Published online 1 August 2007

Nucleic Acids Research, 2007, Vol. 35, No. 15 ***e96***
doi:10.1093/nar/gkm562

Meta-prediction of protein subcellular localization with reduced voting

Jie Liu[1], Shuli Kang[1], Chuanning Tang[1], Lynda B.M. Ellis[2] and Tongbin Li[1,*]

[1]Department of Neuroscience and [2]Department of Laboratory Medicine and Pathology, University of Minneapolis, MN 55455, USA

Received May 15, 2007; Revised June 27, 2007; Accepted July 9, 2007

ABSTRACT

Meta-prediction seeks to harness the combined strengths of multiple predicting programs with the hope of achieving predicting performance surpassing that of all existing predictors in a defined problem domain. We investigated meta-prediction for the four-compartment eukaryotic subcellular localization problem. We compiled an unbiased subcellular localization dataset of 1693 nuclear, cytoplasmic, mitochondrial and extracellular animal proteins from Swiss-Prot 50.2. Using this dataset, we assessed the predicting performance of 12 predictors from eight independent subcellular localization predicting programs: ELSPred, LOCtree, PLOC, Proteome Analyst, PSORT, PSORT II, SubLoc and WoLF PSORT. Gorodkin correlation coefficient (GCC) was one of the performance measures. Proteome Analyst is the best individual subcellular localization predictor tested in this four-compartment prediction problem, with GCC = 0.811. A reduced voting strategy eliminating six of the 12 predictors yields a meta-predictor (RAW-RAG-6) with GCC = 0.856, substantially better than all tested individual subcellular localization predictors ($P = 8.2 \times 10^{-6}$, Fisher's Z-transformation test). The improvement in performance persists when the meta-predictor is tested with data not used in its development. This and similar voting strategies, when properly applied, are expected to produce meta-predictors with outstanding performance in other life sciences problem domains.

INTRODUCTION

In the past decade, increased availability of large amounts of life sciences data, including low-throughput data accumulated by generations of scientists over a half-century, and high-throughput data acquired through newly developed biotechnologies, has coincided with great advances in data analysis and modeling techniques, most notably in the machine-learning area, leading to an increase in computational prediction programs in various important domains in life sciences research. In more and more problem domains, multiple prediction programs have emerged from independent efforts by different groups. These programs differ by what data features they use, or by what the methods or algorithms they apply in the classification tasks, or by both. These prediction programs may be complementary; i.e. one program performs better for one type of data under one set of circumstances, but another prediction program performs better for another type of data or under other circumstances. By proper exploitation of the combined strengths of these prediction programs, it may be possible to construct *meta-predictors* whose performance surpasses that of all existing prediction programs.

The meta-prediction problem is one that seeks to construct a prediction program (termed a *meta-predictor*), which makes predictions by organizing and processing the prediction results of a number of other prediction problems (termed *element predictors*). The meta-predictor takes the output of element predictors as its sole input. No explicit attention is paid to the feature definition or the underlying classification algorithms of the individual element predictors. Rather, the strengths and weaknesses of each element predictor, and the similarities and differences between different element predictors are visible to the meta-predictor only through the prediction results they make. The hope of meta-prediction is to develop a meta-predictor, which can combine the strengths of the element predictors and produce more accurate predictions than any of the element predictors. In this study, we focus on the meta-prediction of subcellular localization of proteins.

Subcellular localization is a key functional characteristic of eukaryotic proteins. Most proteins must be localized to the correct subcellular compartment or organelle in order to properly execute their biological function(s). Cooperating proteins must be present in the same location in order for them to interact. Since Nakai and Kanehisa's pioneering work (1), a large number of computational

*To whom correspondence should be addressed. Tel: +1 612 626 3481; Fax: +1 612 626 5009; Email: toli@biocompute.umn.edu

prediction programs have been developed in this field [see recent reviews, e.g. (2,3)]. These programs use many different data features, such as N-terminal signal sequence information [TargetP (4), PSORT (5) and iPSORT (6)]; amino acid composition [NNPSL (7), PLOC (8), FUZZY_LOC (9) and SubLoc (10)]; evolutionary information obtained by multiple sequence alignment or PSI-BLAST, and/or calculated physicochemical properties of the proteins [Proteome Analyst (11), LOCSVMPSI (12), ESLpred (13) and LOCtree (14)]; 3D structural data [LOC3d (15)]; or even gene expression data (16). They also use many different classification methods, such as expert systems (PSORT, iPSORT); artificial neural networks (ANN) (LOCnet, LOC3d, TargetP); *k*-nearest neighbor (*k*-NN) [PSORT II (5)]; Naive Bayes (NB) classifier (Proteome Analyst); fuzzy *k*-NN (FUZZY_LOC); or support vector machines (SVM) (SubLoc, LOCSVMPSI, PLOC, ELSpred and LOCtree).

These different data features and classification methods may give these prediction programs different, complementary strengths. In this study, we develop meta-predictors that harness the combined strengths of these individual element predictors. We first compiled an unbiased subcellular localization dataset that does not overlap with any data used in the development of these predictors; we then examined the performance of these predictors using this unbiased dataset; and explored several voting-based strategies for constructing meta-predictors. We show that, using a simple reduced voting strategy, an excellent meta-predictor can be developed, with a predicting performance substantially exceeding that of all element predictors, and that this meta-predictor's excellent performance persists with data not used in its development.

MATERIALS AND METHODS

Compilation of MetaSCL06 dataset

In this study, we focus on the subcellular localization predictions of animal proteins. The unbiased protein subcellular localization dataset MetaSCL06 was compiled from Swiss-Prot Release 50.2 (12 July 2006). The compilation procedure consisted of four steps: (1) assembling an unbiased set of proteins, (2) assigning class labels to the proteins based on gene ontology (GO) annotations in Swiss-Prot entries, (3) assigning class labels to the proteins based on the comment field in Swiss-Prot entries, and (4) manual reconciliation of protein sets from steps 2 and 3 (Figure 1).

Step 1: Assembling an unbiased set of proteins. For unbiased testing, the dataset compiled for this study should not contain data used in the development of any element prediction programs. An approach similar to (17) was taken, where the original report of each individual prediction program was carefully examined for descriptions about the data sources used in the development of the program. In the original reports of all but two prediction programs (PSORT and PSORT II), the Swiss-Prot database was explicitly stated as the original data source used in development, and the release numbers of the database were also provided. The latest release used in the development of these prediction programs was 45.0 (used in the development of WoLF PSORT), with release date 25 October 2004. This date was chosen as the cutoff date. For PSORT and PSORT II, since these two predictors were developed much earlier than other programs, and the web servers for these two programs have not been updated since November 1999 (nearly five years prior to the cutoff date), it is highly unlikely that any data used in the development of these two programs would be included in our protein dataset.

