2009 Ninth International Conference on Intelligent Systems Design and Applications

Efficient Construction of Multiple Geometrical Alignments for the Comparison of Protein Binding Sites

Thomas Fober[1], Gerhard Klebe[2], and Eyke Hüllermeier[1]
[1] *Department of Mathematics & Computer Science*
{thomas, eyke}@mathematik.uni-marburg.de
[2] *Department of Pharmaceutical Chemistry*
klebe@mailer.uni-marburg.de
Philipps-Universität, Marburg, Germany

***Abstract*—We proceed from a method for protein structure comparison in which information about the geometry and physico-chemical properties of such structures are represented in the form of *labeled point clouds*, that is, a set of labeled points in three-dimensional Euclidean space. Two point clouds are then compared by computing an optimal spatial superposition. This approach has recently been introduced in the literature and was shown to produce very good similarity scores. It does not, however, establish an *alignment* in the sense of a one-to-one correspondence between the basic units of two or more protein structures. From a biological point of view, alignments of this kind are of great interest, as they offer important information about evolution, heredity, and the mutual correspondence between molecular constituents. In this paper, we therefore developed a method for computing pairwise or multiple alignments of protein structures on the basis of labeled point cloud superpositions.**

***Keywords*-protein binding sites; labeled point clouds; alignment; conserved patterns;**

I. Introduction

Geometric representations of objects in the form of point sets in three-dimensional Euclidean space can be found in many fields, including structural bioinformatics. A well-known example of a representation of this kind is the *Molfile* format [6], 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 [4], we therefore introduced 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. Based on this representation, the method of *labeled point cloud superposition* (LPCS) has been developed [4], which, by finding an optimal spatial superposition, allows for the computation of a similarity score between two objects.

LPCS was shown to produce very good similarity scores in the context of comparing protein binding sites. Besides, it has a number of advantages over alternative methods that are commonly employed for protein structure comparison [11, 9, 13, 8, 1], such as graph-based approaches. In particular, LPCS is quite efficient from a computational point of view. Yet, in contrast to methods for *multiple graph alignment* as recently introduced in [14], LPCS does not establish a one-to-one correspondence between the basic units of two or more protein structures. From a biological point of view, alignments of this kind are of great interest, as they offer important information about evolution, heredity, and the mutual correspondence between molecular constituents. In this paper, we therefore develop a method for computing pairwise or multiple alignments of protein structures on the basis of labeled point cloud superpositions.

The remainder of the paper if organized as follows. Subsequent to a brief introduction to protein binding sites and their representation in Section II, we introduce the concept of multiple geometric alignment in Section III. Section IV is devoted to the experimental validation of the approach, and Section V concludes the paper.

II. Modeling Protein Binding Sites

In this paper, our special interest concerns the modeling of protein binding sites. More specifically, our work builds upon CavBase [12], 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 113,718 hypothetical binding sites that have been extracted from 23,780 publicly available protein structures using the LIGSITE algorithm [7].

The geometrical arrangement of the pocket and its physicochemical properties are first represented by predefined *pseudocenters* – spatial points that represent the 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

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.210

rules [12]. 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 physicochemical properties. Obviously, just like in the case of Molfile, this representation is already in the form of a labeled point cloud: pseudocenters are given with their coordinates and labels, so that no further transformation is needed.

III. Multiple Geometrical Alignment

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 physicochemical 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 physicochemical 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.

In this paper, we derive alignments from labeled point cloud superpositions and, therefore, refer to the latter as *geometric alignments*. Formally, a labeled point cloud P is a set of points $\{p_1, \ldots, p_n\}$ with two associated functions: $l_c : P \to \mathbb{R}^3$ maps points to coordinates in the Euclidean space, and $l_t : P \to \mathcal{L}$ assigns a label to each point.

Definition 1 (Multiple Geometrical Alignment): Let $\mathcal{P}$ be a set of m point clouds $P_i = \{p_1^i, \ldots, p_{n_i}^i\}$, $i = 1, \ldots, m$. A multiple geometrical alignment 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 $i = 1 \ldots m$ 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 in which mutually assigned points have the same label and nearby spatial positions.

A. Construction of pairwise alignments

To construct a pairwise alignment of two point clouds P_1 and P_2, we reduce the alignment problem to a problem of optimal assignment. 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 our LPCS method. Roughly speaking, this method searches for a superposition which, as far as possible, guarantees the following property: For each point in one structure, there exists a point in the other cloud which is spatially close and has the same label. To this end, P_1 is held fix while P_2 is moved via a translation vector $\delta = (\delta_1, \delta_2, \delta_3) \in \mathbb{R}^2$ (which means that δ is added to each point $p \in P_2$) and rotated by three angles θ_1, θ_2, and θ_3 (the label information is of course left unchanged). The quality of a spatial superposition is specified by means of a proper measure, taking into account both label and distance information, and this objective function is maximized using an evolution strategy.

