Classifying Protein Fingerprints

Melanie Hilario[1], Alex Mitchell[2], Jee-Hyub Kim[1],
Paul Bradley[2], and Terri Attwood[3]

[1] Artificial Intelligence Laboratory, University of Geneva, Switzerland
{Melanie.Hilario,Jee.Kim}@cui.unige.ch
[2] European Bioinformatics Institute, Hinxton, Cambridge CB10 1SD, UK
{mitchell,pbradley}@ebi.ac.uk
[3] School of Biological Sciences, University of Manchester, UK
attwood@bioinf.man.ac.uk

Abstract. Protein fingerprints are groups of conserved motifs which can be used as diagnostic signatures to identify and characterize collections of protein sequences. These fingerprints are stored in the PRINTS database after time-consuming annotation by domain experts who must first of all determine the fingerprint type, i.e., whether a fingerprint depicts a protein family, superfamily or domain. To alleviate the annotation bottleneck, a system called PRECIS has been developed which automatically generates PRINTS records, provisionally stored in a supplement called prePRINTS. One limitation of PRECIS is that its classification heuristics, handcoded by proteomics experts, often misclassify fingerprint type; their error rate has been estimated at 40%. This paper reports on an attempt to build more accurate classifiers based on information drawn from the fingerprints themselves and from the SWISS-PROT database. Extensive experimentation using 10-fold cross-validation led to the selection of a model combining the ReliefF feature selector with an SVM-RBF learner. The final model's error rate was estimated at 14.1% on a blind test set, representing a 26% accuracy gain over PRECIS' handcrafted rules.

1 Motivation and Background

Protein fingerprints are groups of conserved amino acid motifs drawn from multiple sequence alignments that are used to characterise protein families. The PRINTS database [1] is a compendium of more than 1800 diagnostic fingerprints for protein families, superfamilies and domains. It provides large amounts of handcrafted annotation, aiming to document the constituent protein families and to rationalise the conserved regions in functional and structural terms. The annotation procedure is exhaustive and time consuming, and consequently PRINTS remains relatively small by comparison with other, largely automatically-derived signature databases.

To address this issue, automation of fingerprint production and annotation has been investigated. The PRINTS group has previously developed PRECIS [9], an annotation tool which generates protein reports from related SWISS-PROT entries. Though this approach has worked well overall, PRECIS has areas of

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 197–208, 2004.

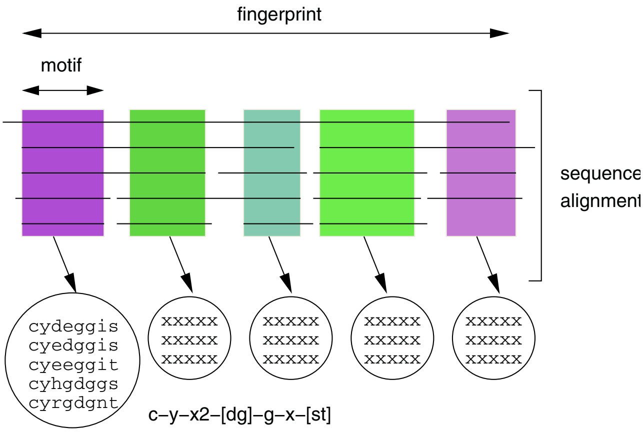

Fig. 1. Schema of a protein fingerprint. Each row is a protein sequence and each column an amino acid. Solid rectangles depict conserved regions or motifs.

limitation. The annotation generated by the tool inevitably lags that which could be derived from current literature. This is due to the fact it is almost entirely dependent on information stored in SWISS-PROT, which, despite the valiant efforts of a team of annotators, cannot be kept up to date. Another limitation of PRECIS is that its relatively simple heuristics often misclassify fingerprints. Broadly speaking, fingerprints may be diagnostic for a gene family or superfamily (united by a common function), or a domain family (united by a common structural motif). The type of fingerprint has implications on the kind of information and the level of detail to be reported within the annotation. Increased accuracy of fingerprint classification would therefore help ensure that the correct information was processed to generate appropriate annotation.

