Protein Fold Recognition Using Geometric Kernel Data Fusion

Pooya Zakeri 1,2, Ben Jeuris 3, Raf Vandebril 3, Yves Moreau 1,2 *

1 Department of Electrical Engineering, ESAT, SCD-SISTA KU Leuven, Leuven, Belgium.
2 Future Health Department, iMinds, Leuven, Belgium.
3 Department of Computer Science, KU Leuven, Leuven, Belgium.

ABSTRACT
Motivation: Various approaches based on features extracted from protein sequences and often machine learning methods have been used in the prediction of protein folds. Finding an efficient technique for integrating these different protein features has received increasing attention. In particular, kernel methods are an interesting class of techniques for integrating heterogeneous data. Various methods have been proposed to fuse multiple kernels. Most techniques for multiple kernel learning focus on learning a convex linear combination of base kernels. In addition to the limitation of linear combinations, working with such approaches could cause a loss of potentially useful information.

Results: We design several techniques to combine kernel matrices by taking more involved, geometry inspired means of these matrices instead of convex linear combinations. We consider various sequence-based protein features including information extracted directly from position specific scoring matrices and local sequence alignment. We evaluate our methods for classification on the SCOP PDB-40D benchmark dataset for protein fold recognition. The best overall accuracy on the protein fold recognition test set obtained by our methods is about 86.7%. This is an improvement over the results of the best existing approach. Moreover, our computational model has been developed by incorporating the functional domain composition of proteins through a hybridization model. It is observed that by using our proposed hybridization model the protein fold recognition accuracy is further improved to 89.30%. Furthermore, we investigate the performance of our approach on the protein remote homology detection problem by fusing multiple string kernels.


Contact: pooya.zakeri@esat.kuleuven.be

1 INTRODUCTION
Knowledge on functions of proteins can be provided by information about their tertiary structure; hence determining this structure is among the most essential objectives in molecular biology, cell biology, proteomics, and bioinformatics. Structural information also provides a much better understanding of protein-protein interaction. Furthermore, this information is potentially useful for drug design studies. Unfortunately, experimentally identifying the three-dimensional structure of proteins is expensive and time consuming. By contrast, recent development in genome sequencing projects have tremendously increased the number of protein coding sequences. Since there is much slower growth in information on 3D structure, there is an increasing gap between the protein sequence information and protein structure information. Despite these problems, knowledge about protein folds can be useful in determining its structural properties. Because of the limitation of homology modelling methods, when there is no sequence similarity to homologous proteins of known structure, the taxonomic approach is usually considered as a trustworthy alternative. This approach is based on the assumption that the number of protein domain folds is restricted (Dubchak et al., 1999; Murzin et al., 1995). Promising results are reported using taxonomic approaches (Ding and Dubchak, 2001; Shen and Chou, 2006; Yang et al., 2011), but they are still far from tackling the classification of protein folds completely. So, fold recognition or protein threading is still among the most challenging tasks in bioinformatics. In many bioinformatics tasks, it is worthwhile to consider several representations of the data, which will not always be vectors. In particular, we should be able to deal with them using the same algorithm, regardless whether they are represented as binary vectors, real vectors on different scales, sequences, graph data, etc. Various approaches based on features extracted from protein sequence and often machine learning approaches have been used to tackle the fold recognition problem. Several informative fold data sources can be constructed based on various representative models of protein features, such as primary structural information (Ding and Dubchak, 2001; Chen and Kurgan, 2007; Yang et al., 2011), local pairwise sequence alignment-based feature spaces (Damoulas and Girolami, 2008), physicochemical properties of constituent amino acids (Ding and Dubchak, 2001; Lin et al., 2013), and sequence evolution information (Shen and Chou, 2009; Kavousi et al., 2013; Chen and Kurgan, 2007; Yang et al., 2011; Sharma et al., 2013). More attention needs to be paid to finding an efficient and cost-effective technique for integrating these different discriminatory data sources for protein fold classification. Nevertheless, to deal with biological data, there are not only a lot of issues in machine

* yves.moreau@esat.kuleuven.be

© The Author(s) 2014. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com
learning algorithms, but also a lot of difficulties in data analysis. Full integration and decision integration are common techniques for fusing proteins fold data sources. In particular, full integration is a fast and easy way to fuse data sources. However, because of the heterogeneity of the biological data, combining data sources at the data level is not always feasible in practice. By contrast, fusing data sources at decision level, such as in the ensemble learning framework, is considered as an intuitive manner to deal with heterogeneous data. Various decision-based integration approaches have been proposed for protein fold classification (Nanni, 2006; Shen and Chou, 2006, 2009; Yang et al., 2011; Kavousi et al., 2011; Lin et al., 2013). In addition to limitations of employing ad hoc ensemble learning, the computational cost of decision-based approaches increases corresponding to the number of data sources.

