Rogers, M. F., Campbell, C., Shihab, H. A., Gaunt, T. R., Mort, M., & Cooper, D. N. (2015). Sequential data selection for predicting the pathogenic effects of sequence variation. In Proceedings - 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015. (pp. 639-644). [7359759] Institute of Electrical and Electronics Engineers, Inc.(IEEE). 10.1109/BIBM.2015.7359759

Sequential Data Selection for Predicting the Pathogenic Effects of Sequence Variation

Mark F. Rogers and
Colin Campbell
Intelligent Systems Laboratory
Merchant Venturers School of Engineering
University of Bristol
BS8 1UB, U.K.
Email: Mark.Rogers@bristol.ac.uk

Hashem A. Shihab and
Tom R. Gaunt
MRC Integrative Epidemiology Unit,
Bristol Centre for Systems Biomedicine
University of Bristol
BS8 2BN, U.K.

Matthew Mort and
David N. Cooper
Institute of Medical Genetics
Cardiff University
CF14 4XN, U.K.

***Abstract*—Recent improvements in sequencing technologies provide unprecedented opportunities to investigate the role of genetic variation in human disease. In previous work we have proposed a machine learning approach to predicting whether single nucleotide variants (SNVs) are functional or neutral in human disease. Many data sources from the Encyclopaedia of DNA Elements (ENCODE) may be relevant to this problem. To integrate these data sources, we applied integrative multiple kernel learning (MKL) that weights each source according to its relevance. Using an MKL optimization that yields sparse weights, we were able to eliminate the least informative data sources from our model. However, when selecting from a wide assortment of data sources, we have found that MKL may not be an efficient method for eliminating uninformative sources. Many data sources related to the human genome are incomplete: this can reduce dramatically the data available for training and the proportion of novel predictions that exploit all relevant sources. Here we introduce a greedy sequential selection method that assesses data sources in a structured fashion prior to MKL weight optimization. This method allows us to eliminate a majority of uninformative data sources prior to assigning kernel weights. When we use this method with our coding-region predictor, we select just five kernels for our final model, yielding increased accuracy over our previous model. In addition, by reducing the amount of data required for novel predictions, we are able to increase by five fold our model's coverage for new predictions.**

I. Introduction

The introduction of fast and inexpensive sequencing technologies is providing many new insights into the role of genetic variation in human disease. In this work we consider single nucleotide variants (SNVs) in the human genome. Predicting which of these are functional, as against neutral, promises to improve our understanding of the molecular mechanisms underpinning human disease. In a recent study we proposed a novel algorithmic approach to predicting the functional consequences of both coding and non-coding SNVs (FATHMM-MKL) [12]. Our approach uses integrative *multiple kernel learning* (MKL), a method that learns to weight different types of data according to their relative informativeness. In our work we use SimpleMKL [9], an MKL implementation that uses an L_1 norm to yield sparse solutions that implicitly exclude data sources by assigning them zero weights. In our previous study, our predictor for non-coding SNVs outperformed competing methods using just 4 out of 10 data sources [12]. Our coding-region predictor, using all 10 data sources, matched competitors' performance when all data were available, but its performance suffered when data were missing from some sources.

With MKL, different types of input data are encoded into kernel matrices that quantify the similarity of data objects. A number of different methods have been proposed for deriving kernel matrices for different types of data objects, including data with discrete and continuous values, sequence data and graph data [10]. With MKL, each constituent data type is encoded into a corresponding base kernel $\mathbf{K}_\ell$ (where $\ell = 1, \ldots, p$ if there are p feature groups), from which we can derive a composite kernel matrix $\mathbf{K} = \sum_{\ell=1}^{p} \lambda_\ell \mathbf{K}_\ell$, where $\sum_{\ell=1}^{p} \lambda_\ell = 1$ and $\lambda_\ell \geq 0$. The λ_ℓ are *kernel weights*. This aggregate kernel can then be used with a kernel-based classifier, such as a Support Vector Machine (SVM) [1], which was the classifier used here.