2 Task and Data Representation

The goal of the work reported in this paper was to replace PRECIS' handcrafted heuristics with classification models extracted from data. These heuristics determined fingerprint type through an analysis of SWISS-PROT database records concerning protein sequences within the fingerprint. Before turning to SWISS-PROT, we decided to investigate whether a fingerprint's physical parameters could be used as discriminators to improve classification. As shown in Fig. 1, a fingerprint is basically a multiple sequence alignment with a number of conserved regions or motifs. A fingerprint can be characterized in terms of three distinct entities: the fingerprint itself and its component motifs and proteins. We are therefore confronted with a multirelational learning problem which can be addressed most naturally using a relational approach. This paper focuses on an alternative approach which propositionalizes the task representation by aggregating protein and motif characteristics over the fingerprint.

Fingerprint. The fingerprint as a whole can be described by the number of motifs and proteins it contains. The coherence of a fingerprint can also be expressed by true and partial positive rates, defined as the proportion of protein sequences that match all or only a part of the motifs in the fingerprint respectively. These fingerprint statistics are summarized in Table 1-I.

Motif. Individual motifs are characterized by their size and degree of conservation. Motif size is assessed in terms of length (number of amino acids) and depth (the number of protein sequences). A motif's depth is used to compute its coverage, i.e., the fraction of protein sequences in the fingerprint that match the motif. We explored two ways of measuring motif conservation. One alternative was to estimate a motif's entropy by averaging over the entropies of its individual columns (residues). Since motifs involving more protein sequences tend to have higher entropies, the result was normalized by dividing the average entropy over the number of sequences. The main objection of proteomics experts to the entropy-based approach was that it takes no account of domain knowledge concerning differential distances between amino acids; entropy computations assume a zero-or-one distance between residues, whereas it is a known biological fact that certain residues are more closely related than others. This knowledge has been codified in substitution matrices, among which Blosum matrices have been shown to achieve better overall performance [7]. On the basis of the Blosum-62 matrix, we computed a motif's blosum score by averaging over the blosum scores of its individual residues. In the absence of strong prior arguments in favor of

Table 1. Predictive information for fingerprint classification.

<table>
<tr><th colspan="2">Description</th><th>Variables</th></tr>
<tr><td colspan="3"><b>I. Fingerprint</b></td></tr>
<tr><td colspan="2">Number of motifs</td><td>nmt</td></tr>
<tr><td colspan="2">Number of proteins</td><td>npr</td></tr>
<tr><td colspan="2">True positive rate</td><td>tpr</td></tr>
<tr><td colspan="2">Partial positive rate</td><td>ppr</td></tr>
<tr><td colspan="3"><b>II. Motif</b></td></tr>
<tr><td colspan="2">Motif length (average, std, median, min, max)</td><td>mlen-A|S|D|N|X</td></tr>
<tr><td colspan="2">Motif coverage (average, stdev, median, min, max)</td><td>mcov-A|S|D|N|X</td></tr>
<tr><td colspan="2">Motif entropy (average, stdev, median, min, max)</td><td>ment-A|S|D|N|X</td></tr>
<tr><td colspan="2">Motif blosum score (average, stdev, median, min, max)</td><td>msco-A|S|D|N|X</td></tr>
<tr><td colspan="2">Intermotif distance (average, stdev, median, min, max)</td><td>mdis-A|S|D|N|X</td></tr>
<tr><td colspan="3"><b>III. Protein sequence</b></td></tr>
<tr><td colspan="2">SWISS-PROT ID: fraction of proteins with an ID</td><td>pSP</td></tr>
<tr><td>- LHS</td><td>frac of proteins whose LHS length $\geqslant$ 3|4 chars
frac of proteins with common first 1|2|3|4 chars in LHS
entropy of LHS averaged over first 1|2|3|4 chars</td><td>pN3, pN4
mj1, mj2, mj3, mj4
e1, e2, e3, e4</td></tr>
<tr><td>- RHS</td><td>frac of proteins with a common RHS (species)
entropy of RHS taken as a unit</td><td>mjr
er</td></tr>
<tr><td>CC</td><td>similarity: sequence belongs to family</td><td>cc-belongs</td></tr>
<tr><td>CC</td><td>similarity: sequence contains domain</td><td>cc-contains</td></tr>
</table>

