Vorolign—fast structural alignment using Voronoi contacts

Fabian Birzele*, Jan E. Gewehr, Gergely Csaba and Ralf Zimmer
Practical Informatics and Bioinformatics Group, Department of Informatics, Ludwig-Maximilians-University, Amalienstr. 17, D-80333 Munich, Germany

ABSTRACT
Summary: Vorolign, a fast and flexible structural alignment method for two or more protein structures is introduced. The method aligns protein structures using double dynamic programming and measures the similarity of two residues based on the evolutionary conservation of their corresponding Voronoi-contacts in the protein structure. This similarity function allows aligning protein structures even in cases where structural flexibilities exist. Multiple structural alignments are generated from a set of pairwise alignments using a consistency-based, progressive multiple alignment strategy.

Results: The performance of Vorolign is evaluated for different applications of protein structure comparison, including automatic family detection as well as pairwise and multiple structure alignment. Vorolign accurately detects the correct family, superfamily or fold of a protein with respect to the SCOP classification on a set of difficult target structures. A scan against a database of >4000 proteins takes on average 1 min per target. The performance of Vorolign in calculating pairwise and multiple alignments is found to be comparable with other pairwise and multiple protein structure alignment methods.

Availability: Vorolign is freely available for academic users as a web server at http://www.bio.ifi.lmu.de/Vorolign
Contact: fabian.birzele@ifi.lmu.de
Supplementary information: Datasets used throughout the article are available at http://www.bio.ifi.lmu.de/Vorolign/supplement.html

1 INTRODUCTION
Protein structure comparison is an essential step to gain a deeper understanding of the interplay of protein sequence and structure evolution to automatically classify proteins into families and to identify similarities that cannot be detected on the sequence level. The problem has been heavily investigated in computational biology for more than two decades and numerous approaches have been developed to address different aspects of the problem. The methods proposed differ in the representation of protein structures in the algorithm, the procedure to identify structurally equivalent residues, the approach to combine similar regions to an alignment, as well as in the treatment of protein structures, either rigid or flexible. Also, while some of the methods address the problem of pairwise protein structure alignment, others align multiple structures. Multiple structure alignment is becoming more and more important with the growing number of protein structures in the PDB (Berman et al., 2000), for example, in order to identify common structural cores of protein families.

Among the well-known methods for pairwise comparison of protein structures are Dali (Holm and Sander, 1993) and CE (Shindyalov and Bourne, 1998), both treating proteins as rigid bodies, and FATCAT (Ye and Godzik, 2003) that is able to align protein structures in a flexible way. Methods for multiple structure alignment are, for example, MultiProt (Shatsky et al., 2004) or POSA (Ye and Godzik, 2005).

The focus of the different methods varies greatly, also owing to the fact that it is not clear how exactly an ‘optimal’ solution to the structure comparison problem has to be defined (Ye and Godzik, 2005). Further, the ‘best’ structural alignment of two or more structures is hard to identify. For rigid body alignment algorithms, there is always a trade-off between the number of residues that can be aligned below a distance threshold (so called number of equivalent residues, N_e) and the overall root mean square deviation (RMSD) of the superimposition of the aligned parts of the structures.

Proteins are dynamic entities and in many cases protein flexibility is essential for protein function. Therefore, treating proteins as flexible instead of rigid can be an interesting and important feature in order to detect similarities between protein structures in different conformational states or between moved domains in multi-domain proteins. On the other hand, allowing for too many flexibilities in the protein chain and thus ignoring the overall structure of the protein can lead to an overestimation of the structural similarity.

In this article, we propose a fast method to flexibly align two or more protein structures. The method is based on the assumption that the environments of two structurally equivalent residues are similar owing to positive selection in order to ensure the structural integrity of the protein. We measure the similarity of the structural environments of two residues by their evolutionary relationship with respect to amino acid and secondary structure exchange scores and use this similarity function to align two protein structures using dynamic programming (Needleman and Wunsch, 1970). In contrast to other protein structure alignment methods that use mostly geometry-based similarity measurements, we take the protein structure only implicitly into account, via the network of neighbouring residues (3D contacts), allowing a larger degree of flexibility of the protein structures being compared. In addition, the more sequence-based similarity scoring function avoids certain common artifacts known from structural alignments where evolutionary equivalent residues are not aligned owing to divergence of the structures (Fig. 1).

