International Journal of Computer Mathematics
Vol. 00, No. 00, Month 200x, 1–15

Invariant Features for Searching in Protein Fold Databases

MAJA TEMERINAC, MARCO REISERT and HANS BURKHARDT

Institut für Informatik, Albert-Ludwigs-Universität, D-79110 Freiburg, Germany

(*v3.2 released Sept 2006*)

The tremendous growth of 3D data models available on the internet requires more skills for fast retrieval and classification algorithms. Especially, the problem of finding structural similarities between proteins automatically, in order to predict their functional similarity is a challenging task. In this paper a new algebraic method for structural comparison between proteins based on invariant features computed by group integration with spherical harmonics and D-Wigner matrices is proposed. Our goal is to achieve good classification without alignment by using intrinsic, pose invariant features. We compare our method to DALI, PRIDE and the Gauss Integral-method in a classification and search task. Additionally we provide a web interface to test the proposed method.

Keywords: Protein Structure; Protein-protein similarity; Protein search and retrieval; Group Integration; Spherical Harmonics; D-Wigner Matrices; Invariants

AMS Subject Classifications: 33C55 (Spherical Harmonics); 14L24 (Geometric Invariant Theory)

1. Introduction

Molecular biologists are often interested in getting a survey of the objects in a biomolecular database making classification one of their basic tasks: To which of the recognized classes in the database does a new molecule belong? Answering this question is one of the basic problems in structural bioinformatics (for a review about structural bioinformatics and its impact on biomedical science see e.g. [9, 10]).

Several classification schemata such as SCOP [1], CATH [2] and FSSP [3] are available in the Internet. When a new object is inserted into the database, the supervision by experts that are very experienced and have a deep knowledge in the domain of molecular biology is necessary in most cases. An efficient classification algorithm is desired, that can speed up the classification process by acting as a fast filter for further investigations.

While SCOP and CATH require classification by human experts, a fully automatic classification is available from the FSSP database (Families of Structurally Similar Proteins), generated by the DALI (Distance matrix ALIgnment) system [3]. The evaluation of a pair of proteins is very expensive, since query processing for a single molecule against the entire FSSP database currently takes an overnight run.

Many efforts have been made to find a suitable algorithm for protein structure comparison. On the one hand we have alignment based methods such as DALI [3] and the Combinatorical Extension algorithm (CE) [23], which are powerful but of high time complexity due to their combinatorial nature. On the other hand we have feature based approaches, which compute representative features and compare the structures solely by their feature representation. They are fast, but mostly of less discriminative power. For example, Pride [4] computes the distribution of $C_\alpha - C_\alpha$ distances. In [5] hierarchical clusters based on indirect coding features from the amino-acid composition sequence are formed with the help of a neural network. The Gauss integral features [6] based on knot theory are the latest attempt to tackle the protein structure comparison problem using invariant features. A good overview of protein structure comparison methods has been given recently by Carugo in [7].

*Corresponding author. Email: reisert@informatik.uni-freiburg.de

International Journal of Computer Mathematics
ISSN 0020-7160 print/ISSN 1029-0265 online
http://www.tandf.co.uk/journals
DOI: 10.1080/0020716YYxxxxxxxx

One of the difficulties of the task is that the results need the approval by the biological community since the quality of the classification algorithm is very difficult to measure. Most molecular biologists use DALI for automatic classification. Furthermore, the user is interested in an alignment of the structure, which is a high time consuming task. In fact, alignment techniques such as contact map overlap [8] are very popular, although their computation is NP-complete. Our goal is to achieve good classification without alignment. Our approach should make an overall fast protein search possible in the first step. In a second step one could think of a refinement by e.g. aligning the remaining structures of high similarity.

In this paper we introduce Group Integration (GI) for structural protein data and apply it to a protein retrieval and search task. The protein features presented in this paper are inspired by the work given in [21, 22], where similar techniques are used for classification of pollen grains and 3D surface models. In general, GI stands in contrast to Normalization techniques, which obtain invariance by computing features relative to a global reference frame. The determination of the reference frame makes Normalization techniques extremely sensitive to noise. Whereas GI is known to be very robust to many kinds of noise. In [11] a detailed overview over GI-techniques is given. Haasdonk [12, 13] applied GI to character recognition and joined the GI-framework with Kernel-techniques. Ronneberger et al [14,15] used GI for the classification of Pollen grains and segmentation of cell nuclei. In [16–18] GI was successfully applied to texture-classification and image retrieval. In [19] algebraic invariants are used for character recognition.

This work is organized as follows: Section 2 introduces GI and explains how it can be expanded by spherical harmonics and D-Wigner matrices. Section 3 discusses the choice of the kernel for GI, while in section 4 a protein-specific implementation of GI is introduced. In section 5 related work is presented. The results of experiments conducted on representative datasets from the PDB and their comparison to related work are discussed in section 6. Finally, in section 7 conclusions and future work is presented.

2. Group Integration Features

Rather than describing two structures relative to each other which is done by alignment, our idea is to find a way to construct a mathematical fingerprint of each structure automatically. This fingerprint is a representation of the structure which can be easily used to construct a similarity measure. Describing three-dimensional structures by feature vectors is a well-known method for three-dimensional structural retrieval. To get a similarity measure that is independent to the relative pose in space of the compared objects the feature vector has to be invariant under Euclidean motion. Group Integration is a constructive way to reach this goal. Starting with a non-invariant simple feature of the structure an averaging over the whole invariance group results in an invariant feature of the structure.

In this work we apply this idea to search and retrieve proteins by their three-dimensional structural properties. Protein structure can be viewed at four hierarchical levels. First, the protein can be seen as a one-dimensional sequence of amino acids. After the protein folding, the one-dimensional sequence folds into a three-dimensional structure. This folded structure is composed of smaller three-dimensional units called secondary structure. The folded structure or simply fold is also called tertiary structure. A protein usually consists of several amino acid chains, thus a fourth level of structure, the quaternary structure exists. Figure 1 displays a protein structure in different visualization modes. Instead of concentrating on one specific hierarchy level, our work tries to incorporate and handle all views in a uniform manner. Thus, the approach has to cope with the sequence and structural information and their interplay in an appropriate way. In this work we show how the GI framework can be used to achieve this goal.