Suppose the training set for a Support Vector Machine consists of vectors $\mathbf{x}_i$ with associated labels $y_i = \pm 1$. The index i labels the training example $(\mathbf{x}_i, y_i)$ and we will assume there are m such training examples in the training set. During the training process for the SVM, the learning parameters α_i are found by maximizing the following convex (quadratic) objective function in α_i [1]:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (1)$$

subject to the linear constraints:

$$\alpha_i \geq 0 \quad , \sum_{i=1}^{m} \alpha_i y_i = 0 \qquad (2)$$

Suppose $\alpha_i^\star$ are the values of the training parameters at the optimum of the objective function stated in (1). From the $\alpha_i^\star$ it is then straightforward to find the *bias*, $b^\star$, or offset in the decision function via:

$$b^\star = -\frac{1}{2}\left[\max_{\{i|y_i=-1\}}\left(\sum_{j=1}^{m}\alpha_j^\star y_j K(\mathbf{x}_i,\mathbf{x}_j)\right)\right. \quad (3)$$

$$\left. + \min_{\{i|y_i=+1\}}\left(\sum_{j=1}^{m}\alpha_j^\star y_j K(\mathbf{x}_i,\mathbf{x}_j)\right)\right] \quad (4)$$

For the binary decision function of an SVM, the predicted label of a novel input $\mathbf{z}$ is then decided by the sign of $\phi(\mathbf{z}) = \sum_{i=1}^{m}\alpha_i^\star y_i K(\mathbf{x}_i,\mathbf{z}) + b^\star$. With a composite kernel of the form $K(\mathbf{x}_i,\mathbf{x}_j) = \sum_{\ell=1}^{p}\lambda_\ell K_\ell(\mathbf{x}_i,\mathbf{x}_j)$ substituted into (1) we see that the learning process now involves optimisation of a linearly constrained linear program in the kernel weights λ_ℓ and a linearly constrained quadratic program in the learning parameters α_i. This is a tractable problem in optimisation theory and can be approached via a wide variety of methods [5], including semi-definite programming or quadratically constrained linear programming (QCLP), for example.

A variety of methods have been proposed for MKL [5] and this approach has been successfully demonstrated with various classification problems in bioinformatics, which use different types of input data e.g. [15]. By using all available data encoded into a set of kernels, MKL classifiers most frequently outperform a single kernel classifier constructed for one type of data. In addition, the kernel weights are adjusted according to the relative informative-ness of the different types of data: this enhances overall performance and interpretation of the model. In its simplest form, all weights in a composite kernel are the same ($\lambda_\ell = \frac{1}{p}$), a form we call an *unweighted aggregate*. When all constituent data sources are at least somewhat informative, an unweighted aggregate may perform as well as one with fully-trained kernel weights. However, when there is disagreement between kernels, performance may decline substantially if uninformative kernels outnumber informative ones. This behavior allows us to evaluate the potential impact of each set of kernels prior to optimizing kernel weights.

MKL optimization methods rely on the assumption that data are available for every kernel for every training example. However, as ENCODE is an ongoing project to annotate the human genome, many data sources are incomplete. As we add data sources it can become increasingly difficult to use MKL to select the most informative ones, as MKL requires training examples common to all of them (see Figure 1). This same restriction can impact novel predictions when data are missing for some sources. In previous work we addressed this by re-weighting the remaining kernels [12] but this may yield lower accuracy than a full-featured prediction (Figure 2). Accordingly, we have developed a novel, greedy approach that pre-selects data sources according to the accuracy of their corresponding kernels. We assess each kernel using cross-validation (CV) and rank the data sources according to their accuracy on a validation set. We then build unweighted aggregate models starting with the two most accurate kernels and sequentially adding the next-lowest-ranked kernel until the aggregate model's performance on the validation set reaches a plateau or declines. Our results suggest that this approach can yield state-of-the-art predictors that dramatically increase the proportion of full-featured predictions.

