

Methodologies for Constructing and Training Large Hierarchical Hidden Markov Models for Sequence Analysis

Chris Pal and Mike Hu, University of Waterloo

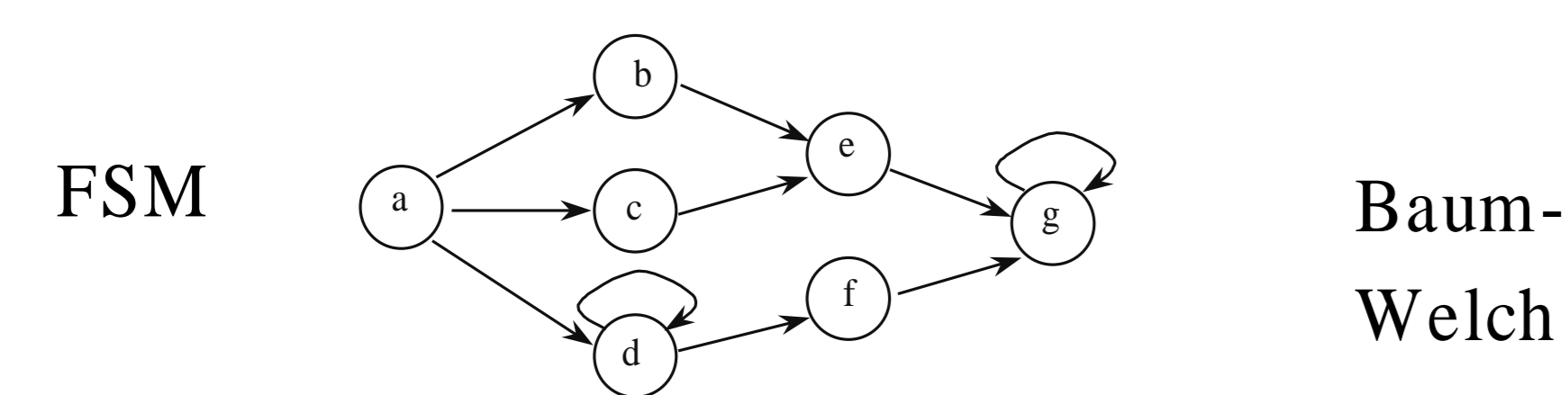
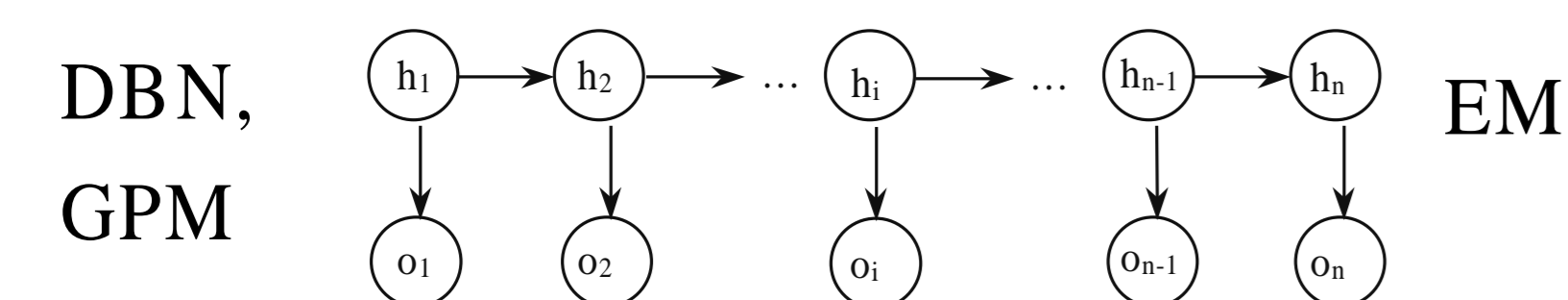
Introduction

Hidden Markov Models (HMMs) are a widely used modeling tool for biological sequence analysis [Burge], [Kulp]. However, for many tasks of interest large hierarchical models must be constructed and optimized using a large amount of training data. We present some methodologies that we have used for constructing and training large HMMs for biological sequence analysis. We illustrate these techniques for the task of splice site prediction in vertebrate genes.

