Conference Paper

Improved functional prediction of proteins by learning kernel combinations in multilabel settings

Author(s):
Roth, Volker; Fischer, Bernd

Publication Date:
2007-05

Permanent Link:

https://doi.org/10.3929/ethz-b-000008235 →

Originally published in:

BMC Bioinformatics 8(Supplement 2), http://doi.org/10.1186/1471-2105-8-S2-S12 →

BMC Bioinformatics

BioMed Central

Research

Improved functional prediction of proteins by learning kernel combinations in multilabel settings

Volker Roth* and Bernd Fischer

Address: ETH Zurich, Institute of Computational Science, Universität-Str. 6, CH-8092 Zurich, Switzerland

Email: Volker Roth* - vroth@inf.ethz.ch; Bernd Fischer - bernd.fischer@inf.ethz.ch

* Corresponding author

from Probabilistic Modeling and Machine Learning in Structural and Systems Biology
Tuusula, Finland. 17–18 June 2006

Published: 3 May 2007

BMC Bioinformatics 2007, **8**(Suppl 2):S12 doi:10.1186/1471-2105-8-S2-S12

This article is available from: http://www.biomedcentral.com/1471-2105/8/S2/S12

Abstract

Background: We develop a probabilistic model for combining kernel matrices to predict the function of proteins. It extends previous approaches in that it can handle multiple labels which naturally appear in the context of protein function.

Results: Explicit modeling of multilabels significantly improves the capability of learning protein function from multiple kernels. The performance and the interpretability of the inference model are further improved by simultaneously predicting the subcellular localization of proteins and by combining pairwise classifiers to consistent class membership estimates.

Conclusion: For the purpose of functional prediction of proteins, multilabels provide valuable information that should be included adequately in the training process of classifiers. Learning of functional categories gains from co-prediction of subcellular localization. Pairwise separation rules allow very detailed insights into the relevance of different measurements like sequence, structure, interaction data, or expression data. A preliminary version of the software can be downloaded from http://www.inf.ethz.ch/personal/vroth/KernelHMM/.

Background

The problem of developing machine-learning tools for protein function prediction has gained considerable attention during the last years. From a machine learning perspective, this task exceeds the "standard" settings of learning problems in that a protein can be involved in several different biological processes exhibiting more than one function. This means that the objects we want to classify (i.e. the proteins) might belong to several classes, a setting which is referred to as *multilabel classification*. From a biological perspective, the information carried in these multilabels might be relevant for extracting correlations of functional classes. When it comes to predicting the function of new proteins, it is therefore desirable to develop tools that can explicitly handle such multiple labeled objects.

In this work we present a multilabel version of a nonlinear classifier employing Mercer kernels. Such kernel methods have been successfully applied to a variety of biological data analysis problems. One problem of using kernels, however, is the lacking interpretability of the decision

functions. In particular, it is difficult to extract further insights into the nature of a given problem from kernel mappings which represent the data in implicitly defined feature spaces. It has been proposed to address this problem by using *multiple* kernels together with some combination rules, where each of the kernels measures different aspects of the data. Methods for learning sparse kernel combinations have the potential to extract *relevant* measurements for a given task. Moreover, the use of multiple kernels addresses the problem of *data fusion* which is a challenging problem in bioinformatics where data can be represented as strings, graphs, or high dimensional expression profiles. Kernels provide a suitable framework for combining such inhomogeneous data under a common matrix representation.

Existing algorithms for combining kernels recast the problem as a *quadratically constrained quadratic program* (QCQP), [1], as a *semi-infinite linear program* (SILP), [2], or within a *sequential minimization optimization* (SMO) framework, [3]. Methods for selecting kernel parameters have also been introduced in the boosting literature, see e.g. [4] or in the context of Gaussian processes, see e.g. [5]. Our method extends these approaches in two major aspects: due to the generative nature of the underlying classification model, it can learn class correlations induced by multilabeled objects and it can be used as a "building block" in hidden Markov models which allow the inclusion of further categorical information, such as the joint prediction of subcellular localization classes and functional classes. We show that these extensions significantly improve the predictive performance on yeast proteins.

Methods

Our classifier is based on an extension of the *mixture discriminant analysis* (MDA) framework, which forms a link between Gaussian mixture models and discriminant analysis [6]. The algorithm for solving multilabel classification problems emerges as a special case of this clustering approach. In the following we will briefly outline the algorithm which is composed of the "building blocks" *Gaussian mixture models*, *discriminant analysis*, *adaptive ridge penalties* and the proper handling of *multilabels*.

Learning Gaussian mixtures by LDA

The dataset is assumed to be given as a collection of n samples $\boldsymbol{x}_i \in \mathbb{R}^d$, summarized in the $(n \times d)$ matrix X. Consider now a Gaussian mixture model with K mixture components which share an identical covariance matrix Σ. Under this model, the data log-likelihood reads

$$l^{mix} = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \Sigma) \right), \qquad (1)$$

where the mixing proportions π_k sum to one, and φ denotes a Gaussian density. The classical EM-algorithm, [7], provides a convenient method for maximizing l^{mix}: in the E-step one computes the assignment probabilities $P(C_k|\boldsymbol{x}_i)$ of objects $\boldsymbol{x}_i$ to classes C_k, while in the M-step the current parameters of the Gaussian modes are replaced by the maximum likelihood estimates.

Linear discriminant analysis (LDA) is a time-honored classifier that (asymptotically) finds the correct class boundaries if the class-conditional densities are Gaussians with common covariance, which is exactly the supervised version of our model (1). This optimality is due to the fact that for given *class labels*, the maximum likelihood parameters of the model (1) can be found by LDA, see e.g. [8]. This result has been generalized in [6], where it has been shown that in the unsupervised "clustering" case the M-step can be carried out via a *weighted and augmented* LDA: class labels are mimicked by replicating the n observations K times, with the k-th replication having observation weights $P(C_k|\boldsymbol{x}_i)$ and the "class label" k.

Following [9], any (standard) LDA problem can be restated as an *optimal scoring* problem. Let the class-memberships of the n data vectors be coded as a matrix Z, the (i, k)-th entry of which equals one if the i-th observation belongs to class k. The point of optimal scoring is to turn categorical variables into quantitative ones: a score vector $\boldsymbol{\theta}$ assigns real numbers to the K levels of the categorical response variable, i.e. to the entries in the columns of Z. The simultaneous estimation of a sequence of scores $\boldsymbol{\theta}_k$ and regression coefficients $\boldsymbol{\beta}_k$, $k = 1,...,K$, constitutes the optimal scoring problem: minimize

$$\sum_{k=1}^{K} || Z\boldsymbol{\theta}_k - X\boldsymbol{\beta}_k ||^2 \qquad (2)$$

under the orthogonality constraint $\Theta ZZ\Theta = I_K/n$, where $\Theta = (\boldsymbol{\theta}_1,...,\boldsymbol{\theta}_K)$ and I_K denotes the $(K \times K)$ identity matrix. In [9] an algorithm for this problem has been proposed, whose main ingredient is a multiple linear regression of the scored responses $Z\boldsymbol{\theta}_k$ against the data matrix X. The algorithm starts with $\Theta = I_K$, and the optimal scores are derived from the solution of the multiple regression problem via an eigen-decomposition. It can be shown that solutions to (2) must satisfy a certain orthogonality constraint which allows us to start with a $(K \times K - 1)$ scoring matrix Θ' that is orthogonal to a K-vector of ones. In the following we will always consider this latter variant which is beneficial since it reduces the number of regressions to $K - 1$. For simplicity in notation we will still write Θ instead of Θ'.

