Predicting Tissue-Dependent Alternative Splicing Using Bayesian Neural Networks

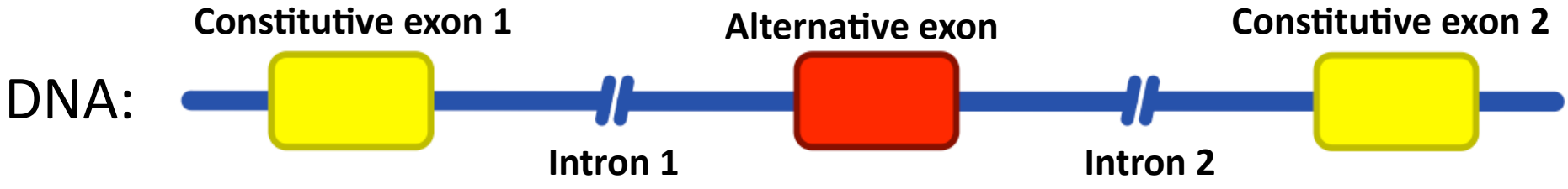
Hui Yuan Xiong

Yoseph Barash and Brendan Frey

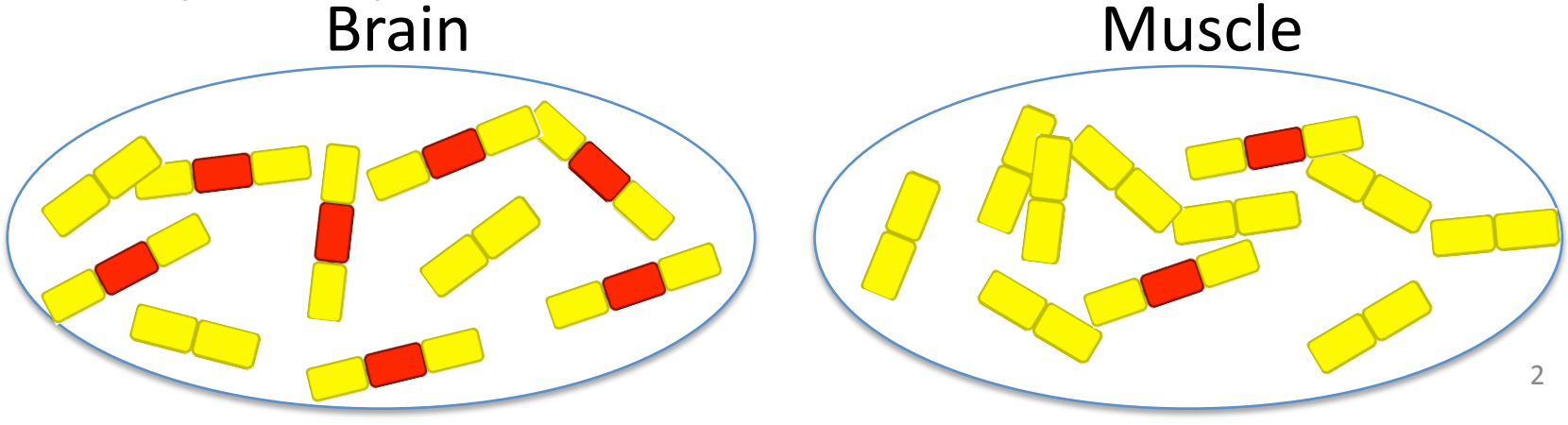
University of Toronto

Alternative Splicing

- One gene produces many transcripts:
  20 thousand genes -> hundreds of thousands of mRNA
- Most common type : cassette splicing

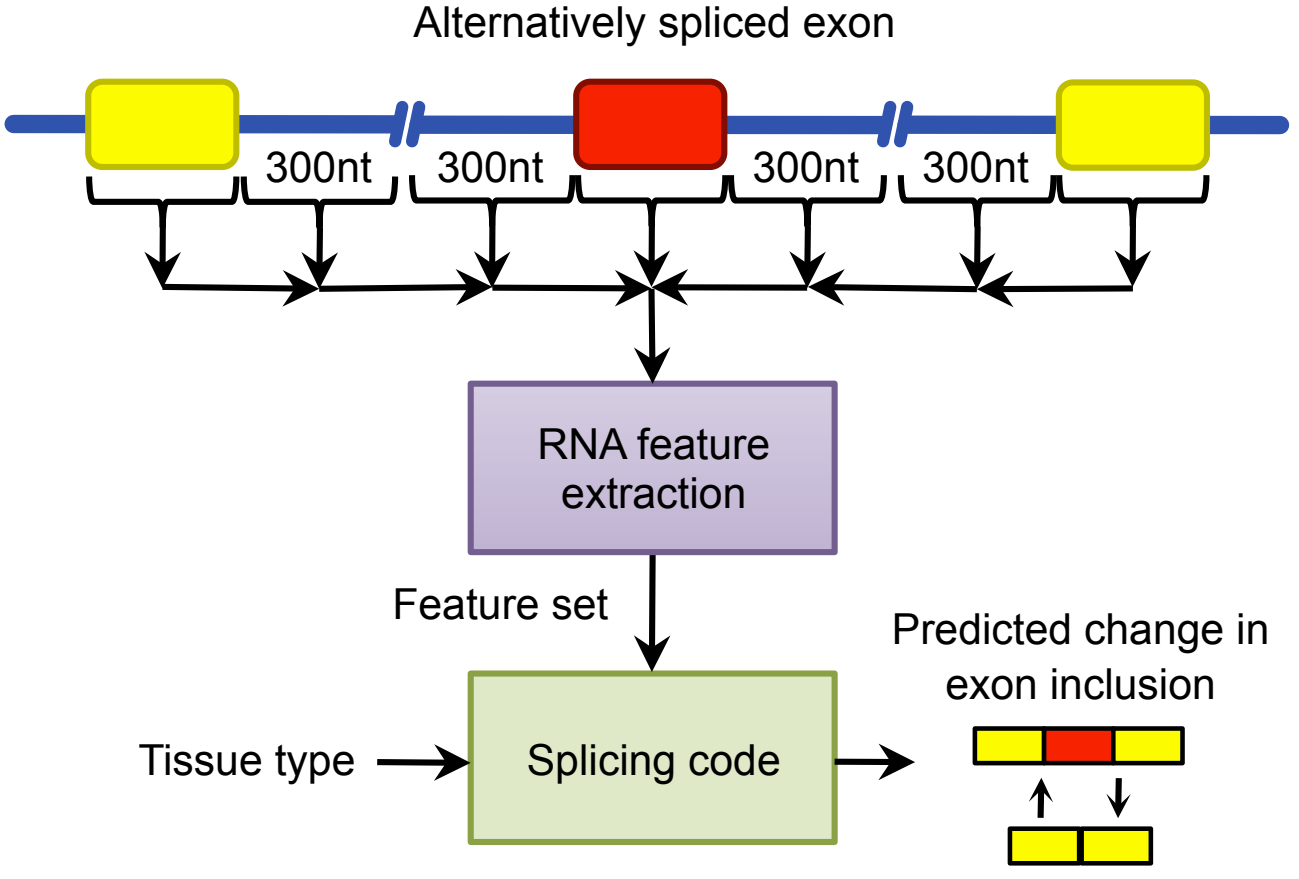


AS pattern (mRNA):