The heterogeneous biological data sources can also be integrated intelligently using partial integration, such as kernel-based data fusion. Using kernel methods is an elegant and versatile strategy because it decouples the original data from the machine-learning algorithms by using a representation of the data as a kernel matrix. The main idea behind kernel methods is, rather than using original data directly to use only a kernel matrix. Symmetric positive definite (SPD) kernel matrices are indeed the nonlinear extension of covariance/correlation matrices and encode the similarity between samples in their respective input space. This implies that the heterogeneous data (binary vectors, real vectors on different scales, graph data) can all be replaced by appropriately scaled kernel matrices, which all have the same size, and thus that the data heterogeneity disappears. Then other algorithms (such as classification, clustering and prioritization) can access the same data, which is currently not possible. Indeed, constructing the same representation for all data sets and integrating these representations systematically is the main intuition behind kernel fusion methods. In the simplest scenario, we can compute kernel matrices separately for each data source and then average them together.

The standard approach for combining kernel matrices is to take the (weighted) arithmetic average. There are several methods for obtaining a valid and fitting kernel by tuning the kernel matrices weights (Gönen and Alpaydin, 2011). Finding such weights from training data and replacing the single kernel by a linear combination of weighted base kernels is usually referred to as multiple kernel learning (MKL). These weights can also be interpreted as their corresponding importance in the fused kernel. During the last decades, several MKL methods have been proposed in the literature (Lanckriet et al., 2004a,b; Bach and Lanckriet, 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Vishwanathan et al., 2010) and are shown to yield good results in various applications, in particular in bioinformatics applications (Lanckriet et al., 2004b; Zien and Ong, 2007; De Bie et al., 2007; Ying et al., 2009; Yu, 2011). Most of these approaches try to learn a linear combination of base kernels, which can be interpreted as the concatenation of the base kernel feature space or an “OR” combination of the individual kernels.

The kernel integration problem is often reduced to a convex optimization problem. In addition to the limitation of linear combinations, solving this optimization problem is only possible for a small number of kernels and small number of data points. Furthermore, since this type of averaging is often sensitive to deal with complementary and noisy kernels, it is not very appropriate for biological data. In fact, going with such approaches could cause a loss of potentially useful latent information in the data.

Recent biological applications have demonstrated that even using uniformly weighted kernel integration can boost the generalization capability of the decision function (Lanckriet et al., 2004b; De Bie et al., 2007; Ying et al., 2009; Daemen et al., 2009). By contrast, the results obtained by employing such averaging of the kernel matrices are comparable to the results of the best existing MKL approaches in general applications (Lanckriet et al., 2004b; De Bie et al., 2007; Ying et al., 2009). Hence, using the uniformly weighted average of the base kernels can be considered as a reliable and computationally more scalable alternative. Uniformly weighted kernel integration can also be considered as the arithmetic mean (AM) of kernel matrices, which is always a generator of a valid Mercer kernel. Similar to the arithmetic mean, other types of means of SPD matrices (such as the harmonic mean (HM), Log-Euclidean mean (LogEM) (Arsigny et al., 2007) and geometric mean (GM)) result in SPD kernels. In this study, we propose and develop several new techniques that combine the Mercer kernel matrices through other types of averaging than convex linear combination. Such averaging of the base kernels can be interpreted as a kind of fusion that expresses the nonlinear relationship between the individual kernels. In particular, we focus on taking the matrix geometric mean of base kernels. However, computing the geometric mean of a general number of SPD matrices is a challenge. In fact, for a general number of SPD matrices, a proper definition of a geometric mean with some natural properties has only recently been developed (Bhatia, 2007). We present two methods for computing the geometric mean. The first approach is focused on computing the actual geometric mean using the definition of the Karcher mean (Jeuris et al., 2012). The second, however, only computes a rough approximation of the actual geometric mean using a proposed, heuristic method based on Arithmetic-Geometric-Harmonic (AGH) mean. We show in the second section that it is a computationally scalable method for computing an approximate geometric mean. We also consider the behaviour of combining kernels by taking HM and log-EM, where this last one can be seen as a consensus between the arithmetic and geometric mean. Moreover, our computational model has been developed by incorporating the functional domain information through the hybridization model. Experimental results on the SCOP PDB-40D benchmark data set (Ding and Dubchak, 2001) demonstrate that our integration technique can effectively improve the accuracy of the state-of-the-art kernel fusion model.