2.1. *Definition*

First we describe the theory for three-dimensional objects in general. In order to represent these objects, we use a kind of intensity function $\mathbf{x} : \mathbb{R}^3 \mapsto \mathbb{R}$ indicating the presence of the object. Since we are dealing with proteins, we will later define a special 'protein-function' that represents the protein. For this, imagine that each protein can be represented by a superposition of blob like functions centered at the atoms' positions. Thus the 'protein-function' can be visualized as a three-dimensional structure constructed of overlapping

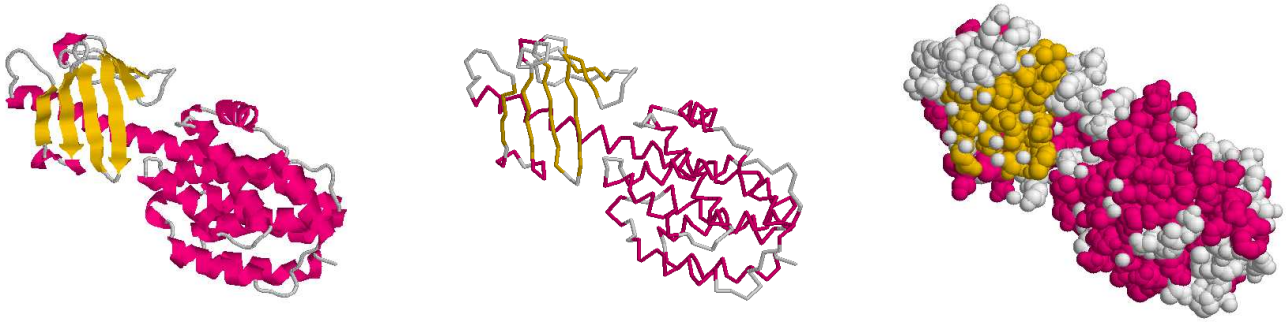

Figure 1. **Example of the three dimensional model for the protein structure with pdbid 1CIS.** All images are produced using RasMol *http://www.umass.edu/microbio/rasmol/*. On the left: cartoon display, one can easily recognize the second-order structure. In the middle: the C_α-backbone of the protein. On the right: All atoms are displayed using small spheres.

normal distributions. We will explain the protein structure model in more detail in section 4. However, for a better understanding of the invariant feature construction, we will think of $\mathbf{x}$ as the intensity function for a three dimensional object at first.

An element g of the Euclidean group $\mathcal{E}$ acts on $\mathbf{x}$ by $g\mathbf{x}(r) \mapsto \mathbf{x}(Rr + t)$, where $r, t \in \mathbb{R}^3$ and $R \in \mathbb{R}^{3\times 3}$ is a orthogonal matrix, i.e. a three-dimensional rotation and/or a reflection. A group integration feature I_k is obtained by integrating a kernel function k over the Euclidean motion

$$I_k(\mathbf{x}) = \int_{\mathcal{E}} k(g\mathbf{x})\ dg. \tag{1}$$

Typically, choices of k are e.g. $k(\mathbf{x}) = \mathbf{x}(0)\mathbf{x}(d)$ or $k(\mathbf{x}) = h(\mathbf{x}(0))\ h'(\mathbf{x}(d))$, where h and h' may be some arbitrary nonlinear functions.

2.2. *Including directional information*

To include directional information, instead of only the values of $\mathbf{x}$ as was mostly done before, one should also consider a local quantity that describes the neighborhood of some point. The gradient $\nabla\mathbf{x}$ is the first choice to capture the configuration of the neighborhood of some point. For computing the group integral, we can use kernels like $k(\mathbf{x}, \nabla\mathbf{x}) = h_1(\nabla\mathbf{x}(0))h_2(\nabla\mathbf{x}(d))$ or further extension, which combine the gradient values with the values of $\mathbf{x}$. Of course, besides the gradient one can also use other local neighborhood operators like the Hessian, or even higher derivatives.

2.3. *Spherical Harmonics*

To give our features more expressiveness we will later combine it with the Spherical Harmonic functions. But first a small review on Spherical Harmonics is given. Any function $f(s)$ defined on the two-sphere S^2 can be orthogonally expanded in terms of the so called Spherical Harmonics (SH).

$$f(s) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_m^l Y_m^l(s), \tag{2}$$

where s is a unit vector and the a_m^l the expansion coefficients, that are computed by projections $a_m^l = \int_{S^2} f(s)\overline{Y_m^l(s)}ds$ on the basis functions. In practice, the infinite sum is truncated at some finite cutoff parameter l_{max}. The Spherical Harmonic Transformation (SHT) is the analogue to the Fourier

transform for the rotation group, i.e. the SHT provides a representation that is invariant to rotations. There are subspaces, which preserve their energy while rotating the function. Moreover, the a_m^l show a nice transformation behavior. Suppose $f(s)$ is rotated by some rotation g, then the a_m^l are transformed by the so-called D-Wigner matrices $D^l(g)$, i.e. $a_m^l \mapsto \sum_{m=-l}^{l} D_m^l(g) a_m^l$. Since an integration over the rotation group can always be separated into an integration over a sphere and a circle, we are able to use the SHT to retain more information. Instead of just integrating the sphere integration out we expand the remaining function in terms of spherical harmonics.

2.4. *D-Wigner expansion*

As the D-Wigner matrices are the irreducible representations of the three-dimensional rotation group, they have another important property. A real function $f(g) : SO_3 \mapsto \mathbb{R}$ defined on the rotation group itself can be orthogonally expanded in terms of D-Wigner matrices:

$$f(g) = \sum_{l=0}^{\infty} tr(D^l(g)\ B_l), \tag{3}$$

where the B_l are some kind 'expansion matrices'. The B_l are obtained by projections of the function $f(g)$ on the D-Wigner matrices $B_l = \int_{SO_3} f(g) D^l(g)^T dR$. Hence we are able to use the projections to retain even more information in our group integration framework. Instead of a simple integration over the rotation group we compute projections on the D-Wigner matrices. Since the D-Wigner matrices are unitary representations of the rotation group one can show that the norms of the columns of the B_l are invariant to right multiplications $f(g) \mapsto f(gg')$ and similar the norms of the rows of the B_l are invariant to left multiplications. Hence we can obtain invariance against rotations by taking the norms of the columns or rows of the 'expansion matrices', respectively. The D-Wigner matrices were already used in [21] to obtain more discriminative features for the retrieval of 3D surface models. This work also proposes an algorithm for the computation of the $D^l(g)$ given the corresponding rotation matrix. This method works linear in the number of coefficients that has to be computed.