To represent the biochemical environment of a residue, we use its nearest-neighbours residues as defined by the Voronoi tessellation (Voronoi, 1908) of the protein structure. Voronoi tessellation has been found to be useful for several tasks in structural bioinformatics including packing analysis (Richards, 1974), protein folding (Gan et al., 2001) and structure comparison (Roach et al., 2005; Ilyin et al., 2004).

In our case, the major benefit of using the Voronoi tessellation, in contrast to distance-based contact definitions, is 2-fold. First, we obtain a well-defined set of nearest-neighbour contacts of a residue.
containing those residues that share a common face in the Voronoi diagram with the residue under consideration. Second, the contact set implicitly takes the geometry of the residue environment into account, since residues do not only have to be close in space but must also be direct neighbours without other residues between them.

The performance of our method, called Vorolign, is evaluated for various applications of protein structure comparison including pairwise and multiple structure alignment as well as the automated assignment of a protein to its corresponding protein family on a comprehensive set of difficult examples, derived from the difference set between the ASTRAL (Brenner et al., 2000) versions 1.67 and 1.65.

2 METHODS

2.1 Voronoi tessellation of protein structures

We use the $C_{\beta}$ atoms (for Glycin the $C_{\alpha}$ atom) of the protein as the input set of points in the Euclidean space for the computation of the Voronoi decomposition. This decomposition partitions the space into convex polyhedra, called Voronoi cells. Each residue cell contains by definition all points that are closer to the corresponding $C_{\beta}$ atom than to all other input nodes. All polyhedra, which are direct neighbours in space, share a common face in the Voronoi decomposition corresponding to a contact of the two input points and, consequently, the two residues.

The Voronoi decomposition of a set of input points can be computed efficiently via standard computational geometry algorithms (O’Rourke, 1993). We use the quickhull algorithm as implemented in the QHULL program (Barber et al., 1996), to obtain the Voronoi tessellation of a given protein structure.

There is a problem, though, with a straightforward application of the standard procedures to obtain the Voronoi decomposition of proteins. Residues especially on the surface of the protein will get unbounded, infinite Voronoi polyhedra and, which is even more unfavourable in our case, can share common faces with residues that are very distant in space. This effect can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft. Such effects can also be observed if the protein structure contains a larger cleft.

2.2 Properties of Voronoi cells

Our structural alignment routine employs the dynamic programming algorithm usually used to compare protein sequences (Needleman and Wunsch, 1970) to align two protein structures. Standard sequence alignment routines use an amino acid exchange scoring matrix, like PAM (Dayhoff et al., 1978), to determine the similarity of two residues. We calculate the evolutionary similarity of two residues based on features of their corresponding Voronoi cells.

A priori, one could think of several, possibly conserved, properties that could be used to calculate the similarity of two cells. Examples are geometrical features like volume, shape or the surface area, biochemical properties of the faces or the nearest-neighbours of a cell, i.e. residues whose cells share a common face with the cell under consideration.

In this study we compute the evolutionary conservation of two Voronoi cells, i.e. their similarity, by using their nearest-neighbour environments and therefore the conservation of Voronoi contacts. Those features implicitly take the structure of the protein into account since they are derived from a structure-based process, i.e. the Voronoi tessellation. Nevertheless, they are sufficiently general to allow a certain degree of flexibility in the two protein structures being compared (see also Section 3.2.3). This is an important feature to detect similarities across more diverse protein structures and for the comparison of multi-domain proteins with domain movements.

2.3 Similarity of Voronoi contacts

Given two proteins $X = x_1 x_2 \cdots x_N$ and $Y = y_1 y_2 \cdots y_N$ for which we want to calculate a structural alignment. The set of nearest-neighbours of a residue $x_i$, as defined by the Voronoi tessellation, is denoted as $N(x_i) = \{x_{i-1}, x_i, x_{i+1}, \ldots, x_N\}$. In order to measure the similarity of two residues $x_i$ and $y_j$ we will calculate the similarity of their corresponding nearest-neighbour sets $N(x_i)$ and $N(y_j)$.

2.3.1 Similarity of two nearest-neighbours

As a first step, we need to define the similarity of two residues $x_i$ and $y_j$ in the nearest-neighbour sets $N(x_i)$ and $N(y_j)$. This similarity will be scored by the weighted sum of two scores as given in the equation below:

$$\text{Sim}(x_i, y_j) = \omega_{A} \text{AA}(x_i, y_j) + \omega_{S} \text{SSE}(x_i, y_j).$$ (1)

In the equation, $\text{AA}(x_i, y_j)$ corresponds to an amino acid exchange matrix score and $\text{SSE}(x_i, y_j)$ scores the similarity of the corresponding secondary structure elements as defined by a secondary structure scoring matrix. The incorporation of the SSE-term avoids biasing the alignment towards equivalencing residues with conserved sequence environments. The combination of both scores leads to better results than achieved by any of the scores alone (data not shown).

