot \min\{\mathrm{INC}(A, B), \mathrm{INC}(B, A)\} \\ &+ (1-\alpha) \cdot \max\{\mathrm{INC}(A, B), \mathrm{INC}(B, A)\} \;\; .\end{aligned} \tag{4}$$

Formally, this similarity measure can be motivated from a fuzzy logical point of view as follows. Considering the $\min$ ($\max$) operator as a generalized conjunction (disjunction), the first (second) combination of the two inclusion degrees is the truth degree of the proposition that A is contained in B AND (OR) B is contained in A. A conjunctive combination of the two degrees of inclusion is obviously more demanding than a disjunctive one, as the former requires equality between A and B while the latter only requires inclusion of A in B or B in A. The measure (4), which formally corresponds to an OWA (ordered weighted average) combination of the two degrees of inclusion [44], achieves a trade-off between these two extreme aggregation modes, which is controlled by the parameter $\alpha \in [0, 1]$: The closer α is to 0, the closer the aggregation is to the maximum, i.e., the less demanding it becomes. The optimal α is application-specific and depends on the purpose of the similarity measure.

5 Solving the LPCS Problem

The computation of the similarity (4) involves the solution of a real-valued optimization problem, namely the problem of finding an optimal vector t in (2) and, thus, an optimal point cloud superposition. The objective function to be maximized here is highly non-linear and multimodal. As an illustration, Figure 2 shows the objective function obtained for the superposition of a randomly generated two-dimensional point cloud A (in which all points have the same label) with itself. This function maps each two-dimensional translation vector $t = (x, y)$ to the corresponding similarity degree between $\mathrm{TF}(A)$ and A (where we used $\alpha = 1$ in (4) and did not consider rotation). As can be seen, there is a sharp peak at $t = (0, 0)$, which corresponds to the optimal superposition. Surrounding this solution, however, there are also many local optima.

The problem of local optima also becomes clear from the small example in Figure 1. Moving the point cloud A from left to right, into the direction of B, has the following effect: First, a good superposition of two sub-clouds will be found, namely the right part of cloud A and the left part of cloud B. This results in a local maximum. Moving A further to the right leads to a larger local maximum (sub-clouds are growing), until the global maximum will eventually be reached.

5.1 Evolution Strategies

To solve the LPCS problem, we resort to *evolution strategies* (ES), a population-based, stochastic optimization method inspired by biological evolution and specifically developed for real-valued optimization problems [5]. An evolution strategy is based on a population, a set of μ (sub-optimal) candidate solutions that are initially spread randomly over the search space. In each generation,

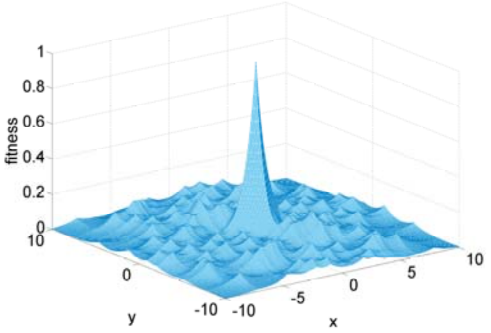


Figure 2: Example of an LPCS objective function.

new solutions are generated by applying the genetic operators *recombination* and *mutation*. Recombination randomly selects ρ individuals from the current population and combines them to a new solution. Mutation takes this solution and shifts it randomly in the search space. An ES produces $\lambda = \lceil \mu \cdot \nu \rceil$ offsprings per iteration, so that this procedure has to be repeated λ times. A selection operator implements the "survival of the fittest" principle by picking the best individuals for the new population. There are two kinds of selection: The *plus*-selection chooses the best μ individuals among the offsprings plus the parents, while the *comma*-selection ignores the parent generation (this requires $\nu > 1$).

A main advantage of the ES is its self-adaptation mechanism that controls the step sizes used in the mutation operator. One property of this mechanism (the advantage during optimization is obvious) is that step sizes decrease dramatically if the optimization reached a maximum. This property can be used as a termination criterion (stop when the largest step size falls below a given threshold).

Population-based optimization methods are especially advantageous for highly multimodal problems. Using a large population leads to an increased probability to generate a candidate solution in a region where the direction of descent points to the global maximum. Choosing the membership function (1) as a strictly monotone decreasing function which converges to zero ensures to have this direction in each point $t \in [0, 2\pi]^3 \times \mathbb{R}^3$ and thus greatly simplifies the maximization problem.

5.2 Complexity

Even though evolution strategies are generally known to be quite efficient solvers, the concrete complexity does of course depend on the application at hand. The application-specific part is the fitness function, i.e., the objective function to be optimized. This function has to be evaluated frequently and, therefore, is an important factor for the runtime. In our case, this function is given by the similarity measure (4), and its evaluation is strongly dominated by the nearest neighbor search which has to be conducted for each single point in both structures (recall that, according to (1), membership degrees are determined by the distance to closest points with the same label).

There exist a lot of data structures for supporting nearest neighbor search; see e.g. [13]. The most efficient among them need time $\mathcal{O}(n \log^2 n)$ for construction and $\mathcal{O}(\log^3 n)$ for answering a query. Unfortunately, we are not aware of an approach that allows for updating a data structure in an efficient and dynamic way. This would be desirable for our problem, in which the point clouds permanently change (the point cloud associated with an individual changes in each iteration). Instead, conventional approaches necessitate a construction from scratch in every iteration.

Figure 3 compares the runtimes, as a function of the number of points, for two approaches: (1) The use of a kd-tree data structure [13], which is reconstructed in each iteration and then used for query processing. (2) The use of a simple linear data structure, in which the points are stored in a fixed order. It needs linear instead of logarithmic time to answer a query but, on the other hand, does not cause additional costs for reconstruction. As can be seen, the use of a more complex approach pays

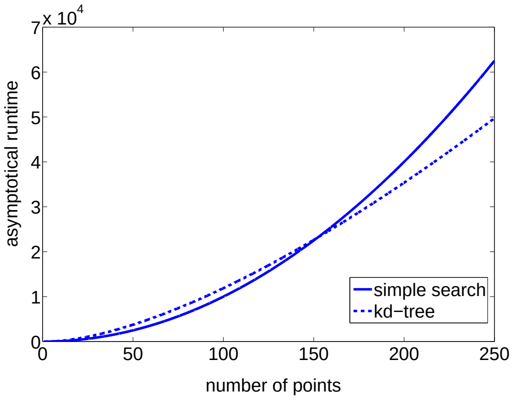


Figure 3: Runtime of a simple procedure and a more complex data structure as a function of the number of points.

off only for sufficiently large point clouds: The kd-tree reaches a break-even point at approximately 150 points.