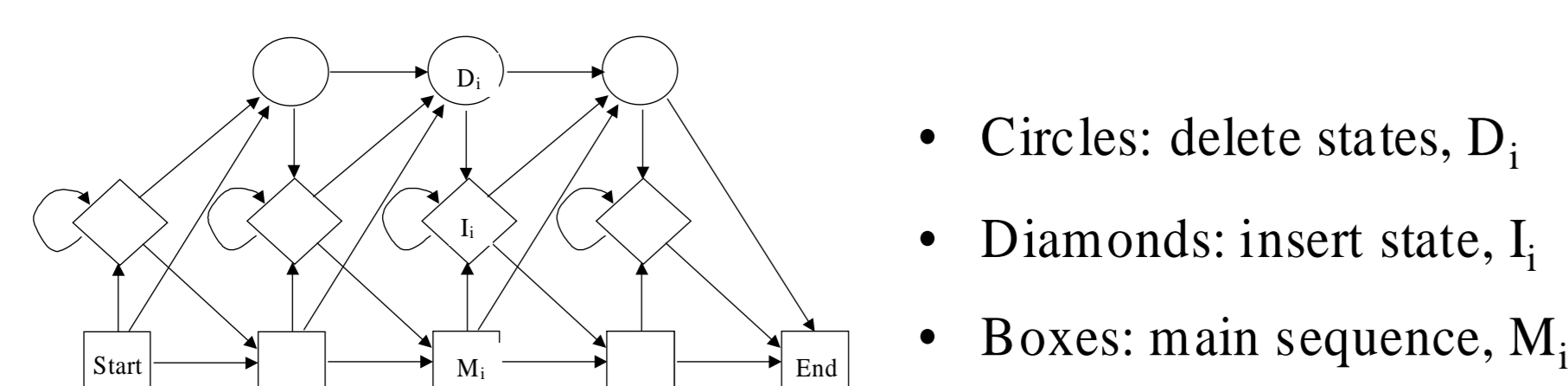
Modeling Methodologies

Hidden Markov models can be described graphically using a number of different types of graphical notation. HMMs can be illustrated using: stochastic finite state machines (SFSMs) [Durbin], dynamic bayesian networks (DBNs) [Russell] and graphical probability models (GPMs) [Frey]. We show how these different representations can be used to help the modeler think about issues such as model structure, parameterization and model complexity.

