BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 2 2011, pages 204–210
doi:10.1093/bioinformatics/btq618

Structural bioinformatics

Advance Access publication November 18, 2010

Improving structure alignment-based prediction of SCOP families using Vorolign Kernels

Tobias Hamp[1], Fabian Birzele[2], Fabian Buchwald[1] and Stefan Kramer[1,*]

[1]Institut für Informatik/I12, Technische Universität München, Boltzmannstrasse 3, D-85749 Garching b. München and [2]Department of Pulmonary Research, Group Genomics, Boehringer Ingelheim Pharma GmbH & Co KG, Birkendorferstrasse 67, D-88397 Biberach an der Riß, Germany

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: The slow growth of expert-curated databases compared to experimental databases makes it necessary to build upon highly accurate automated processing pipelines to make the most of the data until curation becomes available. We address this problem in the context of protein structures and their classification into structural and functional classes, more specifically, the structural classification of proteins (SCOP). Structural alignment methods like Vorolign already provide good classification results, but effectively work in a 1-Nearest Neighbor mode. Model-based (in contrast to instance-based) approaches so far have been shown to be of limited values due to small classes arising in such classification schemes.

Results: In this article, we describe how kernels defined in terms of Vorolign scores can be used in SVM learning, and explore variants of combined instance-based and model-based learning, up to exclusively model-based learning. Our results suggest that kernels based on Vorolign scores are effective and that model-based learning can yield highly competitive classification results for the prediction of SCOP families.

Availability: The code is made available at: http://wwwkramer.in.tum.de/research/applications/vorolign-kernel.

Contact: kramer@in.tum.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 8, 2010; revised on October 1, 2010; accepted on October 30, 2010

1 INTRODUCTION

With the ever increasing number of known protein structures, the assignment of individual domains to structural classes has become a crucial task in computational biology. For this purpose, several protein databases like SCOP (Murzin *et al.*, 1995) and CATH (Orengo *et al.*, 1997) have been created over the past few years, trying to categorize protein structures into various (hierarchical) classes that reflect certain common evolutionary or structural properties. Ranging from manual inspection of protein topologies and sequences to fully automatic assignment methods, they are built upon diverse approaches and employ various criteria and orders of generality to assign the class of a target protein.

One of the most important and best regarded databases is SCOP, Structural Classification of Proteins, an originally completely manually built hierarchical class structure. Although a certain degree of automation has been introduced since its first release in order to cope with the heavily increasing number of newly discovered protein structures, it still features the highest degree of human expert knowledge (Andreeva *et al.*, 2007). This has ambivalent effects: on the one hand, it has made SCOP become a point of reference for comparing the quality of protein classification methods in a variety of research areas, and it also comes with the disadvantage of leading to a slow update process; since its publication, SCOP followed about an annual release cycle so that novel protein structures are not available for prediction efforts relying on protein structure classifications.

With the introduction of structural alignment algorithms like *Vorolign* (Birzele *et al.*, 2007) and *PPM* (Csaba *et al.*, 2008), it was recently shown for comprehensive test sets that automatic predictions solely relying on structure alignment scores can achieve classification accuracies beyond 90% for the higher levels of SCOP and around 85% on the family level. Another recent method, *AutoSCOP* (Gewehr *et al.*, 2007), directly aims at predicting the future SCOP classification by employing a sequence-pattern based filter and Vorolign as a plug-in and achieved 92% accuracy also on the family level.

In the cases those protein similarity measures have been used as classifiers so far, it has always been in an instance-based way, i.e. the structure with the highest similarity was determined and its respective class assigned to the target protein. Model-based machine learning methods were assumed not to be suitable, as classes with only one or a few members would have to be generalized. This is why Melvin *et al.* (2008) recently introduced a simple classification scheme called *punting*, which combines models and instance-based learning by trying to use models for larger classes and a best-hit approach otherwise. In the following, we not only show how punting can be improved, but also introduce a variety of alternatives, including one that discards instance-based learning altogether. Their subsequent evaluation leads to classification accuracies representing statistically highly significant improvements over any previous approach used for the same datasets so far.

It should still be noted that although this article exclusively uses Vorolign as an alignment algorithm and SCOP families as the prediction target, the proposed methods can be applied to basically any structural similarity measure and classification problems with small class sizes (e.g. protein function prediction, comparison of SCOP and CATH). Vorolign was only chosen for its good performance and comparably low computational requirements,

*To whom correspondence should be addressed.

while SCOP represents a gold standard in terms of protein domain classification.

In summary, the contributions of this article are as follows: first, we present a new kernel-based machine learning approach based on a highly predictive structural alignment method (Vorolign). While the structural alignment method already gives excellent results in classification, we show that it is possible to optimize performance using statistical machine learning on top of it. For this purpose, we test and evaluate various methods for indefinite kernels based on structure alignment scores in SVM learning. Second, we explore the design space of classification algorithms for large and small classes of proteins between instance-based (k-NN and variants) and model-based learning. Traditionally, instance-based learning was used almost exclusively due to the existence of many small classes in the SCOP hierarchy. Third, we present highly competitive classification results for structure alignment-based prediction of SCOP families. The best methods presented here improve with a high statistical significance over the underlying structural alignment method (Vorolign) and others (both in predicting SCOP 1.67 from 1.65 and SCOP 1.75 from 1.73; see Table 3).