We also tested the incorporation of other features like the Euclidean or sequential distance of the two residues to their central residue, the Euclidean distance of $x_i$ and $y_j$ with respect to a reference frame (Pennec and Ayache, 1994) as well as PSI-BLAST profiles (Altschul et al., 1997), into the similarity scoring function. This did not lead to a significant change in the alignment accuracy indicating that our two features are sufficient to characterize the evolutionary relationship of the nearest-neighbour environments for our needs.

2.3.2 Similarity of nearest-neighbour sets

Having defined the similarity of two nearest-neighbour residues, we estimate the similarity of the two nearest-neighbour sets $N(x_i)$ and $N(y_j)$. For this we need a matching of the residues in the $N(x_i)$ and $N(y_j)$ sets. Once we have found such a correspondence, the final score (similarity) of the two nearest-neighbour
environments is simply given by the sum of the similarities of the residues matched onto each other minus a penalty score for each unmatched residue.

In principle such a matching of the two nearest-neighbour sets could be calculated by different methods to solve matching problems in bipartite graphs like the Hungarian algorithm (Kuhn, 1955). But since the possible matchings of the neighbours onto each other are constrained by their order in the protein sequence, the optimal solution can efficiently be computed using dynamic programming. The position of $x_i$ (or $y_j$ respectively) is taken into account using two independent sets of nearest-neighbours, one containing all neighbours that are found to the left of the residue under consideration ($x_i$ or $y_j$) in the protein sequence, the other set containing all residues found to the right of the residue in the sequence.

The similarity of two nearest-neighbour sets can be computed using dynamic programming with respect to the similarity function of two nearest-neighbour residues given in Equation (1) and an additional penalty for unmatched residues $p_u$. This corresponds to calculating a global alignment with linear gap costs of the residues in the two nearest-neighbour sets. The entries $S(k,l)$ in the dynamic programming matrix $S$ are filled using the following equation:

$$S(k,l) = \max \left\{ S(k-1,l-1) + \text{Sim}(x_k,y_l), \frac{S(k-1,l) - p_u}{S(k,l-1) - p_u}, \frac{S(k,l-1) - p_u}{S(k,l)} \right\}$$

The final score of that alignment, i.e. the entry in the last row and column of the matrix $S$, is used to score the similarity of the two environments of the residues $x_k$ and $y_l$ and is denoted as $\text{Sim}(S(x_k),S(y_l))$ in the following.

### 2.4 Pairwise alignment of protein structures

Having defined a function to measure the similarity of two residues $x_k$ and $y_l$, we can align two protein structures using any dynamic programming algorithm and the similarity function $\text{Sim}(S(x_k),S(y_l))$. The method is summarized in Figure 2. The choice of the algorithm (global, free-shift, local) as well as the gap penalty model (linear or affine) depends on the application and all standard methods are available in the Vorolign program. Other features that influence the alignment quality are the scoring matrices chosen to measure the amino acid and secondary structure element similarity of two nearest-neighbour residues together with their corresponding weights $w_u$ and $w_o$ [Equation (1)] as well as gap penalties for the low-level ($p_u$) and the high-level (gap open: $G_o$ and gap extend: $G_e$) dynamic programming steps.

All parameters have been optimized for four different amino acid exchange matrices using a genetic algorithm (Aleksandra Djurisic et al., 1997) as discussed in Section 2.7.

The idea of using a low level dynamic programming step in order to fill a higher level dynamic programming matrix is also known as ‘double dynamic programming’ (Taylor and Orengo, 1989) and has already been used in the context of protein structure alignment (Taylor and Orengo, 1989; Taylor, 1999). Their way to calculate the similarity in the low-level matrix differs substantially from ours. For a pair of residues $(i,j)$, the two structures are centered and superimposed with respect to the adjacent residues. With respect to that superimposition all pairs of residues are examined and their similarity is measured using mainly geometric features. The score of an alignment of all residues of the two protein structures is used to fill the high-level matrix entry for the pair $(i,j)$. In comparison, Vorolign concentrates on the nearest-neighbour residues to compute the similarity score using amino acid and secondary structure element conservation. This significantly reduces the costs of each similarity computation and allows Vorolign to calculate flexible structural alignments (see also Section 3.2.3).

