Bayesian Hierarchical Classification

Seminar on Predicting Structured Data

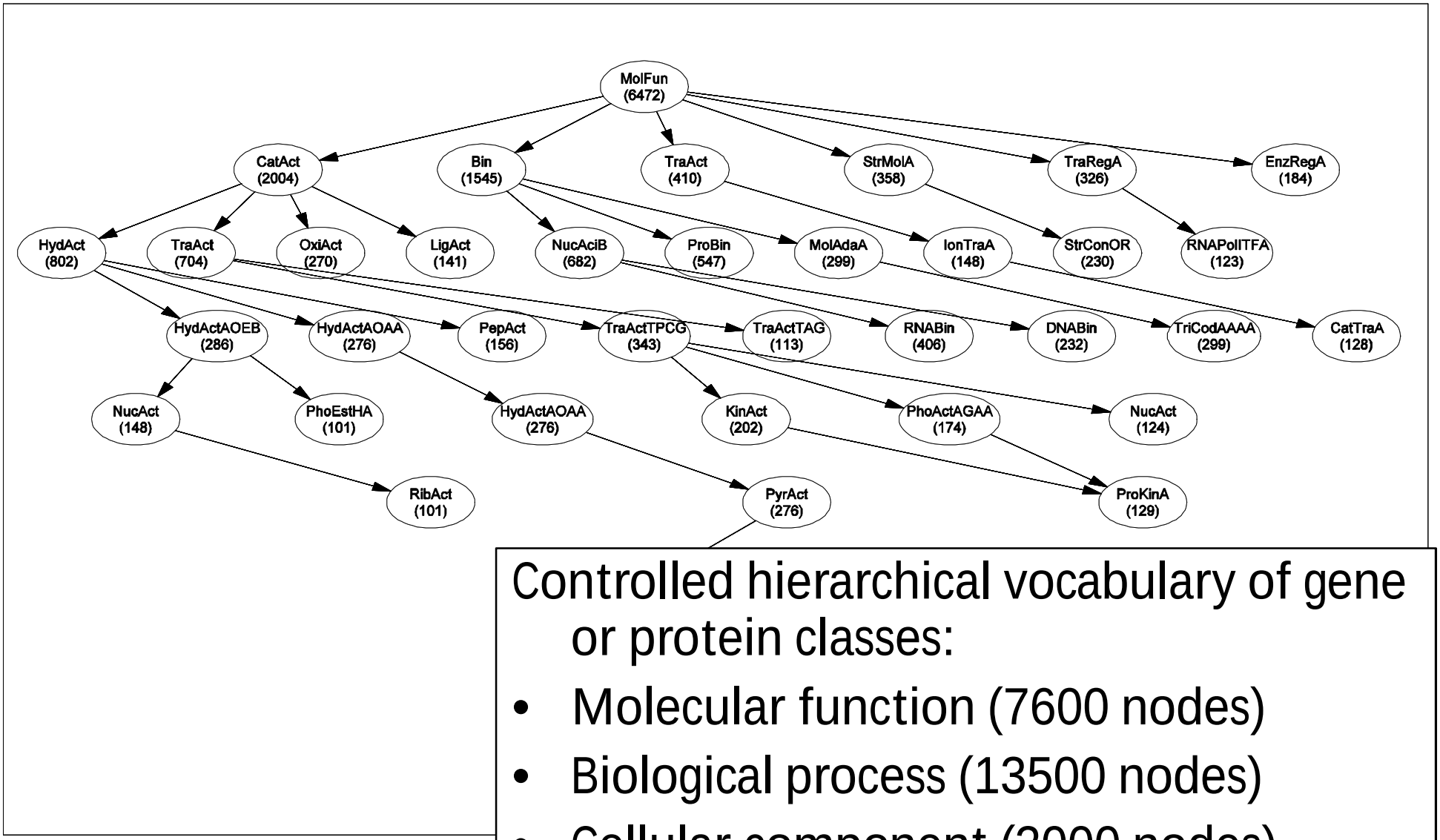
Jukka Kohonen

17.4.2008

Overview

- Intro: The task of hierarchical gene annotation
- Approach I: SVM/Bayes hybrid
  Barutcuoglu et al: Hierarchical multi-label prediction of gene function (Bioinformatics 2006)
- Approach II: Joint Bayesian model
  Shahbaba & Neal: Gene function classification using Bayesian models with hierarchy-based priors (BMC Bioinformatics 2006)
- Discussion