2 METHODS

While structure alignment methods like Vorolign by Birzele *et al.* (2007) are already highly accurate in distinguishing the various hierarchical SCOP classes, their actual use (best hit or 1-Nearest Neigbor/1NN) is still comparably simple. Replacing 1NN by more advanced machine learning techniques, however, is not easy, as the structural space of proteins is very heterogeneous: many proteins share similar features while some others are so particular that single-member classes need to be introduced. Although we present a novel approach that allows to abstain from instance-based learning (see Sections 2.2–2.4), it is commonly considered to be highly improbable to find good models based on classes with less than about a handful of members. Thus, we not only aimed for accurate models, but also developed a number of approaches to integrate them with 1NN. Before we describe the three methods (*Global Separation*, *Threshold Learning* and *Multi Model Separation*; see Section 2.5) in detail, we present the foundation of all methods in this article: the Vorolign structure alignment method and kernels based on Vorolign alignment scores.

2.1 Vorolign

Vorolign is a fast method to flexibly align two or more protein structures. It 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. The environment of a residue is supposed to be captured by a c_α-based Voronoi tessellation, which finally leads to a protein sequence where each residue is represented as a Voronoi cell instead of its former amino acid. Each cell implicitly contains information on protein structure and, given a similarity function for a pair of cells, corresponding sequences can efficiently be aligned using a primary dynamic programming.

The similarity measure is computed by Vorolign as the amino acid and secondary structure conservation of the residues in the neighborhood of a residue: two residues in the neighborhoods of two Voronoi cells are compared to amino acid and secondary structure exchange matrices while the final similarity of two cells is computed as the best alignment of the neighboring residues determined by a second dynamic programming step (see Fig. 1).

In this article, we slightly changed the calculation of the final alignment score compared to Birzele *et al.* (2007), leading to a second Vorolign variant called *Vorolign** (see Supplementary Material S1 for details). It was used in any subsequent learning scheme.

The ability of Vorolign to classify protein structures has been shown to be among the top performing protein structure alignment methods (Birzele *et al.*, 2007; Csaba *et al.*, 2008). Its score for a pair of protein structures appears to be a reliable predictor of similarity and taking the highest scoring template in a set of proteins accurately identifies the correct SCOP family for a given query structure. This last method will be referred to as *Vorolign-1NN* or *Vorolign*-1NN*, in case Vorolign* was used.

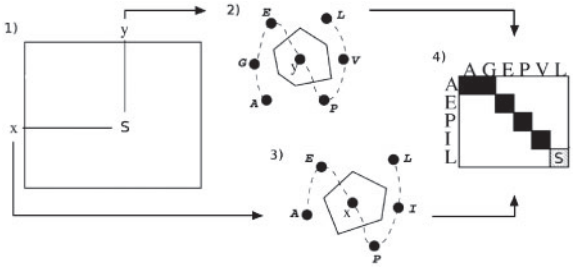

Fig. 1. Vorolign starts with a first dynamic programming for all Voronoi cells. In the figure, the score S of the two Voronoi cells corresponding to amino acid residues x and y is to be computed (1). To this end, all neighboring residues of the two cells are collected (2,3), ordered according to their position in the original protein sequence and aligned in a second dynamic programming step (4).

2.2 Eager learning with Vorolign

The concept of a *kernel* has undergone some change in recent years. It is now referred to as basically any function giving the similarity of two input instances. The reason lies in the fact that, with the right transformation, any similarity measure can be made compatible with traditional kernel classifiers like Support Vector Machines (SVMs). In computational biology, these transformations were usually carried out by use of the *Empirical Kernel Map* and the *psd-Shift* (see Supplementary Material S2). In parallel, the mathematical interpretation of indefinite kernels has experienced a lot of interest, resulting in many more methods to transform a similarity into a kernel value. A good and up-to-date overview is given by Chen *et al.* (2009b). Note, however, that while results certainly show that on average there is progress toward better performance, classical methods like k-Nearest-Neighbor or psd-Shift exhibit accuracies only insignificantly worse or even better than the new approaches [Chen *et al.* (2009a, b)]. Furthermore, while there is no clear winner among the new transformations, publication dates also show that the battleground is far from being closed. Together with the plain number of candidate solutions introduced by now and the growing number of real-world datasets, their value still has to be shown on a case-by-case basis.

In the end, we decided to use the Empirical Kernel Map, psd-Shift and an own special variant of Shift, *c-Shift* (see Supplementary Material S2.3), resulting in the kernels k_{Map}, $k_{psd-Shift}$ and $k_{c-Shift}$, respectively. SMO (Platt, 1999a) was chosen as the SVM training procedure.

