Technical Report No. 0606, Department of Statistics, University of Toronto

Gene Function Classification Using Bayesian Models with Hierarchy-Based Priors

Babak Shahbaba
Dept. of Public Health Sciences, Biostatistics
University of Toronto
Toronto, Ontario, Canada
babak@stat.utoronto.ca

Radford M. Neal
Dept. of Statistics and Dept. of Computer Science
University of Toronto
Toronto, Ontario, Canada
radford@stat.utoronto.ca

5 May 2006

Abstract. We investigate the application of hierarchical classification schemes to the annotation of gene function based on several characteristics of protein sequences including phylogenic descriptors, sequence based attributes, and predicted secondary structure. We discuss three Bayesian models and compare their performance in terms of predictive accuracy. These models are the ordinary multinomial logit (MNL) model, a hierarchical model based on a set of nested MNL models, and a MNL model with a prior that introduces correlations between the parameters for classes that are nearby in the hierarchy. We also provide a new scheme for combining different sources of information. We use these models to predict the functional class of Open Reading Frames (ORFs) from the *E. coli* genome. The results from all three models show substantial improvement over previous methods, which were based on the C5 algorithm. The MNL model using a prior based on the hierarchy outperforms both the non-hierarchical MNL model and the nested MNL model. In contrast to previous attempts at combining these sources of information, our approach results in a higher accuracy rate when compared to models that use each data source alone. Together, these results show that gene function can be predicted with higher accuracy than previously achieved, using Bayesian models that incorporate suitable prior information.

1 Introduction

Annotating genes with respect to the function of their proteins is essential for understanding the wealth of genomic information now available. A direct approach to identifying gene function is to eliminate or inhibit expression of a gene and observe any alteration in the phenotype. However, analysis of all genes for all possible functions is not possible at present. Statistical methods have therefore been employed for this purpose. One statistical approach attempts to predict the functional class of a gene based on similar sequences for which the function is known. The similarity measures used for this task are produced by computer algorithms that compare the sequence of interest against all other sequences with known function. The most commonly used algorithms are BLAST (Altschul *et al.* 1997) and FASTA (Pearson and Lipman 1988).

A problem with using such similarity measures is that a gene's function cannot be predicted when no homologous gene of known function exists. To improve the quality and coverage of prediction, other sources of information can be used. For example, King *et al.* (2001) used a variety of protein sequence descriptors, such as residue frequency and the predicted secondary structure (the structure of hydrogen binding between different residues within a single polypeptide chain). DeRisi *et al.* (1997), Eisen *et al.* (1998) and Brown *et al.* (2000) used gene expression data, on the assumption that similarly expressed genes are likely to have similar function. Marcotte *et al.* (1999) recommended an alternative sequence-based approach, called the "Rosetta stone" method, which regards two genes as similar if they are together in another genome. Deng *et al.* (2003) predict the function of genes from their network of physical interactions. To address some of the problems associated with similarity-based methods, such as their non-robustness to variable mutation rates (Eisen 1998; Rost 2002), annotation of protein sequences using phylogenetic information has been suggested by some authors (e.g., Eisen *et al.* 1998; Sjölander 2004; Engelhardt *et al.* 2005). In this approach, the evolutionary history of a specific protein, captured by a phylogenetic tree, is used for annotating that protein (Eisen *et al.* 1998).

The above-mentioned sources of data can be used separately, or as proposed by several authors (e.g., King *et al.* 2001; Pavlidis and Weston 2001; Deng *et al.* 2004), they can be combined within a predictive model. A variety of statistical and machine learning techniques for making such predictions have been used in functional genomics. These include neighbourhood-count methods (Schoikowski *et al.* 2000), support vector machines (Brown *et al.* 2000), decision tree models (King *et al.* 2001), and Markov random fields (Deng *et al.* 2003). A common feature of these models is that they treat classes as unrelated entities without any specific structure.

The assumption of unrelated classes is not always realistic. As argued by Rison *et al.* (2000), in order to understand the overall mechanism of the whole genome, the functional classes of genes need to be organized according to the biological processes they perform. For this purpose, many functional classification schemes have been proposed for gene products. The first such scheme was recommended by Riley (1993) to catalogue the proteins of *Escherichia coli*. Since then, there have been many attempts to provide a standardized functional annotation scheme with terms that are not limited to certain types of proteins or to specific species. These schemes usually have a hierarchical structure, which starts with very general classes and becomes more specific in lower levels of the hierarchy. In some classification hierarchies, such as the Enzyme Commission (EC) scheme (IUBMB 1992), levels have semantic values (Rison *et al.* 2000). For example, the first level of the EC scheme represents the major activities of enzyme like "transferaces" or "hydrolases". In some other schemes, like the ones considered here, the levels do not have any uniform meaning. Instead, each division is specific to the parent nodes. For instance, if the parent includes "metabolism" functions, the children nodes could be the metabolism of "large" or "small" molecules. Rison *et al.* (2000) surveyed a number of these structures and compared them with respect to their resolution (total number of function nodes), depth (potential of the scheme for division into subsets) and breadth (number of nodes at the top level).