2 GEOMETRIC KERNEL FUSION (GKF)

To improve the efficiency of kernel data fusion through the convex combination of kernel matrices, there are several complex convex optimization-based approaches (Lanckriet et al., 2004a; Bach and Lanckriet, 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Vishwanathan et al., 2010) that try to optimize the kernel weights based on different optimization criteria. The optimized weights of kernel matrices reflect the relative importance of the different data set in the fused kernel. It is expected that the kernel matrices that have more information than others receives higher weights in such weighted convex linear combination. However, convex combination of kernel matrices often leads to mixed results. Moreover, it has also been shown that optimization of weights...
causes an improvement in performance only when dealing with redundant or noisy kernel matrices (Lanckriet et al., 2004b). Indeed, linear convex combination of kernel matrices often fails to fully capture all the information for kernels containing complementary, non-redundant information. This is however a typical situation in biological applications. This is also affirmed by the equal weights theorem (Wainer, 1976), which states when all optimized weights are uniformly distributed on the interval [0.25; 0.75], the performance is barely changed using equal weights. Therefore, when dealing with many data sources, which are all not informative, a more practical scenario could be to select the reliable data sources and discard the rest, then take an unweighted averaging between kernel matrices. Using the Euclidean distance on a convex cone whose interior contains all SPD matrices \(\mathcal{P}(n)\), we can obtain the arithmetic mean. For a given set of SPD kernel matrices \(K_1, K_2, ..., K_n\), the arithmetic mean is given by \(A(K_1, K_2, ..., K_n) = \frac{1}{n} \sum_{i=1}^{n} K_i\). By contrast, since it has been shown that this type of averaging mixed the result and has usually sensitive behaviour in dealing with complementary and noisy kernels, Euclidean distance on SPD matrices might not be appropriate. Moreover, SPD matrices form a convex cone and not a vector space. This has an effect on the “natural” geometry of SPD matrices, which may not be Euclidean, but rather should rely on concepts from Riemannian geometry. This motivates us to think about other means between SPD matrices that are not relative to the Euclidean distance on \(\mathcal{P}(n)\) and necessarily a linear combination of SPD matrices. For example, the mean corresponding to Riemannian distance on \(\mathcal{P}(n)\) is the geometric mean. For a given set of SPD kernel matrices \(K_1, K_2, ..., K_n\), the geometric mean \(G(K_1, K_2, ..., K_n)\) is the unique solution of the nonlinear matrix equation \(\sum_{i=1}^{n} \log(K_i^{-1}K) = 0\). Because of the non-commutative property of matrix multiplication, the equation can not be solved in closed form. However, the geometric mean of two SPD kernel matrices \(K_1\) and \(K_2\) can be defined explicitly as (Bhatia, 2007):

\[
G(K_1, K_2) = K_1^{\frac{1}{2}} (K_1^{-\frac{1}{2}} K_2 K_1^{-\frac{1}{2}})^{\frac{1}{2}} K_1^{\frac{1}{2}}. 
\]  

(1)