Even though there is no reason why other structure alignment algorithms also could not benefit from a transformation into a kernel, a few points still favored particularly Vorolign for model construction: highly competitive results in the prediction of SCOP classifications in 1NN mode combined with good RMSD values (Birzele *et al.*, 2007); a comparably strong influence of protein sequence on the structure alignment, thus rendering it especially suited for the SCOP family level; fast running time in comparison with the second best classifier in 1NN mode, CE (Birzele *et al.*, 2007); a comparably small number of negative Eigenvalues in the similarity matrices, indicating a similarity measure quite comparable to an inner product of a Hilbert space, the feature space of a traditional kernel (Haasdonk, 2005) (data not shown).

2.3 Stratified cross-validation and parameter selection

The only free parameters of our methods are the complexity parameter C of SMO and, in case of $k_{c-Shift}$, the value c, representing the number to

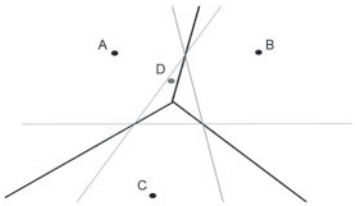

Fig. 2. The figure shows a setting with three input points A, B and C, where each point corresponds to an own class. The gray lines display the hyperplanes of the underlying binary base classifiers. The black lines depict the class boundaries induced by 1vsAll using those hyperplanes. Point D is an example of a target that does not lie on the '1'-side of any binary classifier. It is assigned to the class A, where it is the most unclear that it belongs to the 'All'-side. Note that even two points P_1 and P_2 of different classes suffice to define a maximum margin hyperplane. It would simply comprise all points having equal distance to both P_1 and P_2 (data not shown).

be subtracted from the kernel diagonals. For their optimization via standard parameter selection or grid search, the standard Weka (Holmes *et al.*, 1994) implementation of a stratified k-fold cross-validation (CV) was used in a number of places. While the principle of stratification should be clear if k is smaller than the largest class, it may not be if this is not the case: when k-fold stratifying a class c, there will be $m = |c| \bmod k$ test folds with $\lceil |c|/k \rceil$ random members of class c and $k-m$ folds with $\lfloor |c|/k \rfloor$ members. If $|c| < k$, there will be $k - |c|$ test folds without a representative member of that class and $|c|$ folds with exactly one random member. Note that having all dataset classes represented in a test fold is not a prerequisite (e.g. consider Leave-One-Out CV).

2.4 Multiclass model

2.4.1 1vsAll To deal with multiple classes, the binary SVM has to be employed in a way so that it can be used to differentiate between all the SCOP families. We experimented with various methods for multi-class and hierarchical classification, but found a classical 1vsAll approach to achieve the best results here. This is probably due to the relative independence of class sizes: if no Platt Scaling (Platt, 1999b) is used, simply the class achieving the largest scaled distance to the hyperplane $(\mathbf{w} \cdot \mathbf{x} + b)$ is predicted.

2.4.2 1vsAll for all classes The extensibility of Platt Scaling in particular allows to employ 1vsAll for single-member classes. SVM theory still holds even in this case (see Fig. 2). Thus, 1vsAll was not only the standard multi-class model for larger classes, but also represented a standalone classifier for the entire dataset. The particular combination of 1vsAll and Vorolign* will be referred to as *Vorolign*-1vsAll*. See Section 3.4 for details of the evaluation, Supplementary Material S3 on how to optimize SVM parameters in such a setting and Supplementary Material S5 for differences to instance-based classification.

2.5 Integration of 1NN and models

Models are traditionally laid out to deal with classes with more than just a handful of members. In effect, it is reasonable to integrate them with 1NN when confronted with problems involving arbitrary class sizes. This can be achieved with a surprisingly large variety of approaches, some of which will be described in the following sections. As an overview, Figure 3 gives a flowchart of the decisions that have to be taken before the label of a target is predicted. Generally, the model has to gain a substantial lead over 1NN on a dataset where small classes are excluded, so that the inevitable performance loss caused by the integration with 1NN for the whole dataset is not as dramatic that it would have been better to exclusively use 1NN from the beginning. This accuracy decrease corresponds to wrong decisions in the two diamonds in Figure 3.

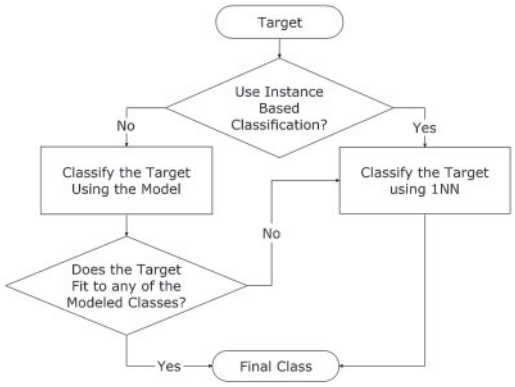

Fig. 3. The general workflow when combining instance- and model-based learning. The decision whether a protein is classified by instance-based learning can be taken before and after the classification with the model. Due to better practical results, we used these decisions mutually exclusive, i.e. only one of them was actually implemented as a learner, the other was fixed to constantly return either 'yes' or 'no'. Decisions before classification can be taken by *Global Separation*, decisions after by *Threshold Learning* or *Multi Model Separation*.