All animal protein sequences bearing an initial entry date after 25 October 2004 in Swiss-Prot 50.2 were the initial start of our unbiased protein dataset. All protein sequences with lengths <30 were discarded, because predictions made on shorter sequences were much less reliable for all element predictors (data not shown). In total, 14 246 proteins were retained at this step.

Step 2: Assigning class labels to proteins based on GO annotations. In this study, we focus on classifying proteins localized in four subcelluar compartments—nuclear, cytoplasmic, mitochondrial and extracellular.

Roughly 20% of the Swiss-Prot entries contain the Category C (denoting 'cellular components') GO annotations in their DR field, based on which a class label indicating one of the four subcellular compartments can be assigned to the corresponding protein. The GO annotations are considered more reliable than the comment annotations in the CC field (see Step 3 below) because they resulted from an additional round of manual curation by Swiss-Prot staff. However, assigning class labels based on GO annotations is not a straightforward task, because multiple GO terms frequently appear in the annotation of a single Swiss-Prot entry, and these terms are often of parent–child relationships in the GO hierarchical structure. In this structure (represented as a DAG, or directed acyclic graph), GO terms corresponding to the four subcellular compartments are interconnected through their parent terms. A procedure was developed to assign class labels to as many proteins as possible with consistent GO annotations (see Supplementary Methods and Supplementary Table 1). At this step, 555 proteins (305 nuclear, 29 cytoplasmic, 105 mitochondrial and 116 extracellular) in the unbiased protein set were assigned class labels.

Step 3: Assigning class labels to proteins based on annotations on comment field. All proteins in the unbiased protein set obtained at Step 1 were fed into a keyword filter, in which the comment field (CC) was checked against a list of keywords. Class labels were assigned to the proteins using an established procedure (see Supplementary Methods and Supplementary Table 2). At this step, 1595 proteins (500 nuclear, 277 cytoplasmic, 154 mitochondrial and 664 extracellular) received class labels.

Step 4: Manual reconciliation. Finally, all proteins that received class labels in the Steps 2 and 3 were subject to

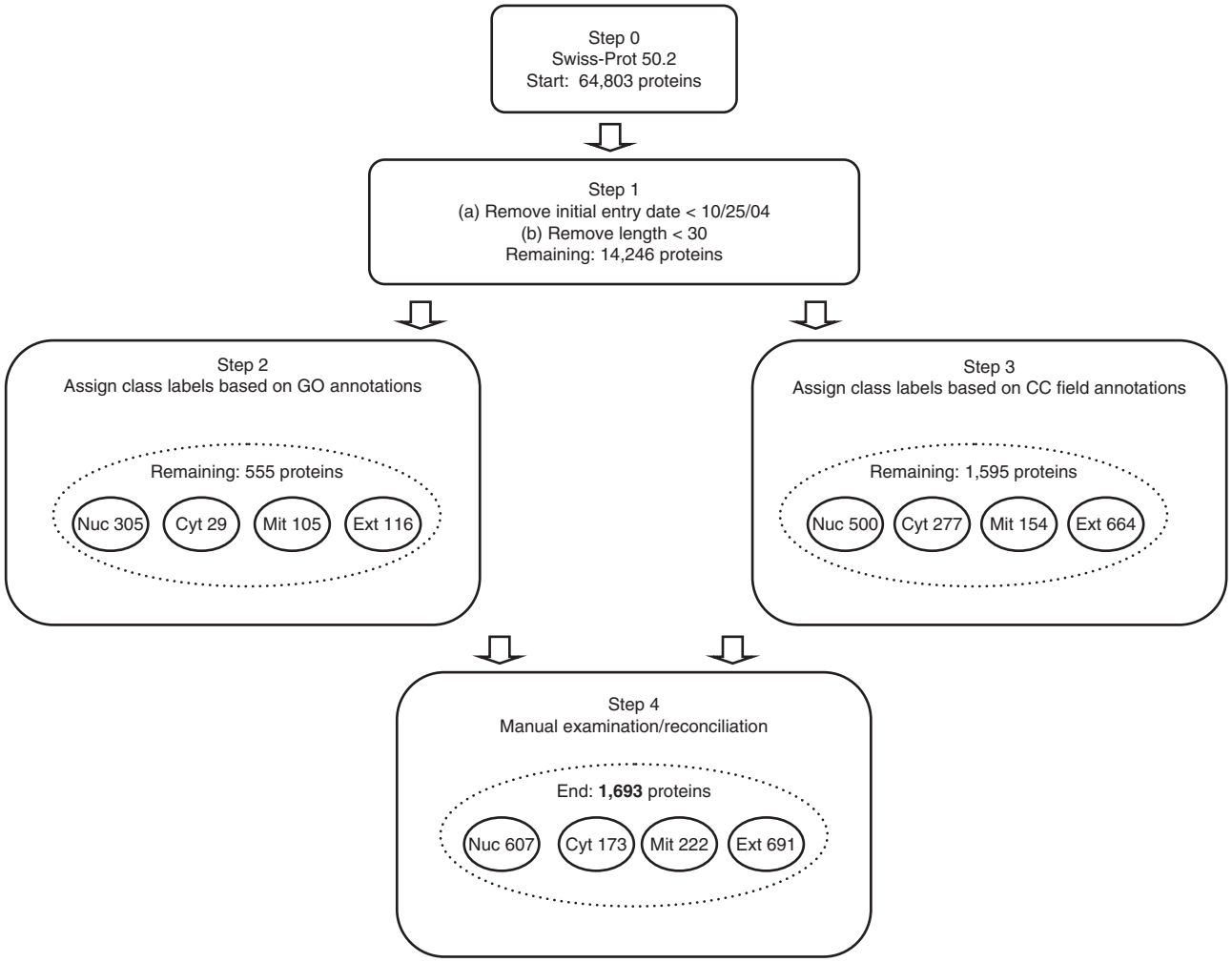

Figure 1. Compiling the MetaSCL06 dataset. Nuc: nuclear; Cyt: cytoplasmic; Mit: mitochondrial; Ext: extracellular.

manual reconciliation. Entries were removed in cases of uncertainty or when there were conflicts between the class labels assigned in the last two steps. The final MetaSCL06 dataset includes 1693 proteins (607 nuclear, 173 cytoplasmic, 222 mitochondrial and 691 extracellular). This dataset is available as Supplementary Table 3.

Inconsistent annotation in GO categories and/or comment fields may represent rare but real cases of individual proteins present in multiple subcellular compartments. For simplicity, these proteins were removed from the MetaSCL06 dataset.