In our application, we are mainly concerned with protein binding sites, which are characterized by around 180 points on average (even though much larger structures do of course exist). The use of a complex data structure did therefore not pay off. Nevertheless, we increased efficiency by hashing the points x_i of a point cloud, using the label $\ell(x_i) \in \mathcal{L}$ as a key. Since nearest neighbors are only searched among points having the same label, this obviously reduces runtime by a factor of approximately $|\mathcal{L}|$.

6 Multiple Point Cloud Alignment

So far, we have mainly been concerned with the problem of similarity measurement: Given two molecular structures, our goal was to determine a numerical degree of similarity between them. Apart from the number itself, however, one is naturally interested in an explanation of the similarity degree. From a biological point of view, it is especially interesting to find out what two (or even more) structures have in common. In this regard, the concept of an *alignment* has established itself as an important tool in bioinformatics, at least in the domain of sequence analysis. A structural counterpart to sequence alignment, called *multiple graph alignment*, was recently introduced in [40]. As the name suggests, the authors made use of graphs to model protein structures (binding sites). In this section, we propose an alternative approach to structure alignment, called *multiple point cloud alignment*, which operates on point clouds instead of graph representations.

Just like in the case of sequence alignment, the goal in structure alignment is to establish a one-to-one correspondence between the basic constituents of the structures under consideration. When comparing homologs from different species in protein cavity space, one has to deal with the same mutations that are also given in sequence space. Corresponding mutations, in conjunction with conformational variability, strongly affect the spatial structure of a binding site as well as its physico-chemical properties and, therefore, its point cloud descriptor. For example, a pseudocenter can be deleted or introduced due to a mutation in sequence space. Likewise, if a mutation replaces a certain functional group by another type of group at the same position, the physico-chemical property of a pseudocenter can change. Finally, the distance between two pseudocenters can change due to conformational differences.

Due to the above reasons, one cannot expect that point clouds of two related binding pockets match exactly. When looking for an alignment of two structures in the form of a one-to-one correspondence between pseudocenters, it is therefore necessary to allow for “mismatches” as well as pseudocenters for which no matching partner is defined. This situation is quite similar to sequence alignment, where mismatches between symbols and the insertion of blanks (to compensate for non-existing matching partners) is also allowed.

Definition 1 (Multiple Point Cloud Alignment, MPCA)**.** *Let $\mathcal{P}$ be a set of m point clouds $P_i = \{p_1^i, \ldots, p_{n_i}^i\}$, $i = 1, 2, \ldots, m$. A multiple point cloud alignment (MPCA) of these point clouds is a subset $\mathcal{A} \subseteq (P_1 \cup \{\perp\}) \times \cdots \times (P_m \cup \{\perp\})$ with the following properties:*

1. *for all $i = 1, 2, \ldots, m$ and for each $p \in P_i$ there exists exactly one $a = (a_1, \ldots, a_m) \in \mathcal{A}$ such that $p = a_i$;*

2. *for each $a = (a_1, \ldots, a_m) \in \mathcal{A}$ there exists at least one $1 \leq i \leq n$ such that $a_i \neq \perp$.*

Here, the symbol $\perp$ denotes a "dummy point" which is needed to compensate for non-existing matching partners.

Each tuple in the alignment represents a mutual assignment of m points, one from each point cloud P_i (possibly a dummy). Thus, the second property in the above definition requires that each tuple of the alignment contains at least one non-dummy point, and the first property means that each point of each point cloud occurs exactly once in the alignment. While these properties can be satisfied by a large number of alignments, we are of course looking for an alignment that reflects structural correspondence in an optimal way.

Our idea is to derive an optimal MPCA from an optimal superposition of the labeled point clouds. More specifically, we define the score of an alignment on the basis of a given superposition. As will be seen below, the problem to find an optimal MPCA thus comes down to solving several linear assignment problems. We proceed in two steps, solving the problem to align two structures before addressing the more general problem of multiple alignment.

6.1 Construction of Pairwise Alignments

To construct a pairwise alignment of two point clouds P_1 and P_2, we reduce the alignment problem to an optimal assignment problem. To this end, we need a square matrix $M = (m_{i,j})$, where $m_{i,j} \in \mathbb{R}$ defines the costs for assigning point $p_i \in P_1$ to point $p_j \in P_2$. According to Definition 1, the maximal length of a pairwise alignment is $n = n_1 + n_2 = |P_1| + |P_2|$. Therefore, to consider all possible alignments, the matrix M has size $n \times n$.

The entries $m_{i,j}$ are derived from the optimal superposition of point clouds P_1 and P_2 as produced by a modification of our LPCS method. The modification concerns the similarity measure to be maximized. Since we are looking for a *mutually* optimal alignment, we do not split the similarity into two optimal degrees of inclusion, as done by the measure (3). Instead, we define similarity in terms of a compromise measure as follows:

$$\mathrm{SIM}_{\mathrm{PCA}}(A, B) = \max_{t \in [0,2\pi]^3 \times \mathbb{R}^3} F(A, B, t) \ , \tag{5}$$

where

$$F(A, B, t) = \frac{1}{2} \left(inc(\mathrm{TF}(B, t), A) + inc(A, \mathrm{TF}(B, t)) \right).$$

Given a spatial superposition optimal in the sense of (5), it makes sense to define the cost $m_{i,j}$ in terms of the associated distance between point $p_i \in P_1$ and $p_j \in P_2$. To account for point-to-dummy mappings, the distance between a point and a dummy is specified by a parameter k. Finally, dummy-dummy assignments are scored by zero, so that these mappings will not influence the construction of the alignment. As an illustration, Table 1 shows a matrix M for two point clouds $P_1 = \{a, b, c, d\}$ and $P_2 = \{a', b', c'\}$.