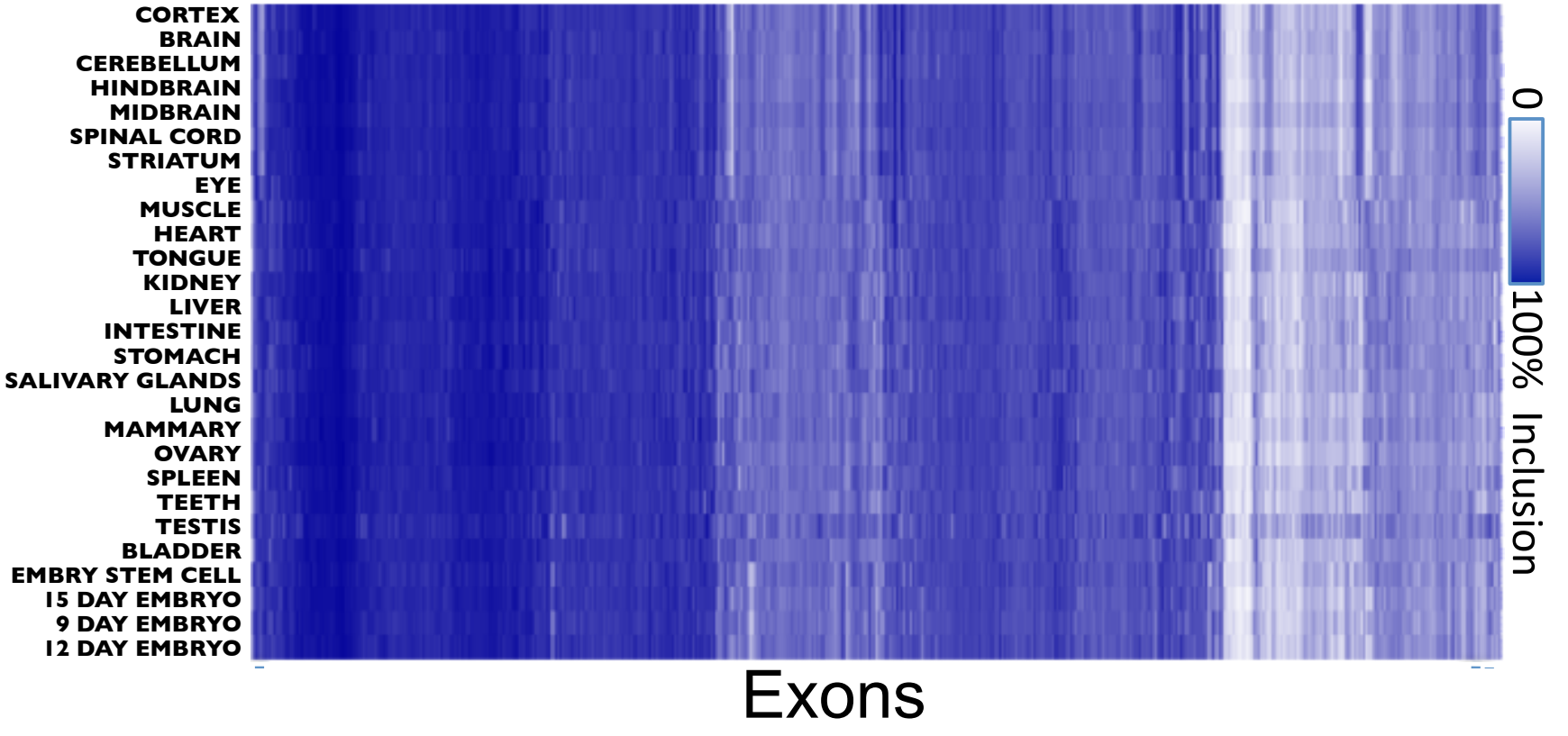
Problem Setup

Goal: find a tissue-dependent "splicing code"



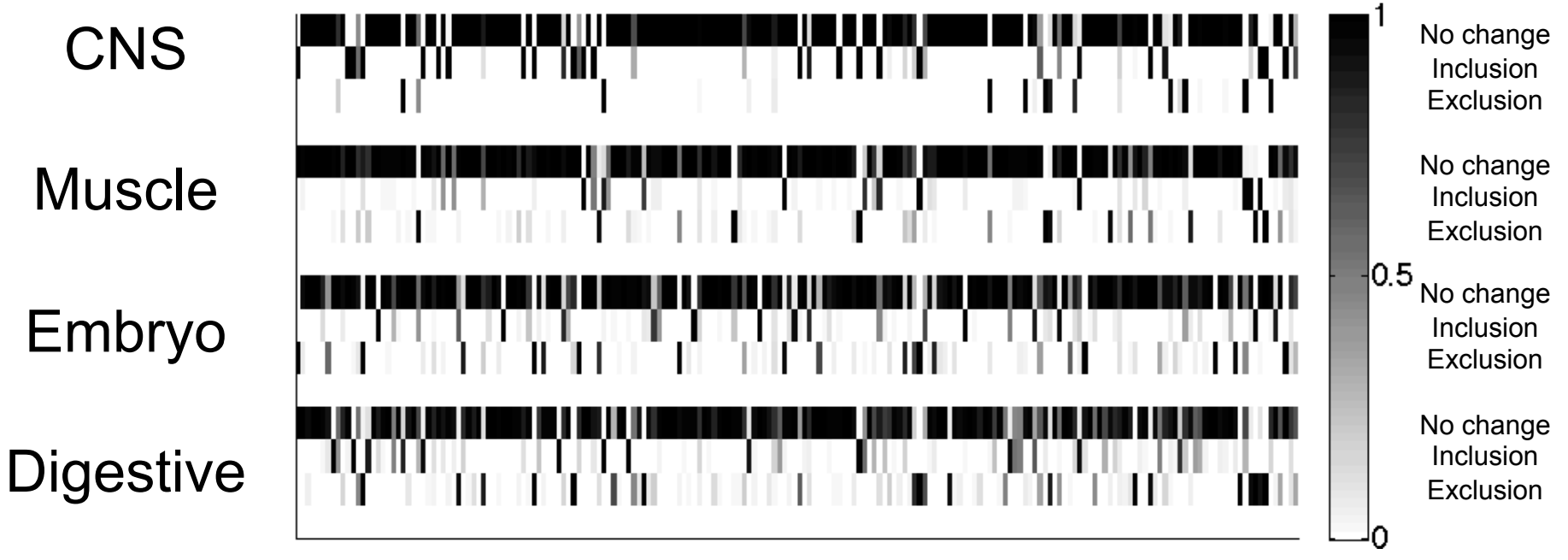
(Y Barash, J Calarco, W Gao, Q Pan, X Wang, O Shai, BJ Blencowe, BJ Frey, Nature 2010)

Microarray Measurements



- 3665 mouse cassette exons were studied
- Percent inclusion in 27 tissues measured by microarray
- Tissues were grouped by a method similar to PCA (Barash *et al.* ISMB 2010)

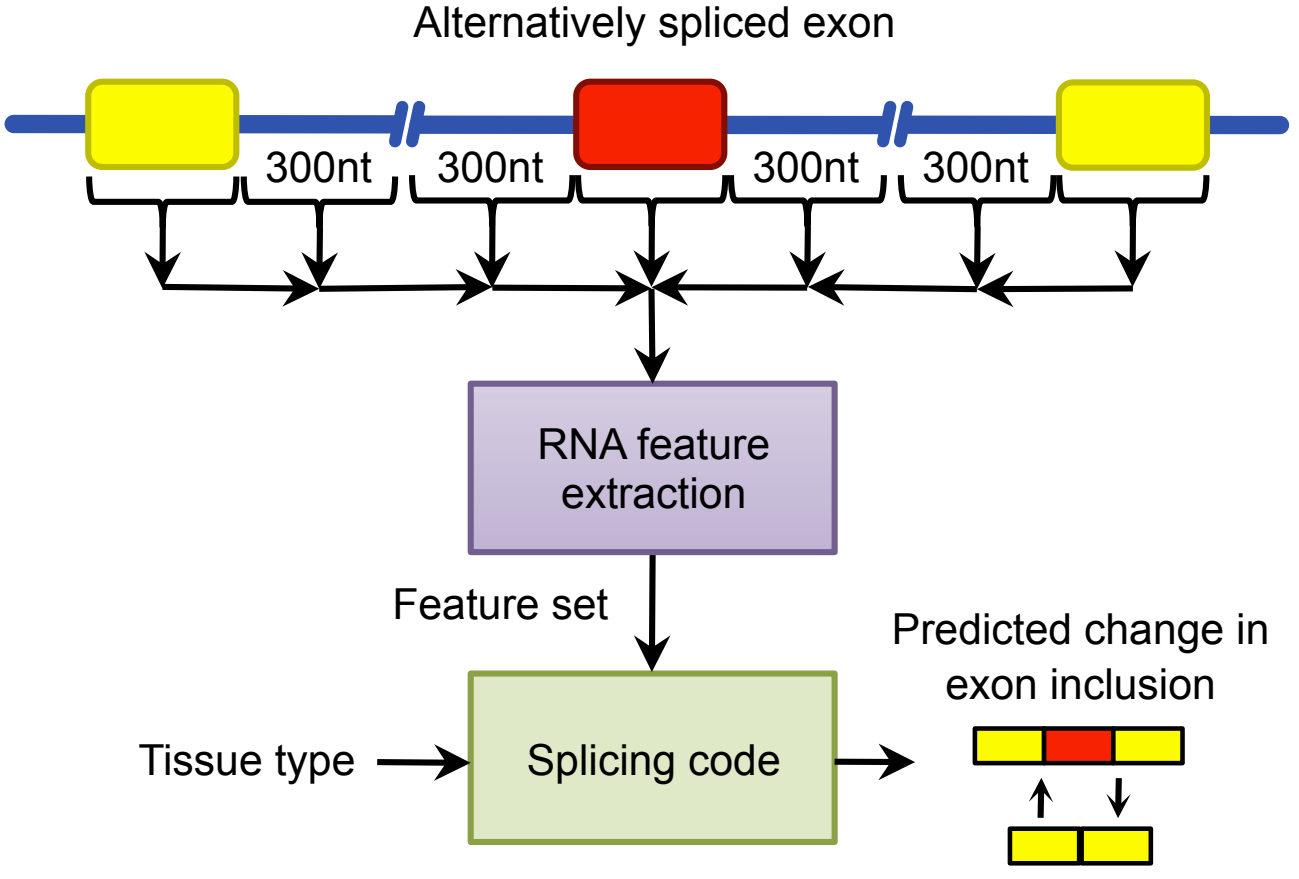
Splicing patterns



- After grouping, inclusion level of an exon is compared to the average inclusion level across tissues
- 'splicing pattern': a multinomial distribution $\{q_{inc}, q_{exc}, q_{nc}\}$, for more inclusion, exclusion, and no change