### 2.5 Multiple alignment of protein structures

Recently, the focus of structure comparison has shifted from pairwise to multiple protein structure alignments and several methods have been proposed (see for example Ye and Godzik (2005) and Shatsky et al. (2004)). We adopted a standard approach of many multiple sequence alignment methods that first calculate all pairwise alignments of the input sequences, in our case structures, and combine the pairwise alignments following a guide tree to retrieve a multiple sequence alignment. We implemented an approach similar to T-Coffee (Notredame et al., 2000) to generate multiple structure alignments from a set of pairwise structural alignments calculated by Vorolign.

The input of the T-Coffee method is a set of pairwise alignments, where the same protein pair can be included several times in the set to account for solutions generated by different alignment strategies. All residues aligned in that set of alignments form the so called ‘primary library’. All aligned residue pairs in the primary library are assigned a weight that represents the belief that the alignment of the two residues is correct. In the original publication (Notredame et al., 2000) this weight is set to the sequence identity of the two sequences as given by the alignment. Since we want to measure not sequential but structural similarity, we use a structure-based score, namely the TM-Score (Zhang and Skolnick, 2004), to measure the quality of a pairwise alignment. The TM-Score scores how well two protein structures that can be superimposed structurally, given an alignment. It is equal to 1 for a perfect structural similarity and <0.17 for random similarities.

The next step of the algorithm is the ‘library extension’. The purpose is to combine information contained in the primary library, such that any pair of residues reflects some of the information contained in the whole library. Therefore, a triplet approach is used. Each aligned pair is tested for consistency via all other alignments in the primary library.

For instance consider the residues $x$, $y$ and $z$ in the proteins $X$, $Y$ and $Z$ respectively as well as three pairwise alignments of the three proteins. We find $x$ and $y$ as well as $x$ and $z$ are aligned. Now, if $y$ and $z$ are also aligned in the third alignment there is an alignment of $x$ and $y$ through $z$. The weight of the pair $x$ and $y$ is increased by the minimum of the weights, i.e. TM-Scores, assigned to the alignment of $X$ and $Z$ or to the alignment of $Y$ and $Z$. The extended library can now be used as a ‘scoring matrix’ for a progressive alignment strategy. The multiple alignment is constructed following a guide tree as produced by a neighbour joining strategy (Saitou and Nei, 1987), with the similarity of two proteins given by their respective TM-Scores.

### 2.6 Fast scan for family members

Protein structure alignment is often used for the automatic detection of structurally similar proteins in a database, given a structurally resolved, but still unclassified, protein. A similar application would be the generation of all-against-all alignments of a set of proteins in the PDB in order to define protein families. For that application, not only the accuracy of the method in detecting proteins that belong to the same family, e.g. a SCOP (Murzin et al., 1995) class, but also the speed of the database scan is important. Since Vorolign is a very fast method for generating structural alignments, we tested the performance of the method in detecting the correct SCOP family...
for a target protein in a database of 4358 family representatives from ASTRAL25. The results are discussed in Section 3.1.

For all database proteins, the Voronoi tessellation and the corresponding nearest-neighbour sets can be pre-calculated in order to speed up the scan. After the calculation of the Voronoi tessellation for the target protein, the protein can be aligned against the database. To avoid the calculation of Vorolign alignments between protein structures that do not show a significant similarity on the secondary structure level, e.g. an all-β protein against an all-α protein, we employ secondary structure element alignment (SSEA) (McGuffin et al., 2001) to pre-filter the template database. In the current setup, we use the 250 highest scoring hits (~5% of the template database) found by SSEA, which are subsequently aligned to the target using the Vorolign method. The results are then ranked with respect to their Vorolign alignment scores and for the best 20 alignments we compute structural superimpositions using the TM-Score program (Zhang and Skolnick, 2004). Please note that the alignments are not re-ranked with respect to their TM-Scores.

2.7 Parameter optimization

Parameters used throughout the experiments have been optimized on a set of 20 randomly chosen protein pairs. None of the training protein pairs has been included in the test cases discussed later on. We used a genetic algorithm (Aleksandra Djurisic et al., 1997) with the average TM-Score of the structural alignments as fitness function. Proteins were repeatedly aligned by Vorolign using the values of the population member until the genetic algorithm converged. As a convergence criterion we used five generations without a change of the fittest population member. The algorithm has been restarted 50 times.