3. The Kernel Choice

It is not a simple question which kernel function one should choose and which non-linearities should be incorporated. The choices are typically guided by the application's demands and complexity considerations. The surface and the outer atoms of the protein play an important role for the biochemical and functional behavior. These regions are of high entropy and contain most of the information about the protein. In fact, these atoms have also the highest temperature within the protein. It is important to emphasize these regions. The magnitude of the gradient $\nabla \mathbf{x}$ of the protein's intensity function gives high responses exactly in this regions. So we choose a kernel of the form

$$k_d(\mathbf{x}) = h(\nabla \mathbf{x}(0))\ h'(\nabla \mathbf{x}(d)) \tag{4}$$

as the basis of our kernel function with width parameter d. But how to choose h and h'? One demand is that both functions should give strong feedback if the gradient is large. The function should also be direction specific to keep the relative directions of the gradients. The simplest idea fulfilling this demands is $h_n(v) = |v^T n|$, where n is some fixed unit vector. We use the absolute value of the dot-product, because experiments have shown that whether the edge is falling or growing is not of the same importance as the actual direction. A disadvantage of the function above is that it is not able to decide whether it has to deal with large, disoriented or a small, oriented gradients v. A more rational choice is

$$h_n(v) = |v| \delta_1 \left(\frac{|v^T n|}{|v|} \right), \tag{5}$$

where δ_1 is the Delta-Distribution[1] giving contribution if its argument is nearby 1 and otherwise zero. The function $h_n(v)$ is unequal to zero whenever $n \parallel v$, i.e. n and v are parallel or antiparallel. Our kernel-function is

$$k_{d,n,n'}(\mathbf{x}) = h_n(\nabla\mathbf{x}(0))\ h_{n'}(\nabla\mathbf{x}(d)), \tag{6}$$

3.1. *Parameter Reduction*

The basis kernel (6) contains the three vector-valued parameters d, n, n'. Of course, after integration of the kernel

$$I_{d,n,n'}(\mathbf{x}) = \int_{\mathcal{E}} k_{d,n,n'}(g\mathbf{x})\ dg \tag{7}$$

the feature $I_{d,n,n'}$ does not depend on the overall seven (the parameters n and n' are unit vectors) real parameters anymore, there are redundancies. Due to the rotation invariance, the feature obviously fulfills the relation $I_{d,n,n'} = I_{Rd,Rn,Rn'}$ for arbitrary orthogonal matrices $R \in \mathbb{R}^{3\times3}$. The relative directions of three vectors up to orthogonal transformations are determined by the three pair-wise dot products of those. So the parameters $\alpha = n^T d/|d|$, $\beta = n'^T d/|d|$ and $\gamma = n^T n'$ uniquely determine the configuration. Using a fourth parameter $\Delta = |d|$ uniquely describes the whole parameter set.

Due to the absolute value in the definition of the h_n, the symmetry $h_n = h_{-n}$ is fulfilled and hence $I_{d,n,n'}$ is also invariant to sign changes of n and n'. Thus, only absolute values of the dot products from above are needed for description.

However, we have still another symmetry. Due to the integration over the rotation and translation group, the feature is also invariant to exchanges of n and n', i.e. $I_{d,n,n'} = I_{d,n',n}$. This fact can be seen easily by reparametrising the integration. Changing the rotation integration by $R \mapsto -R$ and shifting the translation integration by $t \mapsto t + d$ exchanges the kernel-factors and hence n and n'. For further consideration we ignore this last symmetry, which just causes additional memory consumption.

Finally, we can reduce the parameter set to four positive real parameters: Three parameters describing the relative configuration of d, n, n' and $\Delta \in [0, \infty)$ giving the norm of the distance vector d.

4. Implementation for Proteins

Typical protein chains do not contain more than 1000 amino acids. For complexity reasons we represent each amino acid just by the location of its C_α-atom. The C_α-atoms represent the backbone of the amino acid chain. Most approaches for comparing and aligning proteins make the same restriction and only consider the backbone. In the beginning we gave an intuition how to understand the 'protein-function' $\mathbf{x}$. As already depicted we interpret the protein function as a superimposition of Gaussians. Each centered at the C_α-locations. Thus, we have

$$\nabla\mathbf{x}(r) = \frac{2}{\sigma^2}\sum_i (u_i - r)e^{-(\frac{u_i - r}{\sigma})^2} \tag{8}$$

where u_i are the C_α-atom coordinates and the index i ranges over the whole point set and are chosen according to the sequence numbers of the amino acids. In Figure 2, the gradient at each C_α-atom is displayed for the protein with the PDB identity 1DLR.

But how to implement the feature computation in practice? In [14] a convolution with a rotation symmetric kernel is used for a fast evaluation of the integral. In our case this is not possible, because our

[1] we write $\delta_y(x)$ for the usual $\delta(x - y)$ of the Delta-Distribution due to space considerations

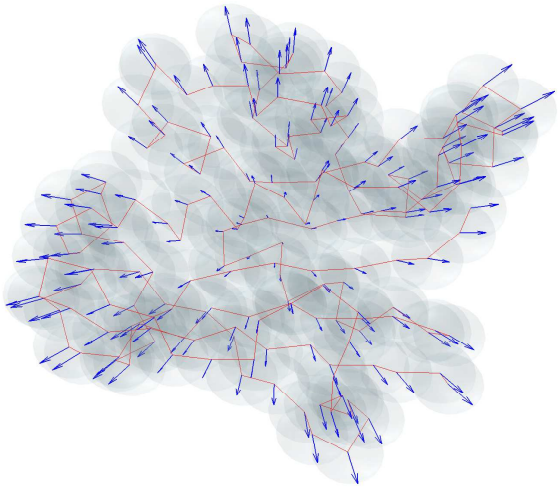

Figure 2. **Protein structure model.** For the protein with the PDB identity 1DLR, the C_α atoms are visualized. The surface defined by the C_α atoms is indicated.

kernel depends on local directional quantities. We want to pursue an alternative direction. We rewrite the integral for the basis kernel

$$I_\Pi = \int\limits_{\mathbb{R}^3} \int\limits_{O_3} h_n(R\nabla\mathbf{x}(u)) h_{n'}(R\nabla\mathbf{x}(u + R^T d))\ du\ dR, \tag{9}$$

where Π denotes the parameter set d, n, n'. We use the relation $f(u) = \int\limits_{\mathbb{R}^3} f(u')\delta_{u'}(u)\ du'$ for second kernel factor and get

$$I_\Pi = \int\limits_{\mathbb{R}^6} \int\limits_{O_3} h_n(R\nabla\mathbf{x}(u)) h_{n'}(R\nabla\mathbf{x}(u'))\delta_{u'}(u + R^T d) du\ du'\ dR. \tag{10}$$