All these hierarchies provide additional information that can be incorporated into the classification model. For example, King *et al.* (2001) attempted to use the additional information from the hierarchical structure of functional classes in *E. coli* by simply using different decision tree models for each level of the hierarchy. Clare and King (2003) expanded this approach by modifying the

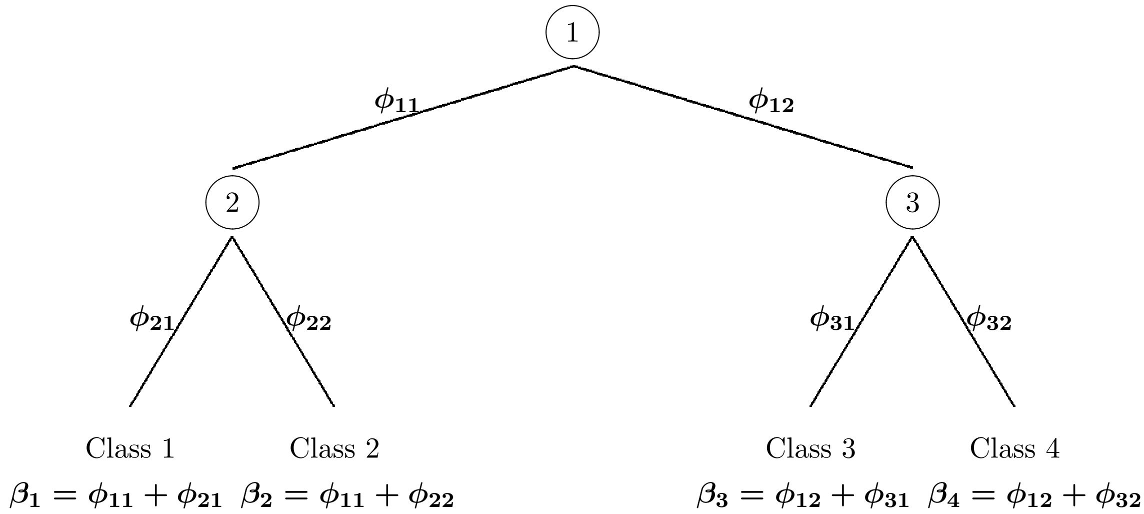

Figure 1: The corMNL model for a simple hierarchy. The coefficient parameter for each class is a sum of parameters at different levels of the hierarchy

original decision tree model so that the assignment of a functional class to a node in the decision tree implied membership of all its parent classes. They evaluated this method based on *Saccharomyces cerevisiae* data and found that the modified version is sometimes better than the non-hierarchical model and sometimes worse.

In a previous paper (Shahbaba and Neal 2005), we introduced an alternative Bayesian framework for modelling hierarchical classes. This method uses a Bayesian form of the multinomial logit model, with a prior that introduces correlations between the parameters for classes that are nearby in the tree. We also discussed an alternative hierarchical model that uses the hierarchy to define a set of nested multinomial logit models. In this paper, we apply these methods to the gene function classification problem.

The rest of this paper is organized as follows. In the next section, we explain our general method using a simple hierarchy for illustration. In section 3, we describe the dataset we used to test our approach. The details of the models we used and their implementation are in sections 4 and 5. The results of our analysis are presented in section 6. Section 7 is devoted to discussion, limitations of the proposed method, and future directions.

2 Methodology

When classes in a classification problem are unrelated, a simple multinomial logit (MNL) model may be appropriate. Consider a classification problem in which we have observed data for n cases, $(x^{(1)}, y^{(1)})$, ...,$(x^{(n)}, y^{(n)})$, where $x^{(i)} = x_1^{(i)}, ..., x_p^{(i)}$ is the vector of p covariates (features) for case i, and $y^{(i)}$ is the associated class. The following is the MNL model, which is also known as "softmax" in the machine learning literature:

$$P(y = j | x, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad = \quad \frac{\exp(\alpha_j + x\boldsymbol{\beta}_j)}{\sum_{j'=1}^{c} \exp(\alpha_{j'} + x\boldsymbol{\beta}_{j'})}$$