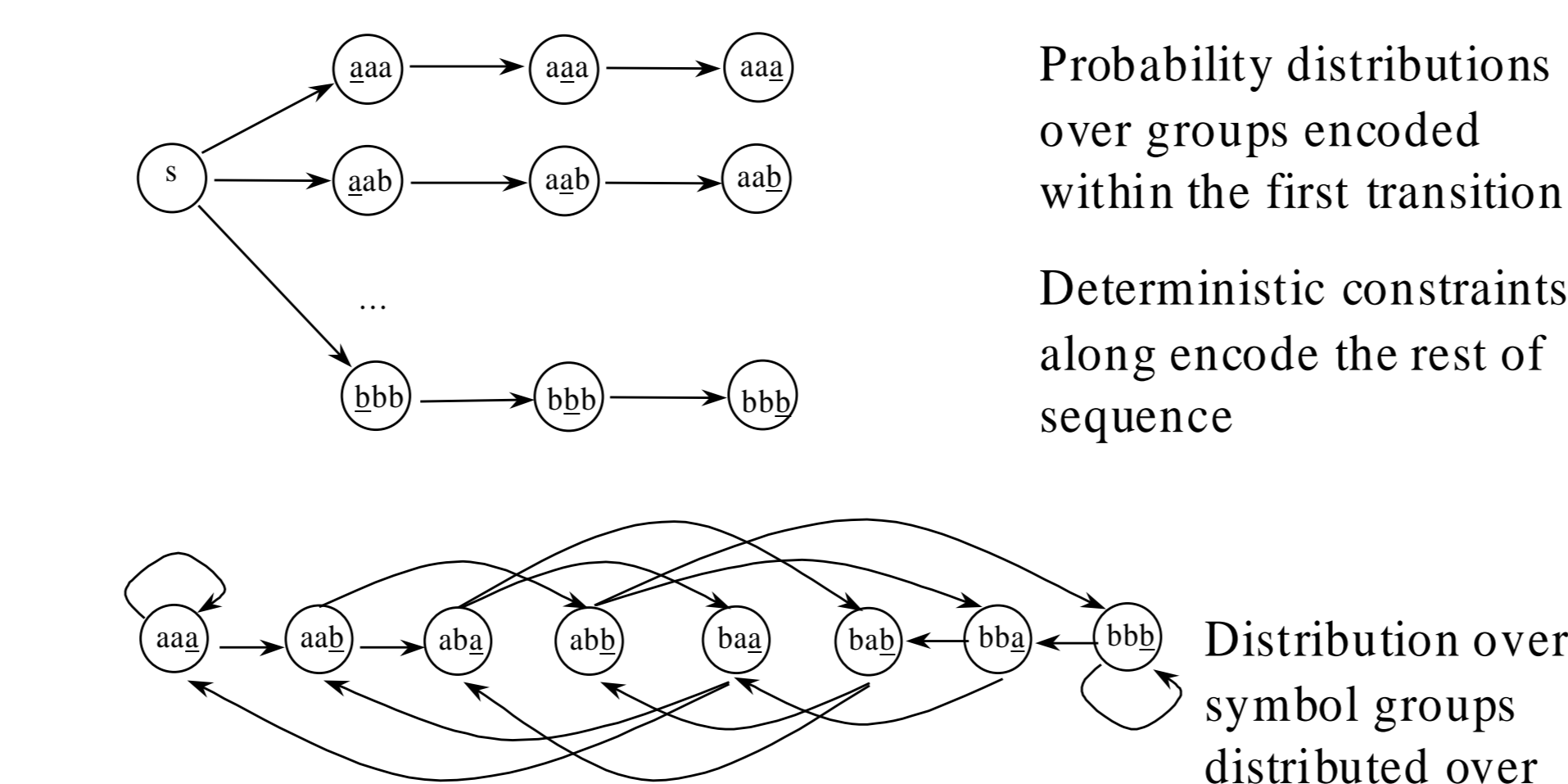
For modeling complex patterns, often it is important to maintain different orders of memory in order to capture various sub-structures of more general patterns. Further, in some cases one may wish to construct models in which symbols are emitted one at one time, while at the same time retaining memory of n-previous symbols. Thus, simple HMMs emitting composite symbols cannot be used. We illustrate some methodologies for constructing HMMs that retain different levels of memory for different regimes of the overall pattern being modeled.



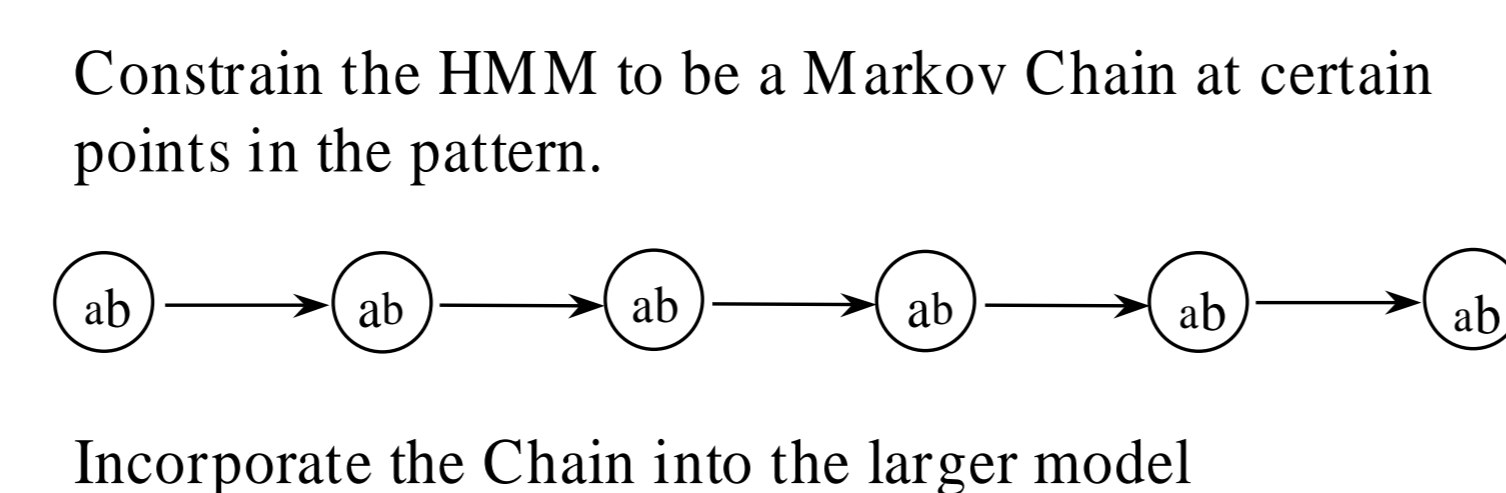
Example: Profile HMMs (FSM Notation)