Returning to the above *weighted and augmented* LDA problem, it has been shown in [6] that the solution for this problem can be found by the standard *optimal scoring* ver-

sion of LDA after replacing the class indicator matrix Z by its "blurred" counterpart $\tilde{Z}$. The rows of $\tilde{Z}$ consist of the class membership probabilities estimated in the preceding E-step of the EM algorithm.

Adaptive ridge penalties and kernelization

In order to find sparse solutions, we take a Bayesian viewpoint of the multiple regression problem (2) and specify a prior distribution over the regression coefficients β. Following some ideas proposed in the Gaussian process literature, we choose *Automatic Relevance Determination* (ARD) priors, [10], which consist of a product of zero-mean Gaussians with inverse variances ω_i:

$$p(\boldsymbol{\beta} \mid \boldsymbol{\omega}) \propto \exp\left[-\sum_{i=1}^{d} \omega_i \beta_i^2 \right]. \qquad (3)$$

The hyper-parameters ω_i encode the "relevance" of the i-th variable in the linear regression. A method called *adaptive ridge regression* finds the hyper-parameters by requiring that the mean prior variance is proportional to $1/\lambda$, cf. [11]: $\frac{1}{d}\sum_{i=1}^{d} \frac{1}{\omega_i} = \frac{1}{\lambda}$, $\omega_i > 0$, where λ is a predefined regularization constant.

The balancing procedure has the effect that some hyper-parameters ω_i go to infinity. As a consequence, the coefficients β_i are shrinked to zero and the corresponding input variables are discarded. Following [11] it is numerically advantageous to introduce new variables $\gamma_{j,i} = \sqrt{\omega_i / \lambda}\,\beta_{j,i}$, $c_i = \sqrt{\lambda / \omega_i}$. Denoting by D_c a diagonal matrix with elements c_i, we have to minimize

$$\sum_{k=1}^{K-1} || \tilde{Z}\boldsymbol{\theta}_k - XD_c\boldsymbol{\gamma}_k ||^2 + \lambda \boldsymbol{\gamma}_k^\top \boldsymbol{\gamma}_k \quad \text{s.t.} \quad \boldsymbol{c}^\top \boldsymbol{c} = d, \quad c_i > 0. \qquad (4)$$

We now consider the case of sharing weights over J blocks containing m regression coefficients each:

$$\boldsymbol{c} = (\underbrace{c_1, \ldots, c_1}_{m \text{ times}}, \ldots, \underbrace{c_J, \ldots, c_J}_{m \text{ times}})^\top. \qquad (5)$$

Note that for given weights $\boldsymbol{c}$, eq. (4) defines a standard ridge-regression problem in the transformed data $\tilde{X} = XD_c$. It is well-known in the kernel literature that the solution vectors $\hat{\boldsymbol{\gamma}}_k$ lie in the span of these input data, i.e. $\hat{\boldsymbol{\gamma}}_k = \tilde{X}\,\alpha_k$, which means that the data enter the model only in form of the Gram matrix (or Mercer kernel) $\tilde{X}\,\tilde{X}^\top$. Since we have assumed that a weight c_i is shared over a whole block of m features, we can decompose this kernel as a weighted sum of J individual kernels:

$$K := \tilde{X}\tilde{X}^\top = \sum_{j=1}^{J} c_j^2 \tilde{X}_{(j)} \tilde{X}_{(j)}^\top =: \sum_{j=1}^{J} c_j^2 K_j. \qquad (6)$$

with $\tilde{X}_{(j)}$ denoting a $(n \times m)$ sub-matrix of $\tilde{X}$ consisting of one block of m input features. With the above expression we have arrived at the desired framework for learning sparse combinations of kernel matrices: the kernel matrices K_j in (6) which have been formally introduced by partitioning an initial feature set into J feature blocks can be substituted by arbitrary kernels fulfilling the positive-semidefiniteness condition of valid dot product matrices. On the technical side, we have to minimize the "kernelized" version of eq. (4)

$$\sum_{k=1}^{K-1} || \tilde{Z}\boldsymbol{\theta}_k - (\sum_{j=1}^{J} c_j^2 K_j)\boldsymbol{\alpha}_k ||^2 + \lambda \boldsymbol{\alpha}_k^\top (\sum_{j=1}^{J} c_j^2 K_j)\boldsymbol{\alpha}_k \qquad (7)$$

subject to $\boldsymbol{c}^\top \boldsymbol{c} = \sum_{j=1}^{J} c_j^2 = d$, $c_i > 0$. The minimizing vectors $\hat{\alpha}_k$, $k = 1,\ldots,K-1$ can be found simultaneously in a very efficient way by employing *block conjugate gradient methods* [12]. The optimal weights $\boldsymbol{c}$ are found iteratively by a fixed-point algorithm similar to that proposed in [11]:

$$(c_j^2)_{\text{new}} = J \frac{\sum_{k=1}^{K-1} c_j^2 \boldsymbol{\alpha}_k^\top K_j \boldsymbol{\alpha}_k}{\sum_{k=1}^{K-1} \sum_{l=1}^{J} c_l^2 \boldsymbol{\alpha}_k^\top K_l \boldsymbol{\alpha}_k}. \qquad (8)$$

Practically, if during the iterations a component c_j becomes small compared to a predefined accuracy constant, c_j is set to zero, and in all regression problems the j-th kernel vanishes.

The algorithm proceeds with iterated computations of kernel weights $\boldsymbol{c}$ and the expansion coefficients α_k, $k = 1\ldots K-1$. We initialize the model with $c_j = 1\ \forall j$. In our experiments, the initialization did not critically influence the final result, as long as the initial c_j's are non-zero and $J < n$. Theoretical uniqueness results, however, are difficult to derive. For the special case without weight sharing (i.e. $J = d$) the above method is equivalent to the *LASSO* model of ℓ_1-penalized regression (see [11]) for which a unique solution always exists if the dimensionality does not exceed the number of samples, $d \le n$. If d exceeds n (which might be the case for the kernel models considered here), there might exist different solutions which, however, share the same globally optimal value of the functional (4). The experimentally observed insensitivity to different initializations is probably due to the weights-sharing constraints that shrink the number of different c_j's from d to J.

A theoretical analysis of the uniqueness of solutions, however, will be subject of future work.