Intro: Gene Ontology (GO)



Controlled hierarchical vocabulary of gene or protein classes:

- Molecular function (7600 nodes)
- Biological process (13500 nodes)
- Cellular component (2000 nodes)

www.geneontology.org

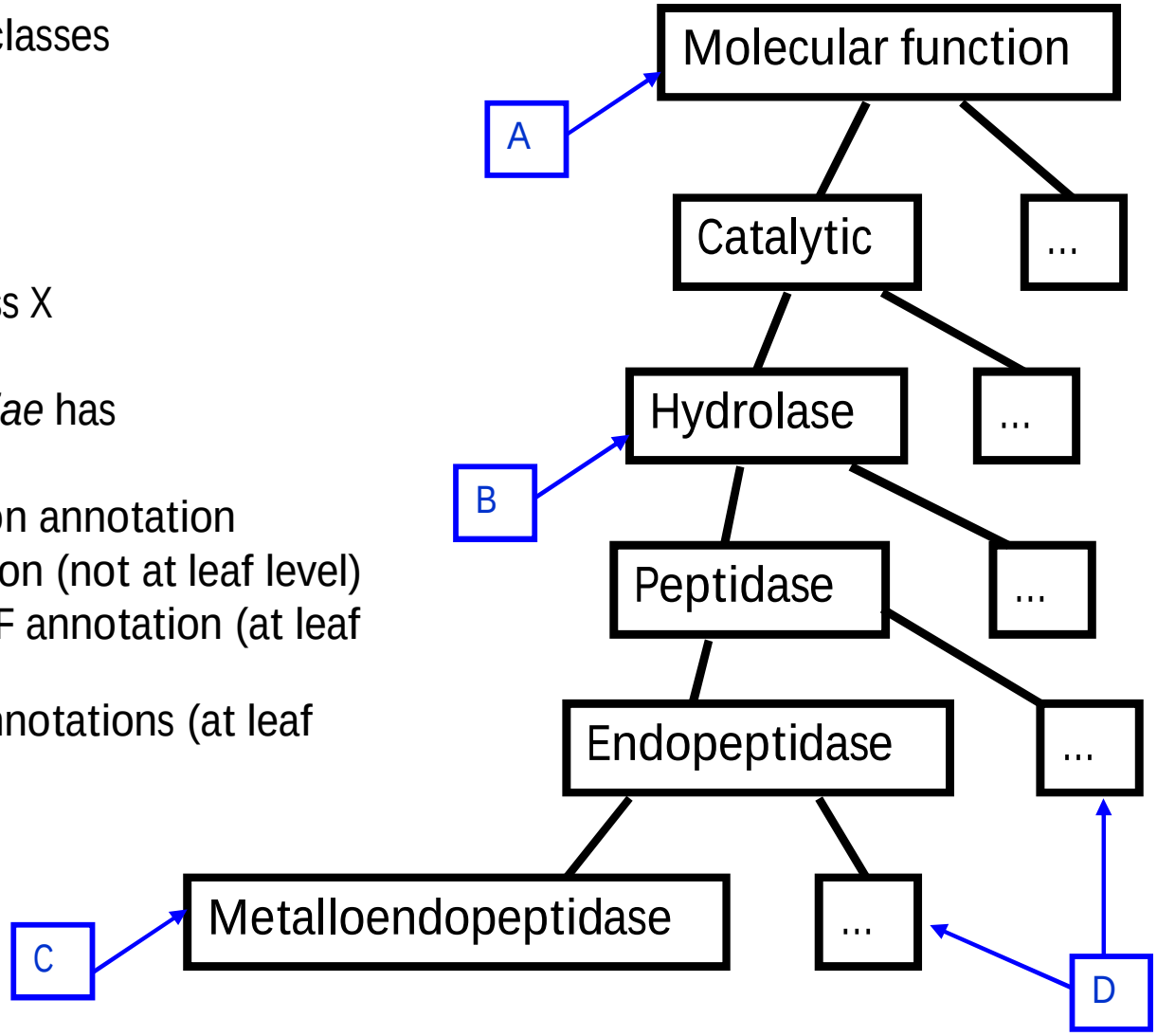
Intro: Hierarchical constraints

Each gene can belong to one or more classes
("multi-label" classification)

Hierarchical constraint:
if gene belongs to class X,