Formally, an assignment (weighted bipartite matching) problem is specified by a graph $G = (V, E)$ with $V = V_1 \cup V_2$ ($V_1 \cap V_2 = \emptyset$) and $E = \{(v_1, v_2) \,|\, v_1 \in V_1,\ v_2 \in V_2\}$. Moreover, each edge $e \in E$ has

Table 1: Matrix representation of the optimal assignment problem.

	a'	b'	c'	$\perp$	$\perp$	$\perp$	$\perp$
a	$d(a,a')$	$d(a,b')$	$d(a,c')$	k	k	k	k
b	$d(b,a')$	$d(b,b')$	$d(b,c')$	k	k	k	k
c	$d(c,a')$	$d(c,b')$	$d(c,c')$	k	k	k	k
d	$d(d,a')$	$d(d,b')$	$d(d,c')$	k	k	k	k
$\perp$	k	k	k	0	0	0	0
$\perp$	k	k	k	0	0	0	0
$\perp$	k	k	k	0	0	0	0

an associated cost value $d(e)$. The goal is to find a subset of edges $M \subseteq E$ solving the following constrained optimization problem:

$$\text{minimize} \quad \sum_{e \in M} d(e)$$

subject to

$$\bigcup_{(v_1,v_2)\in M} \{v_1\} = V_1, \qquad \bigcup_{(v_1,v_2)\in M} \{v_2\} = V_2,$$

and such that $(v_1, v_2), (v_1', v_2') \in M$ with $(v_1, v_2) \neq (v_1', v_2')$ implies $v_1 \neq v_1'$ and $v_2 \neq v_2'$. In other words, M defines a bijection between V_1 and V_2. In our case, the sets V_1 and V_2 represent, respectively, the points in point cloud P_1 (supplemented with $|P_2|$ dummy points) and the points in cloud P_2 (supplemented with $|P_1|$ dummy points). Moreover, the cost $d(e)$ of an edge $e = (v_i, v_j)$ is given by the corresponding matrix entry $m_{i,j}$. See Figure 4 for an illustration.

To solve the weighted bipartite matching problem, we use the Hungarian algorithm [24] that needs time $\mathcal{O}(n^3)$. Once a cost-minimal assignment has been found, the point cloud alignment is defined by the corresponding node-to-node and node-to-dummy assignments, while dummy-to-dummy assignments are ignored.

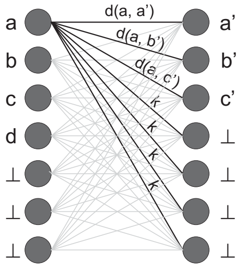


Figure 4: Illustration of the weighted bipartite graph matching problem.

6.2 Construction of Multiple Alignments

Instead of restricting to pairwise alignments, it is also interesting to look for a *multiple* alignment in the sense of a simultaneous alignment of a set of $m > 2$ structures. Alignments of this type can be useful, for example, to discover conserved patterns in a family of evolutionary related proteins.

To derive a multiple point cloud alignment (MPCA) of m point clouds, we resort to the idea of a star alignment [40]: One of the point clouds, say, P_1, is selected and aligned in a pairwise way with all other clouds P_i, $i = 2, \ldots, m$. The pairwise alignments are then “merged” by using P_1 as a pivot structure. Thus, if $p_{ij} \in P_i$ denotes the point (possibly a dummy) aligned with $p_{1j} \in P_1$ in the alignment of P_1 and P_i, then a single assignment in the multiple alignment is of the form

$$a = (p_{1j}, p_{2j}, p_{3j}, \ldots, p_{mj}) \ , \qquad (6)$$

and the cost caused by this assignment is given by

$$\sum_{1 \leq i < k \leq m} d((p_{ij}, p_{kj})) \ .$$

The sum of the costs of all these assignments finally determines the cost of the multiple alignment. Since the quality of a multiple alignment thus defined is strongly influenced by the choice of the pivot structure, we try each point cloud as a pivot and adopt the best result. Thus, $m(m-1)/2$ pairwise alignments have to be computed in total.

Another interesting way to obtain a multiple alignment in one step is proposed in [34], were a k-partite pivot graph is used instead of a bipartite graph. Since the problem to find a matching on that graph is NP-hard, the authors make use of a simple greedy heuristic. Again, since the solution depends on the pivot point cloud, it is reasonable to try each point cloud as a pivot.

The selection of pivot clouds is completely avoided by tree-based approaches. Here, we adapt an approach from [41]: In a first step, an UPGMA-tree is calculated based on the LPCS-scores of all pairwise comparisons. This tree defines the bottom-up order in which the alignments are merged. Leaves in the tree can be merged like in the pairwise case. Merging two inner nodes, both of which correspond to (multiple) alignments $a_i = (a_{1i}, \ldots, a_{ni})$ and $b_j = (b_{1j}, \ldots, b_{mj})$, respectively, is accomplished by calculating the averaging pairwise distances:

$$d(a_i, b_j) = \sum_{\substack{k=1,\ldots,n \\ l=1,\ldots,m}} d(a_{ki}, b_{lj}) \, ,$$

where $d(a_{ki}, b_{lj})$ is the distance between points a_{ki} and b_{lj} in the optimal superposition. If necessary, the distance matrix thus constructed is filled by dummy assignments to obtain a square matrix as input for the Hungarian algorithm.

6.3 Conserved Patterns

A multiple alignment of several (protein) structures can be analyzed in various ways. From a biological point of view, it is especially interesting to look for conserved patterns, i.e., substructures which can be found, at least approximately, in all structures.

Again, conserved patterns of that kind can be defined in various ways. Here, we simply define it in terms of the union of all conserved assignments (6), where we call an assignment conserved if

$$\begin{aligned} cons(a) &= \frac{|\{i \,|\, a_i \neq \perp\}|}{m} \geq \omega \\ maj(a) &= \max_{l \in \mathcal{L}} \frac{|\{i \,|\, a_i = l\}|}{|\{i \,|\, a_i \neq \perp\}|} \geq \xi \end{aligned}$$

for thresholds $\omega, \xi \in (0, 1]$, and with $cons(a)$ and $maj(a)$ denoting, respectively, the relative number of non-dummy nodes in a and the relative frequency of the most frequent label.

7 Experimental Results

In our experimental studies, we perform three types of experiments, in which we compare our methods with existing approaches applicable to the same type of problems, both graph-based and geometrical. In the first and second study, we focus on the aspect of similarity measurement, whereas the third study is devoted to the problem of structural alignment.

7.1 Data

The assessment of a similarity measure for biomolecular structures, such as protein binding sites, is clearly a non-trivial problem. In particular, since the concept of similarity by itself is rather vague and subjective, it is difficult to evaluate corresponding measures in an objective way. To circumvent this problem, we propose to evaluate similarity measures in an indirect way, namely by means of their performance in the context of nearest neighbor (NN) classification. The underlying idea is that, the better a similarity measure is, the better should be the predictive performance of an NN classifier using this measure for determining similar cases. To realize this idea, we compose a suitable classification data set in Section 7.1.1.

A second, much smaller data set is compiled in Section 7.1.2. The purpose of this data set is to check the performance of our method on the level of concrete structures. For this data set, a natural grouping into subsets is known, and we are interested in the question whether this grouping can be reproduced by clustering the structures according to the LPCS similarity measure. We compare the result not only with the ground truth, but also with clustering according to sequence similarity. Thus, we seek to show that some relationships not visible on the sequence level can still be recognized on the structural level.