Higher Order Models (SFSM Notation)



Distributions over Blocks of Symbols



Training Methodologies

Large HMMs can be composed of smaller sub-components. Additionally, it is sometimes convenient to think of large HMMs as being hierarchically structured. However, for implementation purposes such hierarchical models can be flattened into a standard HMM. Using this approach standard algorithms can be used for training and the analysis of new sequences.

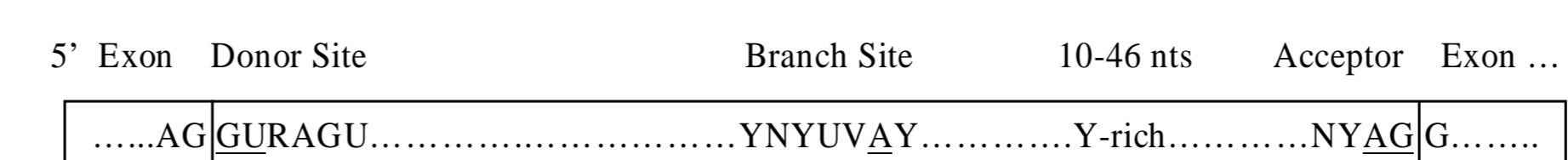
When large, conceptually hierarchical HMMs are trained there are a number of techniques that can be employed to more accurately fit model parameters. The standard procedure for learning parameters in a HMM may be viewed as an instance of the Expectation Maximization EM algorithm [Bilmes] or the Baum-Welch algorithm [Durbin]. Here, we also show some advantages of viewing the parameter estimation problem as an instance of the EM algorithm.

When training a HMM it is helpful to utilize all information that is known about the sequence. For example, in our splice site prediction experiments a data set of genes with the intron and exon boundaries indicated is used as training data. Two separate HMMs for introns exons were constructed. The intron model was trained on intron nucleotide symbols where each segment of exon symbols was replaced by a single symbol indicating an exon. The intron model thus contained a single state for exons that deterministically emitted the exon symbol. A similar construction was used for training the exon model. In this way, during the training phase, the model was assured to always utilized the correct intron-exon boundaries. Without this step, the training phase the algorithm could easily infer an incorrect segmentation.

Vertebrate Gene Splice Site Prediction

We illustrate the modeling methodologies and training techniques described previously for the task of predicting vertebrate gene splice sites. Here the problem is a matter of determining how to use prior knowledge of the patterns found in introns and exons in order to predict intron and exon boundaries. We compare our results with other sequence analysis models.