Multilabel classification

In multilabel classification problems, an object $\boldsymbol{x}_i$ can belong to more than one class, i.e. it might come with a set of labels Y_i. We treat these multilabels in a probabilistic way by assigning to each observation a set of class-membership probabilities. These probabilities might be given explicitly by the supervisor. If such information is not available, they might be estimated uniformly as $1/|Y_i|$ for classes included in the label set Y_i, and zero otherwise. After encoding these probabilities in the "blurred" response matrix $\tilde{Z}$ (which corresponds to a single E-step in the EM algorithm), we run one optimization step (i.e. M-step) described above.

Kernel discriminant analysis is a generative classifier which implicitly models the classes as Gaussians in the kernel feature space. The effect of multilabels on the classifier during the training phase can be understood intuitively as follows. If there are many objects in the training set which belong to both the classes C_i and C_j, the respective class centroids μ_i, μ_j will be shrunken towards the averaged value $1/2 \cdot (\mu_i + \mu_j)$. In this way, the classifier can learn the correlation of class labels and favor the co-prediction of class *i* and *j*.

For discriminant analysis it is straightforward to compute for each object a vector of assignment probabilities to the individual classes C_k, see e.g. [9]. In a traditional two-class scenario we would typically assign an object to class C_1 if the corresponding membership probability exceeds 1/2 (for equal class priors). In multilabel scenarios, however, an object can belong to different classes so that we have to find a suitable way of thresholding the output probabilities. In analogy to the classical two-class case, we propose to sort the assignment probabilities in decreasing order and assign an object $\boldsymbol{x}_i$ to the first *k* classes in this order such that

$$\sum_{j=1}^{k} p^{\text{sorted}}(C_j \mid \boldsymbol{x}_i) \geq \tau, \qquad (9)$$

where τ is a predefined threshold (e.g. $\tau = 1/2$). By varying τ one can record the usual precision-recall curves, cf. Figure 1 for an example.

Algorithm 1 Multilabel kernel learning via AdR regression

Training: /* we start with one E-step */

compute "blurred" ($n \times K$) response matrix $\tilde{Z}$ encoding the membership probabilities.

/* now follows one single M-step */

compute initial ($n \times K$ - 1) scoring matrix Θ_0, see [9] for details;

Initialize $c_i = 1$, $\forall\ i = 1,...,J$;

repeat

compute α_k, $k = 1,...,K$ - 1 as the solution of the linear systems (7);

recompute the kernel weights $\boldsymbol{c}^{(t+1)}$, see (8);

until $||\boldsymbol{c}^{(t+1)} - \boldsymbol{c}^{(t)}|| < \varepsilon$

compute fitted values and projection matrix for the discriminant analysis subspace, see [9,13];

Prediction:

compute projected test object $\tilde{\boldsymbol{x}}_*$ and Mahalanobis distances to class centroids;

compute class membership probabilities $P(C_k|\boldsymbol{x}_*)$ and extract multilabels according to eq.(9).

Model selection

For the purpose of model selection, we assume that we are given a set of kernel matrices over multi-labeled objects. We further assume that the rule of deriving "fractional labels" from the multilabels is given. In the absence of further prior knowledge (which is probably the case in most real-world applications), we assume that the fractional labels are derived by averaging over all classes to which an object is assigned. Under these assumptions, the model contains only two free parameters, namely the regularization constant λ in eq. (7) and the threshold τ for predicting multilabels in eq. (9). In our experiments, the former is estimated via cross-validation on the training sets. A unique multilabel-threshold can also be estimated by cross-validation, but in this work we report the full precision-recall-F1-curves obtained by varying this threshold, see Figure 1.

Functional classes and subcellular localization

In several classification tasks in bioinformatics, more than one classification scheme is available to assign the data to certain groups. Proteins, for instance, can be classified not only according to their *function*, but also according to their *subcellular localization*. For the yeast genome, such a local-

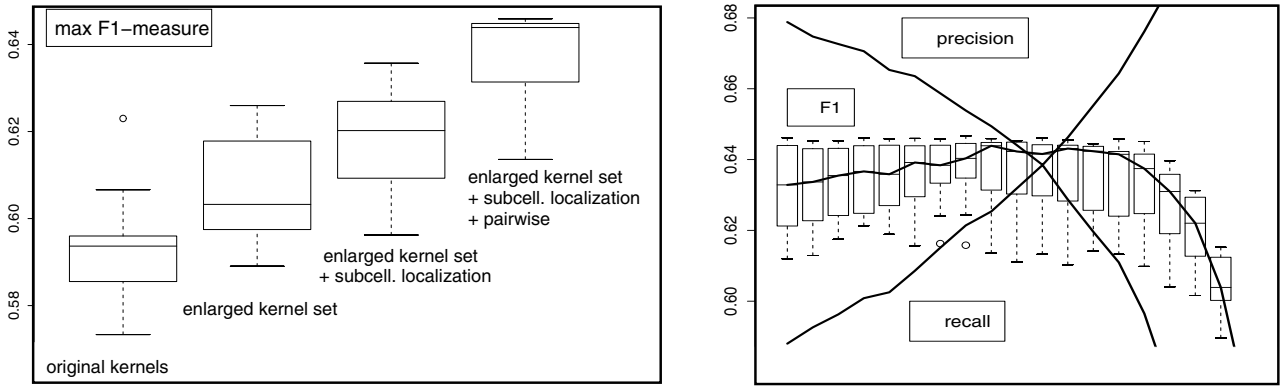

Figure 1

Performance of different classifiers. Performance of different classifier variants under cross-validation. Left panel: boxplots of maximum F1 measure. Left to right: 8 original kernels from [1], extended kernel set, combined subcellular location and functional prediction, pairwise learning of the kernel weights (in the combined model). Right panel: precision, recall and F1 for the pairwise classifier used as "building block" in the HMM under variation of the threshold τ which controls the label multiplicity.

ization-based scheme is available from CYGD, see Table 2. If for some proteins their subcellular localization can be predicted with high reliability, and if we find high correlations between the corresponding localization classes and certain functional classes, we can potentially exploit this prior knowledge to increase the performance of the functional classification.

We combine both classification schemes by way of a *hidden Markov model* (HMM). This choice was guided by the observation that even the standard *K*-class discriminant analysis model can be viewed as a simple HMM consisting of *K* emitting nodes and a silent *begin* and *end* node, see the left panel of Figure 2. The *K* emitting nodes represent the values of the hidden variable $\mathcal{F}$ ("functional class"). The Gaussian emission probabilities are derived from the classifier. The transition probabilities are estimated by the empirical frequencies of class membership in a training set.