Here, c is the number of classes. For each class, j, there is an intercept α_j and a vector of p unknown parameters $\boldsymbol{\beta}_j$. The inner product of these parameters with the covariate vector is shown as $x\boldsymbol{\beta}_j$. The entire set of regression coefficients $\boldsymbol{\beta} = \boldsymbol{\beta_1, ..., \beta_c}$ can be presented as a $p \times c$ matrix. This representation is redundant, since one of the $\boldsymbol{\beta}_j$'s can be set to zero without changing the set of relationships expressible with the model. In methods based on maximum likelihood estimation, it is common to set either β_1 or β_c to zero. However, in a Bayesian framework, the redundant representation is preferred, since removing this redundancy would make it difficult to specify a prior that treats all classes symmetrically. Priors such as the following are typically used:

$$
\begin{aligned}
\alpha_j|\eta &\sim N(0, \eta^2) \\
\beta_{jl}|\tau &\sim N(0, \tau^2) \\
log(\eta) &\sim N(v, V^2) \\
log(\tau) &\sim N(w, W^2)
\end{aligned}
$$

where $j = 1, ..., c$ and $l = 1, ..., p$. Here and later, independence is assumed unless conditioning variables are shown.

For problems such as gene function classification, the assumption of unrelated classes does not always hold. In many cases, classes have a hierarchical structure. The importance of using the hierarchy in classifiers has been emphasized by many authors (e.g., Sattath and Tversky 1977; Fox 1997; Koller and Sahami 1997). One approach for modelling hierarchical classes is to decompose the classification model into nested models (e.g., logistic or MNL). For hierarchical classification problems with simple binary partitions, Fox (1997) suggested using successive logistic models for each binary class. In Figure 1 below, for example, these partitions are {12, 34}, {1, 2}, and {3, 4}. The resulting nested binary models are statistically independent. The likelihood can therefore be written as the product of the likelihoods for each of the binary models. For example, in Figure 1 we have

$$
P(y = 1|x) \quad = \quad P(y \in \{1, 2\}|x) \times P(y \in \{1\}|y \in \{1, 2\}, x)
$$

Restriction to binary models is unnecessary. At each level, classes can be divided into more than two subsets and MNL can be used instead of logistic regression. We refer to Bayesian models in which the tree structure is used to define a set of nested MNL models as treeMNL. Consider a parent node, m, with c_m child nodes, representing sets of classes S_k, for $k = 1, ..., c_m$. The portion of the treeMNL model for this node has the form:

$$
\begin{aligned}
P(y \in S_k|x, \boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) &= \frac{\exp(\alpha_{mk} + x\boldsymbol{\beta}_{mk})}{\sum_{k'=1}^{c_m} \exp(\alpha_{mk'} + x\boldsymbol{\beta}_{mk'})} \\
\alpha_{mk}|\eta_m &\sim N(0, \eta_m^2) \\
\beta_{mkl}|\tau_m &\sim N(0, \tau_m^2) \\
log(\eta_m) &\sim N(v_m, V_m^2) \\
log(\tau_m) &\sim N(w_m, W_m^2)
\end{aligned}
$$

We calculate the probability of each end node, j, by multiplying the probabilities of all intermediate nodes leading to j.

In contrast to this treeMNL model, Mitchell (1998) showed that the hierarchical naive Bayes classifier is equivalent to the standard non-hierarchical classifier when probabilities are estimated by maximum likelihood (ML). To improve the hierarchical naive Bayes model, McCallum *et al.* (1998) suggested smoothing the parameter estimate for each end node by shrinking its ML estimate towards the estimates for all its ancestors in the hierarchy. More recently, new hierarchical classification models based on Support Vector Machines (SVM) have been proposed (Dumais and Chen 2000; Dekel *et al.* 2004; Cai and Hoffmann 2004; Tsochantaridis *et al.* 2004; Cesa-Bianchi *et al.* 2006). An important aspect of these models is the use of a modified loss function which reflects the taxonomy of classes.

For modelling hierarchical classes, we introduced a new method which has a MNL form with a prior that introduces correlations between the parameters of nearby classes (Shahbaba and Neal 2005). Our model includes a vector of parameters for each branch in the hierarchy. We assign objects to one of the end nodes using a MNL model whose regression coefficients for class j are represented by the sum of the parameters for all the branches leading to that class. Sharing of common parameters (from common branches) introduces prior correlations between the parameters of nearby classes in the hierarchy. This way, we can better handle situations in which these classes are hard to distinguish. Our simulation results show that when the hierarchy actually provides information about how distinguishable classes are, our model, which we call corMNL, outperforms both the non-hierarchical MNL model and the nested treeMNL model (Shahbaba and Neal 2005). When an inappropriate hierarchy is used, the penalty for corMNL is significantly less than for the alternative treeMNL model.