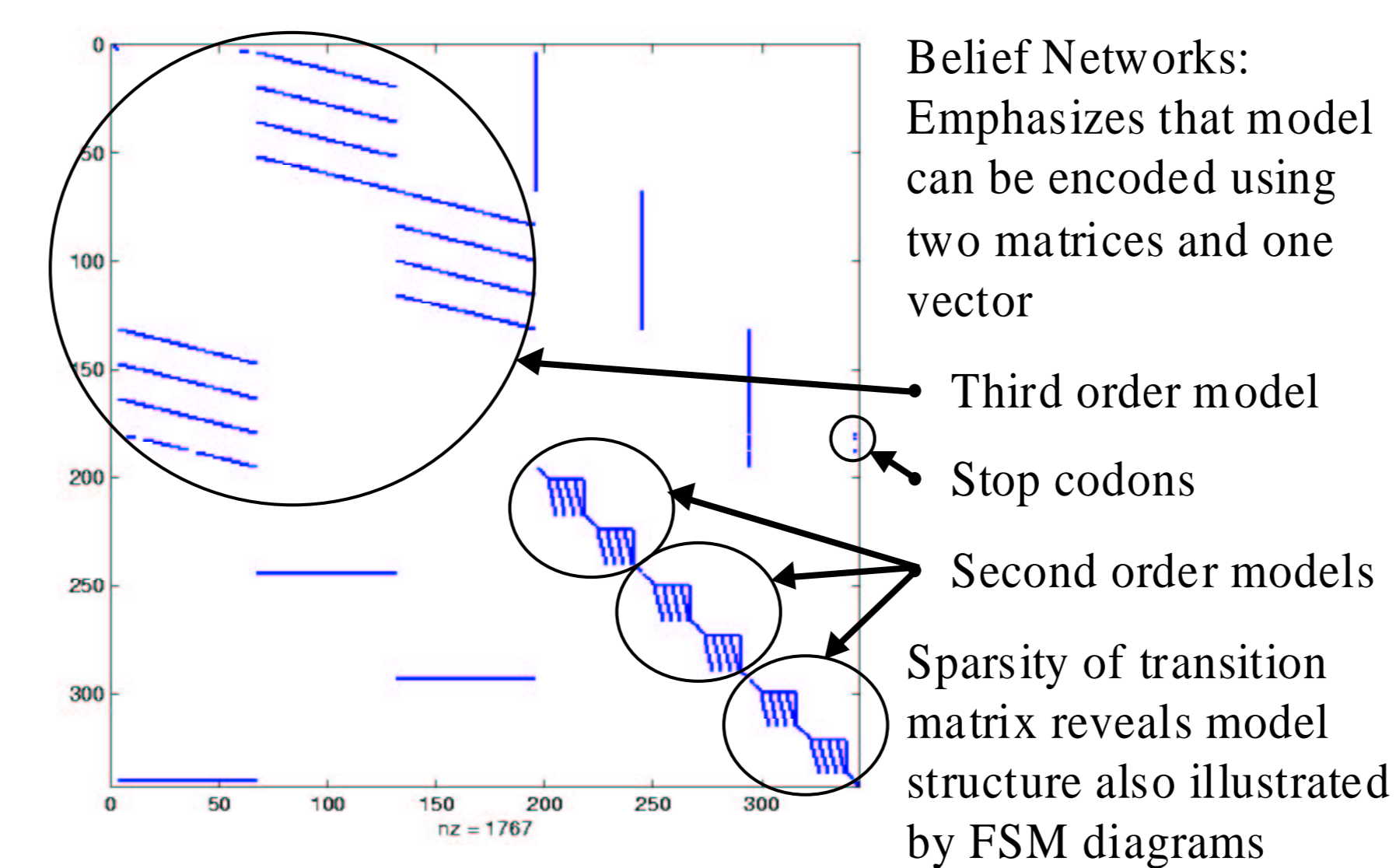
Modeling Features of the Splice Site



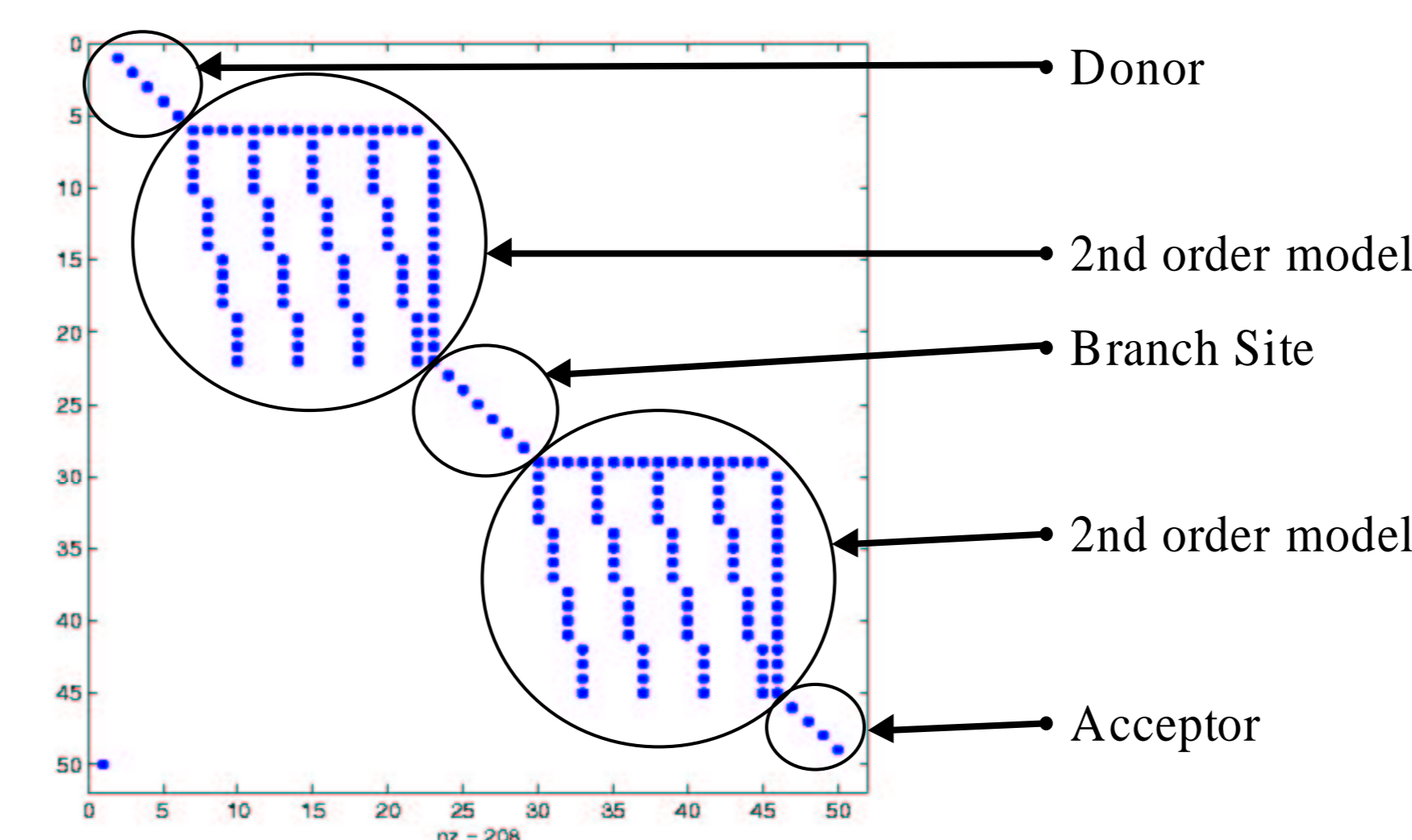
R = G or A N = A, C, G or T
Y = U or C V = G, C or A

Underlined nucleotides are required features

Complete Transition Matrix Structure



Intron Model Transition Matrix



Results

- Tested on remaining 20% of data
- Per nucleotide tests (Intron or Exon)
- True Positives (TP), False Negatives (FN) and False Positives (FP)
- Specificity (S_p) and Sensitivity (S_n)

$$S_p = \frac{TP}{TP + FP} = \frac{\text{true coding nucleotide } s}{\text{total predicted to code}}$$

$$S_n = \frac{TP}{TP + FN} = \frac{\text{true coding nucleotide } s}{\text{all nucleotide } s \text{ analyzed}}$$

	S_n	S_p
Our Model	.74	.88
GRAIL 2	.69	.85
GeneID	.58	.78
GeneID+	.85	.85
GeneParser3	.83	.91

- Note: Larger Task, Predicting coding regions
- Newer Systems: Genie (Haussler, UC Santa Cruz), and GeneScan (Burge, Stanford)

Acknowledgements

The authors would like to thank the following members of the University of Waterloo community for their assistance with this work: Chris Ingram and Sajani Swamy (Computer Science), Monica Sirski (Statistics), and Cheryl Patten (Biology).

This work was supported in part by funding from: the Natural Sciences and Engineering Research Council of Canada (NSERC) and Communications and Information Technology Ontario (CITO).

References

Bilmes, J. 1998. *A Gentle Tutorial of the EM Algorithm*, UC-Berkeley TR-97-021.

Burge, C. and Karlin, S. 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Bio.* 268, 78-94.

Frey, B. 1998. *Graphical Models for Machine Learning and Digital Communication*. MIT Press: Cambridge, MA.

Durbin, et. al. 1998. *Biological Sequence Analysis*. Cambridge University Press.

Kulp, et. al. 1996. A Generalized HMM for the Recognition of Human Genes in DNA. *ISMB 1996*, 4:134-142.

Russell, S and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

A Complete View of the HMM Structure Illustrated as a Stochastic Finite State Machine