2.5.1 Global Separation This method attempts to separate instances in large classes from instances in small classes globally. For this purpose, all instances from large classes are combined in a new class A, while all instances from small classes are combined in class B. Subsequently, any binary learner can be used to predict whether a target supposedly belongs to a small or a large class. The workflow in Figure 3 is implemented by letting Global Separation decide what to do in the first diamond and always returning *yes* in the second. For the evaluation in Section 3.3, this task was carried out by 1NN and SMO with k_{Map} and $k_{psd-Shift}$.

2.5.2 Threshold Learning This method derives a threshold t_C for each large class that defines the probability a target must at least have to be considered a reliable prediction. If found unreliable, the target is classified with 1NN. To go back to Figure 3, we always chose model-based classification in the first diamond and use the t_C-based decision in the second. For implementation details, see Supplementary Material S4.1.

It has to be mentioned that the idea behind Threshold Learning is the same as in Melvin *et al.* (2008), where it is called *punting*. There are a few differences as well. We use cross-validation to obtain probabilities for the samples of large classes instead of splitting the training set into two sets, we use all samples even those from small classes as the negatives in the binary classifiers and discard the user given parameter ρ to basically learn it via LOO-CV. As Vorolign does not provide E-values, we also do not use *double punting*, but always predict a class.

2.5.3 Multi Model Separation This method uses two models to decide whether an instance is sent to 1NN: first, a multi-class model of the large classes is applied. Second, a binary classifier particularly trained for the 'winning' class from step one decides whether it really belongs to that class.

The decisions in the diamonds of Figure 3 are taken analogously to Threshold Learning. Implementation details are given in Supplementary Material S4.2.

3 RESULTS AND DISCUSSION

3.1 Two datasets

As our goal is to compare the performance of best-hit, also referred to as 1-Nearest-Neighbor (1NN), with other methods, we mainly used

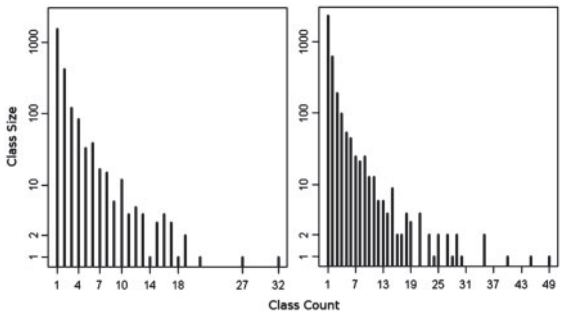

Fig. 4. A histogram of the distribution of family sizes (left: SCOP 1.65, right: 1.73). Small families play a predominant role in both datasets while some exceptionally big families are always retained. In general, this should present a setting highly beneficial for instance-based learning approaches. When considering the whole version of SCOP 1.65, the overall shape of the distribution is conserved, but the *y*-axis has a much smaller range, e.g. for SCOP 1.65 there are only 360 families of size one (data not shown here).

Table 1. Accuracies for minimal family size 10 (upper part) and 5 (lower part)

	k_{Map} (%)	$k_{psd-Shift}$ (%)
1NN	–	95.8
1vsAll	94.0	96.4
1vsAll: P. Scal.	95.8	95.2
1vsAll: C-opt.	97.6	97.6
1vsAll: C-opt. & P. Scal.	96.4	95.8
1NN	–	86.8
1vsAll: C-opt. & P. Scal.	92.5	94.3
1vsAll: C-opt.	94.0	97.7

C-opt refers to the *C* parameter optimization for SVMs and *P. Scal.* to *Platt Scaling*.

the non-redundant dataset from the original Vorolign publication for our evaluations. It contained 4357 training proteins from SCOP 1.65 and 977 test proteins from SCOP 1.67. To make sure our results are still applicable to current data, we decided to also test on the latest releases of SCOP (1.73 and 1.75), applying the same method to reduce redundancy as for the first set. This resulted in 6981 training and 852 test proteins. The distributions of family sizes are shown in Figure 4.

3.2 Minimal family size 10 and 5

As a first attempt to outperform 1NN, all families with less than 10 members were excluded from the original dataset, leaving 586 training and 167 test proteins distributed among 41 families. Two kernel matrices induced by k_{Map} and $k_{psd-Shift}$ (see Section 2.2) were evaluated together with the influence of an optimization of the SVM complexity parameter *C* via standard CV parameter selection. Furthermore, 1vsAll was trained with and without Platt Scaling to get a general impression of its influence in a setting with extremely small families (see Section 2.4.1). Results are given in the upper part of Table 1.

Generally, only very little can be said about significant differences among the kernels, since most values lie closely together. Platt Scaling is apparently not essential already at this minimal family size and even seems to be slightly harmful. The absence of *C* optimization (i.e. $C = 1$) produced the overall worst accuracy with 94.0%, so that this variant was no longer evaluated.