Problem Setup

Goal: find a tissue-dependent "splicing code"

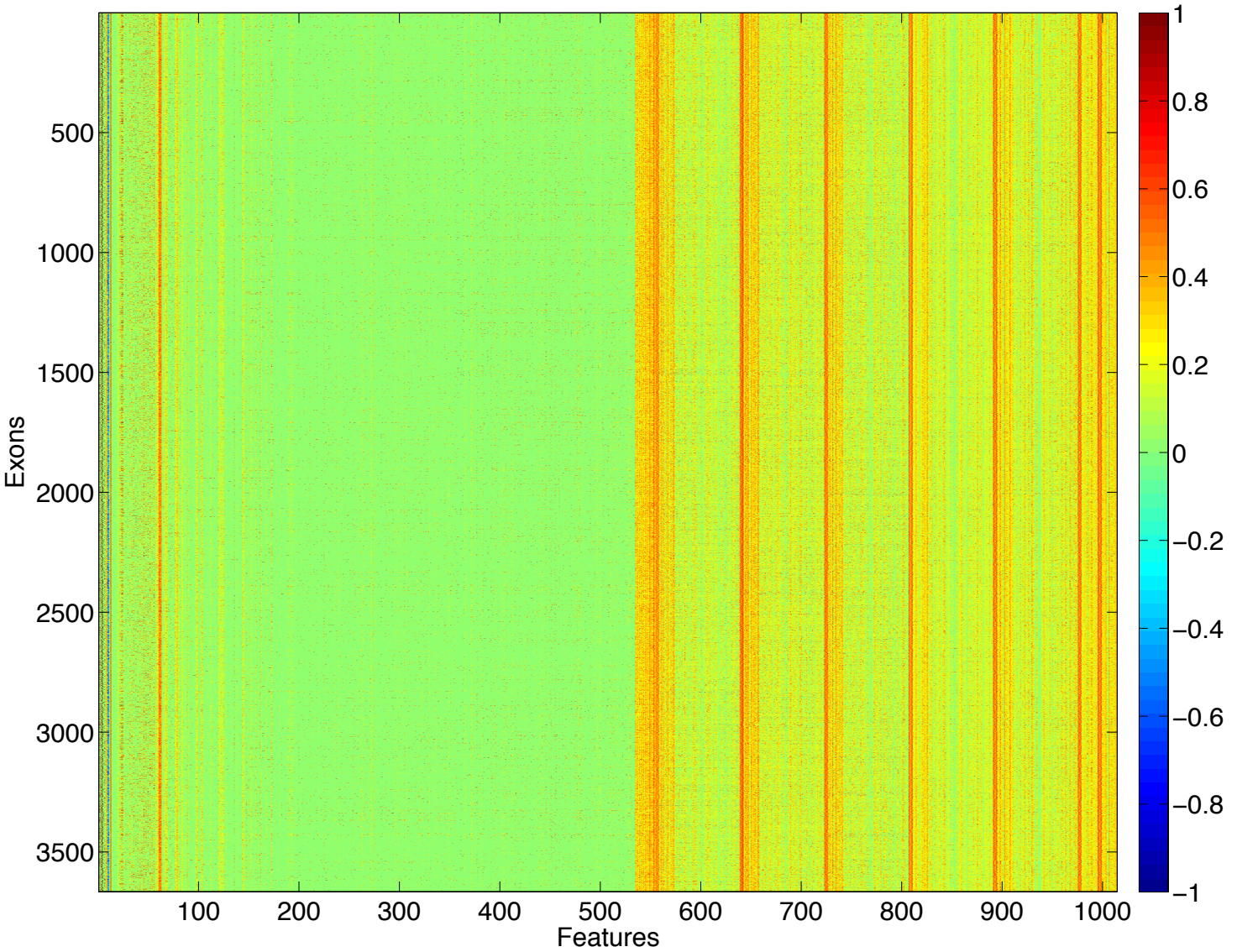


(Y Barash, J Calarco, W Gao, Q Pan, X Wang, O Shai, BJ Blencowe, BJ Frey, Nature 2010)

Alternative Splicing Dataset: Features

- 1014 features extracted from sequence near the alternatively spliced exon, consists of:
  - known motifs (171)
  - computationally identified putative motifs (326)
  - 1-3nt short sequence occurrences (460)
  - transcript structure (57)
- Examples:
  - Fox 1, Fox 2 motif (correlated) (sparse integer valued)
  - Exon length (dense integer valued)
  - Frequency of ‘A’ and ‘AC’ in upstream intron (real valued)

Alternative Splicing Dataset: Features



Objective function

$$\mathcal{H}^t = \sum_{e=1}^{E} \Big(D_{KL}(\mathbf{q}^{t,e} \| \bar{\mathbf{q}}) - D_{KL}(\mathbf{q}^{t,e} \| \mathbf{p}^{t,e}) \Big)$$

$$= \sum_{e=1}^{E} \sum_{s \in \{inc, exc, nc\}} q_s^{t,e} \log_2 \left(\frac{p_s^{t,e}}{\bar{q}_s} \right).$$

$\mathcal{H}^t$: Code quality of tissue t $\quad$ $\mathbf{q}$: target splicing pattern

$\mathbf{p}^{t,e}$: predicted splicing pattern for exon e

- 'Code quality': total difference in KL divergence to the target between a prediction and a naïve guesser
- Maximum code quality is the entropy of the targets (~2000 bits/tissue)

Bayesian Neural Networks

Prior:

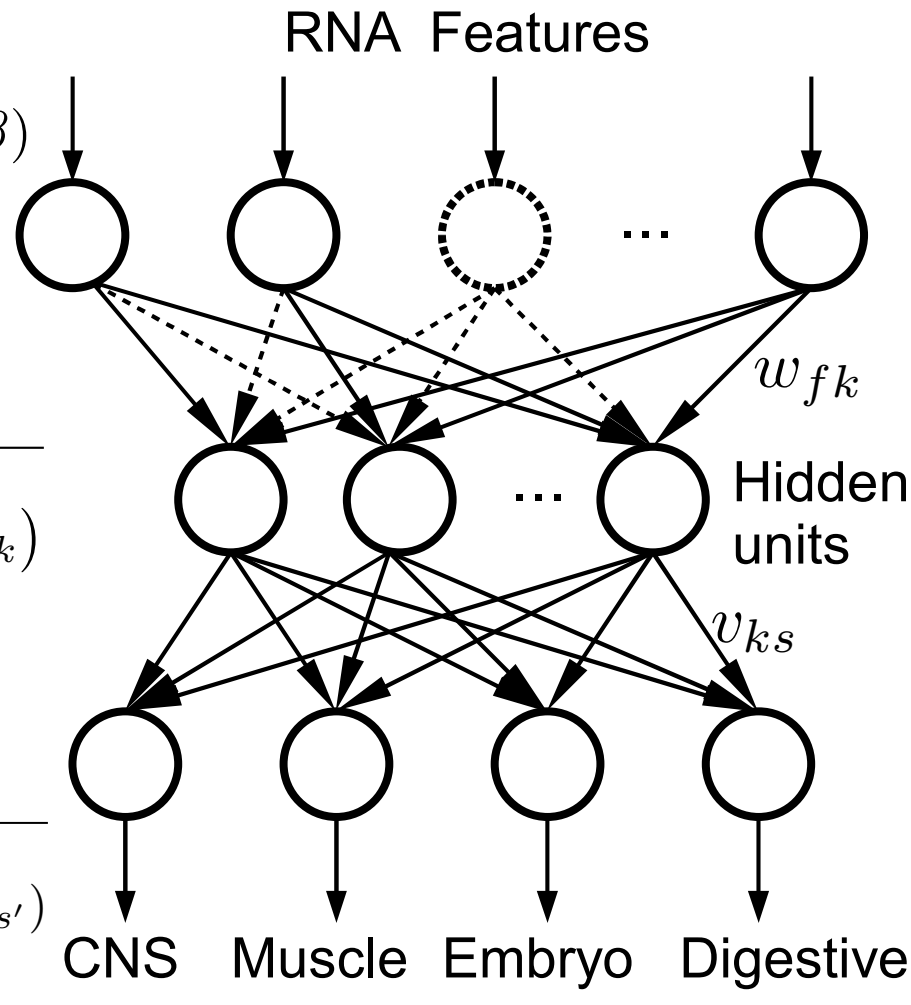
$$p(w_{fk}) = \alpha\delta(w_{fk}) + (1-\alpha)\mathcal{N}(w_{fk}; 0, \beta)$$

$$p(v_{ks}) = \mathcal{N}(v_{ks}; 0, \beta)$$

Model:

$$h_k^e = \frac{1}{1 + \exp(\sum_{f=1}^{F} x_f^e w_{fk})}$$

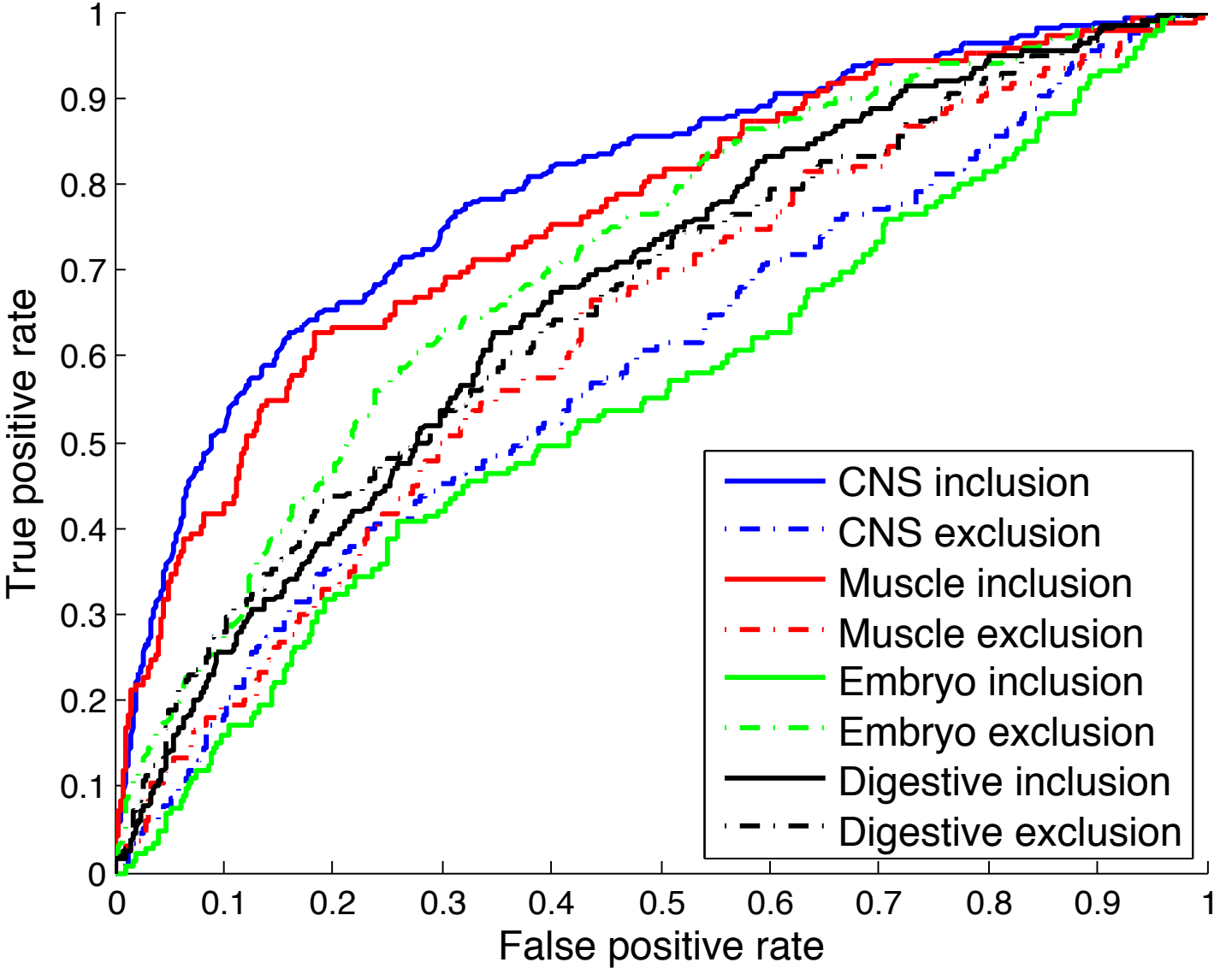
$$p_s^e = \frac{\exp(\sum_{k=1}^{K} h_k^e v_{ks})}{\sum_{s' \in \{inc, exc, nc\}} \exp(\sum_{k=1}^{K} h_k^e v_{ks'})}$$



MCMC Algorithm

- Gibbs sampling used to sample discreteized parameters for fast mixing and easy implementation
- Step size: 0.1 Range: [-5,5]
- 5-fold cross validation was used
- α=0.9 (90% features are inactive) and β=1 (std of the prior)
- 90% parameters are zero in the parameter samples (agrees with prior)

ROC Curves


True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
CNS inclusion
CNS exclusion
Muscle inclusion
Muscle exclusion
Embryo inclusion
Embryo exclusion
Digestive inclusion
Digestive exclusion

Results

Method	Total	Relative Improvement	Tissues			
			CNS	Muscle	Embryo	Digestive
Boosting	393	baseline	187	69	76	61
SVM with 40 PCA	466	19%	211	93	93	69
Non-Bayesian NN (K = 2)	483	23%	211	112	82	78
Non-Bayesian NN (K = 10)	483	23%	215	112	90	66
Non-Bayesian NN (K = 10) (Bagged)	541	38%	229	125	98	89
Bayesian NN (K = 2)	486	24%	230	95	99	62
Bayesian NN (shared) (K = 2)	538	37%	235	111	108	84
Bayesian NN (shared) (K = 10)	598	52%	256	122	123	97

Results

Method	Total	Relative Improvement	Tissues			
			CNS	Muscle	Embryo	Digestive
Boosting	393	baseline	187	69	76	61
SVM with 40 PCA	466	19%	211	93	93	69
Non-Bayesian NN (K = 2)	483	23%	211	112	82	78
Non-Bayesian NN (K = 10)	483	23%	215	112	90	66
Non-Bayesian NN (K = 10) (Bagged)	541	38%	229	125	98	89
Bayesian NN (K = 2)	486	24%	230	95	99	62
Bayesian NN (shared) (K = 2)	538	37%	235	111	108	84
Bayesian NN (shared) (K = 10)	598	52%	256	122	123	97

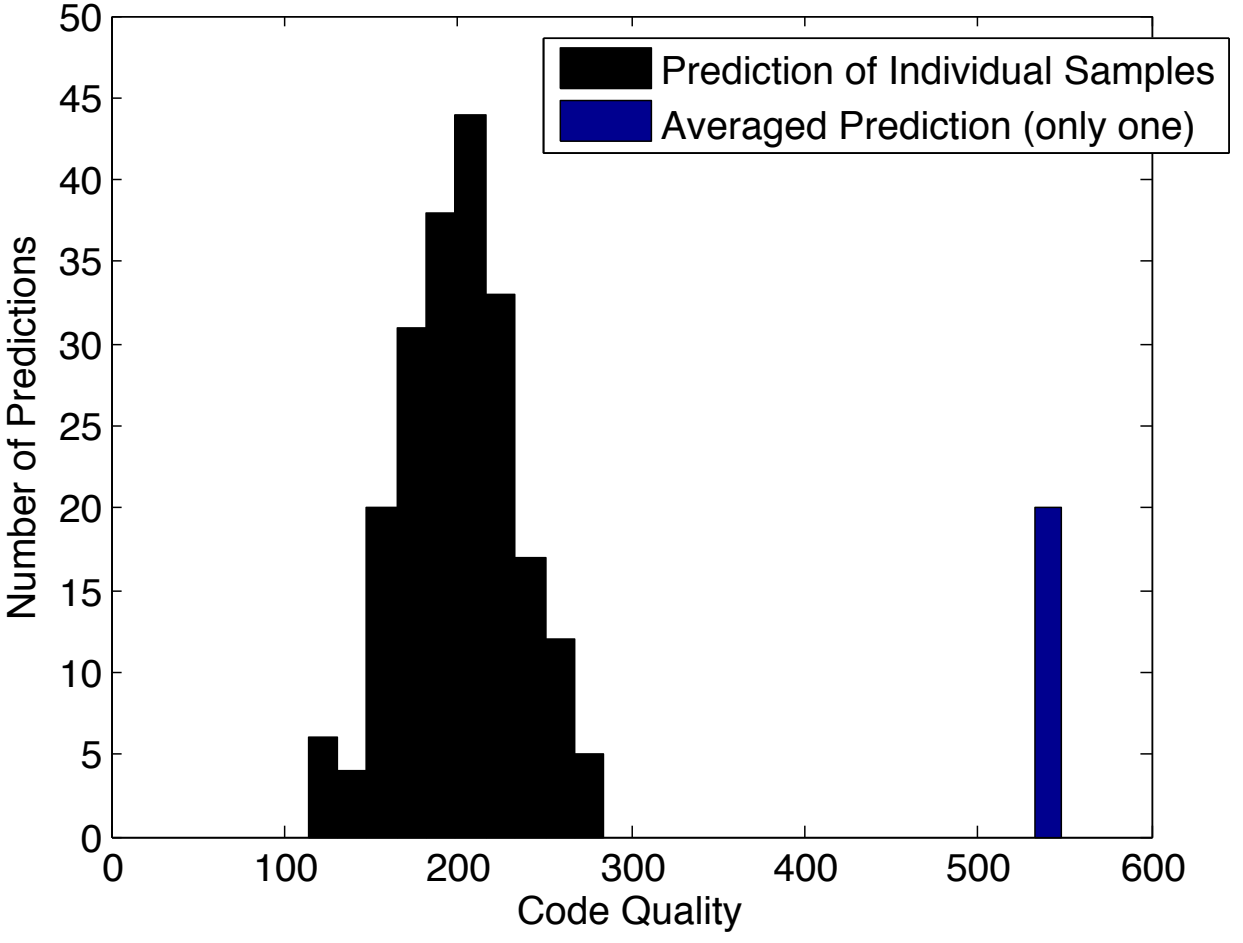
Results

Method	Total	Relative Improvement	Tissues			
			CNS	Muscle	Embryo	Digestive
Boosting	393	baseline	187	69	76	61
SVM with 40 PCA	466	19%	211	93	93	69
Non-Bayesian NN (K = 2)	483	23%	211	112	82	78
Non-Bayesian NN (K = 10)	483	23%	215	112	90	66
Non-Bayesian NN (K = 10) (Bagged)	541	38%	229	125	98	89
Bayesian NN (K = 2)	486	24%	230	95	99	62
Bayesian NN (shared) (K = 2)	538	37%	235	111	108	84
Bayesian NN (shared) (K = 10)	598	52%	256	122	123	97

