REACTION KERNELS
Structured Output Prediction Approaches for Novel Enzyme Function

Katja Astikainen, Esa Pitkänen, Juho Rousu
Department of Computer Science, University of Helsinki, PO Box 68, Helsinki, Finland

Liisa Holm
Institute of Biotechnology, University of Helsinki, PO Box 56, Helsinki, Finland

Sándor Szedmák
Electronics and Computer Science, University of Southampton, SO17 1BJ, Southampton, U. K.

Keywords: Bioinformatics, Machine learning, Kernel methods, Enzyme function prediction.

Abstract: Enzyme function prediction problem is usually solved using annotation transfer methods. These methods are suitable in cases where the function of the new protein is previously characterized and included in the taxonomy such as EC hierarchy. However, given a new function that is not previously described, these approaches arguably do not offer adequate support for the human expert.
In this paper, we explore a structured output learning approach, where enzyme function—an enzymatic reaction—is described in fine-grained fashion with so called reaction kernels which allow interpolation and extrapolation in the output (reaction) space. Two structured output models are learned via Kernel Density Estimation and Maximum Margin Regression to predict enzymatic reactions from sequence motifs. We bring forward two choices for constructing reaction kernels and experiment with them in the remote homology case where the functions in the test set have not been seen in the training phase. Our experiments demonstrate the viability of our approach.

1 INTRODUCTION

Enzymes are the workhorses of living cells, producing energy and building blocks for cell growth as well as participating in maintaining and regulation of the metabolic states of the cells. Reliable assignment of enzyme function, that is, the biochemical reactions catalyzed by the enzymes, is a prerequisite of high-quality metabolic reconstruction (Palsson, 2006).

In literature, the enzyme function prediction problem comes in two general formulations: annotation transfer or classification by machine learning. In the first approach, given an unannotated protein, a similar annotated protein with experimentally verified function is searched for in databases, and the annotation is transferred to the new protein. In the second approach, a model is trained to classify the new protein into one of the predefined functional classes such as four-level hierarchical EC classification of enzymatic functions.

The success of the above approaches depends on the set of previously characterized and catalogued enzymatic functions. If the new protein belongs to the existing function classes, annotation transfer or classification learning may work. If the new protein, however, posseses a function that is not pre-existing, correct function cannot be predicted even in principle.

Given the diversity of the tree of life, it is likely that completely new functions are encountered as sequencing and annotation efforts widen. Tools, which can give accurate predictions of what the new functions might be, could expedite these efforts. In this paper, we develop a structured output prediction approach that, to our knowledge is the first enzyme function prediction tool to possess the capability of prediction previously unseen functions. The key component of our method is the representation of enzyme function in fine-grained fashion with the so called reaction

Astikainen K., Pitkänen E., Rousu J., Holm L. and Szedmák S. (2010).
REACTION KERNELS - Structured Output Prediction Approaches for Novel Enzyme Function.
In *Proceedings of the First International Conference on Bioinformatics*, pages 48-55
DOI: 10.5220/0002741700480055

kernels, that allow interpolation and extrapolation in the space of enzymatic function.

The organization of the paper is the following. In section 2 we briefly describe main existing approaches in enzyme function prediction. In section 3, we review structured output prediction approaches, in particular Kernel Density Estimation and Maximum Margin Regression which are applied in the subsequent sections. In section 4, we describe representations for structured output prediction of enzyme function. We put forward two reaction kernel variants that allow us to interpolate and extrapolate in the space of enzymatic reactions. Section 5 describes experiments validating our approach. Section 6 discusses the relative merits of the current and competing methods, and outlines directions for future work.

2 ENZYME FUNCTION PREDICTION

Protein function prediction is recognized as one of the key problems in bioinformatics, and hence there is a large number of approaches to tackle this problem. Most enzyme function prediction methods are instantiations of the more general protein function prediction problem. Here we give a brief overview of protein function prediction approaches. For more information, we refer the interested reader to the recent survey of (Punta and Ofran, 2008).