As in the actual form of our kernel function the integral over the rotation group contributes a non-zero value whenever $n \parallel R\nabla\mathbf{x}(u)$, $n' \parallel R\nabla\mathbf{x}(u')$, $d \parallel R(u - u')$ and $|d| = |u - u'|$. The first 3 conditions are fulfilled if n, n' and d have the same configuration up to rotation as $\nabla\mathbf{x}(u)$, $\nabla\mathbf{x}(u')$ and $u - u'$. So we rewrite

$$I_\Pi = \int\limits_{\mathbb{R}^6} \theta_{d,n,n'} |\nabla\mathbf{x}(u)||\nabla\mathbf{x}(u')|\delta_\Delta(|u - u'|) du\ du', \tag{11}$$

where $\theta_{d,n,n'}$ denotes the orientation specific part. It has a non-zero value whenever n, n' and d have the same configuration up to rotation as $\nabla\mathbf{x}(u)$, $\nabla\mathbf{x}(u')$ and $u - u'$. As already mentioned, this is the case, when the pair-wise dot-products $\alpha = d^T n/|d|, \beta = d^T n'/|d|$and $\gamma = n^T n'$ are the same as for the observed gradients and difference. So we parameterize the parameter space by solely the dot-products α, β, γ and $\Delta = |d|$. To make the computation of the integral feasible we make now a strong simplification. Instead of integrating over the whole continuous domain $\mathbb{R}^6$, we let vanish the gradient function for all points that are not a position of a C_α atom. That is

$$(\nabla\mathbf{x})_{approx}(r) = \sum_j \delta(u_j - r)\nabla\mathbf{x}(r) \tag{12}$$

One might argue that this approximation is very crude, which is actually true if the width of the gaussian is large. But actally we are not really interested in a very good approximation, as long as the invariance

is not violated, which is actually the case. And the approximation of the gradient still carries information about the neighborhood of a C_α atom, which was the initial motivation. If we now replace the gradient function in (11) with this approximation we get a double sum

$$I_\Pi = \sum_{i,j} \theta_{d,n,n'} \delta_\Delta(|u_i - u_j|) |\nabla \mathbf{x}(u_i)||\nabla \mathbf{x}(u_j)|. \tag{13}$$

where the indices i and j are both ranging over the whole set of C_α atoms. Let us try to interpret this equation. As already mentioned we can determine the parameter set Π to be the relative cosines α, β, γ and the distance Δ, because the value of the integral only depends on the relative configuration due to the invariance obtained by the group integration. Now we fix all this parameters $\Pi = \{\alpha, \beta, \gamma, \Delta\}$. We have to consider all pairs u_i, u_j of C_α positions and check whether they have a certain distance Δ up to some tolerance. This corresponds to the integrand $\delta_\Delta(|u_i - u_j|)$. Further we have to check whether the two gradients at position u_i and u_j and the connecting vector $u_i - u_j$ have a relative configuration corresponding to α, β, γ. This corresponds to the orientation specific part $\theta_{d,n,n'}$. If both conditions are fulfilled we can add the contribution $|\nabla \mathbf{x}(u_i)||\nabla \mathbf{x}(u_j)|$ to the integral and proceed to the next pair of atoms. It would be quite time consuming to consider for each parameter configuration the set of all possible atom pairs. Thus, the actual implementation is a kind of upside down. Instead of fixing a parameter configuration beforehand, we only run just once over all pairs of atoms. For each pair we compute the parameter values Π for which this pair would give a contribution and accumulate the integral for the appropriate parameters. Algorithm 1 shows this in pseudo code.

Algorithm 1 GI Algorithm

Initialize $I_\Pi = 0$ for all parameter configurations Π.
for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    Compute
    $\alpha = \frac{\nabla x(u_i)}{|\nabla x(u_i)|}^T \frac{(u_i - u_j)}{|(u_i - u_j)|}$, $\beta = \frac{\nabla x(u_j)}{|\nabla x(u_j)|}^T \frac{(u_i - u_j)}{|(u_i - u_j)|}$, $\gamma = \frac{\nabla x(u_i)}{|\nabla x(u_i)|}^T \frac{\nabla x(u_j)}{|\nabla x(u_j)|}$, $\Delta = |u_i - u_j|$
    Let $\Pi = \{\alpha, \beta, \gamma, \Delta\}$
    Update $I_\Pi \to I_\Pi + |\nabla x(u_i)||\nabla x(u_j)|$
  end for
end for

Here N is the number of residues in the protein. Of course, the parameter space has to be discretized in a certain way. Later, in the experimental section, we give details how this was actually done. Actually, we compute some kind of four dimensional histogram. We compute the frequency of occurrence of two gradients in a specific distance with a particular relative configuration. This issue is very interesting since it shows a very close connection of GI-features with invariant histograms. In fact the so called Shape Distributions proposed in [20], a kind of distance histogram, may be seen as a GI-feature.

4.1. *Using the SHT*

As already mentioned we want to use the SHT to retain more information about the structure. Rewriting (11) by evaluating $\delta_\Delta(|u - u'|)$ leads to the sphere integral

$$I_\Pi = \int\limits_{\mathbb{R}^3, S^2} \theta_{d,n,n'} |\nabla \mathbf{x}(u)||\nabla \mathbf{x}(u + \Delta s)| \, du \, ds, \tag{14}$$

where s ranges over the unit-sphere S^2. Instead of simply integrating the expression above we now compute the projection of it on $Y_m^l(s)$, i.e.

$$I_{\Pi}^{lm} = \int\limits_{\mathbb{R}^3,S^2} \theta_{d,n,n'} |\nabla \mathbf{x}(u)||\nabla \mathbf{x}(u+\Delta s)| \overline{Y_m^l(s)}\; du\; ds. \tag{15}$$

For $l = 0$ the integral is exactly the same as (11). For $l > 0$ the implementation of the above integral is very similar to the computation of (11). Instead of a simple accumulation, the contributions are weighted by the complex factor $Y_m^l(\frac{u-u'}{|u-u'|})$. So, the update rule in line 9 of Algorithm 1 translates to $I_{\Pi}^{lm} \rightarrow I_{\Pi}^{lm} + Y_m^l(\frac{u-u'}{|u-u'|})|\nabla \mathbf{x}(u)||\nabla \mathbf{x}(u')|$, where the result array gets two additional indices l and m. After computation, the results are made invariant by computing the bandwise energy $\sqrt{\sum_{m=-l}^{l} |I_{\Pi}^{lm}|^2}$.

4.2. *Using the D-Wigner expansion*

For the SHT transform we had to reduce the integral over the rotation group to an integral over a sphere. For the D-Wigner matrices we can directly apply the idea to the group integral. Let us denote the integrand of (10) by $K(R, u, u')$, then we have to compute the projections of K on the D_l matrices as follows

$$\int\limits_{\mathbb{R}^6} \int\limits_{O_3} K(R, u, u') D_l^T(R)\; dR\; du\; du'. \tag{16}$$

Here the weighting of the integrand directly depends on the rotation, which turns the configuration of the gradients $\nabla \mathbf{x}(u), \nabla \mathbf{x}(u')$ and the difference vector $u - u'$ into the parameter-configuration d, n, n'. Until now the actual parameters d, n, n' were not of interest; we only used angles within them. Now we need an actual coordinate representation. To get a non-redundant representation we have to use a standard representation of the three parameters such that no configuration appears twice.