Results

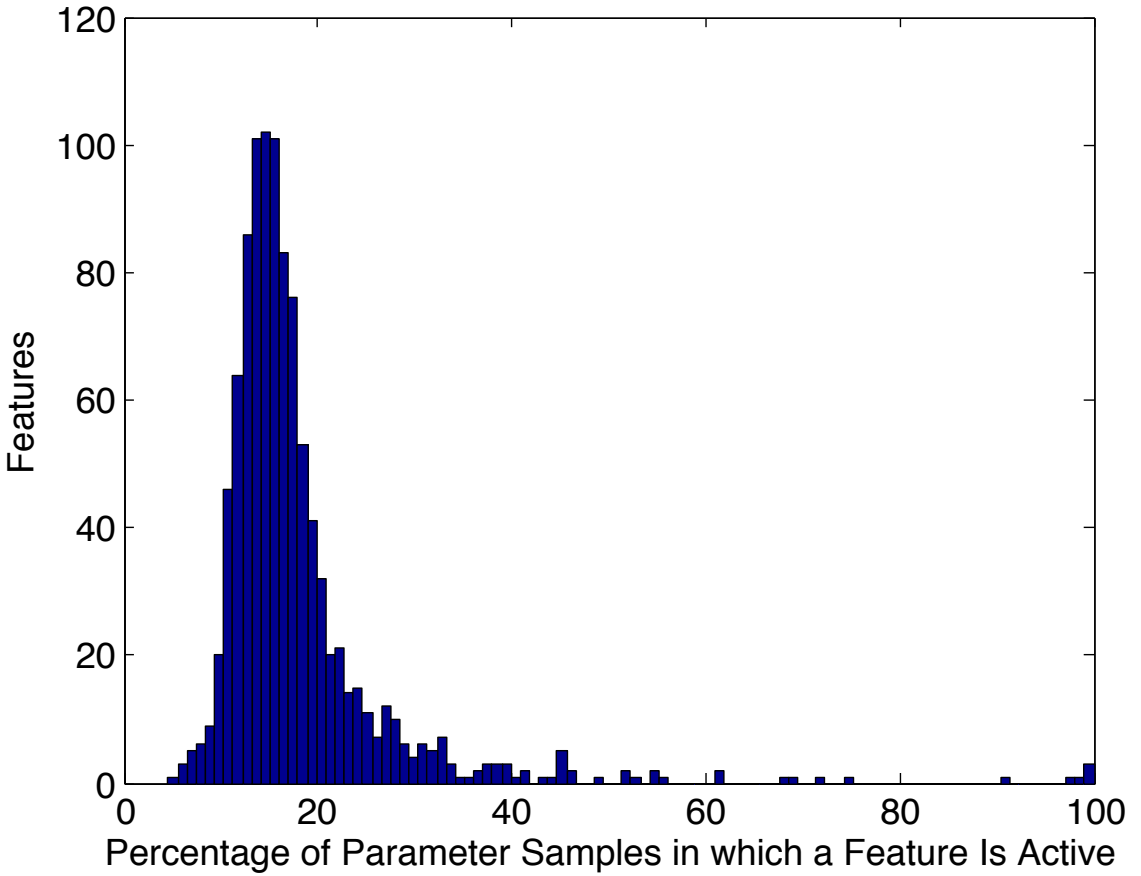
Method	Total	Relative Improvement	Tissues			
			CNS	Muscle	Embryo	Digestive
Boosting	393	baseline	187	69	76	61
SVM with 40 PCA	466	19%	211	93	93	69
Non-Bayesian NN (K = 2)	483	23%	211	112	82	78
Non-Bayesian NN (K = 10)	483	23%	215	112	90	66
Non-Bayesian NN (K = 10) (Bagged)	541	38%	229	125	98	89
Bayesian NN (K = 2)	486	24%	230	95	99	62
Bayesian NN (shared) (K = 2)	538	37%	235	111	108	84
Bayesian NN (shared) (K = 10)	598	52%	256	122	123	97

Prediction of sampled models



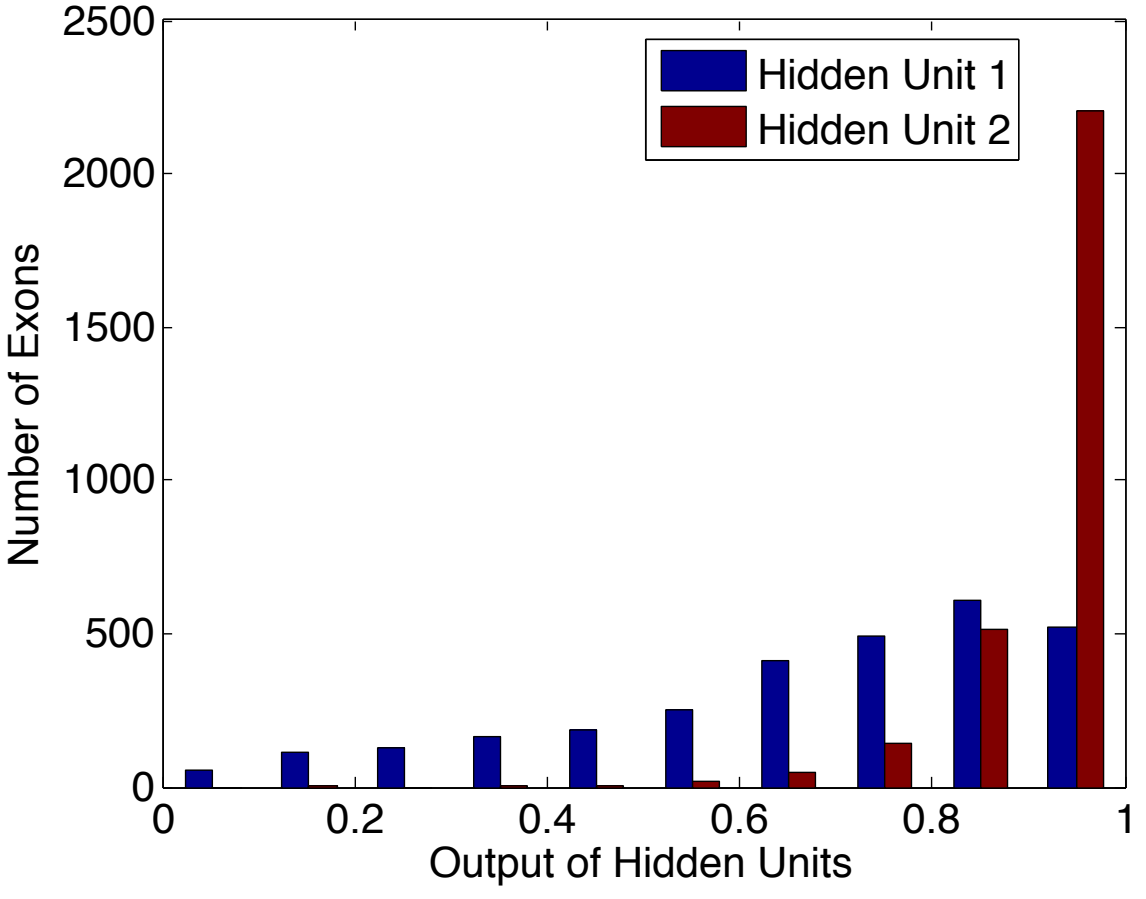
- Averaging predictions dramatically increase performance