then it also belongs to parent(s) of class X

Example: yeast *Saccharomyces cerevisiae* has
~ 6500 named proteins, of which…

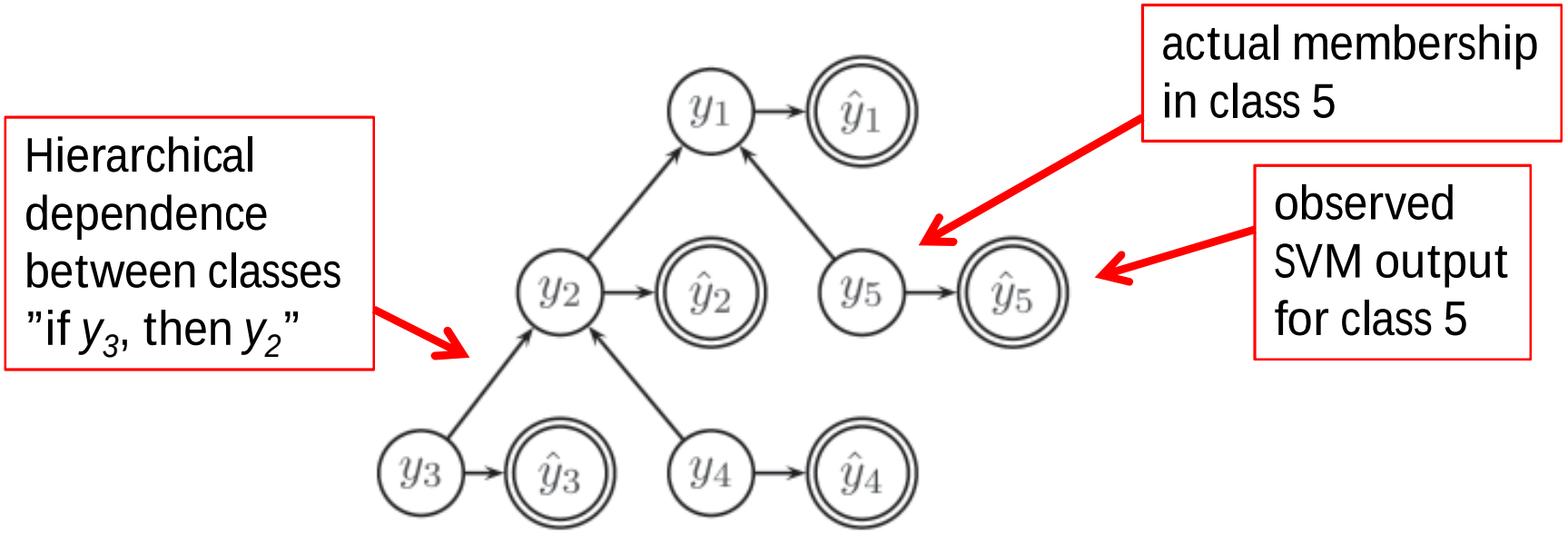
A. ~ 2200 have no molecular function annotation
B. ~ 1600 have only partial annotation (not at leaf level)
C. ~ 2300 have a single complete MF annotation (at leaf level)
D. ~ 300 have multiple complete annotations (at leaf level)