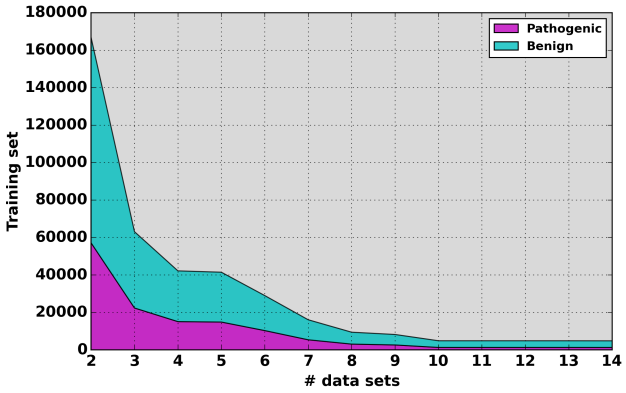

Fig. 1. Training data available for MKL as a function of the number of data sources (kernels) being used. Shown are the total number of examples, along with the number of positive (pathogenic) and benign (negative) examples. The top two kernels (FATHMM conservation scores, Table III) have nearly 167,000 examples in common, including 57,000 positives. As we add kernels, the number drops dramatically, until with 10 or more kernels we have fewer than 5,000 examples, of which just 1,300 are positive.

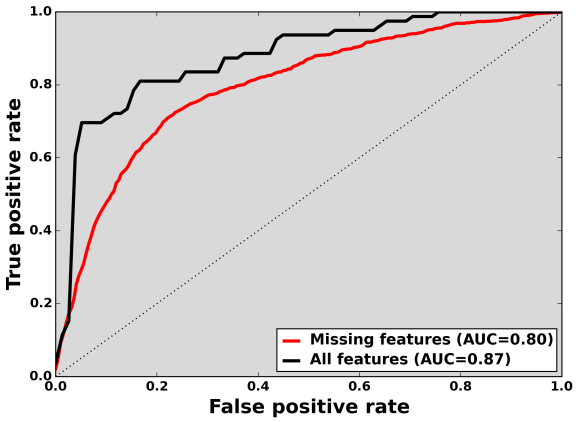

Fig. 2. When all feature groups are available, prediction accuracy remains high (black line) but when only a subset are available, accuracy may be impacted (red line).

II. Methods

For this study we used the same positive and negative examples as in our previous work [12]: a set of pathogenic SNVs derived from the Human Gene Mutation Database [4], and a control set of neutral SNVs from the 1,000 Genomes Project [2]. Many data sources may be relevant to predicting whether a variant is functional in disease. In particular, we used data from the ENCODE consortium, who have assembled approximately 1,640 datasets comprising 24 different experimental approaches in 147 cell lines under various conditions [3]. To leverage this plethora of data, prediction methods

must integrate data from diverse sources and identify the most informative of them. Our FATHMM-MKL method used integrative MKL to weight kernels constructed from different data sources, where zero weights implicitly exclude the least informative sources. However, this may not be a practical way to handle a large number of datasets: as we add new datasets, training an MKL model becomes difficult because many examples are not represented in all data sources. For example, in our previous work only 2,146 out of 87,518 training examples (2.5%) were represented in all 10 of the datasets we used [12]. In addition, a parsimonious model that uses a few well-chosen data sources may generalize better than a model constructed from many sources.

To assess each data source, we train a single-kernel support vector machine (SVM) and compute its average accuracy in 5-fold CV. By isolating each data source in this manner, we can assess its performance across the full spectrum of SNVs represented in the data. To make results comparable between data sources, we use balanced sets of 1,000 positive (pathogenic) and 1,000 negative (neutral) examples in each case. We then rank the data sources according to their average CV accuracy on a validation set.

To identify the data sources we will include in our final model, we construct an unweighted aggregate model using the two top-ranked data sources and record its accuracy on a validation set. We build subsequent models by adding data sources in descending order of accuracy, constructing an aggregate for each combination of data sources. For each combination we establish separate training, validation and test sets: a training set to train the model; a validation set to determine optimal parameters and to record accuracy, and a test set we leave aside to test the final MKL model. We select as our final combination the data sources associated with the unweighted aggregate where accuracy on validation data reaches a plateau or declines. At this point we optimize MKL weights and evaluate the final model on the test set.