This leads to the fusion of \(K_1\) and \(K_2\) as \(F(K_1, K_2) = G(K_1, K_2)\). The geometric mean has several properties that make it useful, of which an important one is its invariance under inversion. On the contrary, the arithmetic mean is not invariant under inversion, which means that if \(K = \frac{1}{2} \sum_{i=1}^{n} K_i\), then in general, \(K^{-1} \neq \frac{1}{2} \sum_{i=1}^{n} K_i^{-1}\). This property becomes interesting when kernel matrices are considered under analogy to covariance matrices of Gaussian distributions. In the Gaussian case, the covariance matrix \(K\) can be used as a positive semi-definite kernel representation of a data sources. But the covariance matrix is not the most interesting object to investigate. For a multivariate normal distribution, the precision matrix \(P\) (which is the inverse of the covariance matrix, \(P = K^{-1}\)) encodes independence relations between variables in the form of partial correlations. Zeros in the precision matrix indicate some notion of partial correlation independence between two variables. Some immediate manipulations result in equalities such as \(G(P_1, P_2)^{-1} = G(P_1^{-1}, P_2^{-1}) = G(K_1, K_2)\) and \(G(K_1, K_2)^{-1} = G(K_1^{-1}, K_2^{-1}) = G(P_1, P_2)\). Hence, computing the geometric mean of the covariance matrices is equivalent to computing the geometric mean between the precision matrices, which is a particularly attractive idea in the case of Gaussian distributions, and may thus be a valuable property when fusing kernels. For a general number of matrices, the fused kernel is obtained by taking the geometric mean

\[
F(K_1, K_2, ..., K_n) = G(K_1, K_2, ..., K_n). 
\]  

(2)

We describe our proposed methods for computing the geometric mean of SPD matrices and some approximations in the following sections.

2.1 Karcher mean and AGH mean

For two SPD matrices \(A\) and \(B\), the geometric mean is given by the explicit formula (1). However, for more than two matrices a proper definition of a geometric mean with some natural properties remained elusive for long. The most popular instance of the matrix geometric mean is considered to be the Karcher mean (Jeuris et al., 2012). The Karcher mean of SPD matrices \(A_1, ..., A_k\) is defined as the barycenter of these matrices on the manifold of SPD matrices with its Riemannian geometry. In practice, this is obtained by searching the minimizer of an optimization problem, given as follows

\[
G(A_1, ..., A_k) = \min_{X \in \mathcal{P}(n)} \sum_{i=1}^{k} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2, 
\]  

(3)

where \(\mathcal{P}_n\) represents the set of SPD \(n \times n\) matrices and \(\| . \|_F\) is the Frobenius norm. To find the minimizer, we use manifold optimization (Absil et al., 2008) (for more details see the Supplementary Data). However, retrieving the Karcher mean can be computationally expensive, which is why we also discuss the AGH mean, which can be considered as an approximation to the Karcher mean. For every two positive scalars, alternatively computing the AM and HM repeatedly will converge to the geometric mean. At the base of the AGH mean lies the observation that the geometric mean of two matrices can be obtained by taking the arithmetic and harmonic mean (for more details see the Supplementary Data) of the matrices and iteratively repeating this procedure with the new matrices (Foster and Phillips, 1984). Generalizing this to more than two matrices, we duplicate the original set of matrices and combine both in arithmetic and harmonic operations, as illustrated by the matrices \(B_i\) and \(C_i\) in Algorithm 1. To counteract the decrease of speed of this technique, a randomization is introduced (last step in Algorithm 1). The result is a rapidly converging algorithm that provides a decent approximation to the Karcher mean.

This approximate mean requires a computational cost of the order \(O(n^3 \log(n))\) per iteration, which is an improvement when compared to the Karcher mean. The stopping criteria of the algorithm are the same as those of the Karcher mean, except when determining the distance between the consecutive iterations, where only the first of the \(B_i\)-matrices is considered. The kernel fusion framework approaches using the Karcher and AGH mean are called Karcher-KF (GKF1) and AGH-KF (GKF2) respectively.

2.2 Log-Euclidean mean

In this section we describe a new approach (Arsigny et al., 2007) to compute a mean of SPD matrices called the Log-Euclidean mean. Given SPD matrices \(S_1, ..., S_k\), their Log-Euclidean Fréchet mean exists and is uniquely given by the explicit formula

\[
F(S_1, ..., S_k) = \log(\exp(\frac{1}{k} \sum_{i=1}^{k} \log(S_i)));
\]  

(4)

We describe our proposed methods for computing the geometric mean of SPD matrices and some approximations in the following sections.
Algorithm 1 The approximate AGH mean algorithm where \( \overline{A} \) denotes the arithmetic mean and \( H \) the harmonic mean