For the third type of experiment, dealing with multiple structure alignment, data sets consisting of many structures sharing a common fragment are needed; such data sets will be introduced in Sections 7.1.3 and 7.1.4.

7.1.1 NADH/ATP

One important problem in pharmaceutical chemistry is the identification of protein binding sites that bind a certain ligand. We selected two classes of binding sites that bind, respectively, to NADH or ATP. This gives rise to a binary classification problem: Given a protein binding site, predict whether it binds NADH or ATP.

More concretely, we compiled a set of 355 protein binding pockets representing two classes of proteins that share, respectively, ATP and NADH as a cofactor. To this end, we used CavBase to retrieve all known ATP and NADH binding pockets that were co-crystallized with the respective ligand. Subsequently, we reduced the set to one cavity per protein, thus representing the enzymes by a single binding pocket. As protein ligands adopt different conformations due to their structural flexibility, it is likely that the ligands in our data set are bound in completely different ways, hence the corresponding binding pocket does not necessarily share much structural similarity. We thus had to ensure the selection of binding pockets with ligands bound in similar conformation. To achieve this, we used the Kabsch algorithm [21] to calculate the root mean square deviation (RMSD) between pairs of ligand structures. Subsequently, we combined all proteins whose ligands yielded a RMSD value below a threshold of 0.4, thereby ensuring a certain degree of similarity. This value was chosen as a trade-off between data set size and similarity. Eventually, we thus obtained a two-class data set comprising 214 NADH-binding proteins and 141 ATP-binding proteins.

7.1.2 Carbonic anhydrases

Carbonic anhydrases (CA, E.C. 4.2.1.1.) catalyze the conversion of carbon dioxide and water to bicarbonate and a proton. They are encoded by four evolutionary unrelated gene families α, β, γ, δ. In case of vertebrates, only the α-class is known to be present. α-CA exhibit a conserved secondary structure of a ten-stranded, twisted beta-sheet. At the bottom of the 15 Å deep, cone-shaped active site a zinc ion is tetrahedrally coordinated to the nitrogens of three histidine residues and a water molecule/hydroxide ion occupies the forth coordination site. The metal ion binding residues and additional 17 residues are found to be invariant in all sequences of α-CA [28]. Up to now, the crystal structures of eight active isozymes have been determined out of 16 isoforms described in literature [36]. Along the catalytic reaction, residues within the CA binding pocket can undergo conformational changes and single amino acid mutations can develop great impact on the catalytic properties. In order to investigate the effect of minimal but critical changes within the active site we have created a data set of 38 entries from 9 active α-CA isozymes.

7.1.3 Benzamidine

For a first proof-of-concept of the MPCA approach, we analyzed a data set consisting of 87 compounds that belong to a series of selective thrombin inhibitors and were taken from a 3D-QSAR study [6]. The data set is suitable for conducting experiments in a systematic way, as it is quite homogeneous and relatively small (the descriptors contain 47 – 100 points, where each point corresponds to an atom). Moreover, as the 87 compounds all share a common core fragment (which is distributed over two different regions with a variety of substituents), the data set contains a clear and unambiguous target pattern.

7.1.4 Thermolysin

Additionally, we used a data set consisting of 74 structures derived from the Cavbase database. Each structure represents a protein cavity belonging to the protein family of thermolysin, bacterial proteases frequently used in structural protein analysis and annotated with the E.C. number 3.4.24.27 in the ENZYME database. The data set is well-suited for our purpose, as all cavities belong to the same enzyme family and, therefore, evolutionary related, highly conserved substructures ought to be present. On the other hand, with cavities (hypothetical binding pockets) ranging from about 30 to 90 pseudocenters and not all of them being real binding pockets, the data set is also diverse enough to present a real challenge for matching techniques.

7.2 Classification

In our experiments, first we compared our novel method (LPCS) with existing graph-based approaches, namely the random walk (RW) kernel [15], the shortest path (SP) kernel [8], and the method of multiple graph alignment (MGA) recently introduced in [40]. Moreover, we included geometric hashing (GH) as a representative geometrical approach [27]. Given two labeled points clouds as input, all these methods produce a degree of similarity as an output. Yet, for the graph-based approaches, it is of course necessary to transform a point cloud into a graph representation in a preprocessing step. This was done as proposed in [40]:

1. each point is transformed into a node with corresponding node label
2. for each pair of nodes:
   (a) the Euclidean distance between both nodes is calculated
   (b) if the distance is below a certain threshold (here 11 Å to ensure connected graphs), an edge with weight equal to this distance is added

LPCS was realized using an evolution strategy as proposed in Section 5. Its parameterization was optimized with the *sequential parameter optimization toolbox* [4] and was chosen as follows: $\mu = 30, \nu = 4, \rho = 6$, plus-selection, termination criteria: largest step size < 0.00001, intermediate recombination for object and discrete recombination for strategy-component. A comprehensive explanation of the different ES parameters and operators can be found in [5].

The step sizes were initialized in $[5, 15]^3$ and $[1, \pi]^3$, respectively. The SP-kernel is parameter-free, the RW-kernel expects a parameter λ that is set to the largest degree of a node in the data set to ensure a geometric series during calculation, which results in a simpler evaluation [7]. Since the geometric information of real-world data is noisy, we also need a tolerance parameter ϵ to decide whether two edges have equal length (difference $\leq \epsilon$) or not; in our experiments, we used $\epsilon = 0.2$. For MGA, we chose the parameterization proposed in [40].

The implementation of GH, based on [27], involves quite a number of technical details that we cannot all explain here. As suggested by the authors, we used seed patterns (k-tuples) consisting of 5 points and defined a hash key based on the discretized distances between these points. The equals predicate, which is used to decide whether two patterns are approximately equal, was implemented in an error-tolerant way through solving an optimal assignment problem: The corresponding distances are assigned in an optimal and mutually exclusive way, and equal is evaluated as true if at least 80% of the assigned distances differ by at most 0.2 Å .

7.2.1 Results

The results of a leave-one-out cross validation, using the simple 1-NN classifier for prediction, are summarized in Table 2. As can be seen, the kernel-based methods (SP and RW) perform very poorly and are hardly better than random guessing. In terms of accuracy, MGA is much better, though still significantly worse than the geometric approaches, GH and LPCS. The latter performs clearly best on this problem. In fact, LPCS is not only better than GH in terms of performance but also much faster in terms of runtime.