III. Results

A. Tests on original data

To assess our proposed method, we compared our original FATHMM-MKL coding-region classifier with models composed of two to ten component kernels, using the same datasets as the original model. Two of these datasets, *100-way conservation* and *46-way conservation*, were constructed from FATHMM [11], PhastCons [13] and Phylop [7] scores, while the remaining data were downloaded from ENCODE [3] (for more details on the datasets and how they were selected, see [12]). These data consist of ten feature groups, with an intersection set of 7,597 examples (2,218 positive and 5,379 negative). To evaluate each component kernel, we performed nested 5-fold CV on balanced sets of 1,774 positive/1,774 negative training examples and 443 positive/443 negative test examples. We used the same training and test examples for all kernels to ensure that we could compare CV statistics between them. The test sets were put aside to test MKL models from the same training data, while the training data were used to evaluate individual kernels. Within each fold, we randomly split the training data into training and validation subsets (80% training and 20% validation). We used the validation set to establish optimal kernel parameters and to determine the maximum accuracy (per fold) for each kernel. We then averaged each kernel's accuracy across folds to yield a kernel ranking (Table I).

Rank	Feature group	Source	Accuracy
1	100-way conservation	FATHMM	**0.812**
2	46-way conservation	FATHMM	**0.805**
3	TFBS (Peak-Seq)	ENCODE	**0.698**
4	Histone (ChIP-Seq)	ENCODE	**0.685**
5	TFBS (SPP)	ENCODE	**0.655**
6	Open chromatin (DNase-Seq)	ENCODE	0.623
7	Open chromatin (FAIRE)	ENCODE	0.575
8	Genome segmentation	ENCODE	0.562
9	DNA footprints	ENCODE	0.555
10	GC content	ENCODE	0.552

TABLE I
5-FOLD CV PERFORMANCE OF INDIVIDUAL LINEAR KERNELS TRAINED AND TESTED ON THE SAME DATA USED TO CONSTRUCT THE ORIGINAL FATHMM-MKL CODING CLASSIFIER. FEATURE GROUPS ARE SHOWN IN DESCENDING ORDER OF ACCURACY AVERAGED ACROSS THE FIVE FOLDS. OUR GREEDY AGGREGATION METHOD SELECTS THE TOP FIVE KERNELS (BOLD) FOR FURTHER MODEL REFINEMENT.

Next we constructed unweighted aggregate models from two or more component kernels and compared their performance to the original FATHMM-MKL on the same test examples. For $k = 2, \ldots, 10$ kernels, we constructed k-kernel aggregates using the top k component kernels according to the rankings shown in Table I. All of the unweighted aggregate models performed better than any of their component kernels, and at $k = 5$ we observed a nominal peak in performance, where the unweighted aggregate performed nearly as well as the original FATHMM-MKL (Table II and Figure 3), The k-kernel aggregate models were trained on 80% of the data and tested on the remaining 20% for five folds, hence each was trained using less data than FATHMM-MKL and none had prior exposure to the test examples. Despite this arguably unfair comparison, the strong performance of these models and the consistent accuracy for models with $k = 5, \ldots, 10$ suggests that we can use fewer datasets than the original model without sacrificing performance.

B. Constructing a new model

For this study we obtained 57,276 pathogenic SNVs from HGMD and identified 109,667 neutral (presumed benign) SNVs from the 1,000 Genomes database for a total of 166,843 SNV examples. We were able to generate FATHMM scores (*100-way conservation* and *46-way conservation*) for all of these examples. Within the 12 additional datasets we selected from ENCODE, we found just one database (*Mappability*) with all of these examples, while *DNA footprints* had just 15,399 (Table III). For our greedy aggregation procedure, we used the data common to all 14 datasets. When we combine all 14 datasets, our training data consists of 4,849 SNVs common to all of them, including 1,300 pathogenic and 3,549 neutral

Model	Acc.	ROC
Original	0.846	0.912
2-kernel	0.820	0.899
3-kernel	0.831	0.892
4-kernel	0.828	0.882
5-kernel	**0.833**	**0.907**
6-kernel	0.829	0.889
7-kernel	0.831	0.903
8-kernel	0.831	0.899
9-kernel	0.831	0.901
10-kernel	0.832	0.901

TABLE II
PERFORMANCE OF UNWEIGHTED AGGREGATE KERNELS ON ORIGINAL TRAINING DATA. SHOWN ARE THE PREDICTION ACCURACY AND ROC SCORE FOR THE ORIGINAL VERSION OF FATHMM-MKL (*Original*) AND FOR AGGREGATES CONSISTING OF UP TO 10 KERNELS ON THE SAME TEST DATA.