Parameters were optimized for free-shift alignment with affine gap penalties ($G_{O}$ and $G_{E}$) as well as for four different structure-based amino acid exchange matrices: the Blake–Cohen matrix (Blake and Cohen, 2001) (BC), a matrix by Prlic et al. (2000) (P), by Azarya-Sprinzak et al. (1997) (N) and the SM_THREADER matrix (Dosztanyi and Torda, 2001) (T). As secondary structure scoring matrix we used the matrix described by Walqvist et al. (2000). The optimization results are shown in Table 1.

### Table 1. Optimized parameter settings for the four different amino acid exchange matrices as found by the genetic algorithm

<table>
<thead>
<tr>
<th>Matrix</th>
<th>$\omega_1$</th>
<th>$\omega_2$</th>
<th>$p_a$</th>
<th>$G_{O}$</th>
<th>$G_{E}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>0.3</td>
<td>2.52</td>
<td>1.23</td>
<td>$-16.58$</td>
<td>$-6.15$</td>
</tr>
<tr>
<td>P</td>
<td>2.03</td>
<td>2.42</td>
<td>1.78</td>
<td>$-13.68$</td>
<td>$-4.83$</td>
</tr>
<tr>
<td>T</td>
<td>0.58</td>
<td>0.71</td>
<td>5.04</td>
<td>$-22.50$</td>
<td>$-6.50$</td>
</tr>
<tr>
<td>N</td>
<td>0.26</td>
<td>0.96</td>
<td>4.45</td>
<td>$-22.866$</td>
<td>$-8.91$</td>
</tr>
</tbody>
</table>

### 3 RESULTS

3.1 Family recognition and pairwise structural alignments

In order to test the ability of Vorolign to detect the correct protein family in a database of representative protein structures, we used 979 proteins from the ASTRAL compendium that are contained in version 1.67 (February 2005) but not in the previous version 1.65 (December 2003). From the >100000 domains contained in the different databases, we could not be sure whether a protein or a domain of a protein could be removed for the sequence identity used in the standard alignment methods against the ASTRAL95 (version 1.65) was found to be >30%, with >30 identical residues. The final set of 979 proteins therefore comprises the non-trivial cases and allows us to assess the ‘blind test’ performance of our and other methods in predicting the SCOP family for a given target.

Those proteins were scanned against the ASTRAL25 database (version 1.65) which includes 4358 proteins using the setup described in Section 2.6. All four amino acid exchange matrices from Section 2.7 are evaluated.

The performance of all methods will be measured as the fraction of proteins that can be assigned to their correct family, superfamily or fold with respect to the SCOP classification. For Vorolign, only the best template due to the Vorolign alignment score is taken into account for the evaluation. To account for the quality of the corresponding structural alignments, we use TM-Score, the percentage of equivalently aligned residues ($N_e$) and the average RMSD of the superimposition of the residues in the $N_e$ subset given the alignment. The results can be found in Table 2.

The performance of Vorolign is compared against several other methods. SSEA, free-shift sequence alignment with affine gap costs (Pam250 matrix, $G_{O} = -12$, $G_{E} = -1$), BLAST (standard options, against ASTRAL25 database) (Altschul et al., 1997), Profile–Profile alignment (PPA) as used in von Ohsen et al. (2004) as well as CE, which is faster than for example Dali and performed best in a recent study of protein-fold comparison servers (Novotny et al., 2004). All methods (except for BLAST) were given the same 250 best hits from the SSEA pre-filter and again only the best template with respect to the method specific quality score is taken into account.

To show the advantage of using Voronoi instead of $C_\beta$ distance-based contacts, we test the performance of the method using neighbour-sets defined by $C_\beta$ contacts (threshold 6.5 Å) together with the SM_THREADER matrix. Parameters have been optimized as discussed in 2.7 and are set to $G_{O} = -20.07$, $G_{E} = -8.76$, $\omega_1 = 0.71$, $\omega_2 = 1.47$, $p_a = 4.76$.

When comparing the performance of the four different amino acid exchange matrices in combination with the Vorolign method, we find the different matrices are differently well-suited to score the evolutionary similarity of residue environments. The SM_THREADER matrix performs best with respect to all quality measurements applied. This finding is interesting, since this matrix is not computed with respect to evolutionary criteria but expresses the contribution of a residue to the total energy of the protein. This could be an interesting starting point for future research, using potentials instead of amino acid exchange scores to score the similarity of two residue environments in combination with the Vorolign method.