2.1 Annotation Transfer Approaches

The most widely used function prediction approach is still annotation transfer based on sequence similarity: given an unannotated protein, using a sequence comparison tool such as BLAST, search for an annotated sequence homolog with an experimentally verified function, and transfer the annotation to the new protein. This approach has well-known pitfalls: sequence similarity does not equate to homology, function is typically determined by a small group of residues whose contribution in the overall similarity may fail to be detected, and the danger is the propagation of the annotation errors.

Sequence motifs or signatures are used to overcome shortcomings of overall sequence similarity. As the protein function is typically dependent on a small region of the sequence (e.g. for enzymes the residues forming the active center), a significant amount of research has been conducted to derive sequence motifs that are predictive of the function (Henikoff and Henikoff, 1996; Falquet et al., 2002; Mulder et al., 2002). In this paper, we apply the Global Trace Graph (Heger et al., 2007) features that can be interpreted as predicted conserved residues. The GTG features are derived from a global alignment of all known protein sequences. In this alignment, GTG features correspond to residues that align consistently within a group of proteins.

Information about the 3D structure is known to be a powerful aid in function prediction, due to the fact that it is ultimately the three-dimensional structure that determines the protein function. Structural similarity of two proteins may indicate common evolutionary origin even in the absence of significant sequence similarity. Numerous structural alignment methods (e.g. (Krissinel and Henrick, 2004; Ye and Godzik, 2004; Holm and Sander, 1996)) have been developed to make use of the 3D structures. Structural motifs are an analogous concept to sequence motifs: a local constellation of residues in the active center of an enzyme may be higly predictive of the function. In this paper, we do not apply 3D information, but leave this as future work.

2.2 Machine Learning Approaches

Machine learning methods are potentially useful in cases where the new protein does not possess significant sequence (or structure) similarity to existing proteins. Given large enough data, machine learning methods are able to distill non-trivial associations between the input features and the function.

In the machine learning setting, enzyme function prediction has been generally defined as a classification problem. The works by Lanckriet et al. (Lanckriet et al., 2004) and Borgwardt et al. (Borgwardt et al., 2005) use the kernel method to predict the main categories in MIPS and EC taxonomies, respectively.

Other works aim to predict the membership in the whole taxonomy. These include the work by Clare and King (Clare and King, 2002) who use decision trees to predict the membership in the MIPS taxonomy. Barutcuoglu et al. (Barutcuoglu et al., 2006) combine Bayesian networks with a hierarchy of support vector machines to predict Gene Ontology classification. Blockeel et al. (Blockeel et al., 2006) use multilabel decision tree approaches to functional class classification according to the MIPS FunCat taxonomy.

Structured output approaches (see below) for hierarchical multilabel classification (c.f. (Rousu et al., 2006)) have been applied to enzyme function prediction by Astikainen et al. (Astikainen et al., 2008) and Sokolov and Ben-hur (Sokolov and Ben-Hur, 2008). In this paper, we take the hierarchical classification against the EC hierarchy (Astikainen et al., 2008) as

one of the comparison methods to the reaction kernel approach.

3 STRUCTURED OUTPUT LEARNING

Our objective is to learn a function that, given (a feature representation of) a sequence, can predict (a feature representation of) an enzymatic reaction.

Learning algorithms that are designed for structured prediction tasks like the above, are many. We concentrate on kernel methods, that let us utilize high-dimensional feature spaces without computing the feature maps explicitly. Structured SVM (Tsochantaridis et al., 2004), Max-Margin-Markov networks (Taskar et al., 2004; Rousu et al., 2006), Kernel Density Estimation (KDE) and Maximum-Margin Regression (MMR) (Szedmak et al., 2005) are learning methods falling into this category.

We consider a training set of (sequence,reaction)-pairs $D^m = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^m$ drawn from an unknown joint distribution $\mathcal{P}(\mathcal{X}, \mathcal{Y})$.