Table 2: Accuracy and runtimes (in seconds with standard deviation, referring to a single comparison) of LPCS ($\alpha = 0.5$, with restarts like described above), MGA, RW, and SP on the NADH/APT data set.

method	accuracy	runtime
MGA	0.7662	121.74 ± 418.02
SP	0.6056	$\mathbf{9.75 \pm 97.77}$
RW	0.5972	65.51 ± 89.07
GH	0.8873	81.71 ± 98.88
LPCS	$\mathbf{0.9352}$	20.04 ± 24.65

Table 3 furthermore shows how the performance of LPCS depends on the choice of the trade-off parameter α in (4). As can be seen, this parameter does indeed have an influence, even though the differences are not extreme. For this data set, α-values around 0.5 yield better results than extreme values close to 0 or 1; the optimal choice would be $\alpha = 0.7$. In practice, α can be considered as a tuning parameter to be adapted to the problem at hand (e.g., by means of a cross-validation on the training data).

Table 3: Accuracy of LPCS for different values of α in (4).

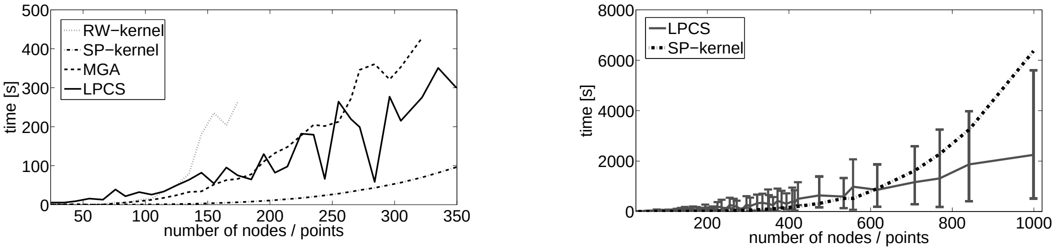
α	accuracy	α	accuracy	α	accuracy
0.0	0.904	0.4	0.927	0.8	0.924
0.1	0.918	0.5	0.935	0.9	0.927
0.2	0.913	0.6	0.935	1.0	0.918
0.3	0.915	0.7	0.938		

7.2.2 Runtime

To investigate the computational complexity of our method, we used the NADH/ATP data set. From this data set we chose protein binding sites of size approximately $s \in \{25, 35, \ldots, 985, 995\}$; this was done by selecting the largest binding site smaller than s and the smallest binding site larger than s.

Again, in addition to our novel approach and MGA, the shortest path and the random walk kernel were included for comparison (especially the SP kernel is known to be fast). Each approach is applied on the protein binding sites mentioned above, and the time for comparing the structures of size s is measured. Since LPCS is based on a stochastic optimizer, we repeated each calculation 10 times and derived the median, minimum and maximum of the runtime.

The results are summarized in Figure 5. Due to their excessive memory requirements, MGA and RW-kernel are not able to compare binding sites exceeding a certain size. For small problems, LPCS has the highest runtime, but the runtime is growing very slowly with the problem size; for point clouds larger than 150 or 200, LPCS is already faster than MGA or RW-kernel. To explain the high variation of the runtime of LPCS, note that we hash the points with equal label to support nearest



(a) runtimes of all methods in the range $[25, 350]$, for LPCS only median was ploted

(b) runtimes of SP-kernel and LPCS (min, median, max) in the range $[25; 1000]$

Figure 5: Runtimes of LPCS, MGA, SP-, and RW-kernel w.r.t. problem size; for RW-kernel and MGA a calculation was posible to a certain size of the problem since the memory requirement was becoming too high

neighbor search. Therefore, the runtime strongly depends on the distribution of the labels, which varies among the data sets: The more uniformly the labels are distributed, the more efficient the search becomes.

The SP-kernel has cubic runtime, so that this method is the most efficient alternative for $s < 600$. LPCS is becoming the most efficient approach for $s > 600$, which is hardly surprising in light of the fact that the dimensionality of the LPCS optimization problem is constant (six parameters have to be optimized) and does not depend on the number of data points. It is true that the size of the point clouds does have an influence on the evaluation of the objective function, which involves a nearest neighbor search for each point. The increase in runtime is at most quadratic, however.

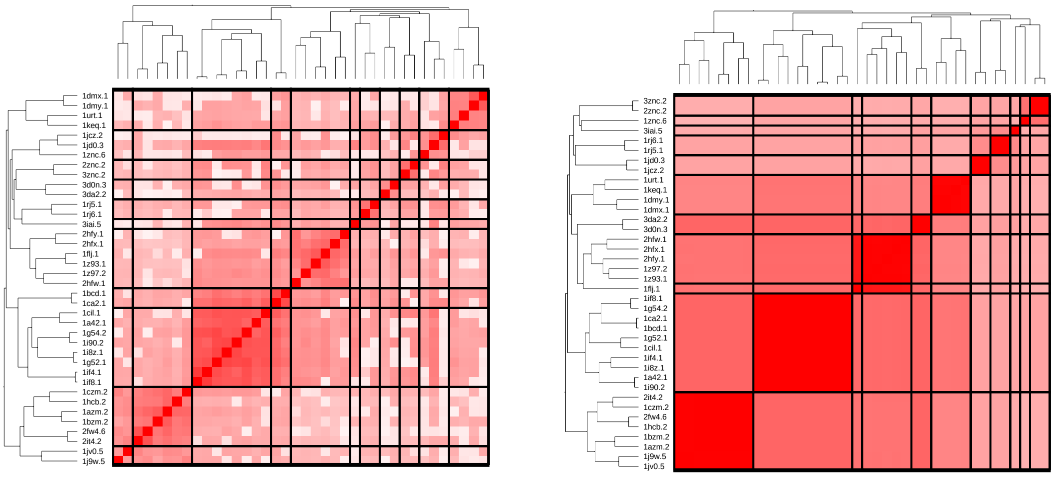
7.3 Clustering

In this study, we compared the cluster structure of α-CA cavities obtained through sequence alignment and LPCS. Protein sequences of the α-CA cavities were mutually aligned using, respectively, FASTA 3.5 [30] in its default settings. An all-against-all comparison of cavities was also performed using LPCS. As a result, two distance matrices were produced, one based on sequence alignment and the other one based on structure comparison. In a second step, a clustering algorithm was applied to the distance matrices. More specifically, we used a partitional clustering method (rbr) in the Cluto package [22]. The number of output clusters, which is a parameter of this method, was set to the expected number of 11.

The overall results of LPCS-based clustering show a clear separation among the different isozymes. Furthermore, CA-I and CA-II are each separated in two clusters respectively (Figure 6(a)). Two CA-I structures (CA-I, Michigan 1), which are grouped in a small cluster, exhibit the amino acid substitution His67Arg. This mutation leads to the coordination of a second zinc ion within the active site, which causes minor conformational changes to several other residues. Due to these modifications within the cavity, the esterase activity of the CA-I Michigan 1 mutant is enhanced toward α- and β- naphthyl acetates [9]. The CA-II isozymes are also split into two clusters reflecting the conformational flexibility of His64. His64 serves as a proton shuttle providing the rate-limiting step of the catalysis. Since, in the considered crystal structures, His64 occurs in two distinct conformations, flexibility of this residue is postulated to be pivotal for the high efficiency of CA II [14]. LPCS is able to detect conformational change of this residue within the active site.