Active features



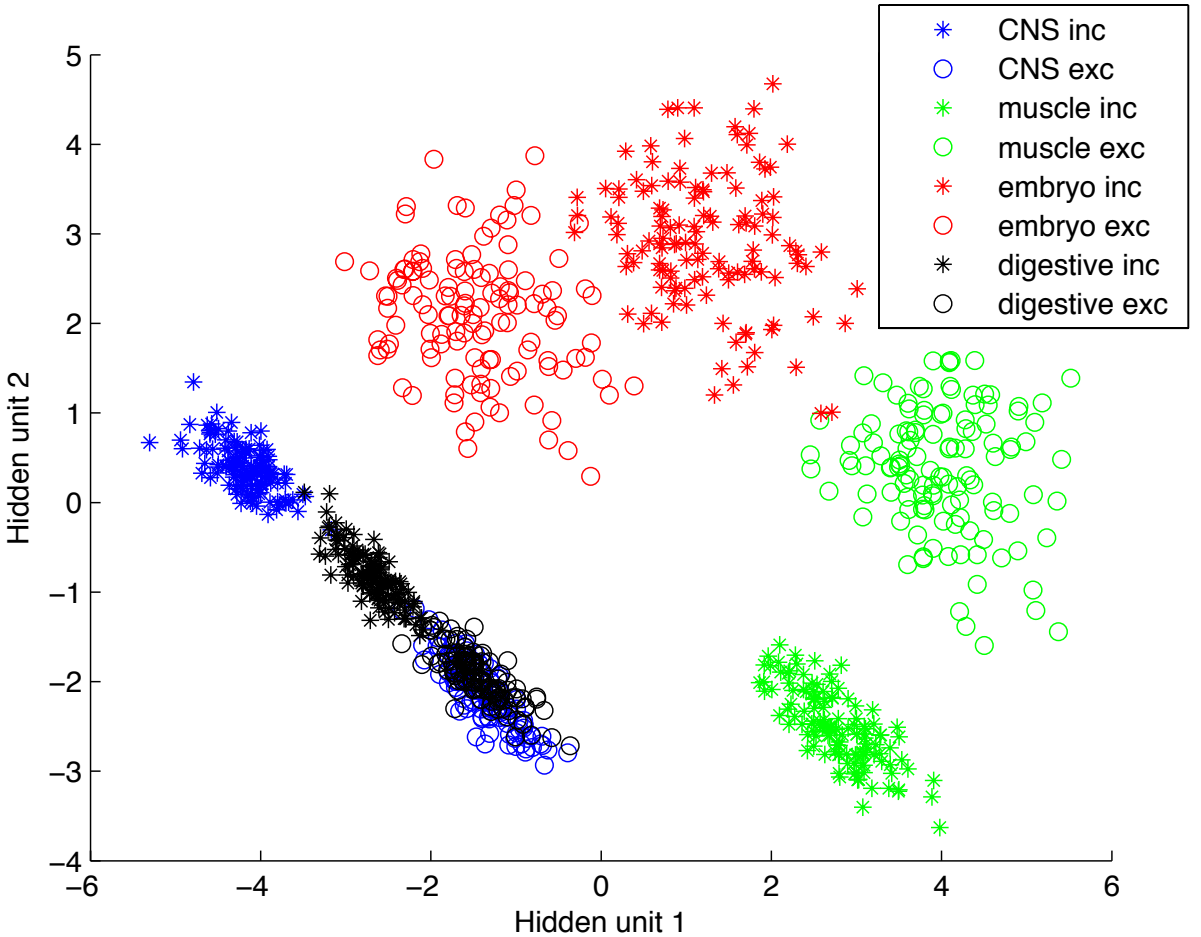
- Most features are not active at any given sample and some features are almost always active
- This identifies important features governing alternative splicing

Non-linear hidden variables



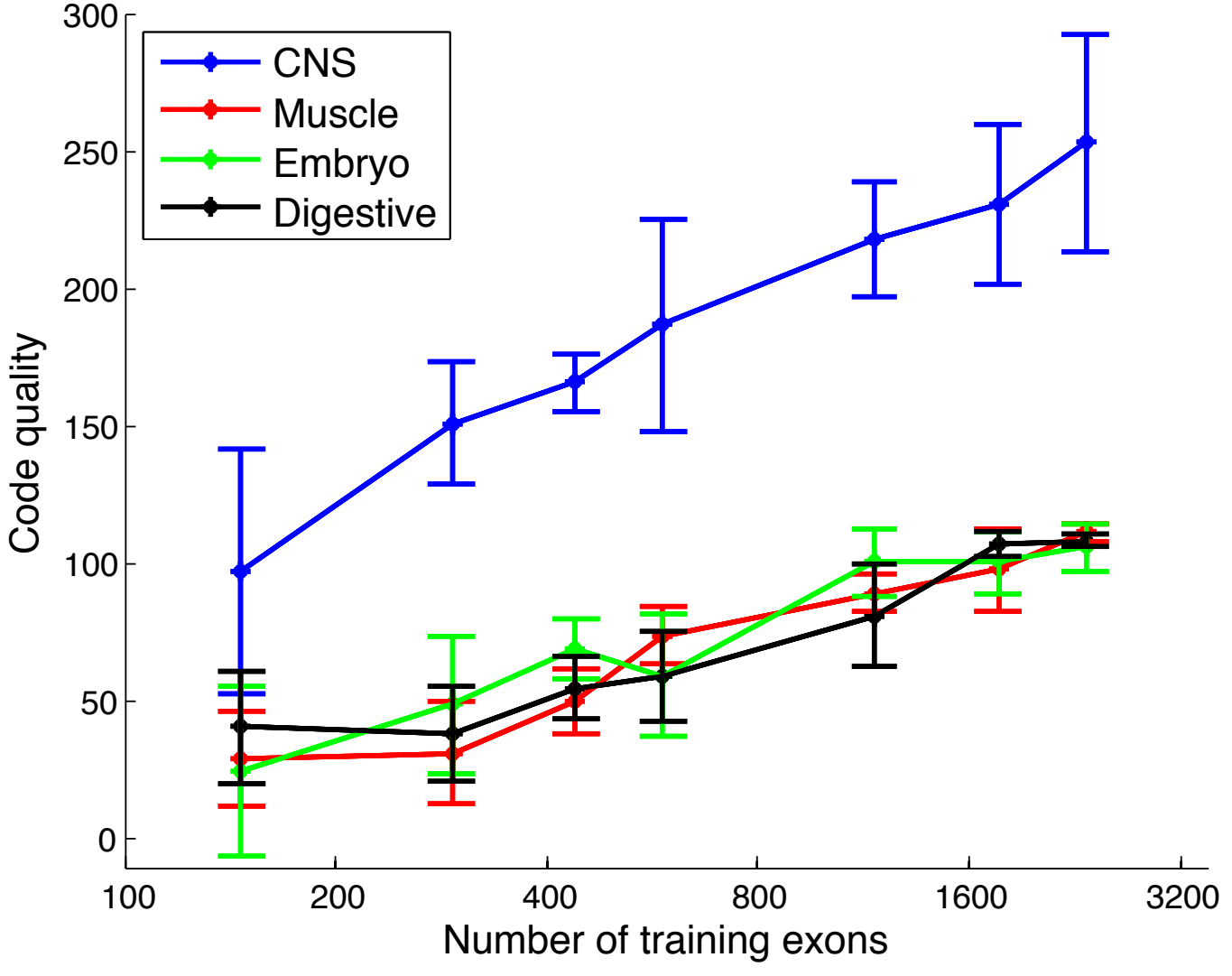
- Exons in the first hidden unit has a wide distribution
- Most exons in the second hidden unit are saturated

Feature combinations



- ‘Meta features’ responsible for AS is identified using hidden variables
- CNS exclusion and digestive inclusion are similar

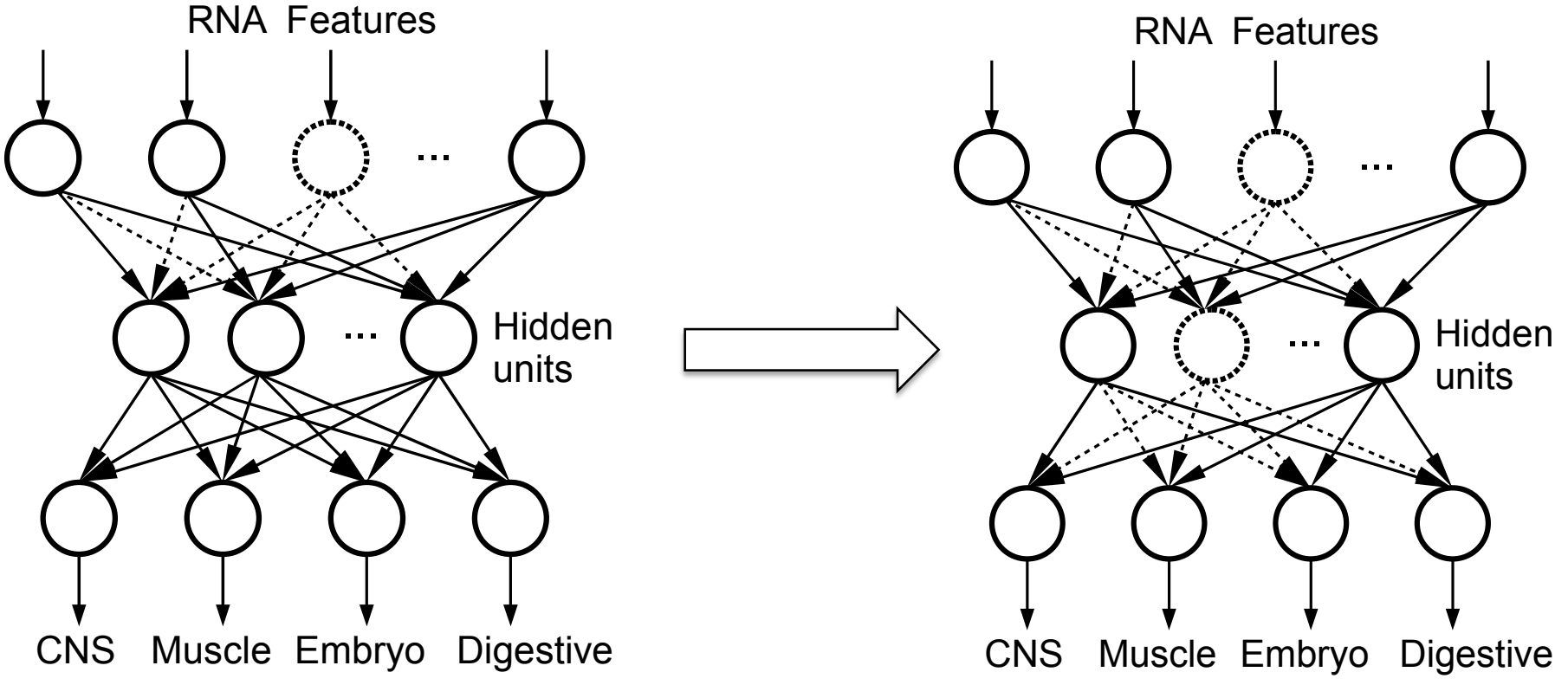
Do we have enough training data?