either entropy or blosum scores, we decided to retain both, leaving it up to the feature selection process (Sec. 3.2) to sort out their relative effectiveness. A final characteristic concerns the distance between a motif and its nearest neighbor in the fingerprint; intuitively, large lengths and small intermotif gaps suggest closely related protein sequences. These motif characteristics are summarized in Table 1-II. For propositional learning, where the training unit is the fingerprint, we summarized each characteristic by computing its average, standard deviation, median, minimum and maximum over all fingerprint motifs.

Sequence. Each protein sequence is uniquely identified by its SWISS-PROT/TrEMBL ID or accession number. In an approach similar to that taken by PRECIS, we use these codes to retrieve the SWISS-PROT entry for the protein and examine this for information concerning the individual protein or the family to which it belongs. The SWISS-PROT ID field itself is particularly informative by virtue of its structure. It is composed of two parts separated by an underscore; the left hand side (LHS) denotes the protein type and the right hand side (RHS) the species. PRECIS' classification heuristics focus on the LHS, which tends to be homogeneous among members of a protein family. PRECIS searches for a common root of at least 2 characters in a set of sequence IDs; if such a root is found in at least 75% of these, the fingerprint is assumed to represent a family.

Rather than imposing fixed thresholds as PRECIS does, we simply isolated features that might correlate with fingerprint type and expressed them in terms of relative frequencies – e.g., the relative frequency of SWISS-PROT IDs in a set of proteins, or the proportion of IDs whose LHS is at least 3 characters long. We used two features to simulate PRECIS' homogeneity heuristic: (1) the majority score, defined as the proportion of LHSs sharing the most frequent common root of 1-4 characters, and (2) entropy as averaged over the first 1-4 characters of the LHS. For the right hand side, homogeneity was also quantified by the majority score and entropy, but computed this time over the RHS as a whole. This asymmetric processing of the 2 ID components aims to mimic unwritten conventions that appear to govern assignment of protein names in SWISS-PROT. In the LHS, biological homogeneity is suggested by the length of the leftmost common substring in a set of protein names; for instance the perfect uniformity of the LHS in JAK1_HUMAN, JAK1_MOUSE, JAK1_BRARE and JAK1_CYPCA suggests a tightly knit baselevel family while the 4th-letter variations in BAXA_HUMAN, BAXB_HUMAN and BAXD_HUMAN reflect interfamily differences within a superfamily. However, these conventions are implicit and short of perfectly consistent, hence the need for adaptive induction from examples rather than formulation as hard and fast rules.

Finally, we follow PRECIS' reliance on SWISS-PROT's CC similarity field, which often contains information about the family membership of a protein. This field's value can take the form belongs to <family-or-superfamily-name> or contains <domain-name>. However the information is not always consistent for all proteins in a fingerprint; rather than a boolean indicating the presence or absence of the flag words 'belongs to' or 'contains', we compute the proportion of proteins containing one or the other (whichever is more frequent).

3 Data Preprocessing and Mining Methods

3.1 Missing Value Imputation

The SWISS-PROT ID field is a valuable source of hints concerning the class of proteins in a fingerprint. Unfortunately, many proteins have no associated SWISS-PROT IDs. As a result for 7.5% of the training examples, all 12 features concerning the LHS and RHS of SWISS-PROT IDs had missing values. These values are clearly not 'missing completely at random' as defined in [8], since their presence or absence is contingent on the value of another feature, the fraction of SWISS-PROT identified proteins. This precludes the use of simple data completion methods such as replacement by means, which have the added drawback of underestimating variance. In addition, the distribution of incomplete features was diverse and far from normal; we thus imputed missing values using a non parametric technique based on K-nearest neighbors [11].