Compilation of MetaSCL07 dataset

The MetaSCL07 dataset is a validation set that was not used in meta-predictor development. This dataset was compiled from Swiss-Prot Release 51.6 (6 February 2007), with the same procedure used in the compilation of MetaSCL06. All entries of proteins bearing an initial entry date on or before 12 July 2006 (date of Release 50.2) were removed. This dataset includes 579 proteins (145 nuclear, 50 cytoplasmic, 144 mitochondrial and 240 extracellular). This dataset is available as Supplementary Table 4.

Selection of element predictors

In order to be usable as an element predictor for the meta-prediction problem, a prediction program needs to be accessible online or be available in downloadable form. Several predicting programs, including NNPSL, FUZZY_LOC, LOCnet and LOCSVMPSI, were excluded from consideration, because the implementations of these prediction programs are no longer available. Since a vast majority of the remaining prediction programs take full-length protein sequences as their only input, we focused on these. Programs requiring structural information (e.g. LOC3d) were excluded. Programs that calculate structure-related features internally (e.g. PSORT, PSORT II, LOCtree and ELSpred) were acceptable because these features are calculated based on the protein sequences, and the latter are the only input needed from the user.

A majority of the prediction programs make predictions on at least four major subcellular compartments: nuclear, cytoplasmic, mitochondrial and extracellular. Thus we focused on prediction of these four compartments. Two prediction programs, TargetP and iPSORT, were excluded because they make predictions on mitochondrial,

Table 1. Summary of the 12 element predictors[a]

Element predictor	Reference	URL	Other subcellular compartments predicted[b]
ELSpred_comp ELSpred_physico ELSpred_dipeptide ELSpred_EuPSI ELSpred_hybrid	(13)	http://www.imtech.res.in/raghava/eslpred/submit.html	None
LOCtree	(14)	http://cubic.bioc.columbia.edu/cgi/var/nair/loctree/query	Organelles
PLOC	(8)	http://www.genome.jp/SIT/plocdir/	Cytoskeleton, Endoplasmic reticulum, Golgi apparatus, Lysosome, Peroxisome, Plasma membrane
Proteome Analyst	(11)	http://pasub.cs.ualberta.ca:8080/pa/Subcellular	Endoplasmic Reticulum, Golgi apparatus, Lysosome, Peroxisome Plasma membrane
PSORT	(5)	http://psort.ims.u-tokyo.ac.jp/form.html	Endoplasmic reticulum, Golgi apparatus, Lysosome, Microbody, Plasma membrane
PSORT II	(5)	http://psort.ims.u-tokyo.ac.jp/form2.html	Cytoskeleton, Endoplasmic reticulum, Golgi apparatus Lysosome, Plasma membrane, Peroxisome, Secretory vesicles
SubLoc	(10)	http://www.bioinfo.tsinghua.edu.cn/SubLoc/	None
WoLF PSORT	(18)	http://wolfpsort.seq.cbrc.jp/	Cytoskeleton, Endoplasmic reticulum, Golgi apparatus, Lysosome, Peroxisome, Plasma membrane

[a]For data features and classification methods, see text.
[b]Other subcellular compartments besides the four compartments focused on in this study: nuclear, cytoplasmic, mitochondria and extracellular.

chloroplast, and secretory pathway, but do not make predictions on nuclear, cytoplamic or extracellular proteins.

The final list of prediction programs and the chosen element predictors are shown in Table 1. A total of 12 element predictors were chosen, derived from eight prediction programs. Each program is discussed below:

ELSpred (13). ELSpred uses the one-versus-the-rest SVM as the underlying classification method. It makes predictions into the four common subcellular localization compartments: nuclear, cytoplasmic, mitochondrial and extracellular. ELSpred provides five prediction options, each corresponding to a different feature formulation scheme: ELSpred_comp uses the compositions of the 20 amino acids as its features. ELSpred_physicochemical uses 33 physicochemical properties as its features. ELSpred_dipeptide defines features using dipeptide compositions. The features used in ELSpred_EuPSI are constructed following three iterations of EuPSI-BLAST through which the similarity between the protein and 2427 eukaryotic proteins is obtained. ELSpred_hybrid, uses a feature scheme that combines all the above four feature schemes. These five prediction options are considered as different element predictors in this study.

LOCtree (14). LOCtree uses amino acid compositions of the proteins as its features. It goes through a three-level binary tree-structured process with a binary SVM model working at each node in the tree. In addition to the four common subcellular localization compartments, LOCtree also makes predictions about whether a protein is an organelle protein, and about whether a nuclear protein is a DNA-binding protein.

PLOC (8). PLOC uses five different types of compositions (amino acids, amino acid pairs, one gapped amino acid pairs, two gapped amino acid pairs and three gapped amino acid pairs) as its features. The predictions are made by one-versus-the-rest SVMs followed by voting. In addition to the four common subcellular localization compartments, PLOC also makes predictions into six other subcellular localization compartments: cytoskeleton, endoplasmic reticulum, the Golgi apparatus, lysosome, peroxisome and plasma membrane.

Proteome Analyst (11). Proteome Analyst adopts features calculated with PSI-BLAST against the Swiss-Prot database, and employs a Naïve Bayes (NB) algorithm for making predictions. Besides the four common subcellular compartments, Proteome Analyst also makes predictions into five additional subcellular compartments: endoplasmic reticulum, Golgi apparatus, lysosome, peroxisome and plasma membrane.

PSORT and *PSORT II* (5). PSORT and PSORT II utilize a large number of features, including the presence of N-terminal sorting signals, the presence of RNA/DNA-binding motifs, amino acid compositions and some calculated structural information. PSORT is a knowledge-based system with a set of 'if-then' rules. PSORT II employs a *k*-NN learning algorithm. Besides the four common subcellular compartments, PSORT also makes predictions into endoplasmic reticulum, the Golgi apparatus, lysosome, microbody, plasma membrane, and PSORT II makes predictions into cytoskeleton, endoplasmic reticulum, the Golgi apparatus, plasma membrane, peroxisome and secretary vesicles.

SubLoc (10). SubLoc uses amino acid compositions as its features. Four one-versus-the-rest SVMs are trained to make predictions on a given protein into one of the four common subcellular localizations.

WoLF PSORT (18). WoLF PSORT defines features using amino acid compositions and N-terminal signals that are encoded by AAindex, and also adopts some PSORT features. A *k*-NN classifier is trained following the WoLF feature selection and weighting procedure.

WoLF PSORT makes predictions into cytoskeleton, endoplasmic reticulum, the Golgi apparatus, lysosome, peroxisome and plasma membrane in addition to the four common subcellular compartments.