Let \( A_1, \ldots, A_k \) be SPD matrices

- For all \( i \) set \( B_i = A_i \) and \( C_i = A_i \);
- while not converged
  - For all \( i \) set \( \tilde{B}_i = H(B_i, C_i(\text{mod } n + 1)) \);
  - For all \( i \) set \( \tilde{C}_i = A(B_i, C_i(\text{mod } n + 1)) \);
  - For all \( i \) set \( C_{p(i)} = \tilde{C}_i, B_{p(i)} = \tilde{B}_i \), with \( p \) a random permutation of \( [1, \ldots, n] \);
- end

\[
\mathbb{E}_{LE}(S_1, \ldots, S_N) = \exp \left( \frac{1}{N} \sum_{i=1}^{N} \log \left( S_i \right) \right).
\]  

The Log-Euclidean mean is similarity-invariant, invariant by group multiplication, inversion and exponential-invariant. The Log-Euclidean mean also has outstanding behavior with respect to the determinant (for more details see the Supplementary Data). Because of the nice properties of the Log-Euclidean mean and the high computational cost of geometric mean, it will also be considered in our fusion framework for combining kernels (LogE-KF).

3 MATERIAL AND METHODS

3.1 Benchmark data

We use the benchmark data set from Ding and Dubchak (DD) (Ding and Dubchak, 2001), which has been widely used for evaluating protein fold recognition predictors. This benchmark data set consists of 27 SCOP fold classes for 694 protein domains (311 proteins for the training set and 383 proteins for the test set). The identity between any two proteins in the classes for 694 protein domains (311 proteins for the training set and 383 proteins for the test set) is known as the integrated FunD database, to identify functionally characterized a new protein sequence. Moreover, we use the InterPro database (Apweiler et al., 2002, 2003) from Position Specific Scoring Matrices (PSSM) (Marchler-Bauer et al., 2007, 2013), which is known as the integrated FunD database, to identify the putative function of a new protein sequence.

Sequence evolution information Recently, sequence evolution information is often used to perform protein fold classification (Shen and Chou, 2009; Yang and Chen, 2011; Kavousi et al., 2011; Sharma et al., 2013) since good results can be obtained when using such information to determine protein secondary structure (Kaur and Raghava, 2003), subcellular localization (Xie et al., 2005; Rashid et al., 2007) and subnuclear localization (Shen and Chou, 2007). In particular, promising results have been reported recently using only the sequence evolution information through a new feature extraction method (Sharma et al., 2013) from Position Specific Scoring Matrices (PSSM) (Schaffer et al., 2001).

A protein sample \( P \) with \( L \) amino acid residues can be represented by its evolutionary information through PSSM or Position Specific Frequency Matrices (PSFM) profiles (Rangwala and Karypis, 2005), which both have \( L \) columns and 20 rows. Each row of PSSM \((M_{i,j})\) represents the log-likelihood of the residue substitution at the corresponding position in the protein sequence. In particular, the \((i,j)\)-th entry of the PSSM matrix \((M_{i,j})\) represents the possibility of the amino acid type \( j \) appearing in the \( i \)-th position of the protein domain during the evolution process. The PSSM entries are obtained using the PSI-BLAST program to search the non-redundant (NR) protein database, like the Swiss-prot database, through 3 iterations with the E-value cutoff set to 0.001. We use four common profile-based representative models of protein sequences:

1. A 400 dimensional feature vector created by summing up each column of the same amino acid in the PSSM and dividing by the length of the protein domain, followed by a normalization \( [\frac{1}{n} + 0.5] \) that scales each score to the range of \([0,1]\) (PS1).  
2. A 20 dimensional feature vector created by summing up each column in the PSSM profile and dividing by the length of the domain (PS2).
3. A 20 dimensional feature vector created by summing up each column in the PSSM profile and dividing by the length of the domain (PS3).
4. The PsePSSM was originally introduced in (Chou, 2001) to avoid complete loss of the sequence-order information (for more details see the Supplementary Data).