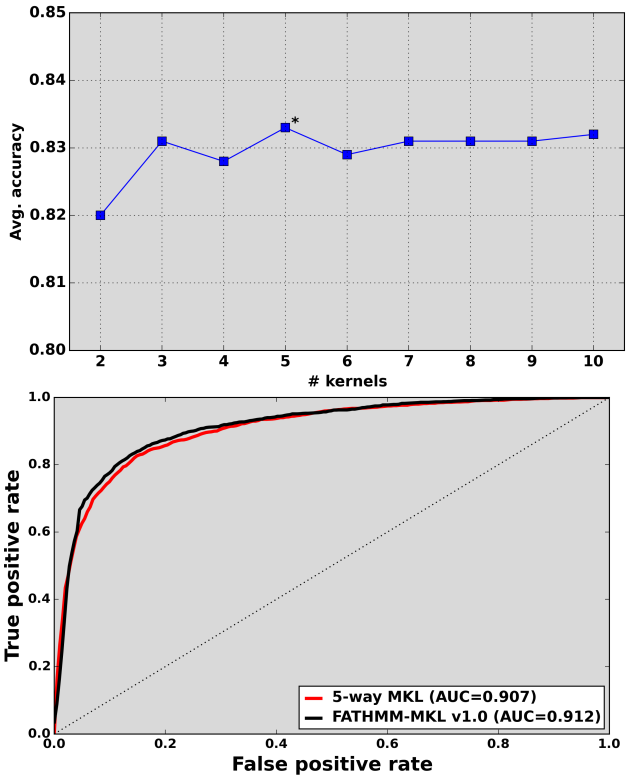

Fig. 3. **Top**: accuracy of aggregate models using the top k kernels in the list, for $k = 2, \ldots, 10$. We reach a nominal peak with 5 kernels, suggesting that a reduced model may yield accuracy at least as high as our original model. **Bottom**: Comparison of the top-performing 5-kernel combination (Table II) with the original FATHMM-MKL. Both models were tested against the data used to train FATHMM-MKL; results for the 5-kernel model are taken from 5-fold CV while results for FATHMM-MKL were taken directly from the database. While FATHMM-MKL may yield slightly better discrimination, there is little evident loss of performance when using the 5-kernel model.

(Figure 1). For each of these datasets we performed nested 5-fold CV using balanced sets of 1,000 pathogenic and 1,000 neutral SNV examples and recorded the average accuracy. For each kernel we used the same examples in each fold, to ensure that the accuracy results were comparable.

As described in Section III-A, we optimised kernel parameters for each fold by splitting the training data into training and validation subsets. We did not find that any of the kernels were linearly separable, so we searched C-values from 10^{-3} to 10^{3} and selected the value that yielded the highest accuracy and still permitted convergence. We then computed average 5-fold CV accuracy for the validation sets to establish kernel rankings. These new rankings are similar to those in our first test (Table III): the two datasets based on FATHMM conservation scores yielded the strongest performance, with accuracy up to 84%. For the ENCODE datasets accuracy ranged from 50% for *Riken CAGE* to 68% for *TFBS (Peak-Seq)*. The new ENCODE datasets altered the rankings slightly, but only *TFBS (Uniform)*, performed well enough to appear in the top half of the rankings.