Obtaining and pre-processing prediction results of element predictors

Prediction jobs were submitted to each of the element prediction programs with the protein sequences in the MetaSCL06 and MetaSCL07 datasets. Some of the element prediction servers, ELSpred, PSORT, PSORT II and PLOC, do not provide a batch-processing option. For these prediction servers, simple Java programs were developed and used to handle the job submission and result retrieval. Other Java programs were developed to parse and analyze the prediction results returned from the element prediction servers. Some prediction programs, including PLOC, ELSpred_EuPSI and ELSpred_hybrid, provide in their output the most likely subcellular compartment, but other prediction programs generate numerical scores in their output, for example, the 'reliability indices' produced by LOCtree and SubLoc, the 'certainty score' produced by ELSpred_comp, ELSpred_physicochemical and ELSpred_dipeptide, and the percentage scores produced by Proteome Analyst and PSORT II. When multiple compartments appeared in the output with numerical scores, the one with the highest value was picked as the predicted compartment. Two of the prediction servers (PLOC and WoLF PSORT) can predict two compartments for a single protein both with the highest scores (e.g. 'nuclear' and 'cytoplasmic'). In these cases, both predictions were considered valid and they were given equal weights when the predicting performance of the prediction program was evaluated.

Performance measures

For a two-class classification problem, commonly used performance measures include sensitivity, specificity, accuracy and Matthew's correlation coefficient (MCC) (19). These measures are defined as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}},$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}},$$

and

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FN} \cdot \text{FP}}{\sqrt{(\text{TN} + \text{FN})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TP} + \text{FP})}}.$$

where TP, TN, FP and FN and denote the numbers of true positive, true negative, false positive and false negative samples in the classification.

For a multi-class classification problem, the definitions of sensitivity, specificity and MCC are no longer valid, but that of accuracy continues to be useful. In addition, Gorodkin (20) defined a correlation coefficient formula, which we will call Gorodkin correlation coefficient (GCC), a measure of predicting performance for a multi-class classifier. GCC calculates the correlation between two $N \times K$ matrices, the observation matrix $\underline{\underline{O}}$ and the prediction matrix $\underline{\underline{P}}$, where N is the number of samples, and K is the number of classes. In the protein subcellular localization prediction problem, K is equal to four for meta-predictors and the element predictors making predictions into the four common subcellular compartments only (e.g. SubLoc). For element predictors that make predictions into the 'other compartment' class (e.g. LOCtree and PLOC), K is equal to 5. An element in the observation matrix, $\underline{\underline{O}}_{ij}$, is set to be 1 if the ith sample is known to belong to class j, and it is set to be 0 if otherwise. An element in the prediction matrix, $\underline{\underline{P}}_{ij}$, is set to be 1 if the ith sample is predicted to belong to class j by the predictor, and it is set to be 0 if otherwise. For PLOC and WoLF PSORT, if a sample is predicted into two equally probable compartments, the corresponding elements in the prediction matrix are set to be 1/2 for both compartments. If no prediction is made for a sample by an element predictor, all corresponding elements in the prediction matrix are set to be 0.

Given the observation matrix $\underline{\underline{O}}$ and the prediction matrix $\underline{\underline{P}}$ GCC is defined as follows:

$$GCC = \frac{COV(\underline{\underline{O}},\underline{\underline{P}})}{\sqrt{COV(\underline{\underline{O}},\underline{\underline{O}})}\sqrt{COV(\underline{\underline{P}},\underline{\underline{P}})}},$$

where $COV(\underline{\underline{O}},\underline{\underline{O}})$, $COV(\underline{\underline{O}},\underline{\underline{P}})$ and $COV(\underline{\underline{P}},\underline{\underline{P}})$ are the covariance of the corresponding matrices, defined as the arithmetic average of the covariance of corresponding columns of the matrices.

GCC has the following desirable characteristics: it has a range of [−1, 1], just like Pearson's correlation coefficient and MCC. The more accurate the prediction is, the closer GCC is to 1. When the number of classes is equal to 2 in the classification problem, the definition of GCC reduces to the familiar MCC formula.

Comparison of element predictors

The element predictors are not completely compatible with one another in the types of predictions they make. First, 6 of the 12 element predictors make predictions for other subcellular compartments than the four we choose to use (Table 1). For simplicity, we lump all other subcellular compartments together for each of these programs and call them 'other compartments'. Since the gold standard dataset (MetaSCL06) contains data for the four chosen compartments only, any predictions made into the 'other compartments' class by any element predictors are classified as wrong predictions. Second, some element predictors do not make predictions for all proteins. For instance, Proteome Analyst made predictions on 1523 of the 1693 proteins in the MetaSCL06 dataset, and ELSpred_EuPSI made predictions on only 624 of the 1693 proteins in the dataset. For the sake of making a fair performance comparison, we classed these 'no prediction' cases as wrong predictions when

calculating the accuracy of these element predictors. However, we adopt an additional performance measure to assess the performance of an element predictor for the proportion of proteins in the dataset where predictions are actually made. We term this accuracy measure the 'relative accuracy', and calculate it as the ratio of the number of corrected predicted proteins and the number of proteins for which predictions are made. Finally, for some proteins, some element predictors (e.g. PLOC and WoLF PSORT) output two predicted subcellular compartments, which are considered equally probable by the predictors. For these samples, the predictions are considered 'half correct' if one of the two compartment output matches the true compartment label for the protein in the dataset.

Unweighted voting strategy

For a given protein in the dataset, the unweighted voting meta-predictor makes prediction P_{uv} as

$$P_{\mathrm{uv}} = \arg\max_i \sum_{j=1}^{12} P(i,j),$$

where i is the index of the subcellular compartments: $i = 1$ denotes 'nuclear', $i = 2$ denotes 'cytoplasmic', and $i = 3$ and $i = 4$ denote 'mitochondria' and 'extracellular', respectively. $P(i,j)$ describes the prediction made by the jth ($j = 1, \ldots, 12$) element predictor: $P(i,j) = 1$ if the predictor made by the jth element predictor is i, and $P(i,j) = 0$ if otherwise. If, for a given protein, two equally probable compartments are output by an element predictor (PLOC or WoLF PSORT), the score is split so that $P(i,j) = 1/2$ for both compartments. The notation 'arg $\max_i$' stands for the 'argument of the maximum', and it returns the value of i that leads to the highest value of the formula that follows (in this case, $\sum_{j=1}^{12} P(i,j)$). That is, for each input protein, the unweighted voting meta-predictor sums the number of element predictors that make positive predictions for each of the four subcellular localization compartments, then picks the compartment with the largest number. When there are two or more compartments with the highest score, one compartment is picked at random.

Weighted voting strategy

The weighted voting strategy differs from the unweighted voting strategy in that the predictions made by element predictors are multiplied by a weight, which varies among predictors, before being summed up to produce the prediction of the meta-predictors. In other words, the prediction made by a weighted voting meta-predictor, P_{wv}, is described as

$$P_{\mathrm{wv}} = \arg\max_i \sum_{j=1}^{12} [P(i,j) \cdot w_j],$$

where w_j is the weight for element predictor j ($j = 1$ through 12).