A comparison of the results of the SM_THREADER matrix in combination with nearest-neighbour sets defined by Voronoi or $C_\beta$ distance finds Voronoi contacts to perform better than $C_\beta$-based contacts according to recognition rates as well as alignment quality. This difference cannot be expected to be large since the nearest-neighbour sets will, of course, overlap, but the advantage of the Voronoi-based contact definition gained in comparison to $C_\beta$ contacts in our opinion justifies the small overhead of computing the Voronoi tessellation in order to retrieve the nearest-neighbour sets.

The results also show that standard sequence alignment methods, i.e. BLAST and Smith–Waterman, are not able to make accurate family, superfamily or even fold assignments for our test set and demonstrate that the set contains the more challenging cases. The structural quality of the alignments is also poor.
Table 2. Results of the family recognition scan and the corresponding average pairwise alignment quality of the methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Family (%)</th>
<th>Superfamily (%)</th>
<th>Fold (%)</th>
<th>Wrong (%)</th>
<th>TM-Score</th>
<th>%Np</th>
<th>Subset RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vorolign, T</td>
<td>86.4</td>
<td>92.4</td>
<td>97.7</td>
<td>2.3</td>
<td>0.74</td>
<td>76.3</td>
<td>1.90 Å</td>
</tr>
<tr>
<td>Vorolign, N</td>
<td>83.7</td>
<td>90.4</td>
<td>96.1</td>
<td>3.9</td>
<td>0.71</td>
<td>74.1</td>
<td>1.93 Å</td>
</tr>
<tr>
<td>Vorolign, P</td>
<td>82.9</td>
<td>88.9</td>
<td>91.4</td>
<td>8.6</td>
<td>0.70</td>
<td>73.1</td>
<td>2.01 Å</td>
</tr>
<tr>
<td>Vorolign, BC</td>
<td>81.6</td>
<td>87.6</td>
<td>89.9</td>
<td>10.1</td>
<td>0.69</td>
<td>73.3</td>
<td>2.04 Å</td>
</tr>
<tr>
<td>CE1</td>
<td>81.8</td>
<td>88.4</td>
<td>90.6</td>
<td>9.4</td>
<td>0.78</td>
<td>77.8</td>
<td>1.93 Å</td>
</tr>
<tr>
<td>CE2</td>
<td>84.6</td>
<td>91.9</td>
<td>94.1</td>
<td>5.9</td>
<td>0.77</td>
<td>78.2</td>
<td>1.95 Å</td>
</tr>
<tr>
<td>CB, T</td>
<td>82.9</td>
<td>89.5</td>
<td>94.4</td>
<td>5.6</td>
<td>0.70</td>
<td>73.2</td>
<td>1.97 Å</td>
</tr>
<tr>
<td>PPA</td>
<td>80.8</td>
<td>87.5</td>
<td>91.9</td>
<td>8.1</td>
<td>0.58</td>
<td>66.2</td>
<td>2.14 Å</td>
</tr>
<tr>
<td>Free-shift alignment</td>
<td>65.8</td>
<td>68.0</td>
<td>69.6</td>
<td>30.4</td>
<td>0.49</td>
<td>49.0</td>
<td>2.16 Å</td>
</tr>
<tr>
<td>SSEA</td>
<td>60.8</td>
<td>68.9</td>
<td>75.6</td>
<td>24.4</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>BLAST</td>
<td>48.9</td>
<td>52.5</td>
<td>52.8</td>
<td>47.2</td>
<td>——</td>
<td>——</td>
<td>——</td>
</tr>
</tbody>
</table>

Methods are evaluated taking the best hit with respect to the method specific score into account. For Vorolign, CE, SSEA, free-shift alignment and PPA those are the respective alignment scores, for BLAST we use the best E-Value and for CE the best Z-score. If CE returns more than one alignment for a target-template pair the Z-Score of the longest alignment (CE1) or the best Z-Score of a pair (CE2) is used. For SSEA no alignment quality is evaluated since the method does not compute residue-based alignments. A similar argument applies for BLAST since it returns only short, local alignments. Best score in each column is shown in bold.

Methods applied in the field of fold recognition, namely SSEA and PPA, gain higher recognition rates. The results of SSEA justify its application as a pre-filter to speed up computationally more expensive methods like Vorolign, CE and PPA. PPA performs surprisingly well with respect to the family, superfamily and fold recognition rates. In >90% of the cases the method is able to identify the correct fold of the target. Nevertheless, the structural quality of the alignments is not comparable with Vorolign and CE with an average subset size of 66.2% and a TM-Score value of 0.58.