For sequences and reactions, respectively, we assume feature mappings ϕ: $\mathcal{X} \mapsto \mathcal{F}_\mathcal{X}$ and ψ: $\mathcal{Y} \mapsto \mathcal{F}_\mathcal{Y}$, mapping the input and output objects into associated inner product spaces $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$. The kernels $K_X(x,x') = \langle \phi(x), \phi(x') \rangle$ and $K_Y(y,y') = \langle \psi(y), \psi(y') \rangle$ defined by the feature maps are called the input and output kernel, respectively. Above $\langle \cdot \rangle$ denotes the inner product. Subsequently, we discuss particular choices for the feature mappings and the kernels suitable for the enzyme function prediction task.

3.1 Joint Kernels

In structured prediction models based on kernels, the associations between the inputs and outputs are typically represented by a *joint* kernel, defined by some feature map joint for inputs and outputs. In this paper, we use a joint feature map

$$\varphi(x,y)\colon \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{F}_{\mathcal{X}\otimes\mathcal{Y}},$$

where $\varphi(x,y) = \phi(x) \otimes \psi(y)$ is the tensor product of input and output feature maps, thus consisting of all pairwise products $\phi_j(x)\psi_k(y)$ between input and output features. This choice gives us the joint kernel representation as elementwise product of the input and output kernels

$$K_{XY}(x,y;x',y') = K_X(x,x')K_Y(y,y').$$

The tensor product kernel is suitable in situations where there is no prior alignment information of input and output features available, but the learning machine is expected to learn the alignments. This is the case in our enzyme function prediction setup.

3.2 Learning Task

Most structured prediction models (Taskar et al., 2004; Tsochantaridis et al., 2004; Szedmak et al., 2005; Rousu et al., 2006) take the form of a linear score function

$$F_w(x,y) = \langle w, \varphi(x,y) \rangle = \langle w, \phi(x) \otimes \psi(y) \rangle$$

in the joint feature space. The model's prediction $\hat{y}(x)$ corresponds to highest scoring output y:

$$\hat{y}(x) = \mathbf{argmax}_y F_w(x,y).$$

For the model learning we use two computational methods. The first method is Kernel Density Estimation (KDE) which uses the joint kernel density function

$$F_w(x,y) = \sum_i K_{XY}(x,y;x_i,y_i) \tag{1}$$

for scoring. This is the simplest model we use for prediction, since there is no weighting vector w for the training examples and all the datapoints are thus equally important.

The second method, Max-Margin Regression (MMR) (Szedmak et al., 2005) aims to separate the training data $\varphi(x_i,y_i)$ from the origin of the joint feature space with maximum margin, thus it can be seen analogous to the one-class SVM (Schlkopf et al., 2001). The primal form of the MMR optimization problem can be written as

$$\begin{aligned} \min \quad & \frac{1}{2}||w||^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & \langle w, \varphi(x_i,y_i) \rangle \geq 1 - \xi_i \\ & \xi_i \geq 0,\ i = 1,\ldots,m. \end{aligned}$$

The dual form of the MMR problem can be expressed as

$$\begin{aligned} \max \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j K_X(x_i,x_j)K_Y(y_i,y_j) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C,\ i = 1,\ldots,m. \end{aligned} \tag{2}$$

MMR, due to its simple form, can be optimized very efficiently which makes, for example, the optimization of kernel parameters a feasible task on medium sized datasets (10^3-10^4 examples), which is not true for most competing approaches (Taskar et al., 2004; Tsochantaridis et al., 2004; Rousu et al., 2006).

Furthermore, as the output representation is kernelized, it is possible to learn in very complex output spaces, as we will demonstrate subsequently.

3.3 Preimage Problem

In all structured output prediction approaches, the prediction of the model needs to be extracted by solving the preimage problem

$$\hat{y}(x) = \mathbf{argmax}_{y \in \mathcal{Y}} F_w(x,y).$$

Depending on the output space, solving the preimage exactly can be computationally challenging or intractable.