Weights are set to reflect the predicting performance of the element predictors: an element predictor with higher predicting performance is given a higher weight.

Reduced voting strategy

Although the prediction results of all element predictors are available to the meta-predictors, it is not necessary for all of them to be used. Indeed, if we exclude from consideration some of the element predictors that do not perform well, it may be possible to obtain meta-predictors with further improved performance. Thus, we applied the so-called 'reduced voting strategy': starting from a full (or 'unreduced') meta-predictor, we iteratively reduce the number of element predictors included in the construction of meta-predictor, by picking the next element predictor with the lowest performance, and setting its weight to 0. This process continues until only one element predictor remains in consideration. There are three performance measures used in evaluating the element predictors—accuracy, reduced accuracy and GCC. Therefore, for each (unreduced) meta-predictor, there are three different ways in which the reduction can be done. They are named accuracy-guided reduction (or AG), relative accuracy-guided reduction (or RAG), and GCC-guided reduction (GG), respectively. In each of these reduction methods, the lowest scoring element predictors are excluded one by one, producing a series of reduced voting meta-predictors.

RESULTS

Predicting performance of element predictors

Table 2 summarizes the predicting performance of the 12 element predictors assessed on the MetaSCL06 dataset. The predictions made by the element predictors vary considerably with one another. Proteome Analyst (accuracy: 0.821, GCC: 0.811) offers the best performance among all predictors, followed by LOCtree (accuracy: 0.746, GCC: 0.663) and WoLF PSORT (accuracy: 0.733, GCC: 0.635). Some predictors from the ELSpred program (ELSpred_hybrid and ELSpred_dipeptide) are ranked among the lowest in predicting performance. Proteome Analyst is also the element predictor that offers the highest relative accuracy (0.913), followed by ELSpred_EuPSI (0.880) and LOCtree (0.766).

Unweighted voting strategy

With the performance of every element predictor assessed using the unbiased MetaSCL06 dataset, we set out to explore strategies to construct meta-predictors on top of these element predictors. First, we attempted a simple unweighted voting strategy. The meta-predictor constructed using the unweighted voting strategy (accuracy: 0.754, GCC: 0.651) offers better predicting performance than the average performance of the 12 element predictors (accuracy: 0.578, GCC: 0.459), but it does not reach the performance of the most accurate element predictor (Proteome Analyst, accuracy: 0.821, GCC: 0.811) (Table 3).

Table 2. Predicting performance of element predictors using the MetaSCL06 dataset

Element predictor	*N* predictions made	*N* correct predictions made	Accuracy	Relative accuracy	GCC	Weights for RAW-RAG-6
ELSpred_comp	1693	951	0.562	0.562	0.359	
ELSpred_physicochemical[a]	1693	1028	0.607	0.607	0.409	0.607
ELSpred_dipeptide	1693	750	0.443	0.443	0.215	
ELSpred_EuPSI[a]	624	549	0.324	0.880	0.458	0.880
ELSpred_hybrid	1693	659	0.389	0.389	0.179	
LOCtree[a]	1649	1263	0.746	0.766	0.663	0.766
PLOC	1692	1014.5[b]	0.599	0.600	0.465	
Proteome Analyst[a]	1523	1390	0.821	0.913	0.811	0.913
PSORT	1692	916	0.541	0.541	0.438	
PSORT II[a]	1687	1013	0.598	0.600	0.464	0.601
SubLoc	1687	973	0.575	0.577	0.409	
WoLF PSORT[a]	1687	1240.5[b]	0.733	0.735	0.635	0.735

Comparison based on the 1693 proteins in the MetaSCL06 dataset.
[a]The six element predictors used and, for each, the weight it was given in the high-scoring RAW-RAG-6 meta-predictor (see text).
[b]PLOC and WoLF PSORT output two most likely subcellular compartments with equal scores for some proteins. If one of them matches the true compartment label, the prediction is deemed 'half correct', counted as 0.5 correct prediction made.

Table 3. Predicting performance of unreduced voting meta-predictors as compared with that of element predictors

Predictor	Accuracy	GCC
Average of all element predictors	0.578	0.459
Best element predictor (Proteome Analyst)	0.821	0.811
UV-UR	0.754	0.651
AW-UR	0.808	0.724
RAW-UR	0.819	0.740
GW-UR	0.838	0.767

Comparison based on the 1693 proteins in the MetaSCL06 dataset. UV: unweighted voting; AW: accuracy weighted voting; RAW: relative accuracy weighted voting; GW: GCC weighted voting. UR stands for 'unreduced'.

Weighted voting strategy

Next, we examined a weighted voting strategy. We looked at three different weighting schemes, which correspond to the three measures we used when assessing the performance of the element predictors: (i) accuracy weighting (or AW), (ii) relative accuracy weighting (or RAW) and (iii) GCC weighting (or GW). In each of these weighting schemes, the value of the respective performance measure for any given element predictor is used as the weights of the element predictor.

As is shown in Table 3, improved performance is achieved in these weighted voting meta-predictors. All three weighted voting meta-predictors show accuracy values that approach or slightly exceed that of the most accurate element predictor, Proteome Analyst. However, using GCC, none of these meta-predictors have reached the level of Proteome Analyst (GCC: 0.811).

Reduced voting strategy

The four voting schemes (unweighted voting and three weighted voting schemes) are combined with the three reduction methods [accuracy-guided reduction (or AG), relative accuracy-guided reduction (or RAG) and GCC-guided reduction (GG)], giving rise to a total of 12 series of reduced voting meta-predictors. In each of these predictor series, the predicting performance (measured in accuracy or in GCC) shows a biphasic relationship with the number of excluded element predictors (Figure 2). When the number of excluded element predictors is small, the predicting performance increases with the number of excluded element predictors, agreeing well with our conjecture that excluding badly performed element predictors may lead to improved predicting performance of the meta-predictors. The predicting performance reaches a peak when about 6–9 element predictors are expelled, then declines as more element predictors are excluded. Apparently, following this critical point, further removing of the more accurate element predictors is detrimental to the predicting performance of the resultant meta-predictor.

As is shown in Table 4, the best predictor in each of the reduced voting meta-predictor series demonstrates better predicting performance than the best performed element predictor (Proteome Analyst) in both accuracy and GCC. Most of these best reduced meta-predictors show significantly higher GCC than that of Proteome Analyst in Fisher's Z-transformation test. The meta-predictor with the best performance was found to be the relative accuracy weighted, reduced by relative accuracy guiding, with six element predictors excluded (denoted as RAW-RAG-6). This meta-predictor makes predictions based on the predictions made by six element predictors: ELSpred_PhysicoChemical, ELSpred_EuPSI, LOCtree, Proteome Analyst, PSORT II and WoLF PSORT (Table 2). RAW-RAG-6 reaches a remarkable accuracy of 0.902, a nearly 8% improvement over Proteome Analyst (A: 0.821); and a GCC of 0.856, significantly higher than that of Proteome Analyst (GCC: 0.811), the best element predictor examined ($P = 8.2 \times 10^{-6}$, Fisher's Z-transformation test).