Functional domain composition To incorporate the available functional domain information (FunD) of proteins, we consider the FunD composition of protein sequences using integrated FunD databases, which contain protein sequences with noted FunD descriptions. A protein sequence can be summarized by its known functional domains. This representative model for a protein sequence was first introduced in (Cai et al., 2002, 2003) and is also considered for protein fold classification (Chen and Chou, 2009), protein structural recognition (Chou and Cai, 2004), protein subcellular localization prediction (Cai et al., 2002), and prediction of protein submichondria locations (Zakeri et al., 2011). In fact, fold information is a useful clue in determining a protein’s tertiary structure, which can facilitate the identification of its function. Hence, the FunD composition features are considered based on the rationale that the function of a protein is often correlated with its structural characteristics. For this purpose, we use the InterPro database (Apweiler et al., 2001; Hunter et al., 2012), which is an integrated database of recognized protein families, domains and functional sites to functionally characterize a new protein sequence. Moreover, we employ the Conserved Domain Database (CDD) (Marchler-Bauer et al., 2007, 2013), which is known as the integrated FunD database, to identify the putative function of a new protein sequence.

4 RESULT AND DISCUSSION

In this section, we discuss the extensive study of integrating multiple informative fold data sources. First, we focus on the individual performance of each protein feature data source. We should note that the geometric mean applies only to SPD matrices. Besides the flexibility of the radial basis function (RBF) kernel function and its good generalization through the non-linear mapping of the input space to the infinite-dimensional feature space,
the RBF kernel function produces SPD matrix. Two types of Gaussian RBF kernel functions are employed for these data sources. Then, classification is performed using a Gaussian support vector machine (SVM) model and its performance is estimated on an independent test set. Parameter selection details are provided in Supplemental data. A one-against-other SVM classifier is constructed on each representative model of the protein samples. To train SVMs, we used LIBSVM-3.1 implementation of the SVM algorithm (Chang and Lin, 2011). The performance of the individual classifiers on the test data is listed in Supplemental Table 2.

Next, to see the advantage of fusing heterogeneous data sources for protein fold classification through intermediate-based data integration, we focus on combining 26 RBF kernel matrices derived from each view on protein domains. The kernel matrices are combined through various types of means like GKF1, GKF2, AM, HM, and LogE. Afterwards, the combined kernel is used to determine the performance. Furthermore, to compare the performance of our proposed approaches, we also consider three types of MKL approaches (MKLdiv-dc and MKLdiv-conv (Ying et al., 2009), and SimpleMKL (Rakotomamonjy et al., 2008)), which have already been used for protein fold classification (Ying et al., 2009). We also consider a heuristic and simple MKL method (Qu and Lane, 2009), which chooses the kernel weights based on the relationship between the kernel matrix and the covariance matrix of the target labels (AK-MKL) (for more details, see the supplemental data). Then, a one-against-other SVM classifier is again constructed, now based on each of the combined kernels. The parameter $C$ is chosen through 5-fold cross validation and is searched over a grid of values $C = \{2^{-1}, 2^0, \ldots, 2^9\}$. Table 1 provides the total prediction accuracies of the existing approaches for classification of protein folds in the DD data set. Table 1 also lists the success rates of our proposed kernel fusion approaches based on averaging of the kernel matrices. According to Table 1, classification results of the combined kernels using Karcher-KF, AGH-KF, and LogE-KF show considerable improvement compared to the state of the art. Figure 2 illustrates the behaviour of integrating kernel matrices using GKF1, GKF2, and LogE-KF. According to Table 1, the performance of GKF1, GKF2, and LogE-KF including all 26 sequence-based features achieves a test accuracy of 86.68%, 86.16%, and 81.72% respectively. This implies that, in terms of similarity between protein samples, the fused kernel based on our proposed alternative algorithm (AGH mean) holds the same information as the fused kernel obtained using the Karcher mean, while the computational cost is lower. Also, promising results are achieved by our alternative fusion approach using the Log-Euclidean mean, which has an even lower computational cost.

In Figure 2, we consider the effect of sequentially incorporating sequence-based features according to the decreasing order of their kernel performances. The performance of uniformly weighted linear combinations of base kernels increases slowly by varying degrees until we include the sixteen most informative data sources, resulting in a best performance of 73.37%. By contrast, its performance decreases continuously if we continue to incorporate less informative protein features. However, there is a slight rise after adding PSp9 and then the performance decreases again when combining all kernels. This observation suggests that sequence-based PsePSSM features that reflect the effect of sequence order carry almost no complementary information with other protein features extracted from the PSSM profile (PS1, PS2, PS3, PSp0). Similar trends are apparent for MKLdiv-dc, MKLdiv-conv and KA-MKL. Contrary to the previous methods, the performance of AGH-KF increases gradually even when adding kernels considered to carry non-complementary information by AM. Its success rate is consistently outperforming other uniformly weighted kernel integration methods and almost always increases until the 26th kernel is included, resulting in the best performance of 86.68%. The experimental results on the SCOP PDB-40D benchmark dataset demonstrate that the geometric-based averaging of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion model. According to Table 2, promising test set accuracy is obtained using each individual FunD information-based feature (FunD-cdd and FunD-InterPro). Now, our