Using kernelized outputs, as in the case of dual MMR (2), the preimage takes an even more challenging form

$$\hat{y}(x) = \mathbf{argmax}_{y \in \mathcal{Y}} \sum_i \alpha_i K_X(x,x_i) K_Y(y,y_i),$$

for which efficient algorithms are hard to come by. However, a difference between MMR and most structured output prediction methods is that there is no need to solve the preimage problem as part of the training, only during prediction. Thus, the computational complexity of the preimage is not as a crucial issue.

In the experiments reported in this paper, we use a trivial preimage algorithm: we enumerate the set of outputs contained in our whole dataset (training and test examples included) $\mathcal{Y}^n = \{y | (x,y) \in D^n\}$. This approach will give us an approximate solution to the preimage problem, that is, the globally best scoring prediction may lie outside the set $\mathcal{Y}^n$. This approach is sufficient for first evaluation of the proposed prediction methods. We leave the development of better preimage algorithms as future work.

4 KERNELS FOR CHEMICAL REACTIONS

In this section, we consider how to build kernels for chemical reactions, using molecule graph kernels as the building blocks.

Let us first introduce some notation used in this section. We denote a basic set of reactions $\mathcal{R}$, where a reaction $\rho(S(\rho), P(\rho)) \in \mathcal{R}$ is given by a set substrates $S(\rho) \subset \mathcal{M}$ and products $P(\rho) \subset \mathcal{M}$[1]. The set of reactants is simply the union of substrates and products $R(\rho) = S(\rho) \cup P(\rho)$. A feature vector describing a reaction ρ is denoted by $\psi(\rho)$ and the feature vector describing a molecule M is denoted by $\phi(M)$.

[1]To fully represent chemical reaction equations, we would also need to consider the stoichiometric coefficients for each reactant; However, we ignore this modelling aspect here

For illustration, consider a chemical reaction $\rho = (\{S_1\}, \{P_1, P_2\})$ converting a substrate molecule S_1 into two product molecules P_1 and P_2, thus defined by the reaction equation

$$\rho\colon S_1 \Rightarrow P_1 + P_2.$$

Consider now a second reaction $\rho' = (\{S'_1, S'_2\}, \{P'_1, P'_2\})$, converting substrates S'_1, S'_2 into products $P'_1 + P'_2$, and back, expressed as

$$\rho'\colon S'_1 + S'_2 \Leftrightarrow P'_1 + P'_2$$

How can we measure the similarity of these reactions via kernels? The approach in this paper is to consider pairwise similarities of the constituent molecules and compute an aggregate on them. While there are many ways how this could be done in principle, two important considerations arise from the (bio)chemical reality:

- **Similarity of Reaction Events vs. Reactants.** We should make a distinction between the similarity of the reaction events versus the similarity of the reactant molecules. For example, enzymes belonging to the amino-transferase group are similar to each other in that they transfer a certain functional group (the amino group) from a reactant molecule to another. However, the reactant molecules need not be similar.

  Conversely, there are many different transformations which can be performed on the same molecule. For example, pyruvate, an important hub metabolite in the central metabolism of all living cells, participates in many reactions. The transformations applied by the reactions may be very different from each other, although they work on the same substrate molecule pyruvate.

  Thus, depending on the application, our kernel should be designed to measure one of these similarity notions, or measure both of them in some proportion.

- **Directionality of Reactions.** The reactions may be defined as unidirectional or bidirectional. As the direction of a reaction depends on thermodynamical conditions, this may or may not be a relevant issue. For example, most enzymatic reactions are bidirectional in principle, but the conditions inside a living cell force unidirectionality.

  When the directionality of reactions is of importance, each bidirectional reaction can be divided into forward and backward reactions. In our example, we would obtain

  $$\rho'_{fwd}\colon S'_1 + S'_2 \Rightarrow P'_1 + P'_2,$$

  and

  $$\rho'_{bwd}\colon P'_1 + P'_2 \Rightarrow S'_1 + S'_2.$$

In this case we would like our kernel to be sensitive to the direction so that forward and backward directions of the same reaction can be discriminated in the feature space.

However, when reaction direction is of no importance, the forward and backward directions of a bidirectional reaction should be treated the same by our kernel.