To increase the model's coverage, find a boundary for its minimal family size and further investigate the kernels' performance, we created a dataset where only families with at least five members are allowed, instead of the former 10. This led to 1278 training and 385 test instances belonging to 153 distinct families. Evaluation results are given in the lower part of Table 1.

In this setting, the 1NN approach was found to have dropped by 9.0% to 86.8%. Models suffered only a minor degradation compared to minimal family size 10, giving them a considerable lead of at least 5.7% over instance-based learning. Among the 1vsAll models, Platt Scaling continued to have detrimental effects, even preventing the best accuracy of all (97.7%) which was achieved in combination with $k_{psd-Shift}$. The latter was constantly on par or better than k_{Map}.

Table 2. Results for the separation of small and large families. PPV, NPV, sensitivity (Sens.), specificity (Spec.) and accuracy on the whole SCOP 1.67 test set are shown

	PPV	NPV	Sens.	Spec.	Accuracy (%)
TL-1vsAll	0.98	0.79	0.57	0.99	87.7
MMS-Map	0.99	0.89	0.80	0.99	88.3
MMS-c-Shift	0.99	0.89	0.80	0.99	88.2
GS-1NN	0.96	0.90	0.84	0.98	87.7
GS-Map	0.90	0.85	0.76	0.94	85.2
GS-c-Shift	0.57	0.62	0.09	0.96	n/a

Methods under evaluation are Threshold Learning (TL) using probabilities obtained via 1vsAll, Multi Model Separation using k_{Map} (MMS-Map) and $k_{c-Shift}$ (MMS-c-Shift) as kernels and Global Separation using 1NN (GS-1NN), k_{Map} (GS-Map) and $k_{c-Shift}$ (GS-c-Shift). In case of a perfect separation of small and large families, the model would be 4.4% better than 1NN, raising the overall accuracy on the whole SCOP 1.67 test set from 87.5% to 91.9%.

3.3 Combination of model- and instance-based classification

The threshold of at least five members per family still appeared to be well suited for model-based classifiers and covered more than a third of the instances of the whole dataset. Thus, we did not lower family sizes any more, but tried to combine 1vsAll with 1NN in order to make predictions for the whole SCOP 1.67 test set (see Section 2.5).

To compare performance, we used sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV) and accuracy on the whole test set. Sensitivity gives the fraction of proteins belonging to a large family that were also predicted as such, and specificity the same for small families. PPV is the ratio of correctly predicted positive samples compared to all positives and NPV the analog for the negatives. Here, sensitivity and PPV are the two most important measures as they reflect coverage and accuracy of the separator, respectively. See Table 2 for results.

1NN showed the best performance of all Global Separators (see 2.5.1); has, however, the disadvantage that it practically functions as a false-positive (FP) filter specifically for instance-based classification. The FPs in the separation of small and large families all correspond to actual family misclassifications later made by the 1NN for all families. Thus, the set of true positives (TPs) will experience a much smaller error rate in 1NN than to be expected

on average. At the same time, all FPs are inevitably inherited by the model, which will have no effect on the performance on TPs. In the end, the approach achieved an accuracy of 87.7%. The SVM as a Global Separator (see 2.5.1); could not even closely reach this performance. The high number of TPs goes along with a large fraction of FPs in the context of k_{Map} so that performance drops to 85.2%. $k_{c-Shift}$ is best explained with coming very close to a coin-flip classifier, its feature space seems in no way suitable for this bisectional hyperplane.

With Threshold Learning (see 2.5.1); PPV and NPV greatly differ, which is mostly due to the intended minimization of the FP rate during training [False-negatives (FNs) are by far not as harmful as FPs, since their classification can be corrected by 1NN; a FP is automatically a misclassification]. 1vsAll produced a large fraction of FNs, thus considerably reducing the amount of targets for the model and this way the chance to gain lead over 1NN. Sensitivity can be increased by introducing different criteria than minimal FP rate, but it never led to a greater lead over 1NN, since the higher FP rate always went along with a lower PPV (data not shown). In the end, Threshold Learning (see 2.5.1); had an error rate of 1.3% among the test instances for large families compared to 0.8% for 1NN, giving it a lead of 0.2% on the whole dataset.

The k_{Map}-based Multi Model Separation (see 2.5.1); takes the position as the most accurate separator. The slight advantage of $k_{c-Shift}$ over k_{Map} seems to have switched in favor of k_{Map}, as it is 0.1% better on both the test set and in terms of the lead over 1NN, which amounts to 88.2% for $k_{c-Shift}$.

At this point, we want to emphasize again that the given results already represent the accuracies of fully equipped classifiers, not leaving out any families of SCOP. For a comparison of their performance in context of the other methods for the whole dataset, see Table 3.

3.4 Models for arbitrarily small families

Rather out of curiosity than expected increase of performance, we investigated how accuracies of 1NN and models develop when shifting the minimal family size between 1 and 10 (see Section 2.4.2). One particularly interesting question is further if the ideal minimal family size for models, i.e. the point where the gap of accuracy between instance- and model-based learning is the largest, can be derived from the training data alone. In our own prior evaluation and basically any former publication about model-based protein classification involving SCOP [e.g. Melvin *et al.* (2008)], the minimal class size was chosen more or less arbitrarily.