Our α-CA data set comprises two murine (2znc, 3znc) and one human (1znc) crystal structure of CA-IV isozyme. Murine structures are found in a separated cluster, whereas human CA-IV falls into the same cluster as CA-XII. Apparently, the cavity of human CA-IV share more similarity in terms of physico-chemical properties, represented by pseudocenters, in common with the cavity of the human CA-XII than with the murine CA-IV binding site.

A sequence-based clustering also leads to separation of different isozymes (Figure 6(b)). Yet, this approach is not able to detect crucial changes expressed by the exposed physico-chemical proper-



(a) LPCS clusters

(b) FASTA clusters

Figure 6: Cluster structure based on pairwise distances of α-CA cavities using LPCS (left) and FASTA (right).

ties within the active site which are induced, e.g., by a single amino acid mutation or a conformational change of an active site residue. Sequence alignment rather discriminates between structures from different species. For instance, CA-IV and CA-III are each divided in two clusters respectively, since, they comprise either human proteins and structures from mouse (CA-IV: 2znc, 3znc) and rat (CA-III: 1flj).

In summary, LPCS-based clustering not only generates a convincing classification of α-CA but also detects local changes within the active site that cannot be reflected by a sequence based analysis. Regarding clustering results of MGA, SP- and RW-kernel, no reasonable classifications could be achieved. All three approaches group different isozymes together and isozymes of the same type are clustered apart from each other.

7.4 Alignment Quality

In the second study, we compared the quality of the alignments calculated, respectively, by MPCA and MGA.[1] To this end, 100 alignments of size 2 were calculated for randomly chosen structures. The quality of a pairwise alignment $\mathcal{A}$ is evaluated in terms of two criteria. The first criterion is the fraction of assignments of pseudocenters preserving the label information:

$$s_1 = \frac{1}{|\mathcal{A}|} \sum_{(a_1,a_2)\in\mathcal{A}} \left\{ \begin{array}{ll} 1, & \ell(a_1) = \ell(a_2) \\ 0, & \ell(a_1) \neq \ell(a_2) \end{array} \right. ,$$

where $\ell(a_1)$ is the label of the pseudocenter a_1. Similarly, the second criterion evaluates to what extent the geometry of the structures is preserved. Since an MGA does not include information about the position of single pseudocenters, this has to be done by looking at distances between pairs of pseudocenters in each structure:

$$s_2 = \frac{1}{N} \sum_{(a_1,a_2),(b_1,b_2)\in\mathcal{A}} \left\{ \begin{array}{ll} 1, & |d(a_1,b_1) - d(a_2,b_2)| \leq \epsilon \\ 0, & |d(a_1,b_1) - d(a_2,b_2)| > \epsilon \end{array} \right. ,$$

where $d(a_1,b_1) = |c(a_1) - c(b_1)|$ is the distance between the coordinate vectors of a_1 and b_1 and $N = |\mathcal{A}|(|\mathcal{A}|-1)/2$. We summarize the evaluation by the vector

$$\boldsymbol{s} = (s_1, s_2) \in [0,1] \times [0,1] \; .$$

[1] Note that the kernel-methods used in the previous study, SP and RW, produce similarity degrees but not an alignment.

To measure the improvement of our method, we calculate the relative improvement

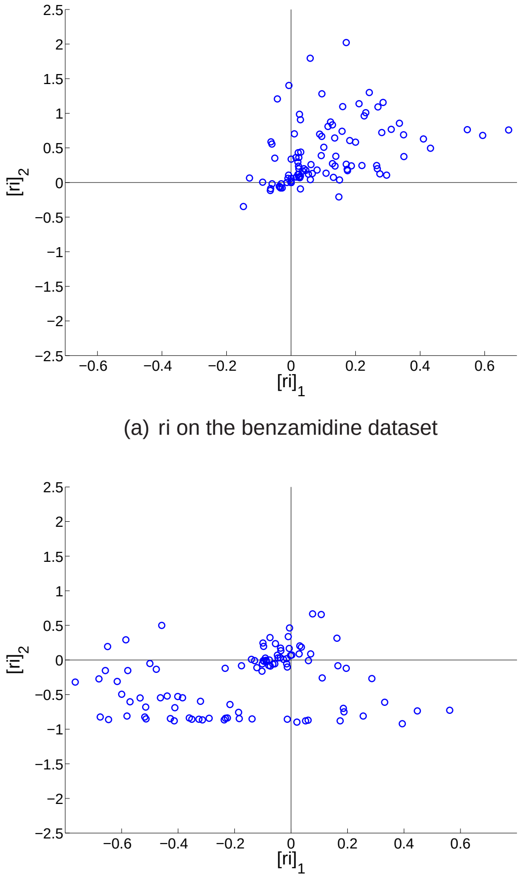
$$\boldsymbol{ri} = \begin{pmatrix} \dfrac{[\boldsymbol{s}_{MPCA}]_1 - [\boldsymbol{s}_{MGA}]_1}{[\boldsymbol{s}_{MGA}]_1} \\[2ex] \dfrac{[\boldsymbol{s}_{MPCA}]_2 - [\boldsymbol{s}_{MGA}]_2}{[\boldsymbol{s}_{MGA}]_2} \end{pmatrix} \tag{7}$$

where $\boldsymbol{s}_{MPCA}$ and $\boldsymbol{s}_{MGA}$ denote, respectively, the evaluations of MPCA and MGA, and$[s]_i$ denotes the i-th element of a vector $\boldsymbol{s}$.

7.4.1 Results

We parameterized MGA as proposed in [40]. For MPCA, we set $k = 6$ and performed experiments like described above. The results for the benzamidine data set are shown in Figure 7 (a), where the relative improvement vectors are plotted. As one can see, most of the ri vectors are lying in the first quadrant, indicating a positive improvement for both criteria.

The corresponding results for the thermolysin data set are depicted in Figure 7 (b). Here, the picture is not as clear, and the number of negative improvements is even slightly higher than the number of positive ones. Apparently, MPCA performs especially good on highly similar structures while not improving on structures that are more diverse. This is hardly surprising, since MPCA strongly exploits information about the geometry of the structures while MGA, as a graph-based approach, is more flexible and can more easily deal with local variations and deformations.