Rank	Feature group	Examples	Accuracy
1	100-way conservation	166,843	**0.819**
2	46-way conservation	166,843	**0.785**
3	TFBS (Peak-Seq)	63,106	**0.706**
4	Histone (ChIP-Seq)	150,882	**0.681**
5	*TFBS (uniform)*	48,882	**0.673**
6	TFBS SPP	32,494	**0.652**
7	Genome segmentation	166,703	0.601
8	Open chromatin (DNase-Seq)	79,540	0.600
9	*Dnase uniform*	79,219	0.586
10	DNA footprints	15,399	0.578
11	Open chromatin (FAIRE)	48,505	0.573
12	*Riken CAGE*	81,004	0.566
13	GC content	164,656	0.553
14	*Mappability*	166,843	0.496

TABLE III
THE FULL SET OF FEATURE GROUPS CONSIDERED FOR OUR NEW MODEL, SHOWN IN DESCENDING ORDER OF ACCURACY. ACCURACY WAS DETERMINED BY SELECTING FROM THE DATA AVAILABLE FOR EACH DATA SET, AS OPPOSED TO THE CROSS-SECTION USED IN OUR ORIGINAL MODEL. THE RESULTING PERFORMANCE, ALONG WITH NEW DATA SOURCES (ITALICS) CHANGED THE RANKINGS FOR SOME FEATURE GROUPS. USING OUR GREEDY SELECTION METHOD, WE IDENTIFIED SIX DATA SETS (BOLD) AS LIKELY TO BE THE MOST INFORMATIVE.

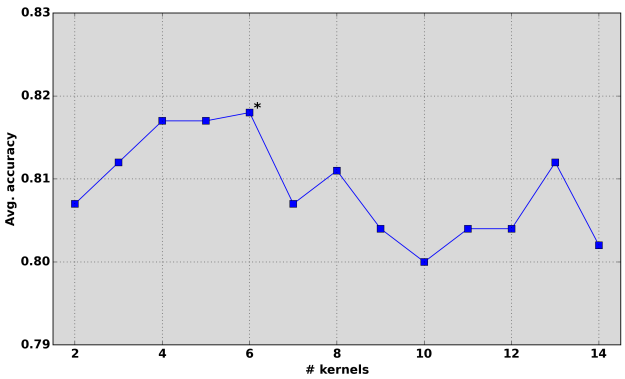

Fig. 4. Five-fold CV performance on the 14 new kernels illustrates how performance for unweighted models can degrade as we add new, less informative, kernels. Performance appears to peak with accuracy of 82% with the top six kernels, after which we see sharp drop and continued weak performance for aggregates with seven to 14 kernels.

When we applied our greedy aggregation procedure, we found that average accuracy increased gradually from 81% for

for $k = 2$ to 82% for $k = 6$, but declined after that. We also see a sharp decline in individual kernel performance between the sixth and seventh ranked kernels (Table III, *TFBS SPP* and *Genome segmentation*), so we selected the top six kernels for MKL optimization. Note that the unweighted aggregates may not perform as well as their best constituent kernels, as conflicting scores from constituent kernels may cancel each other out. However, the model can realize substantial gains once optimum weights have been learned.

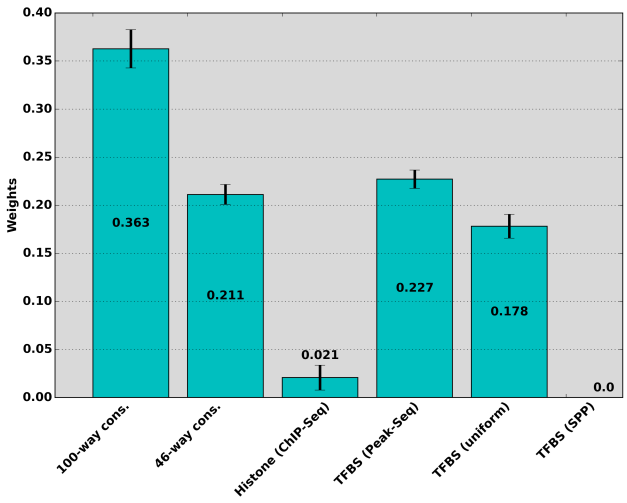

Fig. 5. We used SimpleMKL to optimize kernel weights for the final model. Weights were highly consistent across the five folds, as indicated by the error bars. This result helped us eliminate an additional data set, *TFBS (SPP)*. Note that although the *46-way conservation* data are ranked higher than *TFBS (Peak-Seq)*, the associated weight is lower. This is likely due to redundancy within the two FATHMM score datasets.