A second classification scheme is included in the automaton by adding another layer with *L* emitting nodes. These additional nodes correspond to a Gaussian mixture model with *L* components that in our case correspond to subcellular localization classes, see the right panel of Figure 2. The emitting nodes in the first layer represent the values of the hidden variable $\mathcal{L}$ ("localization"), whereas the nodes in the second layer represent the values of the second hidden variable $\mathcal{F}$ ("functional class"). There are now two "observed data" variables $\mathcal{X}_1$, $\mathcal{X}_2$ that are assumed conditionally independent given the states of the two hidden variables. This independence assumption might be justified by the use of sparse kernel selection rules in each layer which typically induce nearly orthogonal feature spaces. The emission probabilities $P(\mathcal{X}_1|\mathcal{L})$ and $P(\mathcal{X}_2|\mathcal{F})$ are learned separately in the two layers. As in the former case, the individual transition probabilities are estimated by empirical frequencies on the training set. For predicting multilabels in the second layer, we compute the posterior probabilities via the forward-backward algorithm. The number of labels to be assigned is again found by thresholding the sum of ordered probabilities, cf. eq. (9). Varying this threshold yields precision-recall curves as depicted in Figure 1.

Locality due to pairwise kernel classifiers

The method introduced above finds a "global" set of kernels for the full multi-class problem. In some applications, however, it might be desirable to further investigate the discriminative power of kernels in a more class-specific or "local" setting. This can be achieved by an alternative approach to multi-class discriminant analysis based on the *pairwise coupling* scheme in [14]. The main idea is to find a *K*-class discriminant rule (with classes $C_1,...,C_K$) by training all $K(K-1)/2$ possible two-class classifiers and coupling the obtained conditional membership probabilities $r_{ij} := Prob(C_i|C_i \text{ or } C_j)$ to a consistent *K*-class assignment probability. In other words, we want to find

Table 2: Top-level hierarchy of subcellular localization classes from CYGD

701	extracellular	745	transport vesicles
705	bud	750	nucleus
710	cell wall	755	mitochondria
715	cell periphery	760	peroxisome
720	plasma membrane	765	endosome
722	integral membrane/endo membranes	770	vacuole
725	cytoplasm	775	microsomes
730	cytoskeleton	780	lipid particles
735	ER	790	punctate composite
740	golgi	795	ambiguous

probabilities $p_1,...,p_K$ such that the quantities $s_{ij} := \frac{p_i}{p_1 + p_j}$ are as close as possible to the estimated r_{ij}.

The probabilities $p_1,...,p_K$ are finally found be minimizing the KL-divergence between r_{ij} and s_{ij}. This pairwise coupling approach can also be adapted for multiple class labels by way of "fractional" class assignments in the training step. If we find many samples in the training set which belong to both the classes i and j, multi-label discriminant analysis will effectively shrink the respective class centroids μ_i and μ_j towards their common mean while keeping the covariances constant, a procedure which will favor the co-prediction of classes i and j. In our experiments, we use this pairwise approach to LDA as a "building block" in the two-layered HMM for simultaneous prediction of subcellular localization and functional class. The advantage of this pairwise method over the "global" approach is that in each of the $K(K - 1)/2$ subproblems a task-specific subset of kernels is learned. Concerning the computational workload, the pairwise approach is very similar to the "global" method, since the increase of classifiers to be learned is compensated by the smaller sample size in the individual two-class problems.

Results and discussion

On the top-level hierarchy, the functional catalog provided by the *MIPS comprehensive yeast genome database* (CYGD) [15] assigns roughly 4000 yeast proteins to several functional classes listed in Table 1. Note that this classification scheme corresponds to an old version of the MIPS functional catalog, whereas the newest version, *funcat 2.0*, further splits some of these 13 classes. To allow a

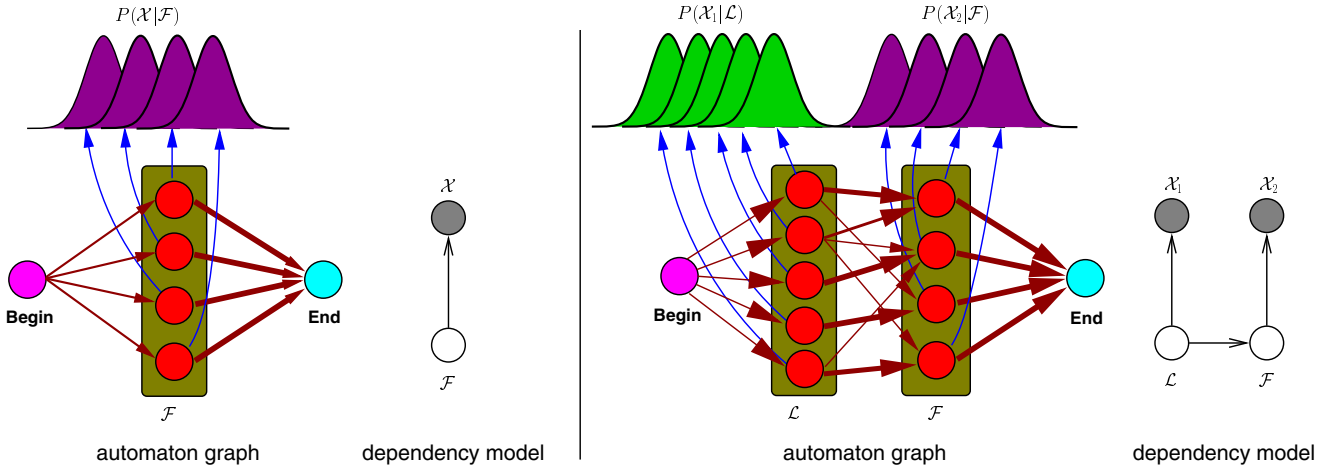

Figure 2

Graphical representation of hidden Markov models. Graphical representation of *hidden Markov models*. Left: standard 4-class discriminant analysis with 4 Gaussian emission probabilities. $\mathcal{F}$ denotes the hidden random variable "functional class", $\mathcal{X}$ represents the observed data. The nodes (red circles) represent the K values of $\mathcal{F}$ (in this example K = 4). Right: Additional layer with random variable $\mathcal{L}$, "subcellular localization class", and two conditionally independent observed variables $\mathcal{X}_1$, $\mathcal{X}_2$. The different widths of the arrows symbolize different transition probabilities.

better comparison with previous results reported in [1], however, we still use the older labeling scheme.

Kernel representation

One of the major advantages of our method is its capability of automatically extracting relevant data sources out of a large collection of kernels presented by the user. The user can, thus, collect as much information sources as possible and let the algorithm decide which to choose. Following this idea, we represent the yeast proteins by previously used kernels that extract information on different levels, like mRNA expression, protein sequences and protein-protein interactions. Moreover, we enrich this set of "basis" kernels by variants thereof resulting from nonlinear feature space mappings. We further investigated additional kernels from several publicly available microarray datasets [16-20].

The "basis" kernels introduced in [1] consist of (i) two kernels which analyze the domain structure: K_{pfdom} and an enriched variant $K_{\text{pf\_exp}}$; (ii) three diffusion kernels on interaction graphs: K_{mpi}(protein-protein interactions), K_{mgi} (genetic interactions), K_{tap} (co-participation in protein complexes); (iii) two kernels derived from cell cycle gene expression measurements: $K_{\text{exp\_d}}$ (binary) and $K_{\text{exp\_g}}$ (Gaussian); (iv) a string alignment kernel K_{SW}. From each of these 8 kernels we derive 3 additional Gaussian RBF variants by computing squared Euclidean distances D_{ij}^2 between pairs of objects (i, j) and deriving new kernels under this nonlinear feature space transform as $K_{ij}^{(l)} = \exp(-\sigma_l D_{ij}^2)$ for the three RBF variants l = 1, 2, 3.