(a) ri on the benzamidine dataset

(b) ri on the thermolysin dataset

Figure 7: Relative improvements (ri) obtained by substituting the MGA approach in MPCA

7.4.2 Parametrization

As an important advantage of MPCA, it deserves mentioning that it only has a single parameter, while MGA has six parameters. In spite of this, we found that it often produces better results, even when trying to parameterize MGA in an optimal way. For example, Figure 8 shows a set of solutions for the benzamidine data that we found by varying the parameters in MPCA and MGA. For ease of exposition, we only plotted the solutions that are Pareto optimal[2] in the two respective sets of solutions; in total, 7776 result vectors $\boldsymbol{s}$ were computed for MGA by varying its 5 parameters in a systematic way. This was done by choosing penalties from -5 to 0 and rewards form 0 to 5 and considering all possible combinations (see [40] for an explanation of these parameters). For MPCA there is only one parameter (threshold k) to be chosen, so that here only 12 results were calculated by considering $k = 0, \ldots, 11$. As one can see in Figure 8, the MPCA solutions are consistently better than the MGA solutions, regardless of the parameterization.

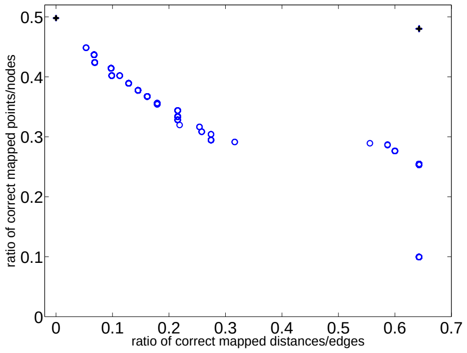


Figure 8: Pareto optimal solutions found by MGA (circles) and MPCA (crosses)

7.5 Structure Retrieval

The focus of the third study is on the ability to detect common substructures in a set of biochemical structures. We randomly selected 100 subsets of c compounds from the benzamidine data set and used MPCA and MGA to calculate an alignment. Then, we checked whether the aforementioned benzamidine core fragment, an amide derivative of benzol which consists of 25 atoms (11 hydrogens), was fully conserved in the alignment, which means that all pseudocenters belonging to the core were mutually assigned in a correct way. For detecting the core fragment we searched for conserved patterns in the alignment and used the parameter $\omega = 1$ and $\xi = 0.9$.

The results, shown in Table 4 for different numbers c, clearly show that MPCA is able to retrieve the core fragment much more reliably than MGA, regardless of the merging technique used.

Table 4: Percent of alignments in which the benzamidine core fragment was fully conserved in the alignment of $c = \{2, 4, 8, 16\}$ structures, and consistency value.

c	2	4	8	16	consist
MGA	0.85	0.38	0.14	0.04	—
MPCA (star)	0.97	0.92	0.80	0.76	0.8686
MPCA (k-partite)	0.93	0.83	0.78	0.67	0.8023
MPCA (tree)	0.97	0.96	0.93	0.90	0.8953

The star and tree-based alignment derive a multiple alignment indirectly, through merging pairwise alignments; *k-partite* obtains the multiple alignment in one step, however, by using a greedy heuristic. Thus, it is natural to ask for the consistency of these methods. To answer this question, we

[2] Given a set of solutions S, a solution $\boldsymbol{s} \in S$ is called Pareto optimal if it is not dominated by any other solution. A solution $\boldsymbol{x}$ dominates another solution $\boldsymbol{y}$ if $\boldsymbol{x}[i] \geq \boldsymbol{y}[i]$ for all i and $\boldsymbol{x}[i] > \boldsymbol{y}[i]$ for at least one i.

compared the pairwise alignments induced by each method, i.e., by the multiple alignment constructed by the method, with the actually optimal pairwise alignments. As a measure of consistency between two pairwise alignments, we used the number of mutually assigned points that are shared by both alignments, divided by the number of such tuples contained in either the first or the second alignment. The average of these consistency degrees over all pairwise alignments is given in Table 4. As can be seen, all techniques perform reasonably well; the tree-based approach even sticks out a bit and achieves the best results.

8 Conclusions

In this paper, we have introduced labeled point cloud superposition (LPCS) as a novel tool for structural bioinformatics, namely as a method for comparing biomolecules on a structural level. The concept of a labeled point cloud, appears to be a quite natural representation for biological structures [33, 34]. In comparison to other approaches, such as the prevalent graph-based methods, the modeling step is hence simplified and does not involve any complex transformations. More importantly, a labeled point cloud preserves the full geometric information and makes it easily accessible to computational procedures.

Taking advantage of suitable fuzzy logic-based modeling techniques, we have defined a reasonable measure of similarity between two labeled point clouds. A labeled point cloud superposition is then defined in terms of a spatial transformation that maximizes this degree of similarity. Like for related problems in bioinformatics, such as sequence alignment, the computation of the similarity between two objects hence involves the solution of an optimization problem. To this end, we have proposed the use of an evolution strategy, an approach from the family of evolutionary algorithms, which appears to be especially suitable for this problem.

First experimental results with classification data are quite promising and suggest that our approach is able to compare protein binding sites in a reasonable way. In terms of classification accuracy, LPCS turned out to be significantly better than existing (graph-based) methods used for comparison. Moreover, even though it is computationally more complex than these methods for small data sets, it scales much better and becomes more efficient for larger data sets. This is due to the fact that, in contrast to graph-based methods, the search space does not depend on the size of the point clouds and remains low-dimensional. LPCS also compares favorably with geometric hashing, another geometric approach to structure comparison, both in terms of performance and runtime. Besides, it is conceptually simpler, with less parameters that need to be defined by the user.

Drawing on the method of labeled point cloud superposition, we furthermore proposed the concept of a multiple point cloud alignment which establishes a one-to-one correspondence between the points from different structures and, therefore, can be seen as a structural counterpart to sequence alignment. Again, first experiments carried out in the context of protein binding site comparison are quite promising and show that our method is competitive, if not even superior, to state-of-the-art graph-based methods for multiple structure alignment.

Still, as mentioned previously, graph-based approaches may have advantages in situations where a high degree of flexibility is needed. As future work, we therefore plan to develop hybrid approaches combining the advantages of both geometrical and graph-based methods.

References

[1] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete and Computational Geometry*, 3:237–256, 1987.

[2] F. R. Bach. Graph kernels between point clouds. In *International Conference on Machine Learning*, pages 25–32, Helsinki, Finland, 2008.

[3] O. Bachar, D. Fischer, R. Nussinov, and H. Wolfson. A computer vision based technique for 3-d sequence-independent structural comparison of proteins. *Protein Engineering*, 6(3):279–287, 1993.

[4] T. Bartz-Beielstein. *Experimental research in evolutionary computation: The new experimentalism*. Springer, 2006.