Below, we will describe a molecule graph kernel matrix $K_{\mathcal{M}}$ which constitutes the basic component of the two alternative reaction kernels described next. For both reaction kernels we also show the underlying feature map which will suffice to show that both of the reaction kernels below are valid Mercer kernels if the underlying molecule kernel is a valid Mercer kernel.

Both of the reaction kernels described below are very fast to compute, given that the molecule kernel $K_{\mathcal{M}}$ is pre-computed: the time complexity of the reaction kernel computation is then linear in the number of the elements in the kernel matrix.

4.1 Kernels for Molecule Graphs

As the molecule kernel $K_{\mathcal{M}}$ underlying the reaction kernels we use a subgraph kernel restricted to small subgraphs (10 nodes or less). The kernel computes the product graph of the two molecule graphs and counts its connected subgraphs. The kernel constructed in this way may in general not be a valid Mercer kernel. However, on our dataset, the kernel matrix was observed to be positive semidefinite.

Enumerating the subgraphs up to the maximum subgraph size d takes $O(m^d)$ time, where m is the number of edges in the product graph. Thus the kernel is quite time-consuming to compute. In practice, we were able to compute the common connected subgraphs of 1767 KEGG LIGAND (Goto et al., 2002) molecules up to subgraph size 10 in a week with approximately 50 Pentium 4 class computers. Considering the computational resources available nowadays in research labs, and the time available to solve a typical problem involving molecular data, the computational complexity hardly presents a prohibitive constraint.

We note that it would also be possible to use a more quickly computable graph kernel based on common walks (Gartner, 2003), that is, sequences of labeled atoms and bonds, which can be thought to approximate common subgraphs (each common subgraph induces a set of common walks). However, we leave exploring this direction as future work.

4.2 Sum-of-Reactants Kernel

A simple kernel, called the Sum-of-Reactants (SoR) kernel, is obtained by defining

$$K_{SoR}(\rho,\rho') = m(\rho)^T K_{\mathcal{M}} m(\rho'),$$

where the vector $m(\rho)$ consists of indicators $m_j(\rho) = \mathbf{1}_{\{M_j \in R_\rho\}}$ for the presence or absence of a molecule M_j in the set of reactants of ρ. The corresponding feature vector is simply the sum of feature vectors of molecule graphs in R_ρ:

$$\psi(\rho) = \sum_{M \in R_\rho} \phi(M)$$

Intuitively, the kernel measures the similarity of reactions in terms of how similar the molecules manipulated by the reactions are on average, rather than the similarity of reaction events. The reaction representation and the kernel can be considered bidirectional as the different roles of reactant molecules are not considered.

4.3 Reactant-Matching Kernels

In the SoR kernel there is an underlying all-against-all matching between the substrate sets (S_ρ, S'_ρ), product sets (P_ρ, P'_ρ) and between the cross-pairs (S_ρ, P'_ρ) and (P_ρ, S'_ρ). This measure implicitly contains spurious matches where one substrate $s_1 \in S_\rho$ is matched against a substrate $s' \in S'_\rho$ while another $s_2 \in S_\rho$ is matched against a product $p' \in P_\rho$. Considering such matches has no biological significance. We can filter out the above spurious mactches by defining a feature map via the tensor product

$$\psi(\rho) = \sum_{M \in S_\rho} \phi(M) \otimes \sum_{M \in P_\rho} \phi(M),$$

which gives us the Reactant-Matching (RM) kernel

$$K(\rho,\rho') = K(S_\rho, S_{\rho'}) K(P_\rho, P_{\rho'}),$$

where we use the shorthand

$$K(S,S') = \sum_{M \in S} \sum_{M' \in S'} K_{\mathcal{M}}(M,M').$$

The above kernel is obviously unidirectional as it matches the reactions in the forward direction. To obtain a bidirectional kernel we compute the backward direction by taking the cross terms

$$K(\rho,\rho') = \frac{1}{2}(K(S_\rho, S_{\rho'}) K(P_\rho, P_{\rho'}) + K(S_\rho, P_{\rho'}) K(P_\rho, S_{\rho'}))$$

We note that the bidirectional kernel still filters out the above mentioned spurious matches, in the second term the other reaction is just flipped around.