Thus, we created the following datasets. For every i in a range from 1 to 11, $\mathcal{D}_i^{1.65}$ represented all proteins from our SCOP 1.65 variant except those belonging to a family with less than i members. For each of those datasets, a set $\mathcal{D}_i^{1.67}$ was established which only contained those proteins from our SCOP 1.67 variant whose family was among the ones in $\mathcal{D}_i^{1.65}$. Each $\mathcal{D}_i^{1.65}$, with i in a range from 1 to 10, was employed as a whole as a training set for both 1NN and 1vsAll. In this setup, $\mathcal{D}_i^{1.67}$ represented the according test set. Furthermore, every $\mathcal{D}_i^{1.65}$, with i in a range from 2 to 11, was also evaluated in a 10-fold stratified CV, again for both 1NN and 1vsAll. Note that in this context, a minimal family size of i actually means dealing with a dataset with a minimal family size of $i-1$ at training time.

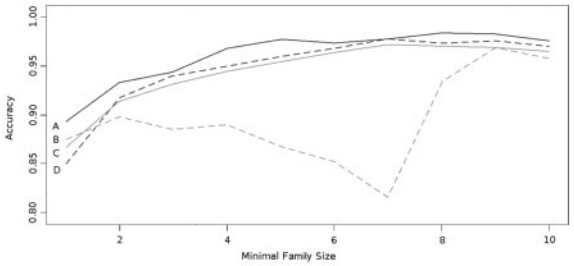

Fig. 5. Family sizes ranging from 1 to 10. Line B (gray dotted) indicates the performance of 1NN on the SCOP 1.67 test set, line C (gray solid) its performance in a 10-fold stratified CV. The model was always set to 1vsAll with $k_{c-Shift}$ and fixed parameters. Line A (black solid) gives the accuracy on the SCOP 1.67 test set, and line D (black dotted) the same in a 10-fold stratified CV.

For all of the resulting 20 1vsAll evaluations, we left the c parameter of $k_{c-Shift}$ at 0 and took a strictly hard margin. Only for the two times where 1vsAll was used in conjunction with minimal family size 1, we additionally applied a grid search optimization for c and C. For the SCOP 1.65-1.67 setup, the optimization dataset was consequently $\mathcal{D}_2^{1.65}$; for the CV, we altered grid search so that all single-member families would be excluded for parameter optimization and then re-added afterwards. See Section 2.3 and Supplementary Material S5 for a description of the optimization routine and Supplementary Material S6 for an analysis and discussion of the resulting models.

All this summed up to 42 individual evaluations. Figure 5 gives an overview of the 40 experiments without parameter optimization. The results of the remaining two are given in the following. For a comparison of the two SCOP 1.65–1.67 evaluations using all data (i.e. minimal family size 1) with other methods, we refer to Table 3.

The most striking part of the plot is arguably the strong alternations in the accuracy of 1NN on the SCOP 1.67 test set. It can only be explained by fluctuating family distributions and, because the corresponding CV has a very continuous accuracy, is a strong indicator that the cross-validated results do not necessarily reflect the classifiers' performance on real-world data. It is additionally confirmed by the fact that the evaluation of the model on the test set almost always shows better accuracies for the SCOP 1.67 test set than for the corresponding CV.

This also gives a very clear answer to our question whether the optimal minimal family size can be learned from the training data: it cannot. While there is a clear correlation between the CV results for 1vsAll and 1NN, it is impossible to predict the performance of 1NN on real world data, given a specific family size threshold. Thus, any lead in accuracy of a model over 1NN for a given family size is highly linked to the chosen minimal family size and so is the performance of any scheme integrating 1NN and models. This can also be seen from the evaluation on a newer SCOP version in Section 3.5.

In context of the CV, the sharp performance loss at family size 1 gives the model an overall accuracy worse than that of instance-based learning. But already the c and C parameter selection prior to each fold renders both methods to be exactly on par at 87.5%.

When comparing model- and instance-based learning on the whole SCOP 1.67 test set, the model seems to be the overall better choice. This even holds true for the smallest family size,

making 1vsAll to be better on real-world data than 1NN *without any separation of small and large families*. The exact value at family size 1 is 89.3% for 1vsAll and 87.5 % for 1NN. When adding grid search parameter selection, the optimal value for c is found at 0.4 and a soft margin discouraged, which is identical to what was obtained for datasets with higher minimal family size. If plugged into 1vsAll, the accuracy is drastically increased to 91.2% on the whole SCOP 1.67 test set.