Multilabels

Since a protein can have several functions, each protein comes with a set Y of functional class labels. Let $|Y|$ denote the cardinality of this set, i.e. the number of classes a certain protein is assigned to. For running *Algorithm 1* in the methods section below we have to translate the label sets Y_i, i = 1,...,n into membership probabilities which form the entries of the "blurred" $n \times K$ indicator matrix $\tilde{Z}$, where n is the number of proteins and K the number of classes. Since no further information is available, for the i-th protein we set $\tilde{Z}_{ik} = 1/|Y_i|$, if the k-th class is a member of the label set Y_i, and zero otherwise. In the following these membership probabilities will also be called "fractional labels".

Performance evaluation

In [1], all 13 classes were trained separately in a one-against-all manner, where a gene is treated as a member of a certain class whenever it has a positive label for that class (irrespective of other labels!). The performance of these classifiers has been evaluated in terms of *area under the ROC curves* (*auc*). Our method, on the contrary, respects the multilabel structure of the problem by explicitly exploiting co-occurrences of class labels. It uses *fractional labels* $\tilde{Z}_{ik}$, and the output is a probability vector for all classes. Thus, even in the optimal case, our classifier will assign a score of $1/|Y_i|$ to a correct class. Since the test set contains genes with *different* cardinalities of label sets, the classifier scores reside on different scales and it will be impossible to find a common threshold when computing a ROC curve.

To overcome this problem, we use two different measures: for each class C_k, $auc_0/1$ measures the area under the ROC curve only on the subset of genes which either do not belong to class C_k, or which exclusively belong to C_k. For this subset ($\approx 2/3$ of the yeast genes) we can directly compute a ROC curve, since there are no scaling problems. The measure auc_{weighted}, on the other hand, uses all test genes and rescales both the fractional label $\tilde{Z}_{ik}$ for class C_k and the corresponding probabilistic classifier score $P(C_k|\boldsymbol{x}_i)$ for the i-th protein by the label set cardinality, $\tilde{Z}'_{ik} = \tilde{Z}_{ik} \cdot |Y_i| \Rightarrow \tilde{Z}'_{ik} \in \{0,1\}$.

Figure 3 depicts the results for the enlarged kernel set consisting of the 8 "basis" kernels and 3 additional RBF kernel variants thereof. For each of the 13 classes three perform-

Table 1: Functional classes from MIPS CYGD

01	metabolism	08	cell rescue, defense
02	energy	09	interaction w/cell.envt.
03	cell cycle & DNA processing	10	cell fate
04	transcription	11	control of cell. org.
05	protein synthesis	12	transport facilitation
06	protein fate	13	others
07	cellular transp. & transp. mech.		

ance values (area under ROC curve) are shown: the result reported in [1] (depicted solely as vertical bars since no variance measurements are provided) and the two measures auc_{weighted} and $auc_0/1$ for our multilabel approach (represented as box-plots). Each of the latter two significantly outperforms the former in most classes (marked red). The measure $auc_0/1$ shows an improved median performance in *all* classes (some are probably not significant, marked orange), auc_{weighted} has worse performance in 2 classes (probably not significant, light blue). A control experiment in which we used only the 8 "basis" kernels yielded a slightly lower performance.

The improvement obtained by using the enlarged kernel set becomes more obvious when computing the *F1-measure* which is the harmonic mean of *precision* and *recall*. The latter are similar to but different from the axes of ROC curves which encode *fallout* and *recall*. Precision is the probability that a predicted category is a true category, whereas *fallout* is the probability that a true absence of a category was labeled a false positive presence. Since the definite absence of a certain protein function can be hardly validated experimentally, the estimated fallout rate will strongly depend on the actual status of experimental coverage. The precision measure, on the other hand, does not so severely suffer from this problem, since the number of present categories might be estimated more reliably even with a small number of carefully designed experiments.

To compute the *F1* statistics for a given threshold τ in eq. (9), we select for each gene the k most probable multilabels such that the sum of the k largest membership probabilities exceeds τ. We then compute *precision* and *recall* up to the rank k and combine both to the *F1-measure*. Variation of τ yields a complete precision-recall-F1-curve. With the enlarged kernel set the maximum *F1* value increases significantly, as can be seen in the two leftmost boxplots in Figure 1.

Figure 4 depicts the learned kernel weights. Each box contains 4 bins corresponding to the original kernels from [1] and three Gaussian RBF kernel variants with decreasing kernel width. It is of particular interest that a RBF variant of the genetic interaction kernel K_{mgi} attains the highest weight, whereas the original diffusion kernel K_{mgi} on the interaction graph seems to contain almost no discriminative information (consistent with [1] where K_{mgi} is the least important kernel). The reason for the improved performance of the RBF kernel variant might be the *local* nature of the Gaussian kernel function. A diffusion kernel encodes transition probabilities for a *random walk* model on a graph G. In the light of this random walk interpretation, the steep decay of the Gaussian kernel accentuates the local graph structure.

Functional classes and subcellular localization

For the yeast genome MIPS CYGD provides a classification scheme with respect to subcellular localization of the proteins, see Table 2. We combine both the functional and the localization-based scheme by way of the *hidden Markov model* (HMM) described in the methods section below. The corresponding automaton model is depicted in Figure 5. The first layer contains 20 emitting nodes corresponding to the top-level localization classes in Table 2. The transition probabilities are estimated from a training set by counting the occurrences of paths in the model. In order to highlight the essential graph structure, only the transitions with probability above 0.1 are shown. Note that several localization nodes have dominant transitions to only one or two functional nodes, see e.g. node "750" (*nucleus*) which has a strong prior for class "04" (*transcription*), the pair "730" (*cytoskeleton*) and "03" (*cell cycle & DNA processing*), or "745" (*transport vesicles*) which strongly votes for functional class "07" (*cellular transport & transport mechanism*).

In order to evaluate the possible advantages of the combined localization-function classifier, we again conducted a cross-validation experiment in which both sets of labels were predicted. According to the results summarized in the left panel of Figure 1, the inclusion of the prediction step for the subcellular localization of a protein indeed improves the prediction of its function.

Towards a "local" model: pairwise kernel weights

While all previous models find a common set of kernels for all classes, the pairwise coupling approach described in the methods section couples pairwise classifiers which find individual kernel weights that are optimal for the "local" problem of separating only two classes. These pairwise classifiers can again learn class correlations induced by objects with multiple labels.

The rightmost boxplot in the left panel of Figure 1 shows that this "local" model significantly improves the predictive power of the HMM-based classifier. The right panel depicts the evolution of precision, recall and F1 under variation of the threshold τ in eq. (9).