Of course, a fair comparison of the abilities of Vorolign can only be made with another structural alignment method. With respect to the recognition rates Vorolign and CE outperform all other methods discussed above. Interestingly, Vorolign in combination with the T and the N matrices is more accurate in predicting the correct family, superfamily or fold of a target than CE, making only 2.3% wrong assignments in comparison with 5.9% wrong assignments made by CE. The cases where Vorolign is correct while CE is wrong can be assigned to two different types of errors. Often CE identifies several hits with high Z-scores in the template database. From a structural point of view all those hits could be regarded as being ‘true’ hits since their structures can be superimposed well. However, since the SCOP hierarchy is not solely based on structural but also other criteria like sequence and function the structure-related Z-score of CE cannot distinguish between the true and the false hits with respect to the SCOP annotation. Here, the more sequence-based similarity score of Vorolign seems to be an advantage, taking not only structural but also sequential knowledge into account. Second, in some rare cases the CE alignment with the true family member is not optimal leading to a bad Z-score for the true family. With respect to the alignment quality, CE performs slightly better than Vorolign according to TM-Score and the number of equivalently aligned residues. This result could have been expected since Vorolign does not attempt to optimize the superimposition of the two structures at any point in the algorithm.

Vorolign is used as a pre-filter to speed up computationally more expensive methods like Vorolign, CE, SSEA, free-shift alignment and PPA. PPA performs surprisingly well with respect to the family, superfamily and fold recognition rates. In >90% of the cases the method is able to identify the correct fold of the target. Nevertheless, the structural quality of the alignments is not comparable with Vorolign and CE with an average subset size of 66.2% and a TM-Score value of 0.58.

Of course, a fair comparison of the abilities of Vorolign can only be made with another structural alignment method. With respect to the recognition rates Vorolign and CE outperform all other programs tested here. Interestingly, Vorolign in combination with the T and the N matrices is more accurate in predicting the correct family, superfamily or fold of a target than CE, making only 2.3% wrong assignments in comparison with 5.9% wrong assignments made by CE. The cases where Vorolign is correct while CE is wrong can be assigned to two different types of errors. Often CE identifies several hits with high Z-scores in the template database. From a structural point of view all those hits could be regarded as being ‘true’ hits since their structures can be superimposed well spanning large parts of the proteins. But since the SCOP hierarchy is not solely based on structural but also other criteria like sequence and function the structure-related Z-score of CE cannot distinguish between the true and the false hits with respect to the SCOP annotation. Here, the more sequence-based similarity score of Vorolign seems to be an advantage, taking not only structural but also sequential knowledge into account. Second, in some rare cases the CE alignment with the true family member is not optimal leading to a bad Z-score for the true family. With respect to the alignment quality, CE performs slightly better than Vorolign according to TM-Score and the number of equivalently aligned residues. This result could have been expected since Vorolign does not attempt to optimize the superimposition of the two structures at any point in the algorithm.

A Vorolign scan of a target against the database (including Vorolign alignments against 250 template proteins) takes on average 1 min on a single CPU. In comparison, the freely available implementation of CE takes on average 30 s per protein and therefore about 2 h per scan in the same environment.

3.2 Multiple alignment of protein structures

We applied the multiple alignment routine described in Section 2.5 to three structural families from the literature (Ye and Godzik, 2005) and compared the performance of the method with the performance of POSA and MultiProt. The examples contain the relatively easy case of 15 proteins from the Globin family, 10 proteins of the NAD(P)-binding Rossmann fold, a highly divergent family, as well as three Calmodulin-like proteins that contain conformational flexibilities to show that our method is able to deal with such a case. Alignments from MultiProt are obtained from the stand-alone program, while POSA alignments are retrieved directly from the web server. All results are evaluated using the same protocol applied to the Vorolign results in order to guarantee a fair comparison.

3.2.1 Globin family

Globins are an extensively studied example of evolution at the molecular level [see (Lecomte et al., 2005) for a recent review] and the family is considered to be a relatively easy case for multiple structure alignment. Vorolign has been used to align a set of 15 globin structures described in Ochagavia and Wodak (2004) and identifies a common core of 72 residues with an average pairwise RMSD of 1.46 Å. In contrast, MultiProt finds a common core, including all 15 input structures, of 75 aligned positions with a pairwise average RMSD of 1.95 Å. POSA identifies 67 positions with an RMSD of 1.12 Å. We find that different programs lead to different structural alignments of various length and RMSD values. Nevertheless, the superimpositions of all programs are very similar. Therefore, the performance of all three programs tested here can be regarded as being comparable.