5 EXPERIMENTS

5.1 Data

The dataset is a sample(sequence, reaction) pairs from the KEGG LIGAND database (Goto et al., 2002). As the input (sequence) representation, we use Global Trace Graph (GTG, (Heger et al., 2007)) features that can be interpreted as predicted conserved amino acids.

We have two separate datasets: the parameter validation set of 1481 enzymes and testing set of 8112 enzymes, which do not have overlapping EC numbers. Parameter validation set is yet divided into two folds, training set of 930 and test set of 551 enzymes. Testing dataset is divided into five folds with average of 1622 enzymes. Members of the folds are chosen such that each of the different EC number exist only in one of the folds, so the training sets have no enzymes with the test set EC number appear. This is to simulate setting where a previously unseen functions are to be predicted.

Both the input (GTG) kernel and the output (reaction) kernels are fed to a polynomial kernel $K_{poly}(x,z) = (K(x,z)+1)^d$ and normalized. The restricted size subgraph kernel is used as the molecule kernel underlying all the reaction kernel variants.

5.2 Compared Methods

We compare the following methods:

- NN(BLAST): This is the baseline annotation transfer method: given a test sequence, find the nearest sequence neighbor in the training set and transfer the annotation to the new protein. Sequence similarity is taken from pre-computed Blast scores from the Pairs-DB server (Heger et al., 2008).
- NN(GTG): This is the annotation transfer methods using the GTG data. Given a test sequence, find the training sequence with the most common GTG features with the test sequence, and transfer the annotation.
- MMR(GTG,Hierarchical): The hierarchical structured output prediction from (Astikainen et al., 2008). The method predicts the membership of the new protein in the EC hierarchy; generally the prediction is a root-to-leaf path in the EC hierarchy.
- MMR(GTG,RM): MMR with GTG as input kernel and Reactant-Matching as output kernel.
- KDE(GTG,RM): KDE with GTG as input kernel and Reactant-Matching as output kernel.

We have beforehand made an experiment where we compared the function prediction accuracy with both of the reaction kernels using degree-6 polynomial kernel over the inputs and degree-20 polynomial kernel over the outputs. F1 score for RM was 27.9% and for SoR it was 25.9%. Since the RM outperformed SoR, we use the RM as output kernel in all the following experiments.

5.3 Measure of Success

To measure accuracy of prediction, for each test instance (x,y), we first compute the set of top-scoring functions $\hat{\mathcal{Y}}(x) = \{y_i \in \mathcal{Y}^n | F(\alpha,x,y_i) \geq F(\alpha,x,y'), \forall y' \in \mathcal{Y}^n\}$, that is the reactions that the prediction model considers the (equally) best. This set is considered as the prediction of the model.

For each function $y' \in \hat{\mathcal{Y}}(x)$, we check how many consecutive digits starting from the left of the EC number associated with y' coincide with digits of the EC number associated with the reference function y. Each such correctly predicted EC digits counts as a true positive, rest of the EC digits counts as a false positive. For example, if the reference function y is 3.1.1.1 and prediction set $\hat{\mathcal{Y}}(x)$ contains two members **3.1**.2.1 and **3.1.1**.10, there are five true positives (marked bold) and three false positives out of 8 EC digits. The EC digit F1 is then the F1 score taken over all EC digit predictions in the test set.

5.4 Results

5.4.1 Effect of Polynomial Kernel Degree

In the first experiment we illustrate the behaviour of the structured output learning of MMR in very high-dimensional joint feature space. We use the GTG kernel (predicted conserved residues) as the base input kernel and the RM kernel as the base output kernel.

In this experiments we use two sets: one for training and second for testing. Figure 1 shows a heat map of the EC digit F1 score. The F1 score improves when the degree either the input, the output or both input and output polynomial kernels increases. The optimum reaches a plateau at input degrees 1-4 and output degrees 8-16 indicating robustness with respect to changes in parameter values.