3.2 Feature Selection

The initial data representation described in Sec. 2 contained a total of 45 features or predictive variables, 30 based on the initial fingerprint and 15 on information culled from SWISS-PROT. It was not obvious which of these features were discriminating or redundant or even harmful. To obtain the minimal feature set needed to obtain reasonable performance, several feature selection methods were investigated and their impact on classification accuracy evaluated. We compared two variable ranking methods based on information gain or mutual information ($I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$), and symmetrical uncertainty ($U(X,Y) = 2\left[\frac{H(X)+H(Y)-H(X,Y)}{H(X)+H(Y)}\right]$). To account for feature interaction, we included ReliefF and correlation-based feature selection (CFS). CFS selects feature sets rather than individual features [6]; while ReliefF scores individual features, its nearest-neighbor based approach evaluates each feature in the context of all others and integrates the impact of irrelevant, noisy or redundant features [10].

3.3 Algorithm and Model Selection

To ensure coverage of the space of possible hypotheses, we investigated learning algorithms with clearly distinct biases. Among the basic algorithms we used were logic-based learning algorithms that build decision trees and rules (J48 and Part [12], variants of C5.0 tree and C5.0 rules respectively; Ltree, which builds oblique hyperplanes contrary to the orthogonal decision borders built by C5.0 [5]); density-estimation based learners like Naïve Bayes (NBayes) and instance-based learning (IBL, a variant of K-nearest-neighbors); linear discriminants (Lindiscr) and their diverse extensions such as multilayer perceptrons (MLPS), and support vector machines (SVMS). Details on each of these learning approaches can be found in, e.g., [4].

These methods represent different points along the bias-variance spectrum: NBayes and LinDiscr are extremely high-bias algorithms; at the other extreme, decision trees and rules, IBL, MLPs and SVMs are high-variance algorithms which can yield simple or very complex models depending on user-tuned complexity parameters. From the viewpoint of feature evaluation, sequential methods like orthogonal decision trees and rules consider predictive features successively while so-called parallel learners like NBayes, IBL, LinDiscr, MLPs, and SVMs evaluate all features simultaneously. Ltree is a hybrid sequential-parallel algorithm as it builds decision tree nodes sequentially but can create linear combinations of features at each node.

This set of basic algorithms was completed by two ensemble methods, boosted decision trees (C5.0boost) and RandomForest [2]. Ensemble methods build multiple models and classify new instances by combining (usually via some form of weighted voting) the decisions of the different models. Both methods use decision trees as base learners but differ in the way they diversify the training set. Boosting produces new variants of the training set by increasing the weights of instances misclassified by the model built in the previous training cycle, effectively obliging the learner to focus on the more difficult examples. RandomForest produces data variants by selecting features instead of examples. At each node, RandomForest randomly draws a subset of K (a user-defined parameter) features and then selects the test feature from this typically much smaller subset.