Consider Figure 1, which shows a hierarchical classification problem with four classes. Parameter vectors denoted as $\boldsymbol{\phi_{11}}$ and $\boldsymbol{\phi_{12}}$ are associated with branches in the first level, and $\boldsymbol{\phi_{21}}$, $\boldsymbol{\phi_{22}}$, $\boldsymbol{\phi_{31}}$ and $\boldsymbol{\phi_{32}}$ with branches in the second level. We assign objects to one of the end nodes using a MNL model whose regression coefficients for a class are represented by the sum of parameters on all the branches leading to that class. In Figure 1, these coefficients are $\boldsymbol{\beta_1 = \phi_{11} + \phi_{21}}$, $\boldsymbol{\beta_2 = \phi_{11} + \phi_{22}}$, $\boldsymbol{\beta_3 = \phi_{12} + \phi_{31}}$ and $\boldsymbol{\beta_4 = \phi_{12} + \phi_{32}}$ for classes $1, 2, 3$ and 4 respectively. Sharing the common terms, $\boldsymbol{\phi_{11}}$ and $\boldsymbol{\phi_{12}}$, introduces prior correlation between the parameters of nearby classes in the hierarchy. Note that the intercept parameters, α_j, are not treated hierarchically.

In our model, $\boldsymbol{\phi}$'s are vectors with the same size as $\boldsymbol{\beta}$'s. We assume that, conditional on higher level hyperparameters, all the components of the $\boldsymbol{\phi}$'s are independent, and have normal prior distributions with zero mean. The variances of these components are regarded as hyperparameters, which control the magnitudes of coefficients. When a part of the hierarchy is irrelevant, we hope the posterior distribution of its corresponding hyperparameter will be concentrated near zero, so that the parameters it controls will also be close to zero. In detail, the simplest form of prior for a corMNL model is as follows:

$$
\begin{aligned}
\alpha_j | \eta &\sim N(0, \eta^2) \\
\phi_{mkl} | \tau_m &\sim N(0, \tau_m^2) \\
log(\eta) &\sim N(v, V^2) \\
log(\tau_m) &\sim N(w_m, W_m^2)
\end{aligned}
$$

Here, ϕ_{mkl} refers to the parameter related to covariate x_l and branch k of node m. For gene function classification, we used a more elaborate prior, discussed in section 4.

3 Data

We used our model to predict the functional class of Open Reading Frames (ORFs) from the *E. coli* genome. *E. coli* is a good organism for testing our method since many of its gene functions have been identified through direct experiments. We used the pre-processed data provided by King *et al.* (2001). This dataset contains 4289 ORFs identified by Blattner *et al.* (1997). Only 2122 of these ORFs, for which the function is known, are used in our analysis. The functional hierarchy for these proteins is provided by Riley and Labedan (1996). This hierarchy has three levels with the most general classes at level 1 and the most specific classes at level 3. For example, lipoate-protein ligase A (lplA) belongs to class 'Macromolecule metabolism' at level 1, to class 'Macromolecule synthesis, modification' at level 2, and to class 'Lipoprotein' at level 3. After excluding categories 0 and 7 at level 1, the data we used had 6 level 1 categories, 20 level 2 categories, and 146 level 3 categories.

It is worthwhile mentioning that since 2001 the function of many new genes have been determined by direct experiment. However, we use the same dataset as King *et al.* (2001), with the same split of data into the training set (1410 ORFs) and test set (712 ORFs), in order to produce comparable results. King *et al.* (2001) further divided the training set into two subsets and used one subset as validation data to select a subset of rules from those produced by the C5 algorithm based on the other part of the training set. Our Bayesian methods do not require a validation set, so we did not subdivided the training set.

The covariates are based on three different sources of information: phylogenic descriptors, sequence based attributes, and predicted secondary structure. Following King *et al.* (2001), we refer to these three sources of data as SIM, SEQ and STR respectively. Attributes in SEQ are essentially based on the composition of singlets and pairs of residues in a sequence. There are 933 such attributes. Information in SIM and STR is derived based on a PSI-BLAST (position-specific iterative BLAST) search with parameters $e = 10$, $h = 0.0005$, $j = 20$ from NRProt 05/10/99 database. King *et al.* (2001) used the Inductive Logic Programming (ILP) algorithm known as Warmr (Dehaspe *et al.* 1998) to produce binary attributes based on the identified frequent patterns (1 if the pattern is present and 0 otherwise) in SIM and STR data. There are 13799 such attributes generated for SIM and 18342 attributes for STR.

4 Models

We first used our models to predict gene function using each data source (SIM, STR and SEQ) separately. Since the numbers of covariates in these datasets are large, we applied principal Component Analysis (PCA). Prior to applying PCA, the variables were centred to have mean zero, but they were not rescaled to have variance one. We selected the first p components with the highest eigenvalues. The cutt-off, p, was set based on the plot of eigenvalues against PCs (i.e., the scree plot). Since there was not a clear cut-off point at which the magnitude of eigenvalues drops sharply, the plots could only help us to narrow down the appropriate values for p. We decided to choose a value at the upper end of the range suggested by the scree plot. We selected 100 components from SEQ, 100 components from STR, and 150 components from SIM.