### Table 1. Comparison of proposed models with the existing predictor and Meta-predictors

<table>
<thead>
<tr>
<th>Methods</th>
<th>PERF description</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>56</td>
<td>(Ding and Dubchak, 2001)</td>
</tr>
<tr>
<td>SE</td>
<td>61.1</td>
<td>(Nanni, 2006)</td>
</tr>
<tr>
<td>PFP-Pred</td>
<td>62.1</td>
<td>(Shen and Chou, 2006)</td>
</tr>
<tr>
<td>PFRES</td>
<td>68.4</td>
<td>(Chen and Kurgan, 2007)</td>
</tr>
<tr>
<td>VDKC</td>
<td>68.1</td>
<td>(Damoulas and Girolami, 2008)</td>
</tr>
<tr>
<td>MKLdiv-dc</td>
<td>73.36</td>
<td>(Ying et al., 2009) 12 PFs</td>
</tr>
<tr>
<td>MKLdiv-conv</td>
<td>71.01</td>
<td>(Ying et al., 2009) 12 PFs</td>
</tr>
<tr>
<td>MKLdiv-dc</td>
<td>75.19</td>
<td>(Ying et al., 2009) 7 PFs</td>
</tr>
<tr>
<td>PFP-FunDisEG</td>
<td>70.5</td>
<td>(Shen and Chou, 2009)</td>
</tr>
<tr>
<td>Classifier Fusion</td>
<td>67.02</td>
<td>(Kavousi et al., 2011)</td>
</tr>
<tr>
<td>MarFold</td>
<td>71.7</td>
<td>(Yang et al., 2011)</td>
</tr>
<tr>
<td>Tax-Fold</td>
<td>71.5</td>
<td>(Yang and Chen, 2011)</td>
</tr>
<tr>
<td>Bi-grams</td>
<td>69.5</td>
<td>(Sharma et al., 2013)</td>
</tr>
<tr>
<td>HPFP</td>
<td>74.21</td>
<td>(Lin et al., 2013)</td>
</tr>
<tr>
<td>MKLdiv-dc</td>
<td>61.1</td>
<td>26 PFs</td>
</tr>
<tr>
<td>MKLdiv-conv</td>
<td>63.70</td>
<td>26 PFs</td>
</tr>
<tr>
<td>AK-MKL</td>
<td>61.88</td>
<td>26 PFs</td>
</tr>
<tr>
<td>SimpleMKL</td>
<td>56.92</td>
<td>26 PFs</td>
</tr>
<tr>
<td>Fisher's Mean</td>
<td>65.80</td>
<td>26 PFs</td>
</tr>
<tr>
<td>Arithmetic Mean</td>
<td>60.57</td>
<td>26 PFs</td>
</tr>
<tr>
<td>Karcher-KF (GeoFold1)</td>
<td>86.16</td>
<td>GFK1 (geometric mean)26 PFs</td>
</tr>
<tr>
<td>AGH-KF (GeoFold2)</td>
<td>86.68</td>
<td>GFK2 (geometric mean)26 PFs</td>
</tr>
<tr>
<td>LogE-KF (LogEGFold)</td>
<td>81.72</td>
<td>LogE (Log-Euclidean mean)26 PFs</td>
</tr>
</tbody>
</table>

![Fig. 1. The architecture of our fusion model for protein fold recognition.](http://bioinformatics.oxfordjournals.org)
Next, to compare the efficiency of the proposed formulations AGH-FK and LogE-KF with other MKL approaches, we consider 201 various convex combinations of two different kernels. For this purpose, we assign different weights to each kernel as follows,

\[ w_i K_1 + (1 - w_i) K_2, \quad 1 \leq i \leq 201, \]

where \( w = [0.1, 0.995, 0.99, \ldots, 0] \). These weights can also be interpreted as their corresponding importance in the fused kernel. In fact, finding such weights is the objective of any MKL approach. As illustrated in Figure 3 and Supplemental 2, we observe better success rates on the majority of the interval of kernel weight pairs for the new approaches. These results indicate the limitation of MKL approaches in terms of their sensitive behaviour in dealing with kernel weights. They also demonstrate that the best linear combination of two kernels usually is the one where we assign more weight to the kernel with a higher performance. This is particularly true when the difference between the performances of the two kernels is considerable. Our results show that the evolutionary-based features and either the S or C convey the considerable complementary information with respect to each other. Moreover, the evolutionary information extracted from PSSM profiles through Psp0 and PS2 carries complementary information with respect to each other features.