Given an optimal spatial superposition, it makes sense to define $m_{i,j}$ by the distance between point $p_i \in P_1$ and $p_j \in P_2$ in the superimposed point clouds. 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 I 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 $E = \{\{u, v\} \mid u \in V_1,\, v \in V_2\}$. The problem is to find a subset of edges $M \subseteq E$ such that $e \cap e' = \emptyset$ for all $e, e' \in M$ (i.e., one point has exactly one mapping partner),

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

Table I
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

and

$$\sum_{e \in M} c(e) \quad \rightarrow \quad \min,$$

where $c(e)$ is the cost associated with edge e. In our case, the sets V_1 and V_2 represent, respectively, the points in point cloud P_1 with additional $|P_2|$ dummy points and the points in cloud P_2 with additional $|P_1|$ dummy points. Moreover, the costs $c(e)$ are given by the corresponding matrix entries $m_{i,j}$. See Figure 1 for an illustration.

To solve the weighted bipartite matching problem, we use the Hungarian algorithm [10] that needs time $\mathcal{O}(n^3)$. Once a cost-minimal assignment has been found, the geometric alignment is defined by the corresponding node-to-node and node-to-dummy assignments, while dummy-to-dummy assignments are ignored.

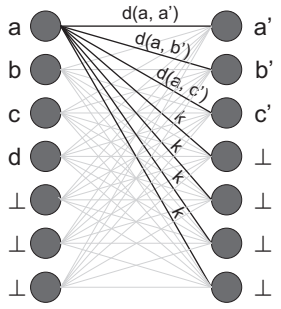


Figure 1. Illustration of the weighted bipartite graph matching problem.

B. Construction of Multiple Alignments

Pairwise alignments can be used, for example, to derive a measure of similarity between two objects. From a biological point of view, however, it is even more interesting to look for a *multiple* alignment, that is, the simultaneous alignment of a set of $m > 2$ structures. Alignments of this type are of interest, for example, to discover conserved patterns in a family of evolutionary related proteins.

To derive a multiple geometrical alignment (3DA) of m point clouds, we resort to the star alignment approach [14]: 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 $p_j \in P_1$ in the alignment of P_1 and P_i, then a single assignment in the multiple alignment is of the form

$$(p_j, p_{2j}, p_{3j}, \ldots, p_{mj}).$$

Since the quality of a multiple alignment 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.

IV. EXPERIMENTAL RESULTS

In our experimental study, we compare the method of multiple geometrical alignment as introduced above (3DA) with the method of multiple graph alignment (MGA) proposed in [14]. Roughly speaking, we thus compare a geometrical with a graph-based approach to aligning protein binding sites.

A. Data Sets

For a first proof-of-concept, 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 [2]. The data set is suitable for conducting experiments in a systematic way, as it is quite homogeneous and relatively small (the graph descriptors contain 47 - 100 nodes, where each node 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.

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 graph matching techniques.

B. Alignment Quality

In the first study, we compared the quality of the alignments calculated, respectively, by 3DA and MGA. To this end, 100 graph alignments of size 2 were calculated for randomly chosen structures. Restricting to pairwise alignments is justified since both 3DA and MGA use the star alignment procedure to derive multiple alignments. 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:

$$s_1 = \frac{1}{|\mathcal{A}|} \sum_{(a_1,a_2) \in \mathcal{A}} \begin{cases} 1, & l_t(a_1) = l_t(a_2)) \\ 0, & l_t(a_1) \neq l_t(a_2)) \end{cases},$$

where $l_t(a_1)$ 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 psedocenters, this has to be done by looking at distances between pairs of pseudocenters in each structure:

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

where $d(a_1,b_1) = |l_c(a_1) - l_c(b_1)|$ and $N = |\mathcal{A}|(|\mathcal{A}|-1)/2$. We summarize the evaluation by the vector

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

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

$$\vec{ri} = \begin{pmatrix} \dfrac{[\vec{s}_{3DA}]_1 - [\vec{s}_{MGA}]_1}{[\vec{s}_{MGA}]_1} \\ \dfrac{[\vec{s}_{3DA}]_2 - [\vec{s}_{MGA}]_2}{[\vec{s}_{MGA}]_2} \end{pmatrix} \qquad (1)$$

where $\vec{s}_{3DA}$ and $\vec{s}_{MGA}$ denote, respectively, the evaluations of 3DA and MGA and where $[s]_i$ gives the i-th element of a vector $\vec{s}$.