We use a modified γ parameter γ', which is computed from the angle of the projections of n and n' on the plane with normal vector d,

$$\gamma' = \frac{P_d n}{|P_d n|}^T \frac{P_d n'}{|P_d n'|}, \tag{17}$$

where P_d is the orthogonal projection on the d-plane. This has the advantage of a compact configuration representation:

$$d = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, n = \begin{pmatrix} \alpha \\ \sqrt{1-\alpha^2} \\ 0 \end{pmatrix}, n' = \begin{pmatrix} \beta \\ \sqrt{1-\beta^2}\gamma' \\ \sqrt{1-\beta^2}\sqrt{1-\gamma'^2} \end{pmatrix},$$

which is consistent to the already given parameter reduction with (α, β, γ').

Let us denote the matrix composed of the three column vectors from above by $V_{\alpha,\beta,\gamma'} = [d\ n\ n']$. So, for any observed $\nabla \mathbf{x}(u), \nabla \mathbf{x}(u')$ and $(u' - u)$ we have to compute the configuration (α, β, γ') and determine the rotation R which turns the standard representation $V_{\alpha,\beta,\gamma'}$ into the observed matrix $M = [\frac{\nabla \mathbf{x}(u)}{|\nabla \mathbf{x}(u)|}, \frac{\nabla \mathbf{x}(u')}{|\nabla \mathbf{x}(u')|}, \frac{(u'-u)}{|u'-u)|}]$, i.e. we have to solve $R\ V_{\alpha,\beta,\gamma'} = M$ for R, which is not difficult since $V_{\alpha,\beta,\gamma'}$ is upper triangular. In Algorithm 2 we outline the modified algorithm.

Now, the result array I_{Π}^l is matrix valued, where each $I_{\Pi}^l \in \mathbb{C}^{(2l+1)\times(2l+1)}$ is a complex-valued matrix. The result array is made invariant by taking norms columnwise. Hence, for each l we obtain $2l+1$ invariant features, instead of one feature as for the SHT-based algorithm.

Algorithm 2 D-Wigner Algorithm

Initialize $I^l_\Pi = 0$
for $i = 1$ to N **do**
  for $j = 1$ to N **do**
    Compute $\Pi = \{\alpha, \beta, \gamma', \Delta\}$
    Determine $R = MV^{-1}_{\alpha,\beta,\gamma'}$
    where $M = [\frac{\nabla x(u_i)}{|\nabla x(u_i)|}, \frac{\nabla x(u_j)}{|\nabla x(u_j)|}, \frac{(u_j - u_i)}{|u_j - u_i)|}]$
    Update $I^l_\Pi \rightarrow I^l_\Pi + |\nabla x(u_i)||\nabla x(u_j)|\ D^l(R)$
  end for
end for

4.3. *Incorporating sequential distances*

As already mentioned, we also want to include the sequential information of the amino acid chain. Proteins are more than point clouds of atoms. They contain sequential information, which should not be neglected. Choosing two C_α-atoms we can assign them a sequential distance. Since the support of our object function is restricted to the atom positions, we are able to give a mapping $I : \mathbb{R}^3 \mapsto \mathbb{R}$ which assigns the appropriate sequence indices to the spatial positions of the atoms. Hence, we extend kernel (6) by

$$k^{(P)}_{d,n,n',\mu}(\mathbf{x}) = k_{d,n,n'}(\mathbf{x})\ \delta_\mu(|\mathbf{I}(0) - \mathbf{I}(d)|), \tag{18}$$

to also incorporate the sequential distance.

After computing one feature vector for each protein domain, the feature vectors are compared using some kind of metric. We tried several L-norm based metrics and some χ^2 based metrics. For our purpose the $L1$ norm yielded the best results.

5. Related work

We compared our results to DALI/FSSP, PRIDE and the Gauss Integral methods. DALI and the authors of the Gauss Integral method provide the software to compute their features. The PRIDE method was implemented by the authors according to the description in [4].

5.1. *DALI/FSSP*

The DALI algorithm trys to align distance matrices that are composed of the pairwise distances of C_α atoms $d_{ij} = |u_i - u_j|$. Usually, the distance matrices are depicted using gray scale images, where black indicates the distance zero and is only present at the diagonal.

How should one compare two distance matrices? The simple idea is to slide one (transparent) matrix over the other and detect similar submatrices. This idea implies a combinatorial optimization problem of merging corresponding similar submatrices to larger blocks of agreement by removing redundant rows and columns. The solution of this optimization problem is computed with the Monte Carlo method. In the trial-and-error method the structurally similar regions are found by defining a cutoff function on the intramolecular distances between two detected submatrices. The result of the alignment is typically reported as an equivalent set of amino acids and visualized as a 3D superposition.

Since algorithms of the alignment of two structures have been known for a long time, the main contribution of DALI was to apply alignment on large data sets in order to compute automatically a complete map of the protein universe. Hence the alignment algorithm should not only compare two structures but induce a global similarity measure between the two. This similarity measure is defined by:

$$S(A,B) = \sum_i \sum_j \left(0.2 - \frac{\left| d_{ij}^A - d_{ij}^B \right|}{d_{ij}^*} \right) e^{-(d_{ij}^*/20A^\circ)^2}, \tag{19}$$

where the summation is over all amino acids of the common core, d_{ij}^* denotes the arithmetic mean of the $C_\alpha - C_\alpha$ distances d_{ij}^A and d_{ij}^B of the proteins A and B, a relative deviation of 0.2 is the threshold of similarity and the exponential factor downweighs contribution from parts at longer distances. The optimal structural alignment is that set of equivalences (i^A, i^B) that maximizes S.

The DALI algorithm performs two steps for searching in large databases. In the first step a fast algorithm is used to compute a group of potential similarity candidates. In the second step a refinement is performed on the set of the previous step using slow but more sophisticated algorithms.

Since the protein structures are too large, they are cut into domains. In the "Dali domain dictionary" each domain is assigned a domain classification number DC_l_m_n_p representing:

(i) a fold space attractor region (l),
(ii) a globular folding topology (m),
(iii) a functional family (n) and
(iv) a sequence family (p).

The finest level of classification is level p and the highest level of the fold classification corresponds to level l. The most evolutionary interesting part of the DALI classification hierarchy is level m. The globular folding topology defines the fold type. Fold types are defined as clusters of structural neighbors in fold space with average pairwise Z-scores above 2. The Z-score is a statistical measure computed on the similarity value S of eq. 19. The Z-score associated with the ith observation of a random variable x is given by:

$$Z_i = \frac{x_i - \overline{x}}{\sigma}, \tag{20}$$

where x_i is the mean and σ the standard deviation of all observations $x_1, ..., x_n$.