Figure 6 shows two examples of the learned kernel weights in the 78 individual two-class models for the 13 functional classes which nicely demonstrate the adaptiveness of the pairwise approach: while for the separation of the classes *metabolism* and *control of cellular organization* the combination of Smith-Waterman sequence alignment kernels ("SW") and protein interaction kernels ("mpi") have a main role, the separation of classes *energy* and *protein synthesis* is dominated by gene expression information ("exp") and protein domain structure ("pfam"). A closer analysis of all 78 classifiers shows some general trends, for

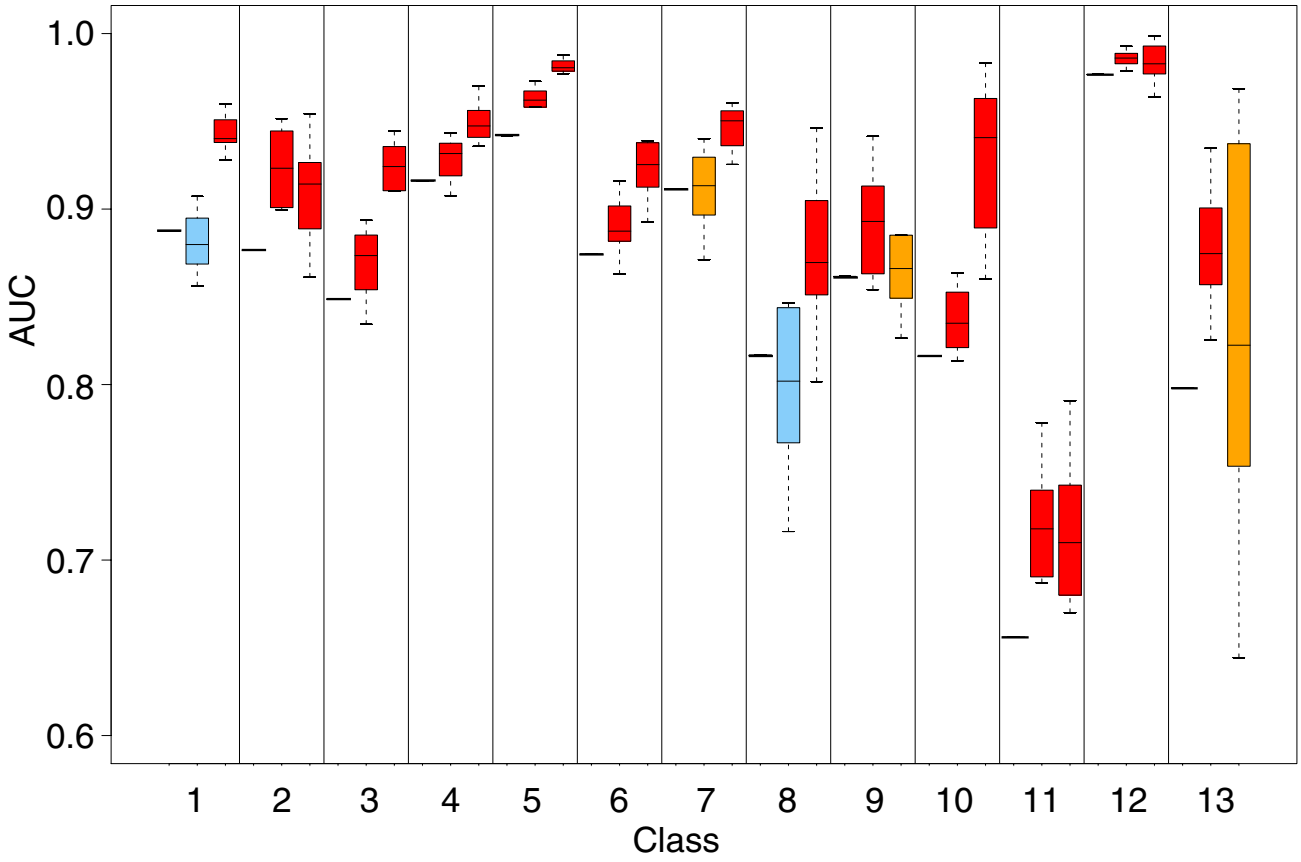

Figure 3
Performance evaluation. Performance Evaluation for the 13 functional classes of yeast proteins. From left to right: results from [1], $auc_{weighted}$ and $auc_0/1$. Red: significant improvements; orange: improvements of unclear significance; light blue: worse performance (unclear significance).

instance the importance of gene expression kernels whenever one of the classes *energy* or *protein synthesis* is involved.

Conclusion

While kernel-based classifiers have been successfully applied to a variety of prediction tasks, their main drawback is the lacking interpretability of the decision functions. One attempt to overcome this shortcoming is to find a weighted combination of multiple kernels, each of which represents a different type of measurement. The idea is that sparse kernel combinations allow the user to identify the relevant influence factors for a given task. In this work, the problem of learning such sparse kernel combinations has been addressed by reformulating classification as an indicator regression problem using adaptive ridge penalties. While the standard adaptive ridge model presented in [11] selects individual input features, our extensions concerning *weight sharing* and *kernelization* lead to a nonlinear model that finds sparse combinations of *kernel matrices*. A probabilistic treatment of multiple labels allows us to apply the classifier to tasks in which the input objects can belong to more than one category. The effect of multilabels can be intuitively understood as shrinking the centroids of classes which share many multilabeled objects towards their average centroid, thus favoring co-prediction of these classes.

The method has been applied to the problem of predicting the function of yeast proteins which defines a classical multilabel setting. From the experiments we conclude that our model compares favorably to the approach in [1]. Two aspects seem to be of particular importance: on the modeling side, our approach directly exploits the multilabel structure of the problem, rather than ignoring class correlations.