Approach I: SVM/Bayes hybrid

For a (sub)hierarchy of 105 classes:

1. Use 105 separate SVM classifiers (one per class) to generate outputs $\hat{y}_i$
2. Use $\hat{y}_i$ as "observations" and infer actual memberships y_i with a Bayes network (with GO structure built in)

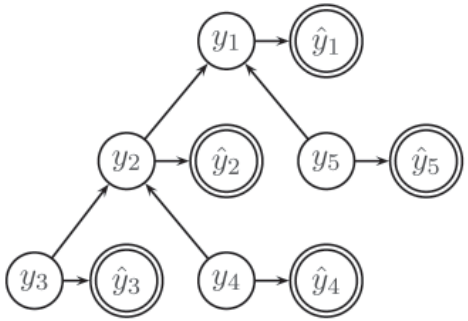


[Barutcuoglu et al, Bioinformatics 2006]

Hybrid: SVM part

- 3465 annotated genes (yeast *S. cerevisiae*)
- Input: 88721 raw features (very sparse; data from pairwise interaction, microarray data, colocalization, and transcription factor sites)
- 5930 features after some pruning
- 105 independent support vector machines (SVM) are trained, using the same input features but different ”yes/no” labelings (according to class membership)

Hybrid: Bayes part

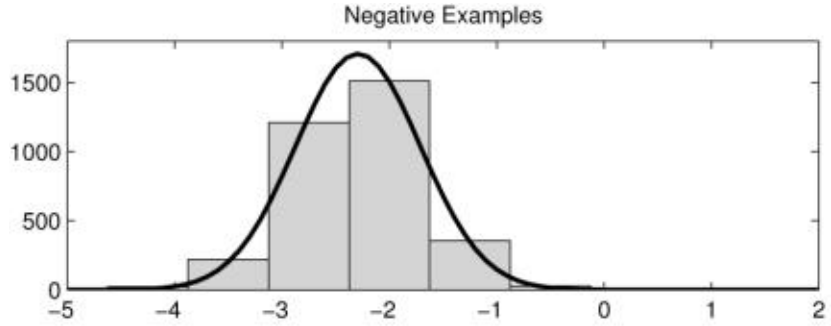


- Class memberships y_i depend on each other according to the GO hierarchy

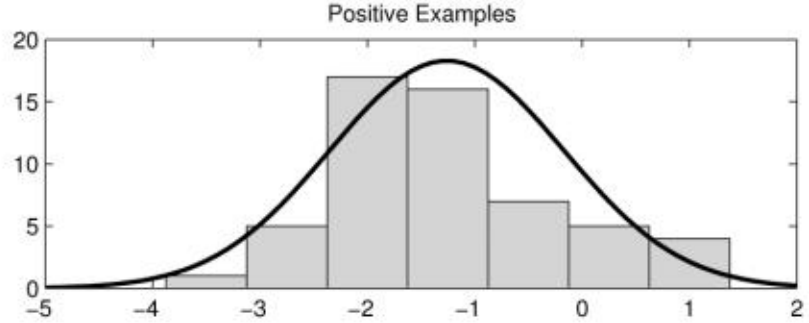
- Each SVM output $\hat{y}_i$ depends only on the true membership y_i
- The two conditional distributions are approximated as Gaussians:
  $(\hat{y}_i \mid y_i = 0) \sim N(..., ...)$
  $(\hat{y}_i \mid y_i = 1) \sim N(..., ...)$

Figure: empirical distributions vs. gaussian model (for one particular class)

Note: not a very good class separation...