3.2.2 NAD(P)-binding Rossmann fold

The Rossmann fold consists of a three-layer sandwich structure with a parallel β-sheet of 6 strands packed between two helices and the sheet consisting of a three-layer sandwich structure with a parallel β-sheet. We align one representative of each of the 10 families in the Rossmann fold (d1heta2, d1ek6a, d1obfo1, d2naca1, d1kyqa1, d2cmd_1, d1np3a2, d1bgva1, d1id1a_ and d1oi7a). Vorolign
identifies a common core of 44 residues with an average RMSD of 1.82 Å, the POSA alignment represents a core of 37 residues with an average RMSD of 1.15 and MultiProt identifies 33 residues with an average RMSD of 1.39. Vorolign finds 3 \( \beta \)-strands (S1, S4 and S5) and 2 helices (H1, H5) to be relatively conserved among all 10 input structures which represent only a small part of the overall structure of the proteins.

### 3.2.3 Calmodulin-like proteins
Calmodulin consists of two domains, each domain having a pair of so called EF-hands which are helix-loop-helix \( Ca^{2+} \)-binding motives. \( Ca^{2+} \) induced domain movements lead to an open or closed conformation of the protein. Different conformations are directly linked to the proteins function. Therefore, proteins from that family are good examples to test the ability of a protein structure alignment method to flexibly align protein chains.

We use three proteins from this family to demonstrate the ability of Vorolign to find good structural alignments even for proteins that show significant conformational changes. Proteins aligned are troponin C from Chicken (1ncx, open conformation), sarcoplasmic calcium-binding protein from Branchiostoma lanceolatum (2sas, closed conformation) as well as EHCABP from Entamoeba histolytica (1jfjA, partly open conformation).

Standard, rigid-body multiple structure alignment routines obtain only small, structurally conserved regions. MultiProt returns an alignment containing 55 positions with an RMSD of 1.58 Å. This result is not surprising since a rigid body superimposition of the molecules is not possible (Fig. 3c).

In comparison, the POSA method returns a full-length alignment of the three structures containing 131 positions with an RMSD of 2.58 Å in a flexible superimposition. The Vorolign method is able to align 124 positions with an RMSD of 2.77 Å. Although the common core is found to be smaller than the core found by POSA the Vorolign alignment spans the whole length of all three proteins, and a flexible superimposition of the three structures (Figure 3a and b) shows that the three proteins can indeed be structurally aligned if flexibilities are incorporated. This exemplifies Vorolign’s ability to align protein structures that contain larger conformational changes.

### 4 CONCLUSIONS
We presented a novel method to address the protein structure alignment problem. The method is based on a similarity function to measure the similarity of two sets of nearest-neighbour residues using amino acid and secondary structure exchange matrices together with double dynamic programming. The nearest-neighbour environment of a residue is defined by the Voronoi tessellation of the corresponding protein structure leading to a well-defined set of nearest-neighbours while taking the geometry of the protein implicitly into account.

Clearly one of the major benefits of the method is its speed (on average five alignments per second, excluding time for Voronoi tessellation) that allows for a rapid comparison of a protein structure against a large database of potential protein templates in order to classify the structure with a very high accuracy as shown in Section 3.1. The method also produces accurate pairwise and multiple structure alignments as described in Sections 3.1 and 3.2, even of highly flexible protein structures (Section 3.2.3), an important feature to detect similarities across the natural variance of protein structures.

Future development will include a further investigation of methods to compute alignments using the similarity function discussed in Section 2.3, as well as extensions taking more features of the Voronoi cells into account. For the multiple alignments, improvements can be expected using more advanced combination methods, e.g. as discussed by Ochagavia and Wodak (2004). The method improves on the T-Coffee protocol using a progressive combinatorial approach during which the pairwise alignments are modified to improve the fit of the multiply aligned proteins. In contrast, T-Coffee does not modify the pairwise alignments presented to the algorithm.

Further work will be on studying the interplay of sequence and secondary structure conservation of environments of structurally
equivalent residues on different levels of sequence and structure similarity. The Vorolign score might also be useful to score the quality of sequence–structure alignments with QUASAR (Birzele et al., 2005).

Flexible, multiple alignments computed by Vorolign for families, superfamilies or folds can be useful to define common cores of proteins which will be an important step towards a deeper understanding of protein structure and sequence evolution.

ACKNOWLEDGEMENTS

This work was partially funded by the DFG (grant PROSEQO II, Zi 616/2) and by the BMBF (grant BFAM, 81621005).

REFERENCES