Conclusion

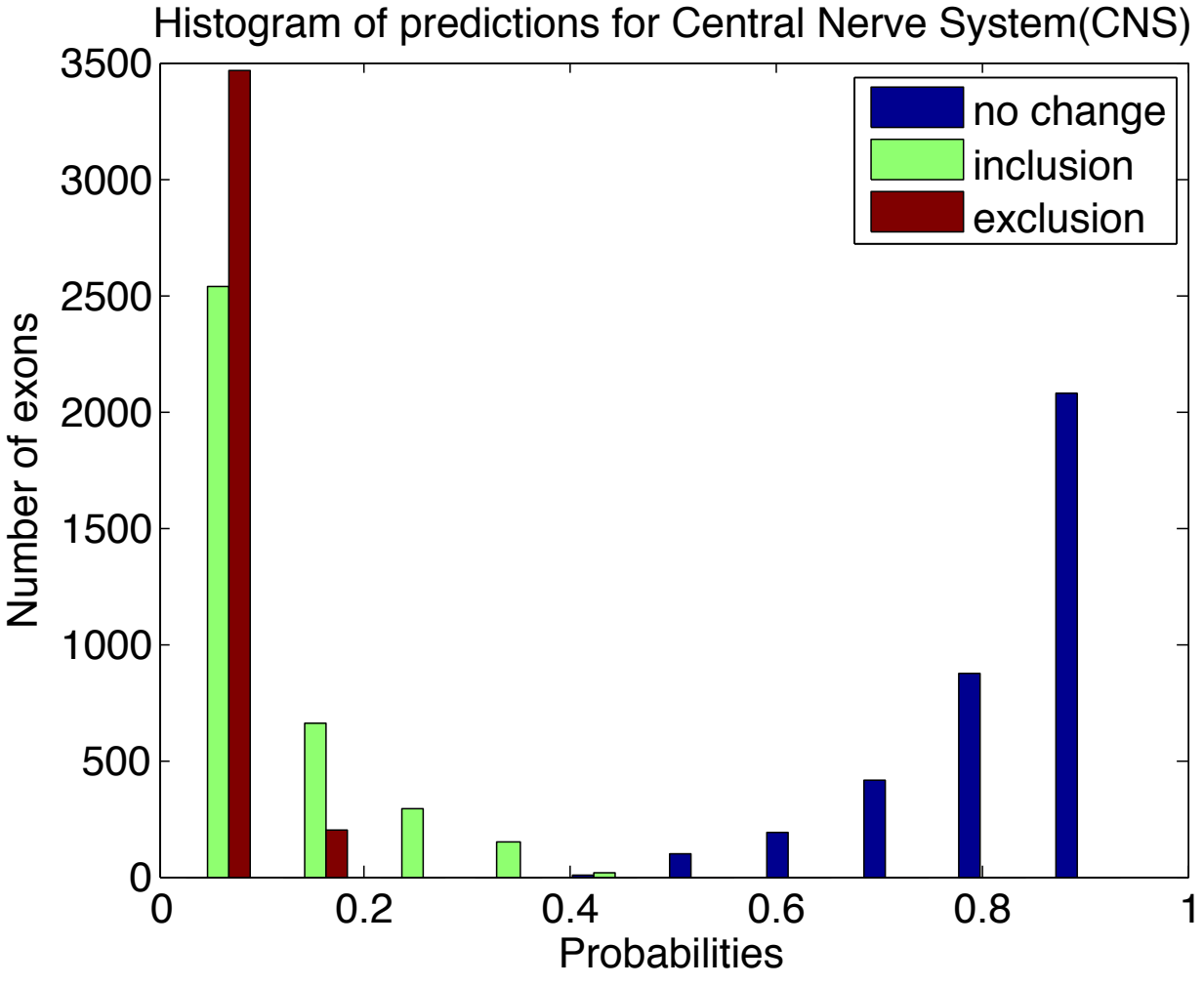
- Bayesian neural network is 50% better at learning the splicing code than boosting
- Useful RNA features for alternative splicing are identified
- ‘Meta features’ were assembled and their influence on splicing is visualized
- Dataset: genes.toronto.edu/wasp

Future Direction



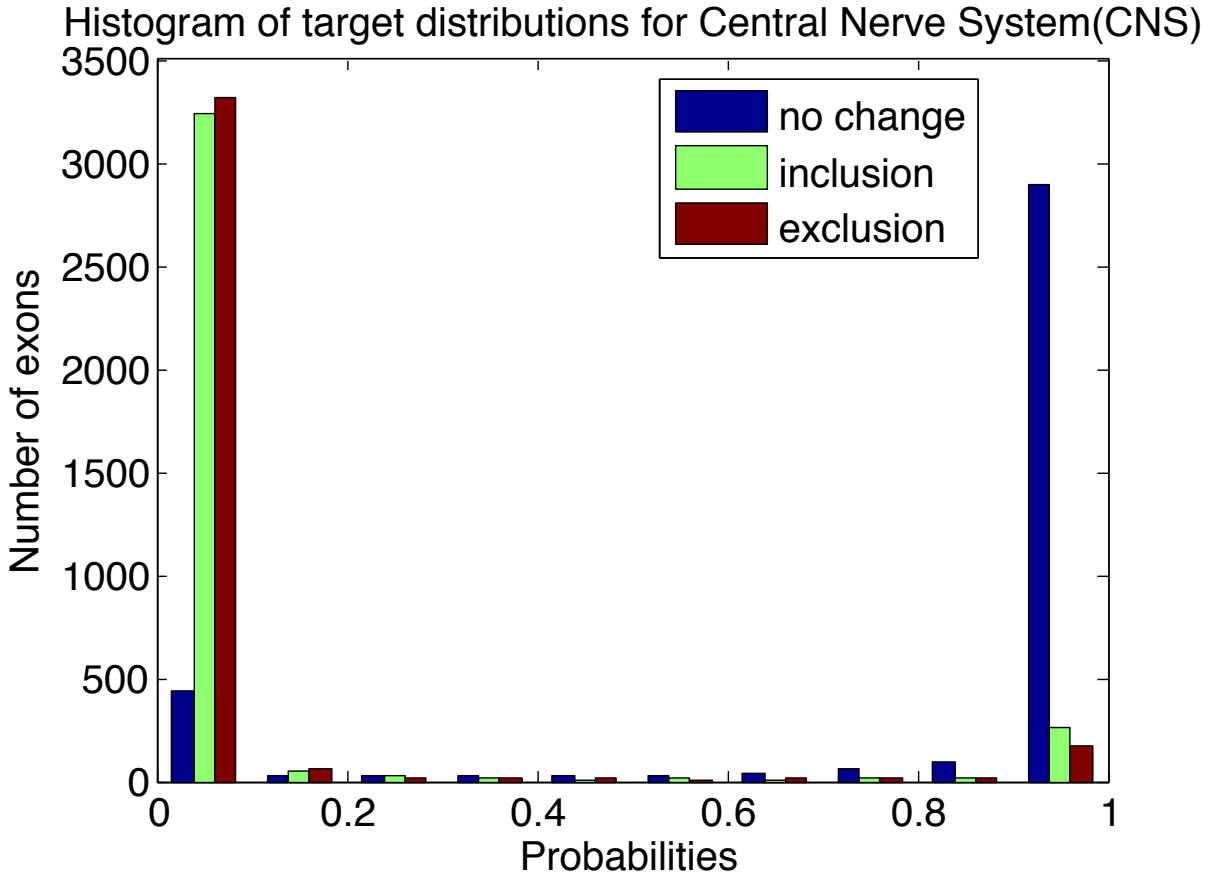
- Allow tissues to selectively share hidden units
- Investigate effects of feature combinations