3.5 Evaluation on SCOP 1.73 and SCOP 1.75

With new insights into the behavior of 1NN when alternating minimal family sizes, it can be expected that the former lead of models over 1NN in the minimal family size 5 case will significantly change for SCOP 1.73 and 1.75. In fact, the biggest differences in accuracy can now be found at minimal family size 2 and 3, minimal family size 5 only produces an advantage of nine instances of 1vsAll over 1NN. As the overall accuracy would only rise by 1% assuming perfect separation of small and large families, we did not follow this approach here, but directly employed the model for arbitrarily small family sizes. 1vsAll with fixed parameters ($c=0$, hard margin) finally produced an accuracy of 89.4% compared to 85.7% for 1NN. When adding grid search, the training phase looks very promising with optima for c and C clearly pointing to 0.5 and 10, respectively. If applied to the test set, however, accuracy is insignificantly reduced by 0.2% (two proteins). A closer examination of the cause revealed that there simply was no other value for c that could have substantially improved accuracy. A final overview of the performances of each method on the full test sets of SCOP 1.67 and SCOP 1.75 is given in Table 3.

3.6 Performance overview

To put all the results of this article on the complete SCOP datasets into relation and get an impression of how methods from other publications have performed, we compiled their results in Table 3. It features the 1vsAll models for all families, schemes integrating 1NN and models and also a collection of the best performing 1NN algorithms besides Vorolign. Each 1NN method was evaluated within the same prediction framework, so that performance differences could only be attributed to the methods themselves. They were all used with standard parameters.

3.7 Running times of model training and testing

Concerning the Vorolign-based experiments, the by far most time consuming task was the structural alignment of all training proteins against each other in order to create the kernel matrices. This took about 1200 CPU hours on an Opteron 2218 (2.6 GHz/1 MB) or 24 wall clock hours with 50 CPUs running in parallel. Training of the models and classification of all test proteins was done in less than 6 CPU hours for each dataset. The number of 1200 CPU hours for the alignments might seem large, but considering the annual release cycle of SCOP and the fact that, in the transition from one SCOP release to the next, solely alignment scores among new and between new and old domains have to be calculated, the approach does not lack practicability.

Table 3. Accuracies of the best methods evaluated so far on the two datasets described in this article

	SCOP 1.67	SCOP 1.75
TM-align-1NN	83.8%	–
CE-1NN	84.6%	–
PPM-1NN	88.3%	–
Vorolign-1NN	86.4%	–
Vorolign*-1NN	87.5%	85.7%
TL-1vsAll	87.7% (2.88e-2)	–
MMS-Map	88.3% (1.53e-3)	–
MMS-c-Shift	88.2% (1.61e-3)	–
GS-1NN	87.7% (6.80e-2)	–
Vorolign*-1vsAll w/o Opt.	89.3% (1.47e-2)	**89.4%** (7.90e-7)
Vorolign*-1vsAll w/ Opt.	**91.2%** (1.44e-5)	89.2% (1.14e-6)

The results of the first four 1NN methods (TM-align, CE, PPM, Vorolign) are cited from Birzele *et al.* (2007) and Csaba *et al.* (2008) and based on various other structural alignment methods besides Vorolign. Vorolign*-1NN is the variant of Vorolign introduced in the last paragraph of Section 2.1 and Supplementary Material S1. Results for TL-1vsAll, MMS-Map, MMS-c-Shift and GS-1NN are taken from Section 3.3. Vorolign*-1vsAll values reflect the accuracies obtained in Sections 3.4 and 3.5 when applying it with and without parameter optimization. Where possible, we carried out a McNemar test to evaluate the significance of the lead over Vorolign*-1NN. Its P-value is given in brackets.

3.8 Comparison and integration with AutoSCOP

The property of self-containment, i.e. the non-use of external tools and databases, allows our models to directly replace Vorolign-1NN in any scheme integrating it with other approaches, like AutoPSI (Birzele *et al.*, 2008). The latter is a publicly available service for the prediction of SCOP classification on both sequence and structure level and represents the combination of AutoSCOP and Vorolign-1NN, together with a few extensions such as the prediction of domain definitions. AutoSCOP can be seen as an orthogonal approach to structure-based SCOP classification and works on sequence level only. It was developed with a particular emphasis on specificity, i.e. to only make predictions in the face of strong evidence for a particular family. When employed as a filter prior to structure-based classification, it reliably predicts targets with high sequence identity to a template protein and delegates the presumably harder cases to another, more sensitive, classifier (e.g. based on Vorolign). To show that the combination of AutoSCOP and Vorolign*-1vsAll directly improves over the already synergistic combination of AutoSCOP and Vorolign*-1NN, we turned to the current version of AutoPSI and investigated the AutoSCOP predictions of all protein chains containing domains from the SCOP 1.67 test set. If a domain featured contradictory or no AutoSCOP predictions, we used a Vorolign-based approach for classification, otherwise we predicted the one found by AutoSCOP. The final accuracies for the 1.67 test set were 90.9% for AutoSCOP&Vorolign*-1NN and 94.0% for AutoSCOP&Vorolign*-1vsAll.