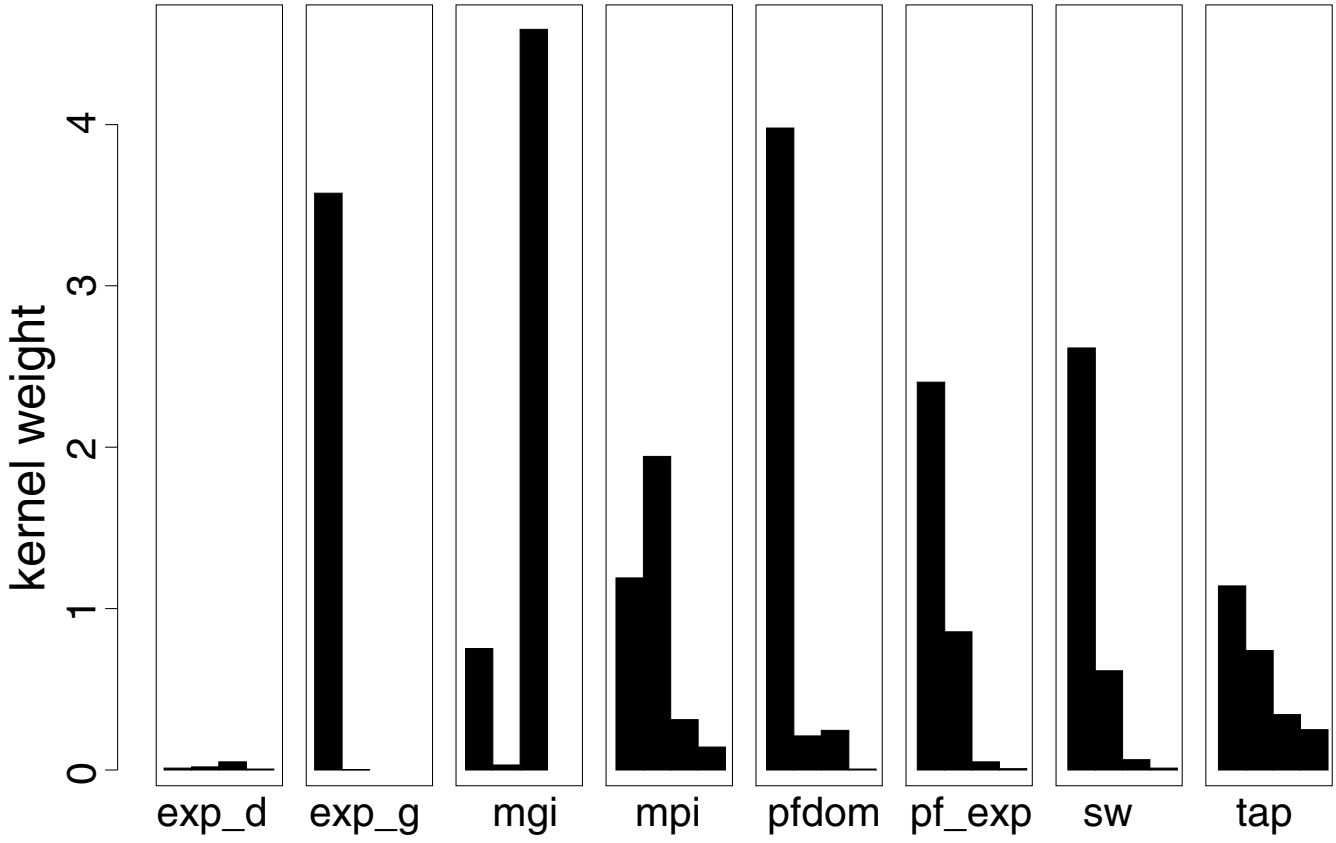

Figure 4
Learned kernel weights. Learned kernel weights. Each box contains one of the 8 "basis" kernels and three Gaussian RBF variants (from left to right).

Concerning the computational aspects, the efficiency of the method allows us to easily enlarge the set of kernels: as long as one single matrix can be hold in the main memory, the algorithm is highly efficient. For yeast proteins, the use of additional kernels has e.g. lead to the insight that genetic interactions are highly discriminative for functional predictions.

Aiming at a still higher classification rate and at more detailed information about the relevance of kernels, we have introduced two further modifications: extending the multilabel classifier to a two-layer *hidden Markov model* (HMM) allows us to combine two different labeling schemes. Multilabel prediction in the HMM naturally translates to reconstructing multiple paths through a graph. It could be shown that the prediction of the subcellular localization of a protein in the first layer helps to identify its functional class in the second layer, the reason for this improvement being the strong correlation of nodes in both layers. Localization in the *transport vesicles*, for instance, gives a strong prior for having a role in the functional class *cellular transport & transport mechanism*.

The second modification concerns the transition from a single "global" prediction model to several "local" models which focus on the separation of *pairs of classes* only. The estimated pairwise membership probabilities are coupled to a consistent set of assignment probabilities over all classes. Using the pairwise classifiers as "building blocks" in the HMM offers the advantage of increased adaptiveness, since the kernel weighting can focus on the individual requirements for separating one class from another. This approach leads not only to a significantly increased classification performance, but it also gives a much more detailed picture on the importance of different data sources for predicting protein function.

Authors' contributions

Both authors developed methods, performed comparisons of methods, and wrote the manuscript.

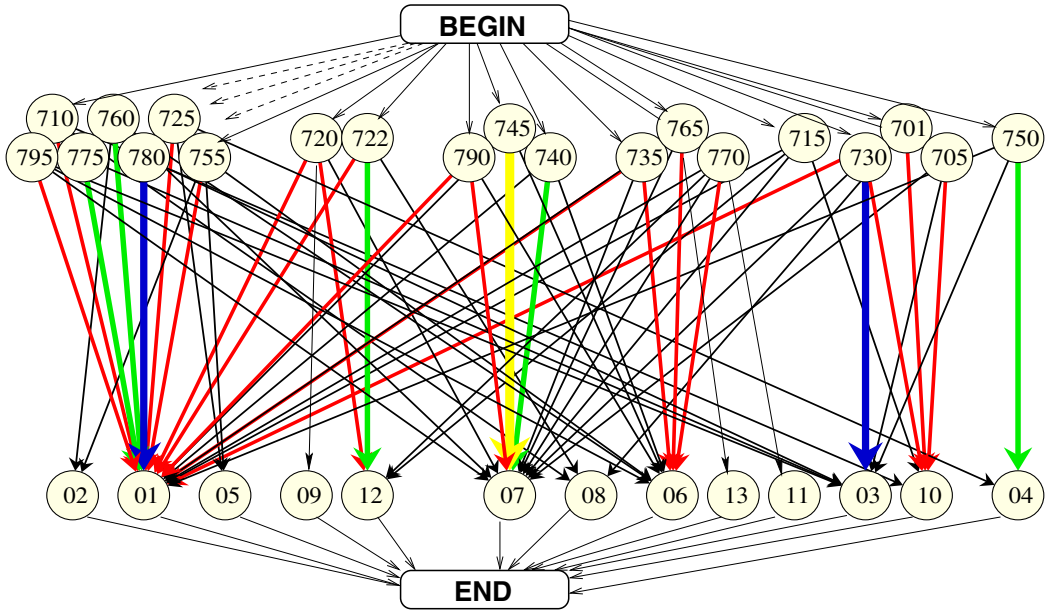

Figure 5
The hidden Markov model. The hidden Markov model. Combined graph for subcellular location classes (upper layer) and functional classes (lower layer). Joint predictions of these two entities means finding (multiple) paths through the graph from *begin* to *end*. The nodes in the two layers encode the values of the hidden random variables *location class* and *functional class*, see also Figure 2. The arrows between the nodes encode "transition" probabilities which are estimated by frequency counts on a training set. For highlighting the main structure of this graph, only transition probabilities with $p > 0.1$ are shown. Width and color of the arrows encode these probabilities: > 0.8 yellow, > 0.6 blue, > 0.4 green, > 0.2 red. For instance, the yellow arrow between the nodes "745" and "07" means that more than 80% of the proteins with subcellular localization *transport vesicles* belong to the functional class *cellular transport & transport mechanism*.