RAW-RAG-6 with data not used in its development

Element predictor performance was evaluated on data not used in their development. To impose this same limit on RAW-RAG-6, the element predictors and RAW-RAG-6 were evaluated using the MetaSCL07 dataset,

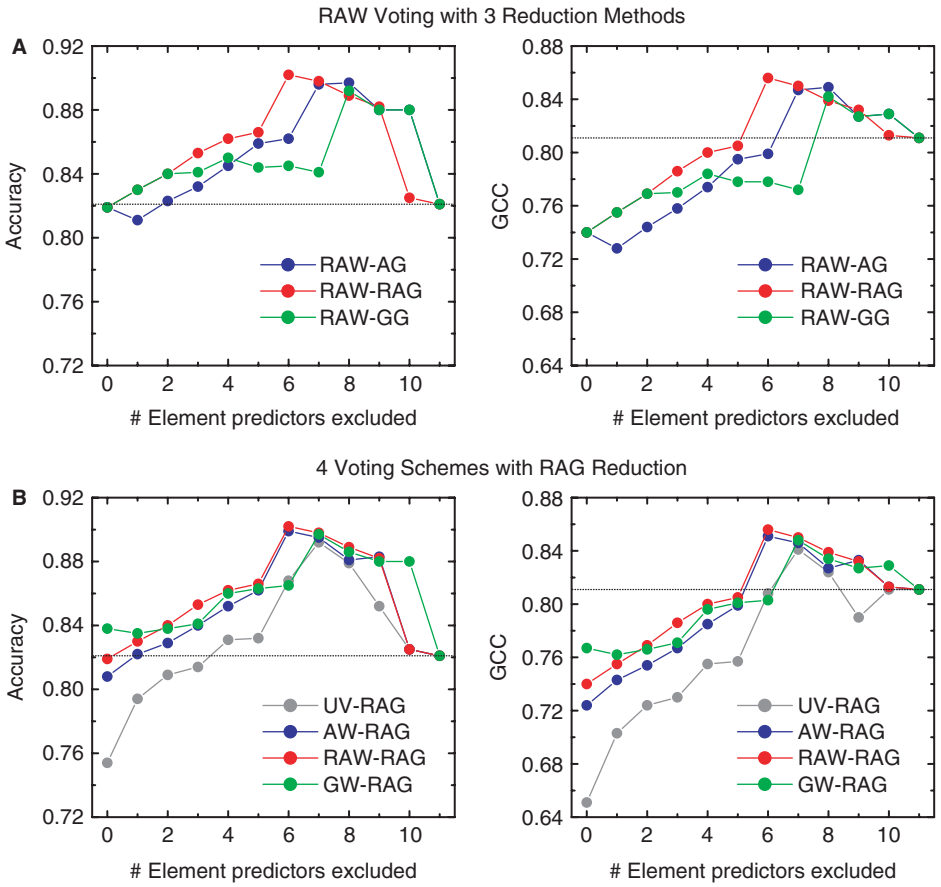

Figure 2. Performance of reduced voting meta-predictors (accuracy on the left, GCC on the right) plotted against the number of excluded element predictors. (**A**) Relative accuracy weighted voting (RAW) combined with three reduction methods—accuracy-guided reduction (AG), relative accuracy-guided reduction (RAG) and GCC-guided reduction (GG), giving rise to three series of meta-predictors. (**B**) Four voting schemes—unweighted voting (UV), accuracy-weighted voting (AW), relative accuracy-weighted voting (RAW) and GCC-weighted voting (GW) are combined with RAG reduction method, making four series of meta-predictors. All curves roughly show a biphasic characteristic—a rising phase followed by a decline phase. Dotted lines indicate performance of the best element predictor (Proteome Analyst, accuracy: 0.821, GCC: 0.811).

Table 4. Predicting performance of reduced voting meta-predictors

	AG	RAG	GG
Accuracy			
UV	0.892 (8)	0.892 (7)	0.863 (9)
AW	0.898 (7)	0.899 (6)	0.882 (8)
RAW	0.897 (8)	**0.902 (6)**	0.892 (8)
GW	0.897 (7)	0.899 (6)	0.883 (8)
GCC			
UV	0.841 (0.003)	0.841 (0.003)	0.816 (0.33)
AW	0.846 (0.00057)	0.851 (8×10^{-5})	0.829 (0.055)
RAW	0.849 (0.00018)	**0.856 (8.2×10^{-6})**	0.842 (0.0022)
GW	0.848 (0.00027)	0.852 (5.2×10^{-5})	0.830 (0.045)

Comparison based on the 1693 proteins in the MetaSCL06 dataset. Best reduced voting meta-predictor (RAW-RAG-6) is shown in bold. UV: unweighted voting; AW: accuracy weighted voting; RAW: Relative accuracy weighted voting; GW: GCC weighted voting; AG: accuracy guided reduction; RAG: relative guided reduction; GG: GCC-guided reduction. Number of excluded element predictors are shown after the accuracy values (enclosed in parentheses). *P*-values of GCC (Fisher's Z-transformation test) are shown after the GCC values (enclosed in parentheses).

containing data not used in RAW-RAG-6 development (Table 5). Proteome Analyst remains the element predictor with the best predicting performance based on GCC (0.783), though LOCtree offers better accuracy (0.829) than Proteome Analyst (0.775) with the MetaSCL07 dataset. The superior performance offered by RAW-RAG-6 persists with this dataset, with an accuracy of 0.888 and GCC of 0.840, significantly better than those of any element predictors.