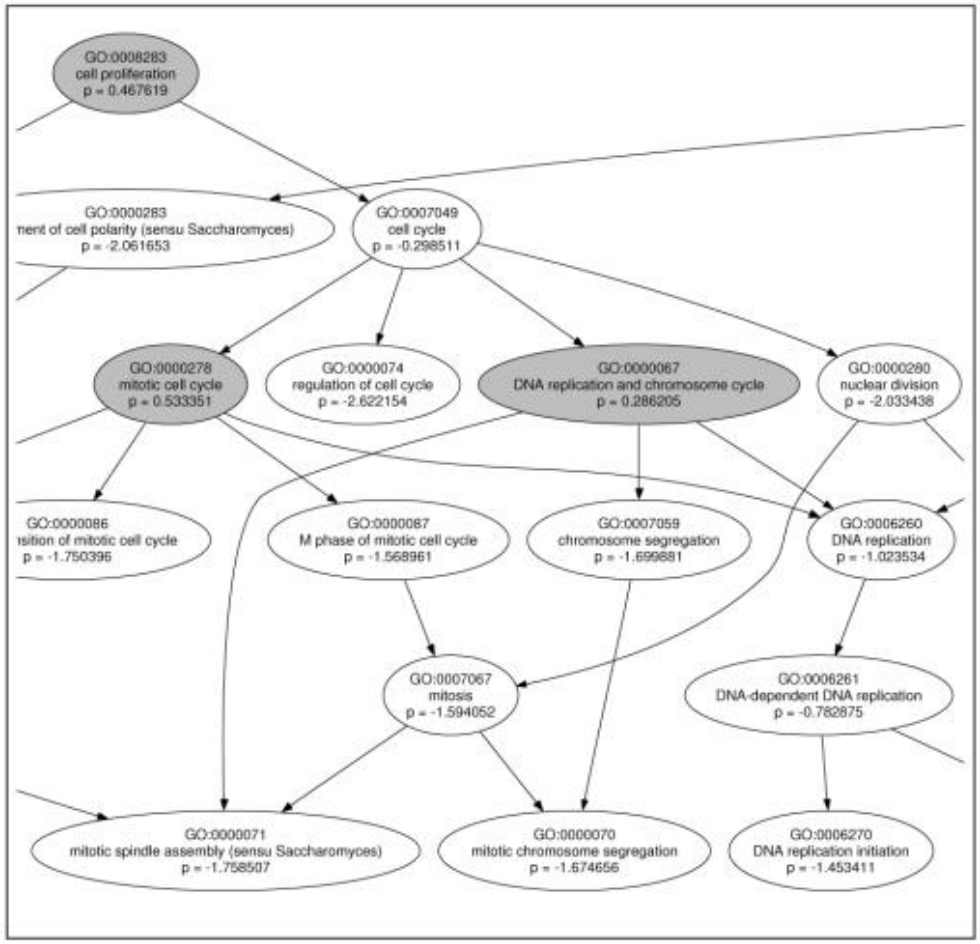
($y_i = 0$)