3.9 Related work

Besides the 1vsAll approaches described in this article, we also experimented with schemes from Zimek *et al.* (2010), but could not find an improvement. Furthermore, we had to leave this and related approaches for arbitrarily small families since it is a common practice in the field of machine learning-driven fold recognition and

remote homology detection to design algorithms solely for larger structural classes. Another candidate group of methods, clusterings trying to adopt the SCOP hierarchy, like e.g. StralCP (Zemla *et al.*, 2007), are not directly applicable as predictors and pose problems such as false splitting (e.g. generation of two families instead of one), false merging (generation of a cluster comprising e.g. two families) or the creation of the correct clusters, but over different hierarchical levels. Also even the rough comparison of error rates is largely not possible because of issues like the restriction of the evaluation to small, not representative fractions of SCOP (Huan *et al.*, 2004) or to levels below family (Madera and Gough, 2002), the creation of training and test sets with only one SCOP release (Melvin *et al.*, 2008) and the prediction of CATH instead of SCOP (Rogen and Fain, 2003). From our perspective, the most complete alternative work is SCOPmap (Cheek *et al.*, 2004). Even though targeting only superfamilies, thus again not directly comparable, their method should be readily applicable also for the family level. In principle, it is a meta classifier combining six other external tools, partially relies on external protein databases and performs a weighted vote to make the final prediction in case simple BLAST has not returned a suitable hit. In contrast, we see our approach as a more light-weight alternative, especially regarding the number of free parameters, which goes into the hundred for SCOPmap.

4 CONCLUSION

In the context of structure-based SCOP domain classification, we have presented a new practical approach to dealing with indefinite similarity measures, several new ways to integrate both model- and instance-based learning and showed that even the exclusive use of models is not only possible but also preferable to combinations with a best-hit approach. Our results for the latter pose statistically highly significant increases in accuracy compared to any other method evaluated on the same data so far. Future work could further investigate the whole new class of transformations of possibly indefinite similarity measures into SVM compatible kernels. The generality of the methods also allows to largely extend their scope of application to e.g. protein function prediction.

Furthermore, while we cannot rule out the influence of bias arising from the redundancy reduction of the datasets, our results indicate that the performance of best-hit on real-world SCOP test data will substantially and unpredictably change if structural classes up to a certain size are excluded. Thus, even if models show better accuracy than best-hit for a certain minimal class size, this might no longer be the case if this size or the dataset is changed. As schemes integrating best-hit and models highly depend on the superiority of models over best-hit for a given minimum class size, we argue that their performance in comparison to the exclusive use of best-hit can only be evaluated by shifting this size and using multiple datasets.

ACKNOWLEDGEMENT

We would like to thank Burkhard Rost for supporting this work.

Conflict of Interest: none declared.

REFERENCES

Andreeva,A. *et al.* (2007) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res.*, **36**, 1–7.

Birzele,F. *et al.* (2007) Vorolign–fast structural alignment using Voronoi contacts. *Bioinformatics*, **23**, e205–e211.

Birzele,F. *et al.* (2008) AutoPSI: a database for automatic structural classification of protein sequences and structures. *Nucleic Acids Res.*, **36**, 398–401.

Cheek,S. *et al.* (2004) SCOPmap: automated assignment of protein structures to evolutionary superfamilies. *BMC Bioinformatics*, **5**, 197.

Chen,Y. *et al.* (2009a) Learning Kernels from indefinite similarities. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, New York, NY, USA, pp. 145–152.

Chen,Y. *et al.* (2009b) Similarity-based classification: concepts and algorithms. *J. Mach. Learn. Res.*, **10**, 747–776.

Csaba,G. *et al.* (2008) Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, **24**, 98–104.

Gewehr,J.E. *et al.* (2007) AutoSCOP: automated prediction of SCOP classifications using unique pattern-class Mappings. *Bioinformatics*, **23**, 1203–1210.

Haasdonk,B. *et al.* (2005) Feature space interpretation of SVMs with indefinite Kernels. *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**, 482–492.

Holmes,G. *et al.* (1994) WEKA: a machine learning workbench. In *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, Brisbane, Australia.

Huan,J. *et al.* (2004) Accurate classification of protein structural families using coherent subgraph analysis. *Pac. Symp. Biocomput.*, **9**, 411–422.

Madera,M. and Gough,J. (2002) A comparison of profile Hidden Markov Model procedures for remote homology detection. *Nucleic Acids Res.*, **19**, 4321–4328.

Melvin,I. *et al.* (2008) Combining classifiers for improved classification of proteins from sequence or structure. *BMC Bioinformatics*, **9**, 389.

Murzin,A. *et al.* (1995) SCOP: a structural classifcation of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Orengo,C.A. *et al.* (1997) CATH–a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Platt,J.C. (1999a) Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods: Support Vector Learning*, pp. 185–208.

Platt,J.C. (1999b) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74.

Rogen,P. and Fain,B. (2003) Automatic classification of protein structure by using Gauss integral. *Proc. Natl Acad. Sci.*, **100**, 119–124.

Zemla,A. *et al.* (2007) STRALCP—structure alignment-based clustering of proteins. *Nucleic Acids Res.*, **35**, e150.

Zimek,A. *et al.* (2010) A study of hierarchical and flat classification of proteins. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **7**, 563–571.