Principal components are derived solely based on the input space and do not necessarily provide the best set of variables for predicting the response variable. In order to find the relevant variables (among the principal components) for the classification task, we use the Automatic Relevance Determination (ARD) method suggested by Neal (1996). ARD employs a hierarchical prior to determine how relevant each covariate is to classification. In the MNL model, for example, one hyperparameter, σ_l, is used to control the variance of all coefficients, β_{jl} $(j = 1, ..., c)$, for covariate x_l. If a covariate is irrelevant, its hyperparameter will tend to be small, forcing the coefficients for that covariate to be near zero. We also use a set of hyperparameters, τ_j, to control the magnitude of the β's for each class. We use a third hypeparameter, ξ, to control the overall magnitude of all β's. This way, σ_l controls the relevance of covariate x_l compared to other covariates, τ_j, controls the usefulness of covariates in identifying class j, and ξ controls the overall usefulness of all covariates in separating all classes. The standard deviation of β_{jl} is therefore equal to $\xi\tau_j\sigma_l$.

For the MNL model we used the following priors:

$$\begin{aligned}
\alpha_j|\eta &\sim N(0, \eta^2) \\
\beta_{jl}|\xi, \sigma_l, \tau &\sim N(0, \xi^2\tau_j^2\sigma_l^2) \\
log(\eta) &\sim N(0, 1) \\
log(\xi) &\sim N(-3, 2^2) \\
log(\tau_j) &\sim N(-1, 0.5^2) \\
log(\sigma_l) &\sim N(0, 0.3^2)
\end{aligned}$$

Since the task of variable selection is mainly performed through PCA, the ARD hyperparameters, σ's, are given priors with fairly small standard deviation. The priors for τ's are set such that both small values (i.e., close to zero) and large values (i.e., close to 1) are possible. The overall scale of these hyperparameters is controlled by ξ, which has a broader prior. Note that since these hyperparameters are used only in the combination $\xi\tau_j\sigma_l$, only the sum of the means for $log(\xi)$, $log(\tau_j)$, and $log(\sigma_l)$ really matters.

Similar priors are used for the parameters of treeMNL and corMNL. For these two models, we again used one hyperparamter, σ_l, to control all parameters (β's in treeMNL, ϕ's in corMNL) related to covariate x_l. We also used one scale parameter τ_k for all parameters related to branch k of the hierarchy. The overall scale of all parameters is controlled by one hyperparameter ξ.

This setting of priors is different from what we used in a previous paper (Shahbaba and Neal 2005), where we used one hyperparameter to control all the coefficients (regardless of their corresponding class) in the MNL model, and we used one hyperparameter to control the parameters of all the branches that share the same node in treeMNL and corMNL. The scheme used in this paper provides an additional flexibility to control β's. In this paper, the hyperparameters are given log-normal distributions instead of the gamma distributions used in Shahbaba and Neal (2005). Using gamma priors has the advantage of conjugacy and, therefore, easier MCMC sampling. However, we prefer log-normal distribution since they are more convenient for formalizing our prior beliefs.

5 Implementation

These models are implemented using Markov chain Monte Carlo (Neal 1993). We use Hamiltonian dynamics (Neal 1993) for sampling from the posterior distribution of coefficients (with hyperparameters temporarily fixed). The number of leapfrog steps was set to 50. The stepsizes were set dynamically at each iteration, based on the current values of the hyperparameters (Neal 1996). In the MNL and corMNL models, new values are proposed for all regression parameters simultaneously. Nested MNL models in treeMNL are updated separately since they are regarded as independent models. The coefficient parameters within each nested model, however, are updated at the same time.

We use single-variable slice sampling (Neal 2003) to sample from the posterior distribution of hyperparameters. At each iteration, we use the "stepping out" procedure to find the interval around the current point and the "shrinkage" procedure for sampling from the interval. The initial values of the ARD hyperprameters, σ's, were set to the inverse of the standard deviation of their corresponding covariates. The initial values of τ's and ξ were set to 1.

Convergence of the Markov chain simulations was assessed from trace plots of hyperparameters. We ran each chain for 5000 iterations, of which the first 1000 were discarded. Simulating the Markov chain for 10 iterations took about 2 minutes for MNL, 1 minute for treeMNL, and 3 minutes for corMNL, using a MATLAB implementation on an UltraSPARC III machine.

6 Results

Table 1 compares the three models with respect to their accuracy of prediction at each level of the hierarchy. In this table, level 1 corresponds to the top level of the hierarchy, while level 3 refers to the most detailed classes (i.e., the end nodes). For level 3, we use a simple 0/1 loss function and minimize the expected loss by assigning each test case to the end node with the highest posterior predictive probability. We could use the same predictions for measuring the accuracy at levels 1 and 2, however, to improve accuracy, we instead make predictions based on the total posterior predictive probability of nodes at levels 1 and level 2.