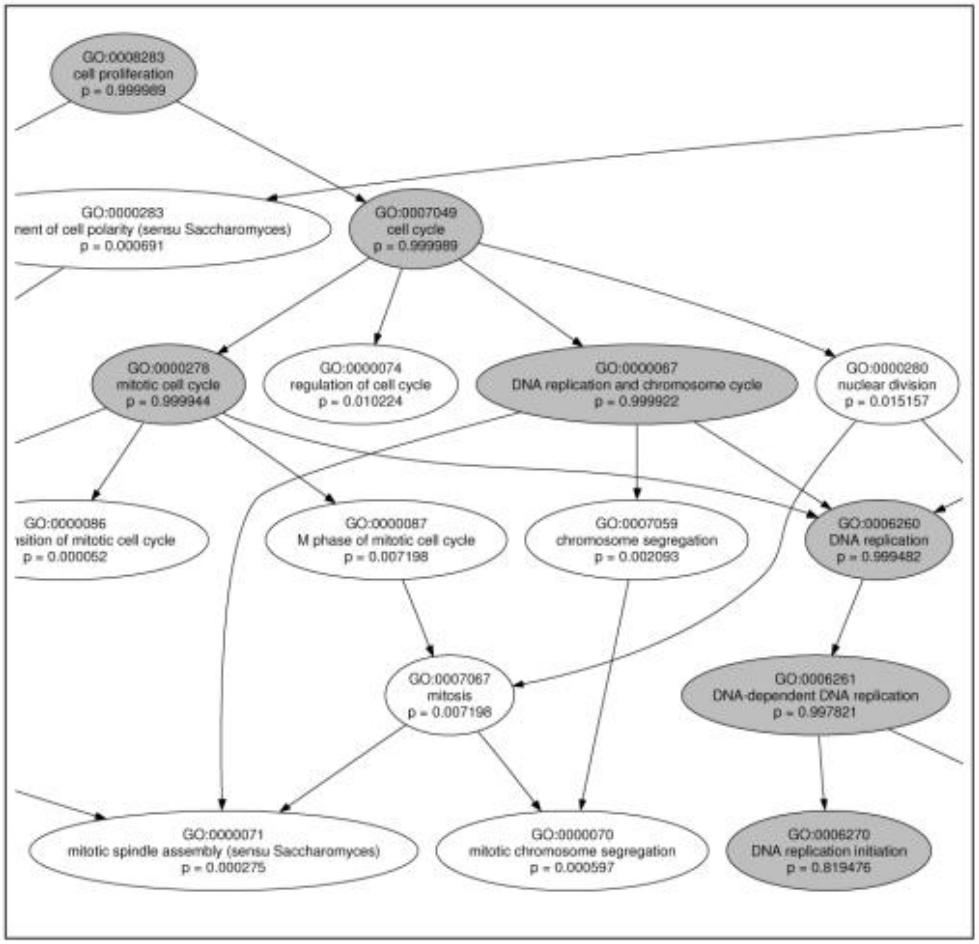
($y_i = 1$)

$\hat{y}_i$ [Barutcuoglu et al]