1) Results: For our calculations we parameterized MGA as proposed in [14], for 3DA we set $k = 6$ and performed experiments like described above. The results for the benzamidine data set are shown in Figure 2, 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.

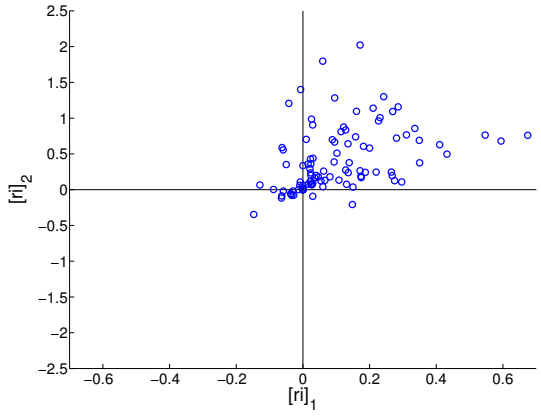


Figure 2. relative improvement (ri) on the benzamidine dataset

The corresponding results for the thermolysin data set are depicted in Figure 3. Here, the picture is not as clear, and the number of negative improvements is even slightly higher than the number of positive ones. Apparently, 3DA performs especially good on highly similar structures while not improving on structures that are more diverse. This is hardly surprising, since 3DA strongly exploits information about the geometry of the structures.

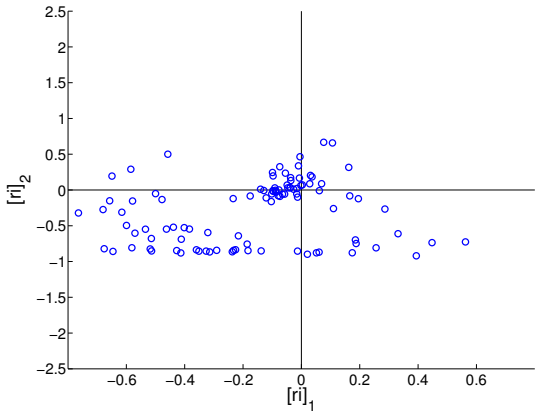


Figure 3. relative improvement (ri) on the thermolysin dataset

C. Parametrization

As an important advantage of 3DA, it deserves mentioning that it only has a single parameter, while MGA has six parameters. In spite of this, we found that if often produces better results, even when trying to parametrize MGA in an optimal way. For example, Figure 4 shows a set of solutions for the benzamidine data that we found by varying the parameters in 3DA and MGA. For ease of exposition, we only plotted the solutions that are Pareto optimal in the two respective sets of solutions; in total, 7776 result vectors $\vec{s}$ were computed for MGA by variation its 5 parameters in a systematic way. For 3DA there was only one parameter (threshold k) to vary, so that here only 12 results were calculated. To have a readable plot we removed results that are not Pareto optimal[1] and plot only the remaining Pareto optimal points. The resulting plot is illustrated in figure 4. As one can see the 3DA solutions were independent of parameterization always better than the MGA results, so that we can claim that our novel method is easy to adjust and will lead to results that are better, even for an optimal adjusted MGA approach.

D. Structure Retrieval

The focus of the second 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 3DA and MGA to calculate an alignment. Then, we checked whether the aforementioned benzamidine core fragment, an amide derivative of benzol

[1]Given a set of results S only such results $\vec{s} \in S$ are called Pareto optimal that are not dominated by other solutions. A vector $\vec{x}$ dominates another vector $\vec{y}$ if $\vec{x}[i] \geq \vec{y}[i]$ for all i and $\vec{x}[i] > \vec{y}[i]$ for some i.

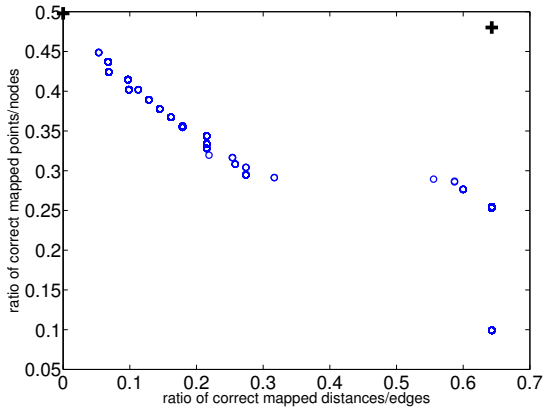


Figure 4. Pareto optimal solutions found by MGA (circles) and 3DA (crosses)

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. The results, shown in Table II for different numbers c, clearly show that 3DA is able to retrieve the core fragment much more reliably than MGA.