In order to find a reasonably good hypothesis, learning algorithms should be assessed in a variety of parameter settings. From the set of candidates described above, only the high-bias algorithms, NBayes and Lindiscr, have no complexity parameters; however NBayes has variants based on whether continuous variables are discretized (D) or not, in which case probabilities are computed either by assuming normality (N) or via kernel-based density estimation (K). For all the others, we tested a number of parameter settings and used only the best settings for inter-algorithm comparison. The main complexity parameter of recursive partitioning algorithms (J48, Part, and Ltree) is the C parameter, which governs the amount of postpruning performed on decision trees and rules. Its default value is 0.25 for J48 and Part and 0.10 for Ltree. We tried values of C from 1 to 50 in increments of 5. In IBL, the parameter K (the number of nearest neighbors to explore) produces a maximal-variance model when assigned the default value of 1. At the other extreme, with K = the number of training instances, IBL degenerates to the default majority rule. We explored the behavior of IBL with K=1, 10, 25, 40, 55, and 70. The topology of MLPs is governed by H, the number of hidden units. We tested H=1, 10, 23, 50, 75, and 100. RandomForest is governed by 2 main parameters – the number of trees which form the commitee of experts (I) and the number of features (K) to select for each tree. We explored combinations of I = 10, 25, 50, and 100 and K = 3, 6, 12, 18, 24. Finally, SVM complexity parameters depend on the type of kernel used. We tried polynomial kernels with degree E = 1 and 2 as well as radial basis function (Gaussian) kernels, with gamma (G) or width = 0.01 (default), 0.05, 0.1, 0.15, and 0.2. In addition, the regularization parameter C governs the trade-off between the empirical risk or training error and the complexity of the hypothesis. We explored values of C from 1 (default) to 100 in increments of 10.

4 Experimentation and Results

4.1 Experimental Strategy

The dataset contained 1842 fingerprint records from version 37 of the PRINTS database. 1487 cases were used as the design set, i.e., as training and validation sets for algorithm, feature, and model selection. The rest (355 cases) was held out for blind testing of the trained models. All experiments were conducted using stratified 10-fold cross-validation. Prior to training, missing values of incomplete examples were imputed as described in Sec. 3.1. Contrary to KNN-based missing value imputation, which is unsupervised, all feature selection techniques used (Sec. 3.2) rely on the training class labels and therefore had to be nested within the cross-validation loop.

4.2 Results on the Initial Feature Set

Table 2, column 3, summarizes the cross validated error rates of the learning algorithms. To provide a basis for comparison, the top give two baseline errors. The first is the traditional default error obtained by assigning all examples to the majority class. The class distribution of the 1842-instance training set is as follows: domain = 0.05, family = 0.54, superfamily=0.41. The majority rule thus yields a baseline error of 45.6% on the training set. A second yardstick, specific to the given task, is the error rate obtained by applying PRECIS' handcrafted classification heuristics. A simulation run of these heuristics on both the design set and the blind test set revealed an error rate of around 40%.

The obvious result is that the error rates of all learning algorithms are significantly better than both the default error of ∼46% and the PRECIS error of ∼40%. The advantage gained from data mining leaves no room for doubt. Note that the lowest errors in this application are obtained by either ensemble

Table 2. Error rates on the full 45-feature set. Each row gives the optimal parameter setting found for the given method, its cross-validation (CV) error on the design dataset, and its final test error on the holdout (HO) set.

Method	Parameters	CV error	HO error
Default		45.60	46.19
PRECIS		39.55	40.28
SVM-RBF	G=0.05, C=50	14.06	14.65
RandomForest	I=100, K=6	14.59	17.46
C5.0boost	B=10, C=0.1	15.13	18.59
MLP	H=10	15.13	16.62
IBL	K=10	15.47	19.44
Lindiscr	-	15.80	17.18
LTree	C=0.05	16.27	17.46
J48	C=0.01	16.48	19.15
Part	C=0.05	19.97	21.69
NBayes	K	23.20	27.07

methods (RandomForest and C5.0boost) or parallel learning algorithms which examine all features simultaneously (SVM-RBF, MLP, IBL). On the contrary, learning algorithms which test individual features sequentially (Ltree, J48, Part) are gathered together at the lower end of the performance scale. NBayes turned out to be the least accurate in all our comparative experiments, with kernel-based density estimation obtaining slightly better results than the variants that discretize continuous features or assume a normal distribution. NBayes aside, this clear performance dichotomy between parallel and sequential algorithms will be observed constantly in this study under different experimental settings.