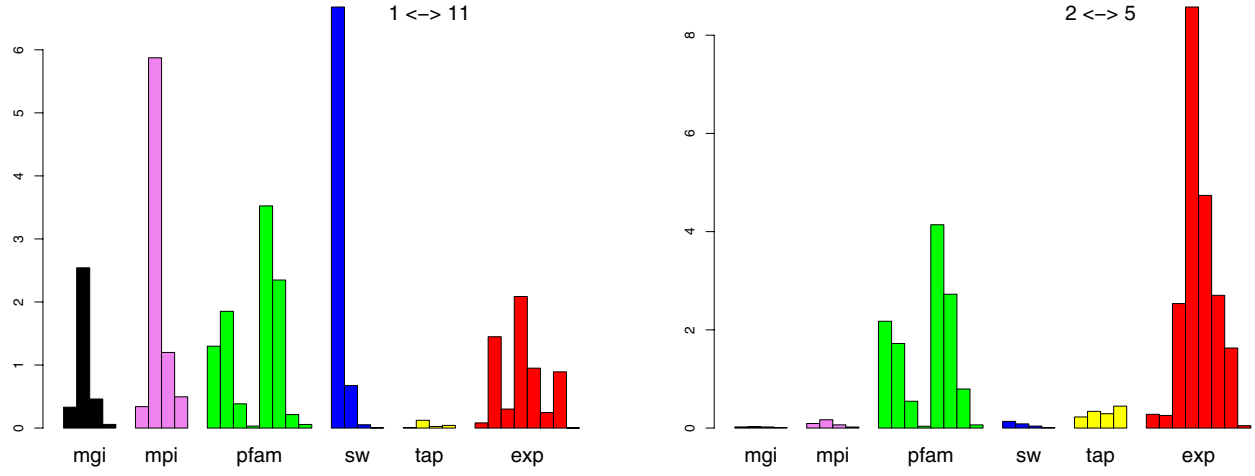

Figure 6
Kernel weights for the pairwise model. Kernel weights for the pairwise model. Left: separating classes *metabolism* and *control of cellular organization*. Right: classes *energy* and *protein synthesis*. The kernels are arranged in groups according to their origin: genetic interaction (mgi), prot.-prot. interaction (mpi), domain structure (pfam) string alignments (SW), protein complexes (tap) and gene expression (exp). The 8 gene expression RBF kernels represent the data in [16-20] and three RBF kernel variants of the data in [21] (left to right).

Acknowledgements

This work was partially funded by ETH grant TH-5/04-3.

This article has been published as part of *BMC Bioinformatics* Volume 8, Supplement 2, 2007: Probabilistic Modeling and Machine Learning in Structural and Systems Biology. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/8?issue=S2.

References

1. Lanckriet G, Deng M, Cristianini N, Jordan M, Noble W: **Kernel-based data fusion and its application to protein function prediction in yeast.** *Pacific Symposium on Biocomputing* 2004:300-311.
2. Sonnenburg S, Rätsch G, Schäfer C: **A general and efficient multiple kernel learning algorithm.** In *NIPS 18* Edited by: *Weiss Y, Schölkopf B, Platt J. MIT Press*; 2006.
3. Bach F, Lanckriet G, Jordan M: **Multiple kernel learning, conic duality, and the SMO algorithm.** *21st Intern Conference on Machine Learning* 2004.
4. Crammer K, Keshet J, Singer Y: **Kernel design using boosting.** In *NIPS 15 MIT Press*; 2002:537-544.
5. Centeno TP, Lawrence N: **Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis.** *Journal of Machine Learning Research* 2006, **7(455-49):**.
6. Hastie T, Tibshirani R: **Discriminant analysis by Gaussian mixtures.** *J Royal Statistical Society series B* 1996, **58:**158-176.
7. Dempster A, Laird N, Rubin D: **Maximum likelihood from incomplete data via the EM algorithm.** *Journal of the Royal Statistical Society series B* 1977, **39:**1-38.
8. Kumar N, Andreou A: **Generalization of linear discriminant analysis in a maximum likelihood framework.** *Proc Joint Meeting of the American Statistical Association* 1996.
9. Hastie T, Tibshirani R, Buja A: **Flexible discriminant analysis by optimal scoring.** *J American Statistical Association* 1994, **89:**1255-1270.
10. MacKay D: **Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks.** *Network: Computation in Neural Systems* 1995, **6:**469-505.
11. Grandvalet Y: **Least absolute shrinkage is equivalent to quadratic penalization.** In *ICANN'98* Edited by: *Niklasson L, Bodén M, Ziemske T. Springer*; 1998:201-206.
12. Dubrulle A: **Retooling the method of block conjugate gradients.** *Electronic Transactions on Numerical Analysis* 2001, **12:**216-233.
13. Roth V, Steinhage V: **Nonlinear discriminant analysis using kernel functions.** In *Advances in Neural Information Processing Systems 12* Edited by: *Solla S, Leen T, Müller KR. MIT Press*; 1999:568-574.
14. Hastie T, Tibshirani R: **Classification by pairwise coupling.** In *Advances in Neural Information Processing Systems Volume 10*. Edited by: *Jordan MI, Kearns MJ, Solla SA. The MIT Press*; 1998.
15. Güldener U, Münsterkötter M, Kastenmüller G, Strack N, van Helden CJ, Lemer , Richelles J, Wodak S, García-Martínez J, Pérez-Ortín J, Michael H, Kaps A, Talla E, Dujon B, André B, Souciet J, Montigny JD, Bon E, Gaillardin C, Mewes H: **CYGD: the Comprehensive Yeast Genome Database.** *Nucleic Acids Research* 2005:D364-348.
16. Hughes T, Marton M, Jones A, Roberts C, Stoughton R, Armour C, Bennett H, Coffey E, Dai H, He Y, Kidd M, King A, Meyer M, Slade D, Lum P, Stepaniants S, Shoemaker D, Gachotte D, Chakrabortty K, Simon J, Bard M, Friend S: **Functional discovery via a compendium of expression profiles.** *Cell* 2000, **102:**109-26.
17. Gasch A, Huang M, Metzner S, Botstein D, Elledge S, Brown P: **Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p.** *Mol Biol Cell* 2001, **12(10):**2987-3003.
18. Yoshimoto H, Saltsman K, Gasch A, Li H, Ogawa N, Botstein D, Brown P, Cyert M: **Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae.** *J Biol Chem* 2002, **277(34):**31079-88.
19. Yvert G, Brem R, Whittle J, Akey J, Foss E, Smith E, Mackelprang R, Kruglyak L: **Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors.** *Nature Genet* 2003, **35:**57-64.
20. Brauer M, Saldanha A, Dolinski K, Botstein D: **Homeostatic adjustment and metabolic remodeling in glucose-limited yeast cultures.** *Mol Biol Cell* 2005, **16(5):**2503-17.
21. Spellman P, Sherlock G, Zhang M, Iyer V, Anders K, Eisen M, Brown P, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.** *Mol Biol Cell* 1998, **9(12):**3273-97.