5.2. *PRIDE*

In this approach by Carugo and Pongor the distribution of the $C_\alpha(i) - C_\alpha(i+n)$ distances in the range of $n = 3, ..., 30$ chain distance is used to describe the protein structure. For each protein the distance distribution is computed for $C_\alpha - C_\alpha$ pairs with a distance of n on the chain.

Hence we get 28 distance histograms associated with one protein structure. These distance histograms are then compared pairwise for two protein structure using contingency table analysis. This analysis answers the question: Is their a dependency between Structure 1 and Structure 2? This PRIDE score of the relative distances ranges between 0 and 1, where 1 implies the maximum dependency between the two structures.

5.3. *Gauss Integrals*

Rogen et al compute a set of 29 writhe-based features associated with the protein backbone structure. The backbone can be parametrized by a polygonal curve μ.

The writhe of a closed space curve γ can be computed by using the Gauss Integral:

$$Wr(\gamma) = \frac{1}{4\pi} \int \int_{\gamma \times \gamma \backslash D} \omega(t_1, t_2) dt_1 dt_2, \tag{21}$$

where

$$\omega(t_1, t_2) = \frac{[\gamma'(t_1), \gamma(t_1) - \gamma(t_2), \gamma'(t_2)]}{|\gamma(t_1) - \gamma(t_2)|^3} dt_1 dt_2, \tag{22}$$

D is the Diagonal of $\gamma \times \gamma$ and $[\gamma'(t_1), \gamma(t_1) - \gamma(t_2), \gamma'(t_2)]$ is the triple scalar product. The writhe can be described as the average signed number of crossings seen when averaged over all directions in 3D-space.

For a polygonal curve μ, the integral is reduced to a sum:

$$Wr(\mu) = I_{(1,2)}(\mu) = \sum_{1<i_1<i_2<N} W(i_1, i_2), \tag{23}$$

where $W(i_1, i_2)$ is the contribution to the writhe coming from the i_1th and i_2th line segment.

28 more writhe-based descriptors are constructed by taking absolute values and looking only at certain interesting configurations. The number of structural descriptors is 30 since the number of C_α atoms is also one of the descriptors.

6. Experiments

6.1. *Datasets*

The Protein Data Bank (PDB) [25] offers a large pool of protein structures. These structures can be classified using several different classification schemes. We decided to use the so called SCOP release 1.67 database [1], a hand-labeled protein archive with nearly 26000 entries. We used a subset of proteins to compute the evaluation within a reasonable amount of time. The SCOP-database is classified in a hierarchical manner by class/fold/superfamily/family. For the first experiment we took one rather large family from each class, resulting in a dataset of 2648 proteins divided in 10 classes, which are all very different ('all-classes'). To evaluate how the features can discriminate between more similar proteins, we created a second dataset with families belonging all to the class 'all-alpha', resulting in a set of 3656 proteins divided in 172 folds. The third dataset in our experiments was chosen as in [5]. It contains 687 proteins from 4 SCOP classes and 27 SCOP folds and is quite difficult to classify automatically, since the data set have less than 40% of the sequence identity for the aligned subsequences longer than 80 residues. The fourth data set 'cath' consists of the connected CATH 2.4 domains selected in reference [6]. The classes are defined as by the CATH 2.4 homology classes.

The number of entries and the classification used for the four data sets is represented in Table 1.

Table 1. **Testing datasets.** Number of domains for the classification of the four testing sets: 'all-classes', 'all-alpha', '27fold' and 'cath'. The '?' in the scopid indicates that all possibilities for this position in the scopid were included in the dataset.

dataset	**# of domains**	**classification level**	# of classification classes
all-classes (scopid ?.1.1.1)	2,650	SCOP-class	7
all-alpha (scopid a.?.?.?)	3,680	SCOP-fold	172
27fold	685	SCOP-fold	27
cath	20,937	CATH-homology	2147

6.2. *Implementation details*

Since our basic feature is a five dimensional histogram we need to choose a quantization for the bins. We have to find a trade off between the size of the features and accuracy of the representation. But, in fact, coarser discretization can sometimes also give better results due to the induced robustness achieved by the more tolerant bin assignments. For the cosines α, β and γ we just used two bins for each. We found that this very coarse quantization gives slightly better results. The distance between the two atoms is quantized

into 16 bins and the sequential distance μ is quantized into 8 bins. Thus, the multidimensional histogram contains $2^3 \cdot 16 \cdot 8 = 1024$ bins. For the Spherical Harmonic and D-Wigner case we have a collection of histograms parametrized over the frequency number l.

The width σ in equation (8) was chosen by 20Å. The sequence-distance μ is recorded in the interval from 0 to 40 and the distance Δ is measured in the interval from 0Å to 50Å. Both ranges are chosen very close to those used by the PRIDE features. For the SH-features we used $l_{max} = 2$. For the D-Wigner features we considered only $D^0(R) = 1$ and $D^1(R) = U^+RU$.

6.3. *Evaluation tools*

The evaluation tools from the Princeton Shape Benchmark (PSB) [24] were used for evaluation of the performance. The PSB provides a suite of tools for comparing shape matching and classification algorithms. The evaluation is based on five statistical measures: Nearest Neighbor (NN), First-Tier, Second-Tier, E-Measure and the Discounted Cumulative Gain (DCG). The same procedure starts the computation for all five measures: Each object of the database is taken as a query object and the distances to all other query objects are computed and stored in a distance matrix. The five statistical measures are computed based on the distance and the class label.

The *Nearest Neighbor* measures the percentage of the closest matches that belong to the same class as the query. This provides an intuition on how well a nearest neighbor classifier would perform. The desired value for this measure is of course 100%.

The *First-* and the *Second-Tier* measure the percentage of models in the query's class that appear within the top K matches, where K depends on the size of the query's class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier, and $K = 2(|C| - 1)$ for the second tier. The optimal result has the value 100%.

The *E-Measure* is a composite measure of the precision P and recall R for a fixed number of retrieved results, where P and R are defined by:

$$P = \frac{|\{relevant\ structures\}| \cap |\{found\ structures\}|}{|\{found\ structures\}|} \tag{24}$$

$$R = \frac{|\{relevant\ structures\}| \cap |\{found\ structures\}|}{|\{relevant\ structures\}|} \tag{25}$$

Since the user is more interested in the query results with a high similarity to the input query, only the first 32 most similar retrieved results are considered. After computing the precision and recall for those results, the E-Measure is obtained by:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}.$$

The higher the E-Measure value the better the result, with the perfect score being 100%.