The statistical significance of the differences in error rate is not clearcut. Without adjustment for multiple comparisons, the difference between the five lowest error rates is not statistically significant at the 1% level. However, after applying the Bonferroni adjustment for a total of around 500 pairwise tests, all statistically significant differences vanish among the first 8 models. Nevertheless, we selected the model with the lowest nominal error – SVM-RBF with a kernel width of 0.05 and a complexity parameter C of 50.

The result of algorithm and model selection was then validated on the blind test set. Since cross-validation produces a different model at each iteration, the selected SVM-RBF parameterization was rerun on the full training set and applied to the blind test set of 355 examples. The error rate obtained was 14.65%, confirming that the observed cross-validation error of 14.06% resulted not from overfitting but from effective generalization. As a countercheck, the other candidate models were also run on the holdout set; the results are shown in the last column of Table 2. The difference between the cross-validation and the blind test error is less than 0.6% for SVM-RBF but varies between 1.2% and 4% for all other algorithms, the highest blind test error (NBayes) exceeding 27%. This remarkable stability of SVM-RBF, added to its predictive accuracy, parsimony, and reasonable computational speed, confirms and magnifies the advantage of SVM-RBF over the other learning methods on this specific classification task.

4.3 Results of Feature Selection

To find the minimal feature set needed for accurate prediction and see which features were truly discriminating, we applied the feature selection methods described in Section 3.2. The number of features to retain was automatically determined by the subset selector CFS in backward search mode but had to be supplied by the user for the three feature rankers ReliefF, InfoGain, and SymmU (we tried 32, 36, and 40 features).

Results are summarized in Table 3. Each row shows the specific combination of model parameters, feature selection method, and number of selected features that produced the lowest cross-validation error (col. 5) for a given learning algorithm (col. 1). The first obvious finding is that feature selection improves performance for all learning algorithms except SVM-RBF, which achieves the same error rate with 36 features as with the initial set of 45 features. Nevertheless, SVM-RBF conserves its top rank; in fact, the most remarkable result is that the overall ranking of learning algorithms remains the same before and after feature selec-

Table 3. Cross-validation and holdout error rates after feature selection.

Method	Parameters	Feature selector	# features	CV error	HO error
SVM-RBF	G=0.05, C=90	ReliefF	36	14.09	14.08
RandomForest	I=25, K=12	InfoGain	40	14.19	16.61
C5.0boost	C=0.3	ReliefF	32	14.79	16.90
MLP	H=10	ReliefF	40	14.86	16.90
IBL	K=10	SymmU	32	14.93	18.31
Lindiscr	-	ReliefF	40	15.40	17.46
LTree	C=0.05	SymmU	32	15.53	18.59
J48	C=0.10	SymmU	32	15.53	19.72
Part	C=0.10	CFS	7-10	17.35	18.03
NBayes	K	CFS	7-10	18.02	23.66

tion. To validate the observed results, we reran these ten learning configurations on the full training set and applied the resulting models to the blind test set. Here again, we observe the same phenomenon as on the initial feature set: the holdout error of SVM-RBF is 14.08%, practically identical to its cross-validation error of 14.06%; for all other algorithms the holdout error was higher than the cross-validation error by an average of 2.84%.

5 Discussion

This section addresses two issues related to the findings described above. First, what is the source of SVM-RBF's generalization power on this particular task? Second, how can we assess the relative impact of domain-specific (e.g. Blosum scores) and domain-independent features (e.g. entropy) on the discriminatory ability of the trained model?