Predictions



- No extreme predictions are made
- More confident in CNS inclusion (same result in ROC)

Targets

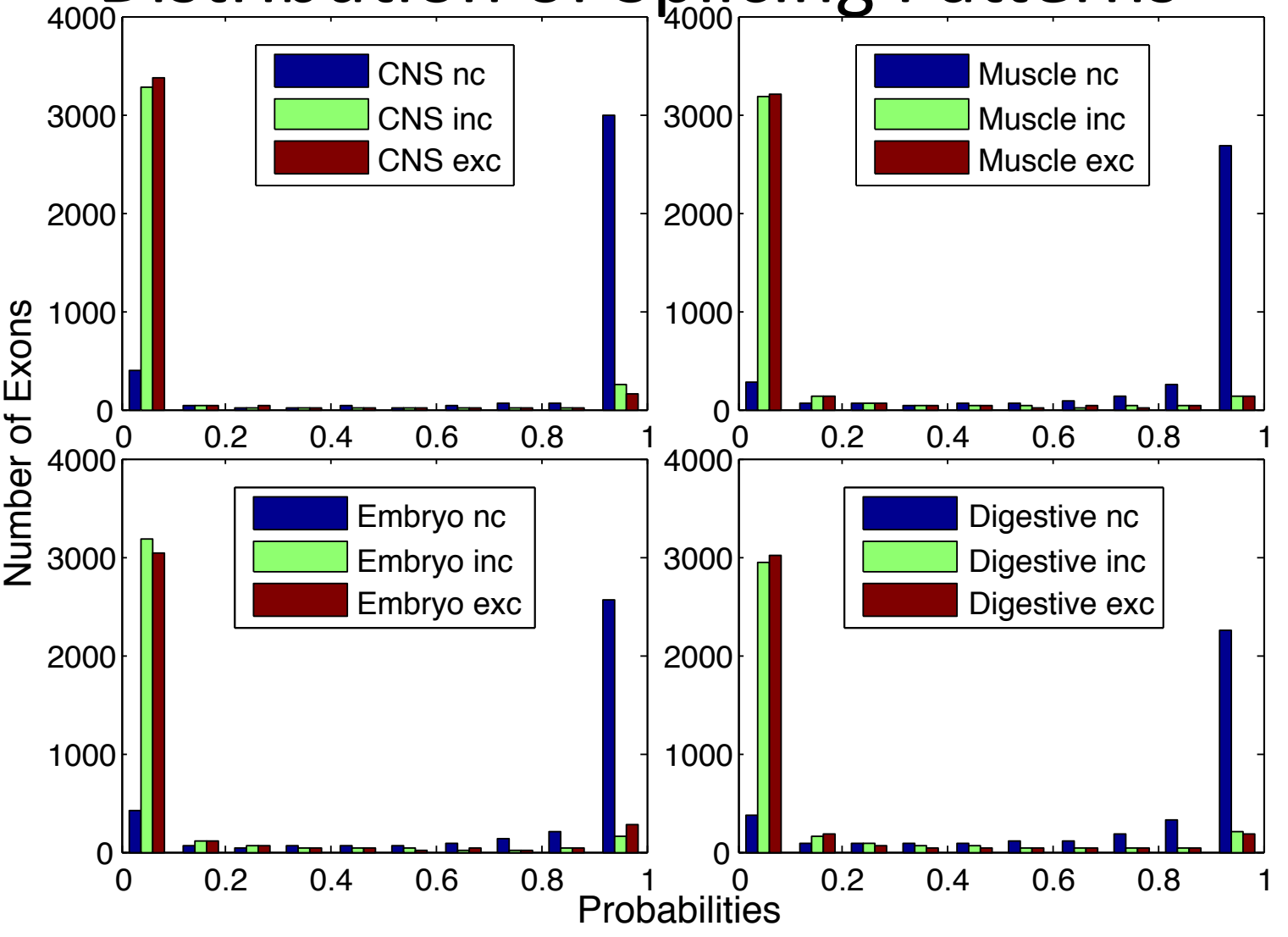


- Most probabilities are extreme

Outline

- Introduction and problem setup
- Alternative splicing dataset (Barash et al., Nature 2010)
- Bayesian Neural Networks with Gibbs sampling
- Comparison of performance
- Visualization of the learned splicing model

Distribution of Splicing Patterns



- Most splicing patterns are extreme
- ‘no change’ is the most frequent category

Algorithm of the First Splicing Code

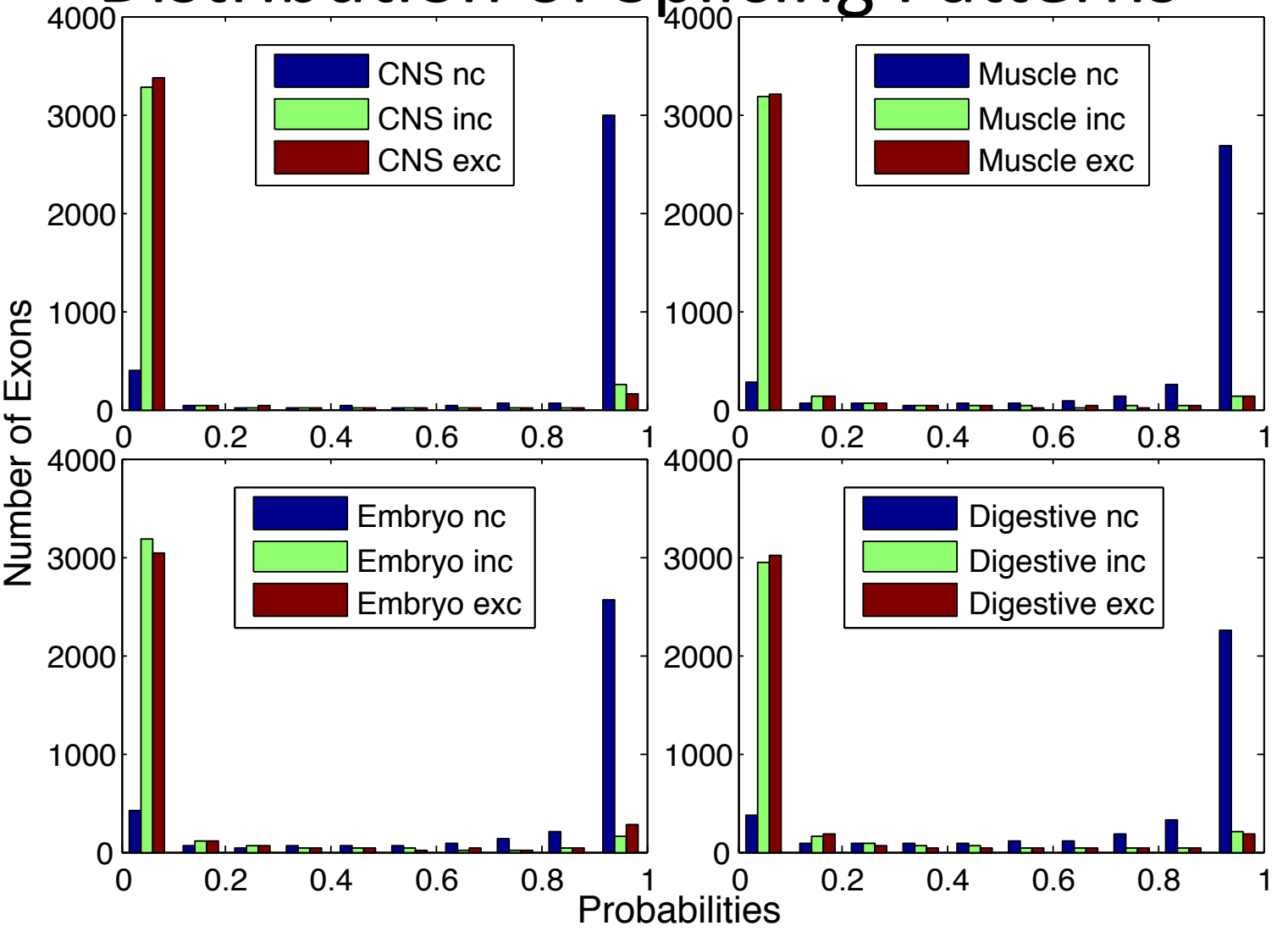
- This algorithm is based on Boosting (Barash, et al)
- Weak classifiers: pick a single feature and optimize a threshold for best code quality
- About 200 weak classifiers used in each tissue
- Overfitting was not observed
- Final test results was about 100 bits/tissue although getting near 100% performance in training (about 2000 bits/tissue)

Discussion

- Bayesian method is better than non-Bayesian method and bagged versions of non-Bayesian method

- Sharing hidden units improves performance, suggesting same set of RNA features influence splicing change across tissues

Distribution of Splicing Patterns



- Most splicing patterns are extreme
- 'no change' is the most frequent category