To provide a baseline for interpreting the results, for each task we present the performance of a model that ignores the covariates and simply assigns genes to the most common category at the given level in the training set.

As we can see in Table 1, corMNL outperforms all other models. For the SEQ dataset, MNL performs better than treeMNL. Compared to MNL, the corMNL model achieves a slightly better accuracy at level 3 and more marked improvements at level 1 and level 2. For the STR dataset, both hierarchical models (i.e., treeMNL and corMNL) outperform the non-hierarchical MNL. For this dataset, corMNL has a slightly better performance than treeMNL. For the SIM dataset, the advantage of using the corMNL model is more apparent in the first and second levels.

King *et al.* (2001) used a decision tree model based on the C5 algorithm for analysing these datasets. They selected sets of rules that had an accuracy of at least 50% with the coverage of at least two correct examples in the validation set. In Table 2, we compare the accuracy of our models to those of King *et al.* (2001). In order to make the results comparable, we used the same coverage values as they used. Coverage is defined as the percentage of test cases for which we make a confident prediction. In a decision tree model, these test cases can be chosen by selecting rules that lead to a specific class with high confidence. For our models, we base confidence on posterior predictive probability. We rank the test set based on these probabilities and for a coverage of g, we

Accuracy (%)	SEQ			STR			SIM		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
Baseline	42.56	21.21	8.15	42.56	21.21	8.15	42.56	21.21	8.15
MNL	60.25	33.99	20.93	50.98	25.14	15.87	69.10	45.79	30.76
treeMNL	59.27	34.13	18.26	52.67	27.39	16.29	67.70	45.93	30.34
corMNL	**61.10**	**35.96**	**21.21**	**52.81**	**27.95**	**16.71**	**70.51**	**47.19**	**30.90**

Table 1: Comparison of models based on their predictive accuracy (%) using each data source separately.

Accuracy (%)	SEQ			STR			SIM		
	Level 1 (20)	Level 2 (18)	Level 3 (4)	Level 1 (10)	Level 2 (1)	Level 3 (5)	Level 1 (29)	Level 2 (26)	Level 3 (16)
C5	64	63	41	59	44	17	75	74	69
MNL	81	79	88	**83**	**100**	67	96	**90**	**84**
treeMNL	81	76	70	70	86	69	95	87	84
corMNL	**84**	**82**	**89**	**83**	**100**	**73**	**97**	**90**	82

Table 2: Comparison of models based on their predictive accuracy (%) for specific coverage (provided in parenthesis). The C5 results and the coverage values are from King *et al.* (2001).

Accuracy (%)	SIM only			Combined dataset single scale parameter			Combined dataset separate scale parameters		
	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
MNL	69.10	45.79	30.76	69.66	48.88	32.02	70.65	**49.16**	33.71
treeMNL	67.70	45.93	30.34	68.26	46.63	30.34	68.82	46.63	31.74
corMNL	**70.51**	**47.19**	**30.90**	**71.49**	**49.30**	**32.87**	**72.75**	**49.16**	**34.41**

Table 3: Accuracy (%) of models on the combined dataset with and without separate scale parameters. Results based on using SIM alone are provided for comparison.

select the top g percent. In Table 2, the coverage values are given in parenthesis. All three models discussed here outperform the decision tree model. Overall, corMNL has better performance than MNL and treeMNL.

King *et al.* (2001) attempted to improve predictive accuracy by combing the three datasets (SEQ, STR and SIM). Although one would expect to obtain better predictions by combining several sources of information, their results showed no additional benefit compared to using the SIM dataset alone. We also tried combining datasets in order to obtain better results. Initially, we used the principal components which we found individually for each dataset, and kept the number of covariates contributed from each data source the same as before (i.e., 100 covariates from SEQ, 100 covariates from STR, and 150 covariates from SIM). Principal components from each dataset were scaled so that the standard deviation of the first principal component was 1. We did this to make the scale of variables from different data sources comparable while preserving the relative importance of principal components within a dataset.

Using the combined dataset, all our models provided better predictions, although the improvement was only marginal for some predictions. We speculated that some of the variables (i.e., PCs) may become redundant after combining the data. That is, some variables are providing the same information. In general, one may obtain better results by removing redundancy and reducing the number of variables. To examine this idea, we kept the number of principal components from SIM as before (i.e., 150) but only used the first 25 principal components from SEQ and STR. The total number of covariates was therefore 200. Reducing the number of covariates from SEQ and STR may also prevent them from overwhelming the covariates from SIM, which is the most useful single source. This strategy led to even higher accuracy rates compared to when we used the SIM dataset alone. The results are shown in Table 3 (middle section). It is worth noting that using 25 principal components results in a lower performance (i.e., lower accuracy rate) when SEQ and STR are used individually in the models (results not shown).