One hypothesis that might explain the performance of SVM-RBF on this task is its approach to multiclass learning. Rather than solve a C-class problem directly like most of the other algorithms studied, it builds a decision frontier by building and combining the responses of $\binom{C}{2}$ pairwise binary classifiers. In this sense SVM-RBF could be viewed as an ensemble method and the rankings given in Tables 2 and 3 would simply confirm the widely observed efficacy of model combination versus individual models in many classification tasks. This hypothesis is however weakened by the fact that Lindiscr follows the same pairwise binary approach to multiclass problems and yet displays worse performance. To see more clearly into the issue, we reran J48, Part, IBL, NBayes and MLP with the same parameters as in Table 2, but this time adopting SVM's pairwise binary classification strategy. Holdout error increased slightly for MLP and J48, and improved somewhat for the others. However, no error improvement led to a performance level comparable to SVM-RBF's 14.65% holdout error. While the binary pairwise approach may have favorably affected accuracy, it cannot be considered the main source of SVM-RBF's generalization performance.

A complementary explanation can be found by comparing the performance of parallel and sequential learners, as noted in Sec. 4.2. Recursive partitioning methods generally fared badly on this problem, as shown clearly in Tables 2 and 3. Exceptions are the two ensemble methods where the myopia of sequential feature testing is compensated by iterative resampling, whether instance-wise (boosting) or feature-wise (RandomForest). These cases aside, parallel algorithms take the top six performance ranks, with or without feature selection. The clear performance dichotomy between parallel and sequential algorithms suggests a strong interaction among the 45 features distilled from fingerprints. This conjecture is further supported by results of sensitivity analyses described below.

The second issue concerns the relative contributions of domain-specific (e.g., Blosum scores, PRECIS heuristics) and domain-independent features (e.g. entropy measures) to the discriminatory power of the final classifier. We examined separately features describing motif conservation and those describing the set of collected proteins. Motif conservation in a fingerprint is depicted by 2 groups of features, one based on domain-specific Blosum scores and another on generic entropy measures (Sec. 2). To compare their relative effectiveness, we removed each feature set at a time and trained the selected SVM-RBF learner on the remaining features. Each time, the resulting increase in error was taken to quantify the impact of the excised feature set on classification performance. Finally, we removed both feature sets simultaneously to estimate their combined impact.

The results are shown in Table 4(a). Error increase was slight for both feature sets and neither appeared to have a convincingly higher impact on accuracy than the other. Even the combined impact on performance differed little from the individual contribution of one or the other. Blosum scores and entropy not only seem to have roughly equivalent and redundant predictive impact; their combined contribution is scarcely greater. We see two possible explanations: either motif conservation has a minor role in discriminating fingerprint types, or an adequate representation of motif conservation remains to be found.

Knowledge-based and knowledge-poor features concerning fingerprint proteins displayed quite different trends. One set of features embodied expert knowledge underlying the PRECIS heuristics while another set comprised less informed

Table 4. Impact of knowledge-based and knowledge-poor features. Error" and "Perf Impact" indicate respectively the error and error increase (wrt to the baseline) entailed by removal of a given feature set.

	CV		HO	
Baseline: Full feature set (SVM-RBF)	14.06		14.65	
(a) Features describing motif conservation				
	Error	Perf Impact	Error	Perf Impact
Uninformed (entropy)	15.33	1.28	16.34	2.28
Knowledge-based (Blosum scores)	14.85	0.81	16.62	2.56
Both	15.06	1.01	16.62	2.56
(b) Protein-related features				
	Error	Perf Impact	Error	Perf Impact
Uninformed (see Section 2)	24.88	10.83	29.86	15.80
Knowledge-based (PRECIS rules)	14.53	0.47	15.49	1.44
Both	27.10	13.05	33.52	19.47

features such as simple statistical and entropy measures on the left and right components of SWISS-PROT protein IDS. We followed the same procedure as above to quantify their respective contributions to generalization power; these are summarized in Table 4(b). When PRECIS-based features were removed, hold-out error increased by 1.44% whereas removal of uninformed features incurred a degradataion of 15.8% (cross-validation errors display the same behavior). It is clear that uninformed features are contributing much more to classification performance (note however that the 3 knowledge-based features are heavily outnumbered by the 12 uninformed features). Remarkably, when both feature sets were deleted, error climbed to 27.10%, much more than the sum of their individual contributions.