Moreover, we investigate the performance of our approach on the newer SCOP database (version 1.75)(Yang and Chen, 2011). As the results on the SCOP PDB-40D benchmark data set suggest, it is interesting to consider only two protein features including predicted secondary structural information of the protein sequence and information extracted directly from position specific scoring matrices. For this propose, the PS2 and predicted secondary structure results from NetSurfP (Petersen et al., 2009) are used. Composition, transition and distribution descriptors as described

(a)
in (Dubchak et al., 1995) are used to construct the feature vector for the representation of S. Table 3 provides the mean percentage accuracy with standard deviation from our proposed data fusion methods using 10-fold cross validation for classification of protein folds in the newBD data set (Yang and Chen, 2011).

It is observed that by incorporating the available functional domain information (interPro) through our proposed hybridization model, we are almost able to completely crack the protein fold recognition problem for 27 folds. In addition, it is observed that using FunFold-cdd, FunFold-InterPro, LogEFold, and GeoFold models, we achieve competitive results compared to the Taxfold webserver (Yang and Chen, 2011).

We also investigate the performance of our geometric kernel fusion approach on the protein remote homology detection problem (Liao and Noble, 2003) by fusing multiple kernels. In the supplemental data, we report the competitive results on this problem.

5 CONCLUSION

In this study, we enhance the fold recognition results on the SCOP PDB-40D benchmark data set through a novel kernel data fusion framework based on the geometric mean of kernel matrices (GFK).

We present two methods (Karcher-KF and AGH-KF) for computing the geometric mean, where the second one is a computationally scalable method that computes an approximate geometric mean. The experimental results demonstrate that the geometric mean of kernel matrices can effectively improve the accuracy of the state-of-the-art kernel fusion model. In addition, we obtain similar results using the Log-Euclidean mean, which is a more cost-effective framework based on the geometric mean of kernel matrices (GFK).

Our meta-predictor is developed by incorporating the available knowledge on functions of protein domains into our kernel data fusion framework, giving a promising total accuracy of 89.30%.

Understanding the relationship between primary and tertiary structure in proteins is one of the main objectives of protein sequence analysis. This relation is still elusive, but our results suggest that combining the evolutionary and secondary structural information could be crucial to elucidate such a latent link. This claim is investigated on the newer SCOP database (version 1.75)(Yang and Chen, 2011), where our new methods again have very good performance. In addition, by incorporating the available functional domain information using our FunGeoFold model, nearly exact protein fold recognition for 27 folds is achieved.

Furthermore, the limitation of convex linear combinations in dealing with fusion of different protein features which carry complementary information is considered. Our proposed fusion frameworks, by contrast, can be used to detect these features with complementary information, which provides an insightful approach for fusing different features of other problems in bioinformatics.

ACKNOWLEDGEMENT

YM is a full professor at KU Leuven, Belgium. This research is supported by KUL. GOA/10/09 MaNet, KUL PFV/10/016 SymBioSys, KUL IOF 3M120274 Immunosuppressive drugs. IWT TBM Haplotyping. Hercules Stichting: Hercules III PacBio RS, iMinds: SBO 2013. EU COST Action BM1006: NGS Data analysis network. FCT Neuroinformatics. The scientific responsibility is assumed by its authors.

This work was partially supported by MIUR grant number 2002014121; by the Research Council KU Leuven, projects OT/11/055 (Spectral Properties of Perturbed Normal Matrices and their Applications), CoE EF/05/006 Optimization in Engineering (OPTEC); by the Fund for Scientific Research–Flanders (Belgium) project G034212N (Reestablishing Smoothness for Matrix Manifold Optimization via Resolution of Singularities); and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization).

REFERENCES