To improve the models even further, we tried using separate scale parameters, ξ, for different sources of information. This way, we allow the coefficients from different data sources to have appropriately different variances in the model. This is additional to what ARD hyperparameters provide. As we can see in Table 3 (right section), this strategy resulted in further improvements in the performance of the models. The posterior distribution of the ξ's reflected the importance of each data source. In the MNL model, the posterior means of the three ξ's were 2.90, 0.86 and 4.67 for SEQ, STR and SIM respectively. The corresponding values in treeMNL were 2.39, 0.85 and 4.56 and in corMNL 2.21, 0.74 and 3.63.

We examined the idea of having separate ξ's and larger numbers of covariates. We found that when we increased the number of principal components for SEQ and STR to 100, the accuracy of predictions mostly remained the same, though a few dropped slightly.

In practice, we might be most interested in genes whose function can be predicted with high confidence. There is a trade-off between predictive accuracy and the percentage of the genes we select for prediction (i.e., coverage). Table 4 shows this trade-off for results on the test set from the corMNL model with three ξ hyperparameters applied to the combined dataset. In this table, the accuracy rates for different coverage values are provided. As we can see, our model can almost perfectly classifies 10% of the genes in the test set.

Accuracy (%)	Coverage (%)					
	5	*10*	*20*	*50*	*90*	*100*
Level 1	100	98	96	92	76	73
Level 2	100	98	96	71	53	49
Level 3	100	97	80	52	36	34

Table 4: Predictive accuracy (%) for different coverage values (%) of the corMNL model using all three sources with three ξ hyperparameters.

The MATLAB programs for MNL, treeMNL and corMNL, as well as the combined dataset for *E. coli*, are available online at http://www.utstat.utoronto.ca/∼babak.

7 Conclusions and Future Directions

In this paper, we investigated the use of hierarchical classification schemes to perform functional annotation of genes. If the hierarchy provides any information regarding the structure of gene function, we expected that this additional information would lead to better prediction of classes. To examine this idea, we compared three Bayesian models: a non-hierarchical MNL model, a hierarchical model based on nested MNL, referred to as treeMNL, and the corMNL model, which is a form of the multinomial logit model with a prior that introduces correlations between the parameters of nearby classes. We found corMNL provided better predictions in most cases. Moreover, we introduced a new approach for combining different sources of data. In this method, we use separate scale parameters for each data source in order to allow their corresponding coefficients have appropriately different variances. This approach provided better predictions compared to other methods.

While our emphasis in this paper was on the importance of using hierarchical schemes in gene classification, we also showed that even the non-hierarchical Bayesian MNL model outperforms previous methods that used the C5 algorithm. Overall, our results are encouraging for the prospect of accurate gene function annotation, and also illustrate the utility of a Bayesian approach with problem-specific priors. For our experiments, we used the pre-processed datasets provided by King *et al.* (2001), who used the Warmr (Dehaspe *et al.* 1998) algorithm to generate binary attributes. It is conceivable that the accuracy of predictions can be further improved by using other data processing methods. Similarly, it is possible that a method other than our use of PCA might be better for reducing dimensionally before doing classification.

In the *E. coli* dataset, each ORF was assigned to only one function. This is not the case for some other hierarchies such as the MIPS functional classification defined for the genome of *S. cerevisiae*, where an ORF may belong to more than one class. For such problems, one can modify the likelihood part of the models described here to handle this additional complexity. For example, if a gene can belong to several classes with equal probabilities, its contribution to the likelihood is proportional to the sum of probabilities of those classes.

The functional hierarchies considered here are simple tree-like structures. There are other hierarchical structures that are more complex than a tree. For example, one of the most commonly used gene annotation schemes, known as Gene Ontology (GO), is implemented as a directed acyclic graph (DAG). In this structure a node can have more than one parent. Our method, as it is, cannot be applied to these problems, but it should be possible to extend the idea of summing coefficients along the path to the class in order to allow for multiple paths.

Our approach can also be generalized to problems where the relationship among classes can be described by more than one hierarchical structure. For these problems, different hyperparameters can be used for each hierarchy and predictions can be made by summing the parameters in branches from all these hierarchies.

Acknowledgements

We thank Ross D. King, Andreas Karwath, Amanda Clare and Luc Dehaspe for providing the processed *E. coli* datasets. This research was supported by the Natural Sciences and Engineering Research Council of Canada. Radford Neal holds a Canada Research Chair in Statistics and Machine Learning.

References

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389–3402.

Blattner, F. R., Plunkett, G. r., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J. D., Rode, C. K., Mayhew, G. F., Gregor, J., Davis, N. W., Kirkpatrick, H. A., Goeden, M., Rose, D. J., Mau, B. and Shao, Y. (1997) The complete genome sequence of Escherichia coli k-12. *Science*, **277**, 1453–1474.