To summarize, these motif- and protein-centered views of fingerprints reveal two distinct feature interaction scenarios. In one case, the domain-specific and domain-independent feature sets have roughly comparable contributions to predictive accuracy; their combination seems to add nothing to either alone, but it is unclear which should be kept. In the second case one feature set is clearly more effective than the other, but their combined contribution to generalization performance suggests a synergy that individual feature rankers or sequential learners are at pains to capture. This could explain the observed superiority of parallel learners like SVMs and MLPS on this particular problem.

6 Conclusion and Future Work

Since this classification task is a preliminary step in the time-consuming process of annotating protein fingerprints, it is important to achieve high accuracy in order to avoid even more tedious backtracking and database entry revision. The approach described in this paper achieved a 26% accuracy increase relative to the performance of expert rules; the goal of ongoing and future work is to decrease further the residual error of 14.1%. There is a diversity of ways to achieve this; we explored 2 alternatives with negative results.

The first unfruitful track is the relational learning approach. As seen in Section 2, protein fingerprints have a natural multirelational flavor since they gather information on diverse object types – the fingerprints themselves and their component motifs and protein sequences. Relational learning thus seemed to be a way of gaining accuracy via increased expressive power. However, our experiments in relational instance-based learning were inconclusive; they incurred considerably higher computational costs but did not yield better performance than the propositional approach reported above.

The second track explored was the combination of multiple learned models. We investigated the efficacy of combining learned models with uncorrelated errors to obtain a more accurate classifier. We measured the pairwise error correlation of the 10 classifiers in Table 2. SVM-RBF, NBayes and Part were among those that had the lowest pairwise error correlations. We built an ensemble model which combined the predictions of these three learners by a simple majority vote. The error rate of the combined model was 15.87% on the training set and 18.03%

on the holdout test set – in both cases higher than that of SVM-RBF alone. Several other model combinations based on low error correlation were explored; they all yielded higher error rates than at least one of their component base learners.

Other hopefully more promising paths remain to be explored. Ongoing work is focused on correcting data imbalance to increase accuracy. Protein domain families represent less than 5% of PRINTS records, and we are adapting to this task a set of class rebalancing techniques that have proved effective in another application domain [3]. Perhaps the biggest remaining challenge is that of bringing more discriminatory information to bear on the classification task. Integrating information from databases other than SWISS-PROT is a feasible solution in the short term. But given the time lag between the production of new data and their availability in structured databases, we may ultimately have to mine the biological literature to gather fresh insights on the 14% of protein fingerprints that currently defy classification.

Acknowledgements

The work reported above was partially funded by the European Commission and the Swiss Federal Office for Education and Science in the framework of the BioMinT project.

References

1. T. K. Attwood, P. Bradley, D. R. Flower, A. Gaulton, N. Maudling, and A. L. Mitchell et al. PRINTS and its automatic supplement, prePRINTS. *Nucleic Acids Research*, 31(1):400–402, 2003.
2. L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
3. G. Cohen, M. Hilario, H. Sax, and S. Hugonnet. Data imbalance in surveillance of nosocomial infections. In *Proc. International Symposium on Medical Data Analysis*, Berlin, 2003. Springer-Verlag.
4. R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley, 2000.
5. J. Gama and P. Brazdil. Linear tree. *Intelligent Data Analysis*, 3:1–22, 1999.
6. M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conference on Machine Learning*, 2000.
7. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. National Academy of Sciences USA*, 89:10915–10919, November 1992.
8. R. J. Little and D. B. Rubin. *Statistical Analysis with Missing Data.* Wiley, 1987.
9. A. L. Mitchell, J. R. Reich, and T. K. Attwood. PRECIS–protein reports engineered from concise information in SWISS-PROT. *Bioinformatics*, 19:1664–1671, 2003.
10. M. R. Sikonja and I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2003.
11. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.
12. I. Witten and E. Frank. *Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 2000.