RAW-RAG-6 in individual compartment predictions

The problem of protein subcellular localization is commonly formulated as a multi-class classification problem. However, it can also be viewed as several individual two-class classification problems, one for each subcellular compartment. This allows one to examine the ability of a given predictor to identify proteins localized in each of the compartments individually. The MetaSCL06 dataset was converted into four variant datasets, each one of which for examining one of the four subcellular compartments: nuclear, cytoplasmic, mitochondria and

Table 5. Predicting performance of element predictors and RAW-RAG-6 using the MetaSCL07 dataset

Element predictor	Accuracy	GCC
ELSpred_comp	0.230	0.145
ELSpred_physicochemical	0.370	0.081
ELSpred_dipeptide	0.434	0.274
ELSpred_EuPSI	0.252	0.424
ELSpred_hybrid	0.332	0.136
LOCtree	0.829	0.757
PLOC	0.446	0.324
Proteome Analyst	0.775	0.783
PSORT	0.494	0.427
PSORT II	0.459	0.348
SubLoc	0.451	0.230
WoLF PSORT	0.654	0.565
RAW-RAG-6	**0.888**	**0.840 ($P = 0.0022$)**

Comparison based on the 579 proteins in the MetaSCL07 dataset. *P*-values of GCC (Fisher's Z-transformation test) are shown after the GCC value of RAW-RAG-6 (enclosed in parentheses).

extracellular, respectively. In the variant dataset for examining nuclear proteins, for instance, all proteins labeled as 'nuclear' were considered as 'positive samples', and all proteins labeled with any of the other compartments (cytoplasmic, mitochondria and extracellular) were lumped together and considered as 'negative' samples. We evaluated the predicting performance of each of the 12 element predictors, as well as that of RAW-RAG-6 meta-predictor, using these variant datasets.

RAW-RAG-6 outperforms each of the 12 element predictors in accuracy and MCC for all four two-class classification problems (Table 6). Comparing with the element predictor with the best performance (Proteome Analyst), the biggest improvement was achieved for the extracellular compartment, with 2.7% increase in accuracy (from 0.956 to 0.983) and 5.7% increase in MCC (from 0.909 to 0.966, $p < 2 \times 10^{-16}$, Fisher's Z-transformation test). It is followed by the nuclear compartment, for which a 2.1% increase in accuracy (from 0.908 to 0.929) and 4.6% increase in MCC (from 0.801 to 0.847, $P = 1.4 \times 10^{-5}$, Fisher's Z-transformation test) are achieved. The smallest improvement is found for the cytoplasmic compartment, where 0.7% increase in accuracy (from 0.922 to 0.929), and 1.3% improvement in MCC (from 0.617 to 0.630, $P = 0.27$, Fisher's Z-transformation test) are observed. Overall, the RAW-RAG-6 meta-predictor achieves remarkable performance in these two-class classification problems, and consistently outperforms every element predictor in identifying proteins localized in each of the four subcellular compartments.

DISCUSSION

Meta-predictors may resolve conflicting predictions

In many life science domains, several prediction programs have emerged that often have different strengths due to different types of data (or different aspects of the same data) used, and/or different classification methods adopted in their development. When more than one of these is used on the same data, they may produce

Table 6. Predicting performance of element predictors and RAW-RAG-6 in two-class predictions for the 4 subcellular compartments

Predictor	Sensitivity	Specificity	Accuracy	MCC
Nuclear				
ELSpred_comp	0.776	0.620	0.676	0.380
ELSpred_physicochemical	0.824	0.681	0.732	0.484
ELSpred_dipeptide	0.901	0.369	0.560	0.291
ELSpred_EuPSI	0.545	0.977	0.822	0.615
ELSpred_hybrid	0.572	0.444	0.490	0.015
LOCtree	0.728	0.935	0.861	0.692
PLOC	0.890	0.622	0.718	0.495
Proteome Analyst	0.781	0.980	0.908	0.801
PSORT	0.611	0.944	0.825	0.611
PSORT II	0.774	0.785	0.781	0.544
SubLoc	0.799	0.757	0.772	0.537
WoLF PSORT	0.768	0.931	0.872	0.719
RAW-RAG-6	0.923	0.932	0.929	**0.847 ($P = 1.4 \times 10^{-5}$)**
Cytoplasmic				
ELSpred_comp	0.515	0.916	0.875	0.391
ELSpred_physicochemical	0.075	0.932	0.845	0.009
ELSpred_dipeptide	0.243	0.897	0.831	0.132
ELSpred_EuPSI	0.434	0.972	0.917	0.485
ELSpred_hybrid	0.705	0.753	0.748	0.305
LOCtree	0.607	0.886	0.857	0.402
PLOC	0.523	0.950	0.906	0.480
Proteome Analyst	0.728	0.944	0.922	0.617
PSORT	0.347	0.835	0.785	0.142
PSORT II	0.538	0.842	0.811	0.289
SubLoc	0.561	0.840	0.811	0.302
WoLF PSORT	0.517	0.921	0.880	0.404
RAW-RAG-6	0.699	0.955	0.929	**0.630 ($P = 0.27$)**
Mitochondrial				
ELSpred_comp	0.189	0.971	0.868	0.247
ELSpred_physicochemical	0.284	0.971	0.881	0.357
ELSpred_dipeptide	0.099	0.969	0.855	0.117
ELSpred_EuPSI	0.144	0.997	0.885	0.331
ELSpred_hybrid	0.225	0.975	0.877	0.306
LOCtree	0.761	0.950	0.926	0.686
PLOC	0.288	0.967	0.878	0.346
Proteome Analyst	0.730	1.000	0.965	0.837
PSORT	0.306	0.963	0.877	0.352
PSORT II	0.243	0.967	0.872	0.296
SubLoc	0.329	0.936	0.857	0.300
WoLF PSORT	0.369	0.970	0.892	0.438
RAW-RAG-6	0.793	0.996	0.969	**0.859 ($P = 0.011$)**
Extracellular				
ELSpred_comp	0.505	0.841	0.704	0.371
ELSpred_physicochemical	0.654	0.826	0.756	0.489
ELSpred_dipeptide	0.201	0.944	0.641	0.224
ELSpred_EuPSI	0.161	0.996	0.655	0.306
ELSpred_hybrid	0.203	0.982	0.664	0.312
LOCtree	0.792	0.968	0.896	0.787
PLOC	0.462	0.979	0.768	0.540
Proteome Analyst	0.909	0.988	0.956	0.909
PSORT	0.604	0.986	0.830	0.665
PSORT II	0.573	0.991	0.820	0.650
SubLoc	0.460	0.888	0.714	0.393
WoLF PSORT	0.873	0.957	0.923	0.840
RAW-RAG-6	0.970	0.993	0.983	**0.966 ($P < 2 \times 10^{-16}$)**

Comparison made based on datasets derived from the MetaSCL06 dataset. *P*-values of MCC (Fisher's Z-transformation test) are shown together with the MCC values (enclosed in parentheses).

conflicting predictions. Users are often confused and frustrated by such conflicting results, because they may lack the knowledge to make a sensible choice among them. If a meta-predictor can be developed with predicting performance exceeding that of any individual element predictors, it may resolve this quandary.

Meta-predictors versus element predictors

Meta-predictors cannot replace element predictors. Rather, they are enhancements. Meta-predictors are constructed from element predictors, and their performance depends on accurate predictions made by element predictors. Without good element predictors, it is not possible for good meta-predictors to be developed. In addition, meta-predictors (in particular, voting-based meta-predictors) are effective only within the scope of the prediction problem that is common to multiple element predictors. Often, element predictors make unique predictions. For example, among the prediction programs discussed in this study, only PSORT II makes predictions about protein localization to secretory vesicles. For unique predictions, one has to rely on an element predictor.