The DCG weighs the results near the front of the list more than correct results later in the ranked list. For details on the computation of DCG see [24]. The best value for the DCG is 1.0.

6.4. *Results*

In Table 2 the results for the 'all-classes' dataset are presented. As expected, the classification rate is very high since division into SCOP classes is quite an easy task. The SH did not improve the already very good classification results. The D-Wigner results are worse than SH, although we would expect them to give better results.

In Table 3, the search is performed on the 'all-alpha' dataset. The results are not as good as on the 'all-classes'-dataset since the classification into folds is a more difficult task. However, 97.8% is a quite high classification rate and the SH improve the GI features.

Table 2. **Results 'all-classes'.** Results on the 'all-classes'-dataset with GI, SH and D-Wigner features.

Feature	1NN	1T	2T	EM	DCG
GI	99.8	86.8	91.4	13.4	96.7
GI with SH	99.8	87.6	92.5	13.4	97.2
GI with D-Wigner	99.5	86.1	89.9	13.3	96.3

Table 3. **Results 'all-alpha'.** Results on the 'all-alpha'-dataset with GI, SH and D-Wigner features.

Feature	1NN	1T	2T	EM	DCG
GI	97.4	84.8	88.6	35.6	94.4
GI with SH	97.8	89.3	92.2	37.4	96.0
GI with D-Wigner	97.4	87.5	90.4	36.8	95.2

In Table 4, the results for the '27-folds'-dataset are presented. They are much worse than the previous two testing sets since the domains of the '27fold' have less than 40% sequential similarity and are thus hard to classify. Also, the number of samples per class is far less than in the previous two sets. The number of samples per class is important, since it increases the probability to find a similar structure in one class.

Table 4. **Results '27fold'.** Results on the '27fold'-dataset with GI, SH and D-Wigner features.

Feature	1NN	1T	2T	EM	DCG
GI	77.3	31.0	41.2	27.2	67.9
GI with SH	78.8	32.4	44.7	28.7	69.3
GI with Dwigner	77.8	29.5	39.1	26.2	66.8

In Table 5, the results for the 'cath' - dataset are presented. They are very good since the homologous are very well populated and therefore one similar structure to the query structure could be always retrieved.

Table 5. **Results 'cath'.** Results on the 'cath'-dataset with SH and D-Wigner features.

Feature	1NN	1T	2T	EM	DCG
GI with SH	98.9	72.6	77.7	41.2	91.1
GI with Dwigner	98.8	71.0	75.2	41.0	89.9

The DALI server [1] provides a standalone application called DaliLite. This program was used to compute the alignments and the pairwise Z-scores of the '27fold' resulting in 235,641 alignments. The results of the evaluation are presented in Table 6. DALI performs better by 6.3% than the proposed method. However, the classification time is one week by DALI as opposed to 2 minutes by the proposed method.

Table 6. **Comparison of results with DALI.** Comparison of the results on the '27fold'-dataset computed by DALI and by the new method.

Feature	1NN	1T	2T	EM	DCG
GI with SH	78.8	32.4	44.7	28.7	69.3
DALI	85.1	59.1	67.8	45.0	82.8

In Table 7, the results of the proposed method are compared to the results obtained by PRIDE and the Gauss Integrals. The implementation of the PRIDE features was performed as in [4]. However, the bins of the histogram were not combined to contain a certain number of samples. Thus, the PRIDE score was not evaluated by contingency table analysis, but just simply using the $L1$ norm. The results obtained by the new method are better, especially for the '27fold' dataset.

For the computation of the Gauss Integral features we used the program [2] provided by the authors of [6]. In fact, the SH features perform better than Gauss integrals. For the difficult '27fold' dataset, SH outperform Gauss features by 8.1%.

[1] http://www.ebi.ac.uk/dali/

[2] http://www2.mat.dtu.dk/people/Peter.Roegen/Gauss_Integrals.html

Table 7. **Comparison with PRIDE and Gauss Integral features.**

dataset	Feature	1NN	1T	2T	EM	DCG
all-classes	GI with SH	99.8	87.6	92.5	13.4	97.2
	PRIDE	99,7	84.8	88.2	13.3	96
	Gauss	99.2	73.3	81.2	12.1	93.6
all-alpha	GI with SH	97.8	89.3	92.2	37.4	96.0
	PRIDE	96.8	80.7	85	34.3	92.7
	Gauss	94.2	63.8	72.9	29.5	87.0
27fold	GI with SH	78.8	32.4	44.7	28.7	69.3
	PRIDE	70.7	29.4	38.9	25.9	65.1
	Gauss	67.6	26.1	35.5	23.2	63.3
cath	GI with SH	98.9	72.6	77.7	41.2	91.1
	PRIDE	98.8	66.8	73.2	39.1	88.8
	Gauss	98.4	69.8	76.4	40.2	90.0

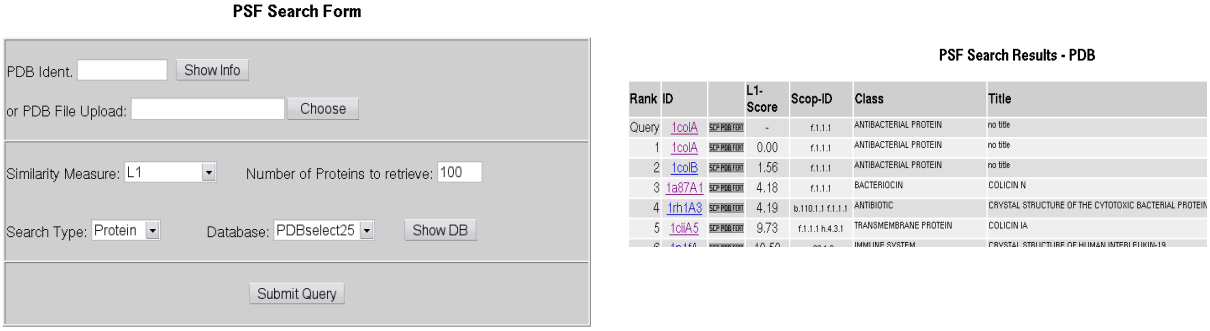

Figure 3. Screenshots from the Web interface. *http://lmb.informatik.uni-freiburg.de/cgi-bin/reisert/PSF2Search.cgi* On the right the Search Form which helps you to make query, on the left, an example for the resultpage.