[5] H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[6] M. Böhm, J. Stürzebecher, and G. Klebe. Three-dimensional quantitative structure-activity relationship analyses using comparative molecular field analysis and comparative molecular similarity indices analysis to elucidate selectivity differences of inhibitors binding to trypsin, thrombin, and factor xa. *Journal of Medicinal Chemistry*, 42(3):458–477, 1999.

[7] K. M. Borgwardt. *Graph Kernels*. PhD thesis, Ludwig-Maximilians-Universit ät München, Germany, 2007.

[8] K. M. Borgwardt and H. P. Kriegel. Shortest-path kernels on graphs. In *International Conference on Data Mining*, pages 74–81, Houston, Texas, 2005.

[9] F. Briganti, S. Mangani, P. Orioli, A. Scozzafava, G. Vernaglione, and C. T. Supuran. Carbonic anhydrase activators: X-ray crystallographic and spectroscopic investigations for the interaction of isozymes I and II with histamine. *Biochemistry*, 36(34):10384–10392, 1997.

[10] H. Bunke and X. Jiang. Graph matching and similarity. *Intelligent systems and interfaces*, 15:281 – 304, 2000.

[11] H. Bunke, X. Jiang, and A. Kandel. On the Minimum Common Supergraph of two Graphs. *Computing*, 65(1):13–25, 2000.

[12] W.J. Christmas, J. Kittler, and M. Petrou. Structural Matching in Computer Vision using Probabilistic Relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.

[13] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, New York, 2000.

[14] M. Ferraroni, S. Tilli, F. Briganti, W. R. Chegwidden, C. T. Supuran, K. E. Wiebauer, R. E. Tashian, and A. Scozzafava. Crystal structure of a zinc-activated variant of human carbonic anhydrase I, CA I Michigan 1: evidence for a second zinc binding site involving arginine coordination. *Biochemistry*, 41(20):6237–6244, 2002.

[15] T. Gärtner. *Kernels for structured data*. World Scientific, Singapore, 2008.

[16] J. Gasteiger and T. Engel. *Chemoinformatics*. Wiley-Vch, Weinheim, 2003.

[17] M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky. Practical methods for approximate geometric pattern matching under rigid motions. In *Annual Symposium on Computational Geometry*, pages 103 – 112, Stony Brook, New York, United States, 1994.

[18] 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*, 15:359–363, 1997.

[19] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg. On dynamic voronoi diagrams and the minimum hausdorff distance for point sets under euclidean motion in the plane. In *Proceedings of the eighth annual symposium on Computational geometry*, pages 110 – 119, Berlin, Germany, 1992.

[20] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of voronoi surfaces and its applications. *Discrete and Computational Geometry*, 9(1):267–291, 1993.

[21] W. Kabsch. A solution of the best rotation to relate two sets of vectors. *Acta Crystallographica*, 32:922–923, 1976.

[22] G. Karypis. CLUTO - family of data clustering software tools v 2.1.1. http://glaros.dtc.umn.edu/gkhome/views/cluto, 2006.

[23] EP. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers, 2002.

[24] H.W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics*, 52(1):7–21, 2005.

[25] Y. Lamdan and HJ Wolfson. Geometric Hashing: A General And Efficient Model-based Recognition Scheme. In *Second International Conference on Computer Vision*, pages 238–249, 1988.

[26] N. Leibowitz, Z. Y. Fligelman, R. Nussinov, and H. J. Wolfson. Multiple structural alignment and core detection by geometric hashing. In *International Conference on Intelligent Systems for Molecular Biology*, pages 169–177, Heidelberg, Germany, 1999.

[27] N. Leibowitz, R. Nussinov, and H.J. Wolfson. MUSTA-A General, Efficient, Automated Method for Multiple Structure Alignment and Detection of Common Motifs: Application to Proteins. *Journal of Computational Biology*, 8(2):93–121, 2001.

[28] S. Lindskog. Structure and mechanism of carbonic anhydrase. *Pharmacology & Therapeutics*, 74(1):1–20, 1997.

[29] F. Mémoli and G. Sapiro. Comparing point clouds. In *Eurographics / ACM SIGGRAPH symposium on Geometry processing*, pages 32–40, Nice, France, 2004.

[30] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 85(8):2444–2448, 1988.

[31] J. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16(7):521–533, 2002.

[32] 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*, 323(2):387–406, 2002.

[33] M. Shatsky, R. Niussinov, and H. J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins: Structure, Function, and Bioinformatics*, 56:143–156, 2004.

[34] M. Shatsky, A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson. The multiple common point set problem and its application to molecule binding pattern detection. *Journal of Computational Biology*, 13(2):407–428, 2006.

[35] P. N. Suganthan, E. Teoh, and D. Mital. Pattern recognition by graph matching using the potts MFT neural networks. *Pattern Recognition*, 28(7):997–1009, 1995.

[36] C. T. Supuran and A. Scozzafava. Carbonic anhydrases as targets for medicinal chemistry. *Bioorganic & Medicinal Chemistry*, 15(13):4336–4350, 2007.

[37] Y. Wang and N. Ishii. Method of similarity metrics for structured representations. *Expert Systems with Applications*, 12:89–100, 1997.

[38] A. Weber, A. Casini, A. Heine, D. Kuhn, C.T. Supuran, A. Scozzafava, and G. Klebe. Unexpected Nanomolar Inhibition of Carbonic Anhydrase by COX-2-selective Celecoxib: New Pharmacological Opportunities due to Related Binding Site Recognition. *Journal of Medical Chemistry*, 47(3):550–557, 2004.

[39] N. Weskamp, E. Hüllermeier, and G. Klebe. Merging Chemical and Biological Space: Structural Mapping of Enzyme Binding Pocket Space. *Proteins*, 76(2):317–330, 2009. major rev.

[40] N. Weskamp, E. Hüllermeier, D. Kuhn, and G. Klebe. Multiple graph alignment for the structural analysis of protein active sites. *IEEE Transactions on Computational Biology and Bioinformatics*, 4(2):310–320, 2007.

[41] T. J. Wheeler and J. D. Kececioglu. Multiple alignment by aligning alignments. *Bioinformatics*, 23(13):i559–i568, 2007.

[42] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. IEEE Computational Science and Engineering, 1997.

[43] L. Xu and E. Oja. Improved Simulated Annealing, Boltzmann Machine, and Attributed Graph Matching. In *EURASIP Workshop on Neural Networks*, pages 151–160. Springer-Verlag London, UK, 1990.

[44] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183 – 190, 1988.

[45] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computing and Mathematics with Applications*, 9:149–184, 1983.