Table II
PERCENT OF ALIGNMENTS IN WHICH THE BENZAMIDINE CORE FRAGMENT WAS FULLY CONSERVED IN THE ALIGNMENT OF $c = \{2, 4, 8, 16\}$ STRUCTURES.

c	2	4	8	16
MGA	0.85	0.38	0.14	0.04
3DA	0.96	0.92	0.80	0.76

E. Runtime

To investigate the computational complexity of our method, we used the NADH/ATP data set [4] consisting of a large set of protein binding sites. From this 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. In addition to our novel approach and MGA, two other approaches were included for comparison, namely the shortest path (SP) and the random walk (RW) kernel [3, 5]. Both approaches yield similarity scores (though do not determine an alignment), and 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 3DA 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 Fig. 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, 3DA has the highest runtime, but the runtime is growing very slowly with the problem size; for point clouds larger than 150 or 200, 3DA is already faster than MGA or RW-kernel. To explain the high variation of the runtime of 3DA, note that we hash the points with equal label to support nearest 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 $s < 600$. 3DA is becoming the most efficient approach for $s > 600$, which is hardly surprising in light of the fact that the dimensionality of the 3DA 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.

V. CONCLUSIONS

In this paper, we proposed an extension of the method of labeled point cloud superposition (LPCS). Originally, LPCS computes an optimal spatial superposition of two labeled point clouds but does not establish a one-to-one correspondence between the points. Motivated by applications in structural bioinformatics, we developed the method of multiple geometric alignment which, based on a given superposition, computes a correspondence of this type. First experiments carried out in the context of protein structure 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. Besides, it was already shown in [4] that LPCS is computationally more efficient than the graph-based approach. All things considered, multiple geometric alignment is therefore a viable option for protein structure comparison and might even be of interest beyond the field of structural bioinformatics.

REFERENCES

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

[2] 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.

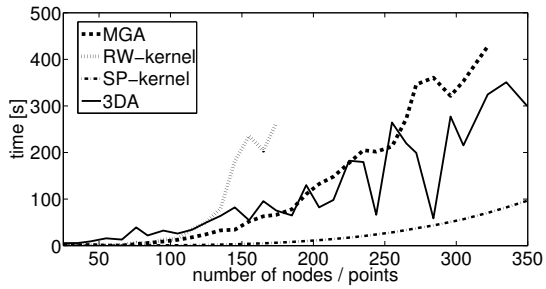
[3] K. M. Borgwardt and H. P. Kriegel. Shortest-path kernels on graphs. In *International Conference on Data Mining*, pages 74–81, Houston, Texas, 2005.

[4] Thomas Fober and Eyke Hüllermeier. Fuzzy modeling of labeled point cloud superposition for the comparison of protein binding sites. In *IFSA World Congress, EUSFLAT World Conference*, 2009.

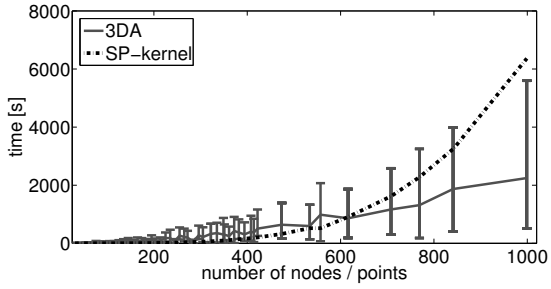
[5] Thomas Gärtner. *Kernels for structured data*. World Scientific, Singapore, 2008.

[6] Johann Gasteiger and Thomas Engel. *Chemoinformatics*. Wiley-Vch, Weinheim, 2003.

[7] 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.



(a) runtimes of all methods in the range $[25, 350]$, for 3DA only median was ploted



(b) runtimes of SP-kernel and 3DA (min, median, max) in the range $[25; 1000]$

Figure 5. Runtimes of 3DA, 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

[8] 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*, 52(2):137–145, 2003.

[9] K. Kinoshita and H. Nakamura. Identification of Protein Biochemical Functions by Similarity Search using the Molecular Surface Database eF-site. *Protein Science*, 12(8):1589–1595, 2003.

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

[11] A. R. Ortiz, C. E. M. Strauss, and O. Olmea. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, 11(11):2606–2621, 2002.

[12] 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.

[13] R.V. Spriggs, P.J. Artymiuk, and P. Willett. Searching for Patterns of Amino Acids in 3D Protein Structures. *Journal of Chemical Information and Computer Sciences*, 43(2):412–421, 2003.

[14] 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.