7. A Web Interface

We also provide a web interface to test the proposed method. It is located under *http://lmb.informatik.uni-freiburg.de/cgi-bin/reisert/PSF2Search.cgi*. The underlying features are a more compact version of those used in this paper. We used histograms over $\alpha, \gamma, \Delta, \mu$ with discretization $2, 2, 8, 8$. This results in a feature histogram of size 256. The search is performed on the whole PDB consisting of approx. 30000 proteins (last update 6.06.05). In Figure 3 we show some screen shots from the interface. One is able to choose different kind of parameters and subsets of the database. It is also possible to upload structures in ordinary pdb format and search for it. For each structure it is possible to visualize the features via the *Show Info* button. Each structure is additionally subdivided into domains, so different kind of search types are possible, protein search, chain search or domain search. To compute a distance between two structures consisting of several domains, all possible pairwise distances are computed. The pair with the smallest distance serve as the distance measure between the two structures. Besides, the ordinary search by query it is also possible to compute distance matrices for a set of structures. This helps to get more intuition about the feature-induced metric. To get a more detailed help and description of the web interface we refer to the help page on the web.

8. Conclusion

We introduced a new method for protein classification based on group integration using spherical harmonics. Our method works very fast while at the same time achieving very good results. Only structural and sequential information is used for the comparison of proteins. No other properties like hydrophobicity, temperature or amino acid class are used. Classification precision is over 99% for discriminating on all SCOP classes and over 97% for discrimating SCOP folds in one SCOP class. Even on the difficult '27 folds' dataset we achive $78,8\%$. The group integration results could be significantly improved using SH-features. However, the high time consuming D-Wigner matrices did not improve the results. SH-features are better than PRIDE and the Gauss integral features and almost as good as DALI. Unfortunately, we could not

test DALI on a larger dataset because of its tremendous time requirements. Our goal is to achieve same accuracy as DALI while at the same time keeping the computational cost low.

For future work, we suggest to add more information to the kernel. For this, the chemical properties of the amino acids can be considered. The hydrophobicity of one amino acid plays an important role for the protein folding and hence for its shape. And finally we would like to explore feature selection techniques on the SH-features.

Acknoledgements

The authors would like to thank anonymous reviewers for their valueable comments.

References

[1] Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C., 1995, SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol,* **247**, 536–540.

[2] Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., and Thornton, J.M., 1997, CATH- A Hierarchic Classification of Protein Domain Structures. *Structure,* **5(8)**, 1093–1108.

[3] Holm, L. and Sander, C., 1998, Touring protein fold space with Dali/FSSP. *Nuc. Acids,* **26**, 316–319.

[4] Carugo, O. and Pongor, S., 2002, Protein Fold Similarity Estimated by a Probabilistic Approach Based on C_α-C_α Distance Comparison. *J. Mol. Biol.,* **315**, 887–898.

[5] Huang, C.D., Lin, C.T. and Pal, N.R., 2003, Hierarchical Learning Architecture With Automatic Feature Selection for Multiclass Protein Fold Classification. *IEEE Trans. on Nanobioscience,* **2(4)**, 221–232.

[6] Rogen, P. and Fain, B., 2003, Automatic classification of protein structure by using Gauss integrals. *Proc.Nat.Sci USA,* **100(1)**, 119–124.

[7] Carugo, O., 2006, Rapid methods for comparing protein structures ans scanning structure databases. *Current Bioinformatics,* 75–83.

[8] Lancia, G., Carr, R., Walenz, B. and Istrail, S., 2001, Optimal PDB structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. *RECOMB,* 193–202.

[9] Chou, Kuo-Chen , 2004, Structural Bioinformatics and its Impact to Biomedical Science *Current Medicinal Chemistry, Volume 11, Number 16, August 2004, pp. 2105-2134(30)*

[10] Chou, Kuo-Chen; Wei, Dong-Qing; Du, Qi-Shi; Sirois, Suzanne; Zhong, Wei-Zhu, Progress in Computational Approach to Drug Development Against SARS *Current Medicinal Chemistry, Volume 13, Number 27, November 2006, pp. 3263-3270(8)*

[11] Burkhardt, H. and Siggelkow, H., 2001, Invariant features in pattern recognition - fundamentals and applications. *In Nonlinear Model-Based Image/Video Processing and Analysis* (John Wiley and Sons), pp. 269–307.

[12] Haasdonk, B., Halawani, A. and Burkhardt, H., 2004, Adjustable invariant features by partial Haar-integration. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR),* **2**, 769–774.

[13] Haasdonk, B., Vossen, A. and Burkhardt, H., 2005, Invariance in Kernel Methods by Haar-Integration Kernels. *Proceedings of the 14th Scandinavian Conference on Image Analysis,* 841–851.

[14] Ronneberger, O., Burkhardt, H. and Schultz, E., 2002, General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants. *Proceedings of the International Conference on Pattern Recognition,* (Quebec, Canada).

[15] Ronneberger, O., Fehr, J. and Burkhardt, H., 2005, Voxel-Wise Gray Scale Invariants for Simultaneous Segmentation and Classification. *In Proceedings of the 27th DAGM Symposium, Vienna, Austria,*.

[16] Schael, M., 2002, Invariant Texture Classification Using Group Averaging with Relational Kernel Functions. *Texture 2002 the 2nd international workshop on texture analysis and synthesis,* 129–134.

[17] Siggelkow, S., Schael, M. and Burkhardt, H., 2001, SIMBA - Search IMages By Appearance. *In Proceedings of the 23rd DAGM Symposium,* 9–16.

[18] Setia, L., Ick, J. and Burkhardt, H., 2005, SVM-based Relevance Feedback in Image Retrieval using Invariant Feature Histograms. *In Proc. of the IAPR Workshop on Machine Vision Applications,* 542–545.

[19] S. M. Shamsuddin, M. N. Sulaiman and M. Darus, 2002, Invarianceness of Higher Order Centralised Scaled-invariants Undergo Basic Transformations result. *International Journal of Computer Mathematics* **79(1)**, 39–48.

[20] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D., 2001, Matching 3D Models with Shape Distribution. *Proceedings Shape Modeling International*.

[21] Reisert, M. and Burkhardt, H., 2006, Irreducible Group Representation for 3D Shape Description. *Proceedings of the DAGM'06, Berlin, Germany, pages 132-142*

[22] Reisert, M. and Burkhardt, H., 2006, Invariant Features for 3D-Data based on Group Integration using Directional Information and Spherical Harmonic Expansion. *Proceedings of the ICPR'06, Hong Kong*

[23] Shindyalov, I.N. and Bourne, P.E., 1998 Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering 11(9) 739-747.*

[24] Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T., 2004, The Princeton Shape Benchmark. *Shape Modeling International, Genova, Italy.*

[25] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhata, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E., 2000, The Protein Data Bank. *Nucleic Acids Research,* **28**, 235–242.