Applying a high-degree polynomial to the base kernel makes the resulting output kernel more sparse, which suggests that the reactant matching kernel alone is too smooth for optimum performance. We note that optimizing the input and output kernels independently can be useful in other structured prediction settings as well.

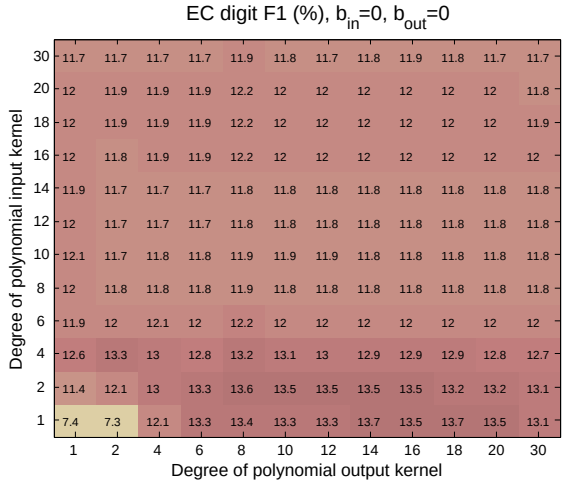

Figure 1: The EC digit F1 score plotted as the function of the degrees of the input and output kernels. The best results are obtained with degree 2 polynomial over the inputs and degree 8 or higher over the outputs.

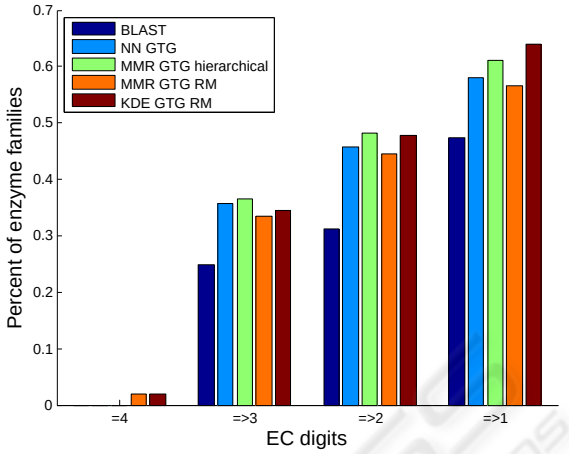

Figure 2: The cumulative distribution of correctly predicted EC digits in the test set (bottom chart). Each member of the top ranking preditions $\hat{\mathcal{Y}}(x)$ contributes one item in the distribution.

5.4.2 Prediction under Remote Homology

In the final experiment, we demonstrate the generalization ability of the structured output prediction methods. We measure how many EC digit are correctly predicted in testing over a five fold set of enzyme families where the four digit EC numbers are not overlapping between folds. Thus the training set contains no enzyme that has exactly the same EC number, but families that have three matching EC digits typically appear in the training.

In this setup is should be clear that the nearest neighbor classifier or the hierarchical classifier cannot ever predict four-digit EC number correctly, as the methods have not seen any examples of that particular family. The reaction kernel approach, however, does not suffer from this limitation: as all possible reactions can be represented in the output space, it is in principle possible to predict the correct function.

Figure 2 shows the results of this experiment. Here, we used a degree 8 polynomial kernel over the RM kernel and degree 2 polynomial kernel over the inputs. In the bottom chart is the cumulative chart depicting the number of enzyme families that have at least certain number of correctly predicted EC digits.

It can be seen that the methods relying on the GTG features (NN(GTG), KDE(GTG,RM) and both MMR methods) are more effective in predicting more than one EC digits correcly. The KDE reaction kernel and MMR hierarchical approach is slightly better in predicting two or more EC digits correctly than the competing approaches. Finally, we note that the reaction kernel approach is the only method that, at times, can get the whole EC number correct. In other words, the set of top-ranking reactions $\hat{\mathcal{Y}}(x)$ contain reactions that possess the exactly correct EC number.