Cross-validation and future performance

We did not perform cross-validation explicitly in this development. However, because all parameters of RAW-RAG-6 (relative accuracy values of element predictors) are calculated as sample statistics, and the latter are insensitive to removal of a small number of samples given that the sample size is sufficiently large, the testing we performed can be considered as being equivalent to cross-validation. To demonstrate, suppose we perform an 'explicit' LOO (leave-one-out) cross-validation, i.e. taking 1692 samples as the 'training dataset', and the remaining sample as the validation data, and do so for 1693 iterations so that each sample is validated once. In each iteration, the parameters of the meta-predictor—which are relative accuracy values of the element predictors—calculated based on the 1692 training samples, would be essentially the same as the relative accuracy values of the whole 1693-sample dataset, because the relative accuracy of each element predictor is a sample statistic, which is insensitive to the removal of one sample from the dataset, given that the sample size is sufficiently large. Therefore, each predictor achieved from the 1693 iterations of LOO cross-validation would be the same as the predictor achieved from the entire 1693-sample dataset.

The validation performed using the MetaSCL07 dataset suggests that the RAW-RAG-6 meta-predictor is robust. Its performance for future, unseen data is expected to be close to what was achieved in this study, assuming no changes are made to the element predictors. However, if changes take place in any of its component element predictors, the reduced voting-based meta-predictor will need adjustment.

Linear voting strategies

The linear voting strategies explored in this study are related to several well-known online learning algorithms, including Littlestone and Warmuth's weighted majority (WM) algorithm (21) and Freund and Schapire's Hedge algorithm (22). Those algorithms are applied to situations where one person is trying to make predictions based on the opinions of several 'experts' from whom he seeks advice. If the weights are properly chosen, there are theoretical bounds of the maximal number of wrong predictions made by the 'master predictor', i.e. the performance of the 'master predictor' will not be 'too much worse' than that of the predictions made by the best 'expert'. The meta-prediction problem discussed in this study differs from those previous studies in that 'batch learning', rather than 'online learning', applies. In other words, the training samples are assumed to be provided together, instead of one at a time. In the same paper in which the Hedge algorithm was discussed (22), Freund and Schapire introduced the well-known Adaboost algorithm, which applies to batch learning. The major difference between meta-prediction problem discussed in this study and Adaboost and other ensemble learning algorithms [e.g. Logitboost (23) and Bagging (24)] is that in ensemble learning, the 'element predictors' are results of an identical training algorithm applied to different samplings of the training data. In meta-prediction, the 'element predictors' are assumed to be known and unchanged, and all training data is used for all element predictors.

CONCLUSIONS

The successful development of RAW-RAG-6 demonstrates the effectiveness of voting-base strategies in the meta-prediction problems. Proper employment of voting-based strategies is likely to lead to good meta-predictors in other life sciences problem domains.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank Drs A. Banerjee and W. Pan at the University of Minnesota for inspiring discussions. We also thank the Supercomputing Institute, University of Minnesota for computational resources, and W. Gong for technical assistance. T.L. acknowledges the support of NIH (1R21CA126209) and Minnesota Medical Foundation. Funding to pay the Open Access publication charges for the article was provided by NIH/NCI.

Conflict of interest statement. None declared.

REFERENCES

1. Nakai,K. and Kanehisa,M. (1992) A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics*, **14**, 897–911.
2. Rost,B., Liu,J., Nair,R., Wrzeszczynski,K.O. and Ofran,Y. (2003) Automatic prediction of protein function. *Cell. Mol. Life Sci.*, **60**, 2637–2650.

3. Donnes,P. and Hoglund,A. (2004) Predicting protein subcellular localization: past, present, and future. *Genomics Proteomics Bioinformatics*, **2**, 209–215.
4. Emanuelsson,O., Nielsen,H., Brunak,S. and von Heijne,G. (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.*, **300**, 1005–1016.
5. Nakai,K. and Horton,P. (1999) PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem. Sci.*, **24**, 34–36.
6. Bannai,H., Tamada,Y., Maruyama,O., Nakai,K. and Miyano,S. (2002) Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics*, **18**, 298–305.
7. Reinhardt,A. and Hubbard,T. (1998) Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res.*, **26**, 2230–2236.
8. Park,K.J. and Kanehisa,M. (2003) Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, **19**, 1656–1663.
9. Huang,Y. and Li,Y. (2004) Prediction of protein subcellular locations using fuzzy k-NN method. *Bioinformatics*, **20**, 21–28.
10. Hua,S. and Sun,Z. (2001) Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, **17**, 721–728.
11. Lu,Z., Szafron,D., Greiner,R., Lu,P., Wishart,D.S., Poulin,B., Anvik,J., Macdonell,C. and Eisner,R. (2004) Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, **20**, 547–556.
12. Xie,D., Li,A., Wang,M., Fan,Z. and Feng,H. (2005) LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. *Nucleic Acids Res.*, **33**, 110.
13. Bhasin,M. and Raghava,G.P. (2004) ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Res.*, **32**, 419.
14. Nair,R. and Rost,B. (2005) Mimicking cellular sorting improves prediction of subcellular localization. *J. Mol. Biol.*, **348**, 85–100.
15. Nair,R. and Rost,B. (2003) Better prediction of sub-cellular localization by combining evolutionary and structural information. *Proteins*, **53**, 917–930.
16. Drawid,A. and Gerstein,M. (2000) A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome. *J. Mol. Biol.*, **301**, 1059–1075.
17. Klee,E.W. and Ellis,L.B. (2005) Evaluating eukaryotic secreted protein prediction. *BMC Bioinformatics*, **6**, 256.
18. Horton,P., Park,K.-J., Obayashi,T. and Nakai,K. (2006) In *The 4th Annual Asia Pacific Bioinformatics Conference APBC06*. Taipei, Taiwan, pp. 39–48.
19. Matthews,B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.
20. Gorodkin,J. (2004) Comparing two K-category assignments by a K-category correlation coefficient. *Comput. Biol. Chem.*, **28**, 367–374.
21. Littlestone,N. and Warmuth,M.K. (1994) The weighted majority algorithm. *Information and Computation*, **108**, 212–261.
22. Freund,Y. and Schapire,R.E. (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J, Comput. System Sci.*, **55**, 119–139.
23. Friedman,J., Hastie,T. and Tibshirani,R. (2000) Additive logistic regression: a statistical view of boosting. *Ann. Stat.*, **28**, 337–374.
24. Breiman,L. (1996) Bagging predictors. *Machine Learning*, **24**, 123–140.