Two of these kernels are associated with FATHMM conservation scores and represent all available training data. Missing data in the remaining sets reduced our overall training set to 28,998 examples—still vastly more than we had for our original study. We used SimpleMKL to establish kernel weights for each fold, using only the training and validation sets. For each kernel we then used the average weight as the weight for the final model (Figure 5). Note that the optimized weights do not track the relative rankings very closely. This is likely due to redundancies between the data sets: the *100-way conservation* and *46-way conservation* scores are closely related, so the *46-way conservation* weight is considerably lower than we might expect given the relatively small difference in their kernel accuracies (Table III). Similarly, one of the three TFBS kernels, *TFBS (SPP)* received zero weight, allowing us to eliminate that source. In turn, this increased the number of examples available to train the final model, to 41,476 examples. This also improved coverage for novel predictions, from under 5% for the original classifier to 24.9% for the new model, a five-fold improvement.

To test our final model, we ran 5-fold CV using a balanced set of 2,000 pathogenic and 2,000 neutral examples. We compared these predictions with those of the original FATHMM-MKL and two other state-of-the-art methods, CADD [6][1] and DANN [8] (Figure 6). Our new model yields substantial improvements over our previous model (Figure 6, top), likely due to the additional training data now available. All of the top competitors yield similar performance (Figure 6, top): the newest version of CADD (v1.3) is the best of these competitors (AUC 0.90) while our new model yields the top AUC score of 0.91. While these results do not suggest a clear winner, they demonstrate that our new model provides accuracy that is competitive with an ever-improving state of the art. In addition, we found that we could obtain scores for all five of our datasets across 24.9% of coding regions in the human genome, a dramatic improvement over the severely restricted coverage we obtained when using a 10-kernel model.

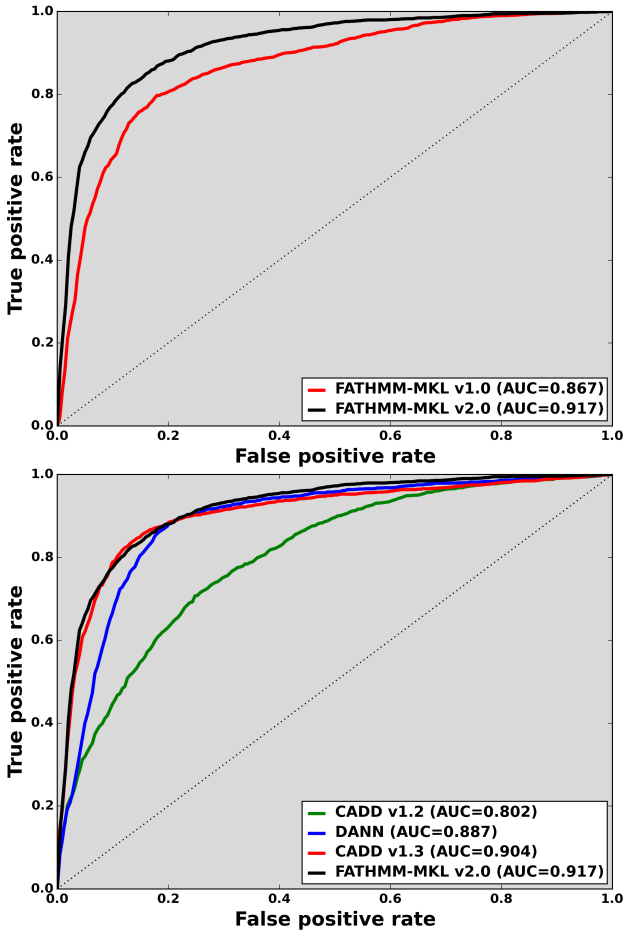

Fig. 6. Five-fold CV performance using the coding dataset and the top-performing aggregate kernel using just the top five kernels. Focusing on just the top five kernels allowed us to use all of our training data instead of the tiny proportion used in our original model. In sum, this improves performance substantially over FATHMM-MKL v1.0 (top) and outperforms the top competing algorithms in our tests (bottom).