Brown, M., Nobel, G. W., Lin, D., Cristianini, N., Walsh, S. C., Furey, T., Ares, M. J. and Haussler, D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci. USA*, **97**, 262–267.

Cai, L. and Hoffmann, T. (2004) Hierarchical document categorization with support vector machines. *ACM 13th Conference on Information and Knowledge Management.*

Cesa-Bianchi, N., Gentile, C. and Zaniboni, L. (2006) Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, **7**, 31–54.

Clare, A. and King, R. D. (2003) Predicting gene function in Saccharomyces cerevisiae. *Bioinformatics*, **19**, ii42–ii49.

Dehaspe, L., Toivonen, H. and King, R. D. (1998) Finding frequent substructures in chemical compounds. In Agrawl, R., Stolorez, P. and Piatetsky-Shapiro, G. (eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Minging*, pp. 30–36. AAAI Press, Menlo Park, CA.

Dekel, O., Keshet, J. and Singer, Y. (2004) Large margin hierarchical classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML).*

Deng, M., Chen, T. and Sun, F. (2004) An integrated probabilistic model for functional prediction of proteins. *Journal of Computational Biology*, **11**, 463–475.

Deng, M., Zhang, K., Mehta, S., Chen, T. and Sun, F. (2003) Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology*, **10**, 947–960.

DeRisi, J., Iyer, V. and Brown, P. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

Dumais, S. T. and Chen, H. (2000) Hierachical classification of web content. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 256–263.

Eisen, J. A. (1998) Phylogenomics: Improving functional prediction for uncharacterized genes by evolutationary analysis. *Genome Research*, **8**, 163–167.

Eisen, M., Spellman, P., Brown, P. and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences (USA)*, **95**, 14863–14868.

Engelhardt, B. E., Jordan, M. I., Muratore, K. E. and Brenner, S. E. (2005) Protein molecular function prediction by Bayesian phylogenomics. *PLoS Computational Biology*, **1**, 432–445.

Fox, J. (1997) *Applied Regression Analysis, Linear Models and Related Methods*. Sage.

IUBMB (1992) *Enzyme nomenclature: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology*. Academic Press, New York.

King, R. D., Karwath, A., Clare, A. and Dehaspe, L. (2001) The utility of different representations of protein sequence for predicting functional class. *Bioinformatics*, **17**, 445–454.

Koller, D. and Sahami, M. (1997) Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*.

Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O. and Eisenberg, D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

McCallum, A., Rosenfeld, R., Mitchell, T. and A., N. (1998) Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 359–360.

Mitchell, T. M. (1998) Conditions for the equivalence of hierarchical and flat bayesian classifiers. URL `http://www.cs.cmu.edu/~tom/hierproof.ps`.

Neal, R. M. (1993) *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. Springer Verlag, New York.

Neal, R. M. (2003) Slice sampling. *Annals of Statistics*, **31**, 705–767.

Pavlidis, P. and Weston, J. (2001) Gene functional classification from heterogeneous data. *Proceedings of the 5th International Conference on Computational Modelcular Biology (RECOMB)*, pp. 249–255.

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences (USA)*, **85**, 2444–2448.

Riley, M. (1993) Functions of the gene products of Escherichia coli. *Microbiology Review*, **57**, 862–952.

Riley, M. and Labedan, B. (1996) E. coli gene products: physiological functions and common ancestries. In Neidhardt, F. N., Curtiss, R. I., Lin, E. C. C., Ingraham, J. L., Low, K. B., Magasanik, B., Reznikoff, W., Riley, M., Schaechter, M. and Umbarger, E. (eds.), *Escherichia coli and Salmonella: cellular and molecular biology, 2nd edition*. ASM Press, Washington, DC.

Rison, S., Hodgman, T. C. and Thornton, J. M. (2000) Comparison of functional annotation schemes for genomes. *Functional and Integrative Genomics*, **1**, 56–69.

Rost, B. (2002) Enzyme function less conserved than anticipated. *Journal of Molecular Biology*, **318**, 595–608.

Sattath, S. and Tversky, A. (1977) Additive similarity trees. *Psychometrika*, **42**, 319–345.

Schoikowski, B., Uetz, P. and Fields, S. (2000) A network of protein-protein interaction in yeast. *Nature Biotechnology*, **18**, 1257–1261.

Shahbaba, B. and Neal, R. M. (2005) Improving classification when a class hierarchy is available using a hierarchy-based prior. Technical Report 0510, Department of Statistics, University of Toronto.

Sjölander, K. (2004) Phylogenomics inference of protein molecular function: Advances and challenges. *Bioinformatics*, **20**, 170–179.

Tsochantaridis, I., Hoffmann, T., Joachims, T. and Altum, Y. (2004) Support vector machine learning for independent and structured output spaces. *Proceedings of the 21st International Conference on Machine Learning (ICML)*.