Hybrid: Example of reconciliation



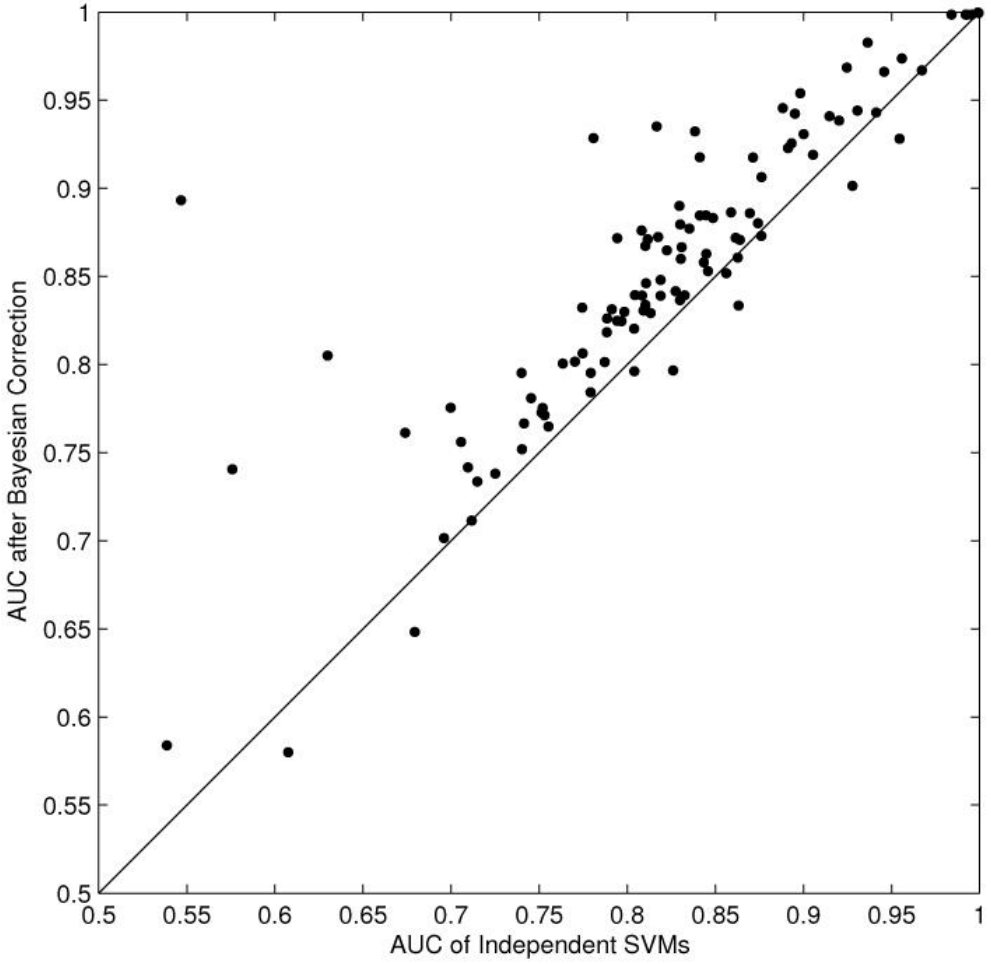
(a) Independent SVMs



(b) Bayesian correction

[Barutcuoglu et al]

Hybrid: Prediction results



Bayesian correction improves the AUC scores slightly

(AUC = area under ROC curve)

[Barutcuoglu et al]

Approach II: Joint Bayesian model

Leaf class probabilities depend on input features x according to a multinomial logit (MNL) model:

$$P(y = j \mid x, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \exp(\alpha_j + x^T \boldsymbol{\beta}_j)$$

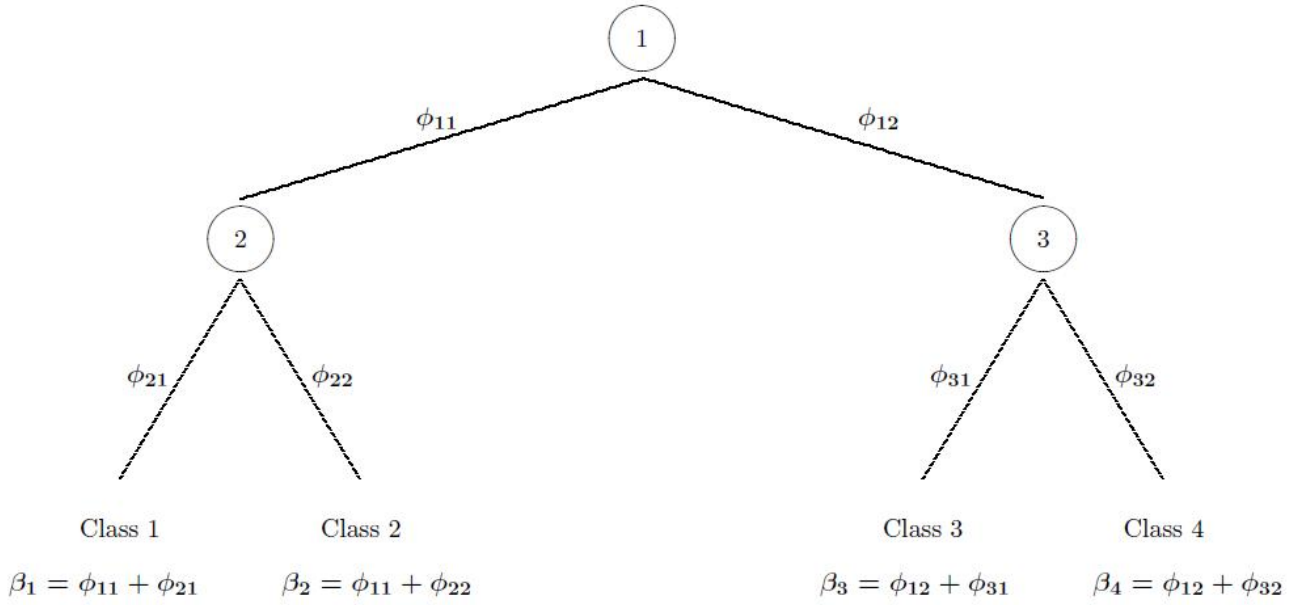


Figure 1
The corMNL model for a simple hierarchy. The coeffcient parameter for each class is a sum of parameters at different levels of the hierarchy.