IV. Conclusions

Motivated by observations with our previous development of MKL methods [16], in this paper we propose a *greedy*

[1]CADD version 1.3 was released as this draft was in preparation, so we present results for versions 1.2 and 1.3.

approach to pre-selecting data sources. Our new model gives greater test accuracy than our original model [12]. This greedy approach suggests that certain types of data can be ignored because the information they contain is implicit in an already learnt data source (encoded into its respective kernel), or because a new data source contains little new information and may also contribute a substantive extent of noise.

These results suggest further promising directions for future work, to improve the data integration procedure. We intend exploring other MKL methods such as those surveyed in [5]. Indeed, the *difference of convex* approach we proposed in [16], achieved up to 6% greater test accuracy over the SimpleMKL method used here, for benchmarking studies with some datasets (though it has an adjustable parameter which must be found via a validation study). The method in [16] also has the advantage that the kernel weights λ_ℓ are found separately from the learning parameters α_i. After deriving the kernel weights, and hence a composite kernel, this means that the composite kernel can be used in any kernel-based learning method. For example, the Core Vector Machine [14] can handle larger datasets than an SVM (computational complexity scales as m rather than the m^3 of the SVM) and, by using more training data, we could improve performance. Rather than integrating component feature groups at the level of the data, via a composite kernel, it would also be possible to integrate classifiers via ensemble learning. In future projects, we shall investigate these potential improvements in addition to devising bespoke predictors for labelling variants in specific disease contexts, such as cancer.

ACKNOWLEDGMENTS

This work was supported by the Medical Research Council MC_UU_12013/8 and G1000427/1. M.R. was supported by an EPSRC grant (EP/K008250/1). MM & DNC gratefully acknowledge the financial support of BIOBASE GmbH. The authors also wish to thank Martin Kircher for making CADD training data available.

REFERENCES

[1] C. Campbell and Y. Ying. *Learning with Support Vector Machines*. Morgan and Claypool, 2011.

[2] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491:56–65, 2012.

[3] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489:57–74, 2012.

[4] Stenson *et al*. The human gene mutation database: building a comprehensive mutation repository for clinical and molecular genertics, diagnostic testing and personalized genomic medicine. *Hum. Genet.*, 133:1–9, 2014.

[5] M. Gonen and E. Alpaydn. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[6] Martin Kircher, Daniela M Witten, Preti Jain, Brian J O'Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, 46(3):310–315, 2014.

[7] K. S. Pollard, M.J. Hubisz, K.R. Rosenbloom, and A. Siepel. Detection of non-neutral substitution rates on mammalian phylogenies. *Genome Research*, 20:110–121, 2010.

[8] Daniel Quang, Yifei Chen, and Xiaohui Xie. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics*, 31:761–763, 2015.

[9] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[10] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[11] Hashem A Shihab, Julian Gough, David N Cooper, Peter D Stenson, Gary LA Barker, Keith J Edwards, Ian NM Day, and Tom R Gaunt. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Human mutation*, 34(1):57–65, 2013.

[12] Hashem A Shihab, Mark F Rogers, Julian Gough, Matthew Mort, David N Cooper, Ian NM Day, Tom R Gaunt, and Colin Campbell. An integrative approach to predicting the functional effects of non-coding and coding sequence variation. *Bioinformatics*, 31:1536–1543, 2015.

[13] A. Siepel, G. Bejerano, J.S. Pedersen, A.S. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L.W. Hillier, S. Richards, G.M. Weinstock, R.K. Wilson, R.A. Gibbs, W.J. Kent, W. Miller, and D. Haussler. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*, 15:1034–1050, 2005.

[14] I.W. Tsang, J. T. Kwok, and P.-M. Cheung. Core vector machines: Fast svm training on very large data sets. *J. Mach. Learn. Res.*, 6:363392, 2005.

[15] Y. Ying, C. Campbell, T. Damoulas, and M. Girolami. Class prediction from disparate biological data sources using an iterative multi-kernel algorithm. *Lecture Notes in Bioinformatics*, 5780:427–438, 2009.

[16] Y. Ying, K. Huang, and C. Campbell. Enhanced protein fold recognition through a novel data integration approach. *BMC Bioinformatics*, 10:267, 2009.