6 DISCUSSION

The present experiments show the potential of structured output prediction using reaction kernels: given a novel, previously unseen enzymatic function, the reaction kernel approach is significantly more accurate than the annotation transfer approach and also compares with a hierarchical classifier trained with structured output learning.

Also we note that the reaction kernel approach is an *enabling* technique: it is possible, albeit not easy, to predict the new function exactly correctly. Interestingly best results are obtained with a highly complex output representation: a high-degree polynomial kernel over reactant matching kernel.

As the result show, using the reaction kernel methods for enzyme function prediction is encouraging way to go, even if the prediction accuracy is still very low for all of the methods used. There are many areas where the methods can be improved. First, we only used predicted conserved residues (GTG) as inputs. Although they work well, augmenting them with other types of data, e.g. structural information should be helpful. Second, the presented reaction kernels certain can be improved and completely different kinds of encodings of enzyme function can be imagined.

Third, a better preimage algorithm will be needed

for novel prediction, brute-force enumeration of reactions, although sufficient for the purposes of this paper, is not a satisfactory approach for a practical system. As simpler output representations may provide more efficient preimage algorithms, it would be tempting to simplify the representations. However, in our view this should not be done at the expense of predictive accuracy.

REFERENCES

Astikainen, K., Holm, L., Pitknen, E., Szedmak, S., and Rousu, J. (2008). Towards structured output prediction of enzyme function. *BMC Proceedings*, 2(S4):S2.

Barutcuoglu, Z., Schapire, R., and Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836.

Blockeel, H., Schietgat, L., Struyf, J., et al. (2006). Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *PKDD*.

Borgwardt, K. M., Ong, C. S., Schnauer, S., Vishwanathan, S. V. N., Smola, A. J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, 21(1):47–56.

Clare, A. and King, R. (2002). Machine learning of functional class from phenotype data. *Bioinformatics*, 18(1):160–166.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C., Hofmann, K., and Bairoch, A. (2002). The prosite database, its status in 2002. *Nucleic Acids Research*, 30(1):235.

Gartner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, 5.

Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M. (2002). Ligand: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Research*, 30(1):402.

Heger, A., Korpelainen, E., Hupponen, T., Mattila, K., Ollikainen, V., and Holm, L. (2008). Pairsdb atlas of protein sequence space. *Nucl. Acids Res.*, 36:D276–D280.

Heger, A., Mallick, S., Wilton, C., and Holm, L. (2007). The global trace graph, a novel paradigm for searching protein sequence databases. *Bioinformatics*, 23(18).

Henikoff, J. and Henikoff, S. (1996). Blocks database and its applications. *METHODS IN ENZYMOLOGY*, pages 88–104.

Holm, L. and Sander, C. (1996). Dali/fssp classification of three-dimensional protein folds. *Nucleic Acids Research*, 25(1):231–234.

Krissinel, E. and Henrick, K. (2004). Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica D Biol Crystallogr*, 60(1 Part 12):2256–2268.

Lanckriet, G., Deng, M., Cristianini, N., et al. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. *PSB*, 2004.

Mulder, N., Apweiler, R., Attwood, T., Bairoch, A., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., et al. (2002). Interpro: An integrated documentation resource for protein families, domains and functional sites. *Briefings in Bioinformatics*, 3(3):225–235.

Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press.

Punta, M. and Ofran, Y. (2008). The rough guide to in silico function prediction, or how to use sequence and structure information to predict protein function. *PLoS Computational Biology*, 4(10).

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *JMLR*, 7.

Schlkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471.

Sokolov, A. and Ben-Hur, A. (2008). A structured-outputs method for prediction of protein function. In *Proceedings of the 3rd International Workshop on Machine Learning in Systems Biology*.

Szedmak, S., Shawe-Taylor, J., and Parado-Hernandez, E. (2005). Learning via linear operators: Maximum margin regression. Technical report, Pascal.

Taskar, B., Guestrin, C., and Koller, D. (2004). Max-margin markov networks. In *NIPS 2003*.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *ICML*.

Ye, Y. and Godzik, A. (2004). Fatcat: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Research*, 32(Web Server Issue):W582.