[Shahbaba & Neal, BMC Bioinformatics 2006]

For each class *j*, different parameters α_j, $\boldsymbol{\beta}_j$ are used.

The $\boldsymbol{\beta}_j$ parameters are constructed in a way that makes them correlated for nearby classes (hence name corMNL)

Joint Bayesian: feature selection

- initial dimension reduction by PCA to ~100
- then Automatic Relevance Detection (ARD)

intercept parameter of MNL (for class *j*)		$\alpha_j \mid \eta \sim N(0, \eta^2)$
slope parameter of MNL (for class *j*, feature *l*)		$\beta_{jl} \mid \xi, \sigma_l, \tau \sim N(0, \xi^2 \tau_j^2 \sigma_l^2)$
hyperparameters	overall scale of **α**	$log(\eta) \sim N(0, 1)$
	overall scale of **β**	$log(\xi) \sim N(-3, 2^2)$
	scale of **β** for class *j*	$log(\tau_j) \sim N(-1, 0.5^2)$
	scale of **β** for feature *l*	$log(\sigma_l) \sim N(0, 0.3^2)$

- Note: if a feature is irrelevant, its coefficients will tend to be near zero.
- For corMNL similar model is applied, but instead of **β**, the model first chooses **ϕ** (from which **β** is calculated, as shown on previous slide).

Joint Bayesian: Data for experiment

- 2122 genes of the bacterium *Escherichia coli* with known function
- Functional hierarchy of 3 levels, 6+20+146 = 172 classes (simpler than GO)
- Three sources of input features (phylogenic, sequence attributes, secondary structure), reduced with PCA to 100, 100, 150 features

Joint Bayesian: Prediction results

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

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**

[Shahbaba & Neal 2006]

Data sources:
SEQ = sequential features, *STR* = secondary structure, *SIM* = phylogenic similarity

Classification methods:
Baseline = simply assign every gene to most common class (at each level)
MNL = non-hierarchical multinomial logit (ignores hierarchy)
treeMNL = nested models (first predict level 1, then level 2, then level 3)
corMNL = joint model with correlated parameters (as described before)

Note: *corMNL* improves over the simple *MNL* (slightly)

Discussion (1/3)

- Very different methods for basically the same problem (and this was just 2 examples)
- In both cases, taking the hierarchy into account improved the prediction results over ”blind” classification (but not much – why?)
- Difficult to compare results between methods:
  - Different feature data, different class hierarchies
  - Different definitions of ”accuracy” of hierarchical classification
- Measuring prediction quality with a ”test set” of known functions might not give a realistic view of true prediction quality:
  the functions of ”unknown” genes might be overall very different from the functions ”known” genes

Discussion (2/3)

- Functional classification seems to be a difficult, ”noisy” classification task
- Input features are often nominally high-dimensional, but extremely sparse; often some feature data is available only for a subset of the genes. For other genes, it is ”missing data” → treat as zero, or impute with KNNimpute, or use something like EM or MCMC.
- Both methods (SVM, and MNL+ARD) are able to use high-dimensional feature data; they assign small ”weights” for irrelevant feature dimensions.

Discussion (3/3)

- Quality of training labels is problematic: e.g. lots of uncertain classifications (which might change between today and tomorrow); lack of explicit ”negative examples”
- Functional classification of genes is a field that needs more work on ”unifying” the definitions, a better definition of what the prediction task really is, and how the prediction quality should be measured!
- Similar hierarchical classification tasks appear in other domains (e.g. document classification).