

# Gene Finding

Partly based on

*Computational Prediction of Eukaryotic Protein-Coding Genes*

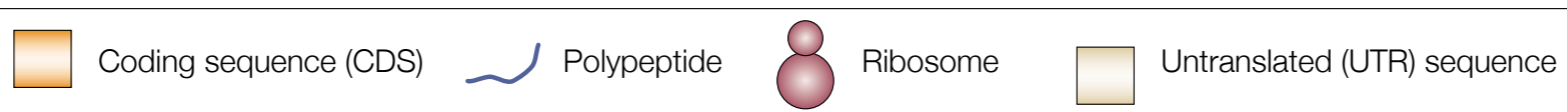
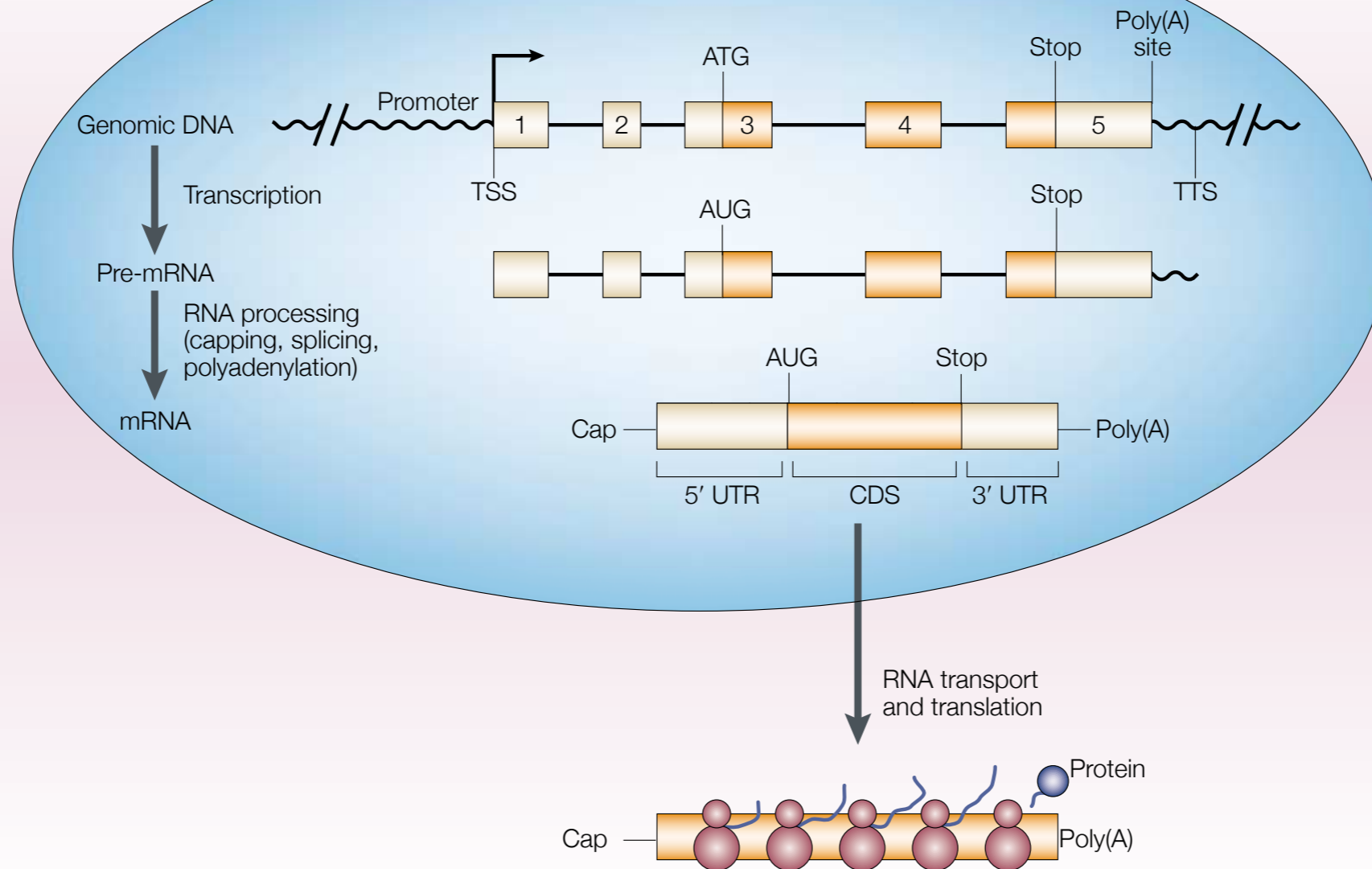
Michael Q. Zhang

Nature Genetics v. 3, pp.698-709, 2002

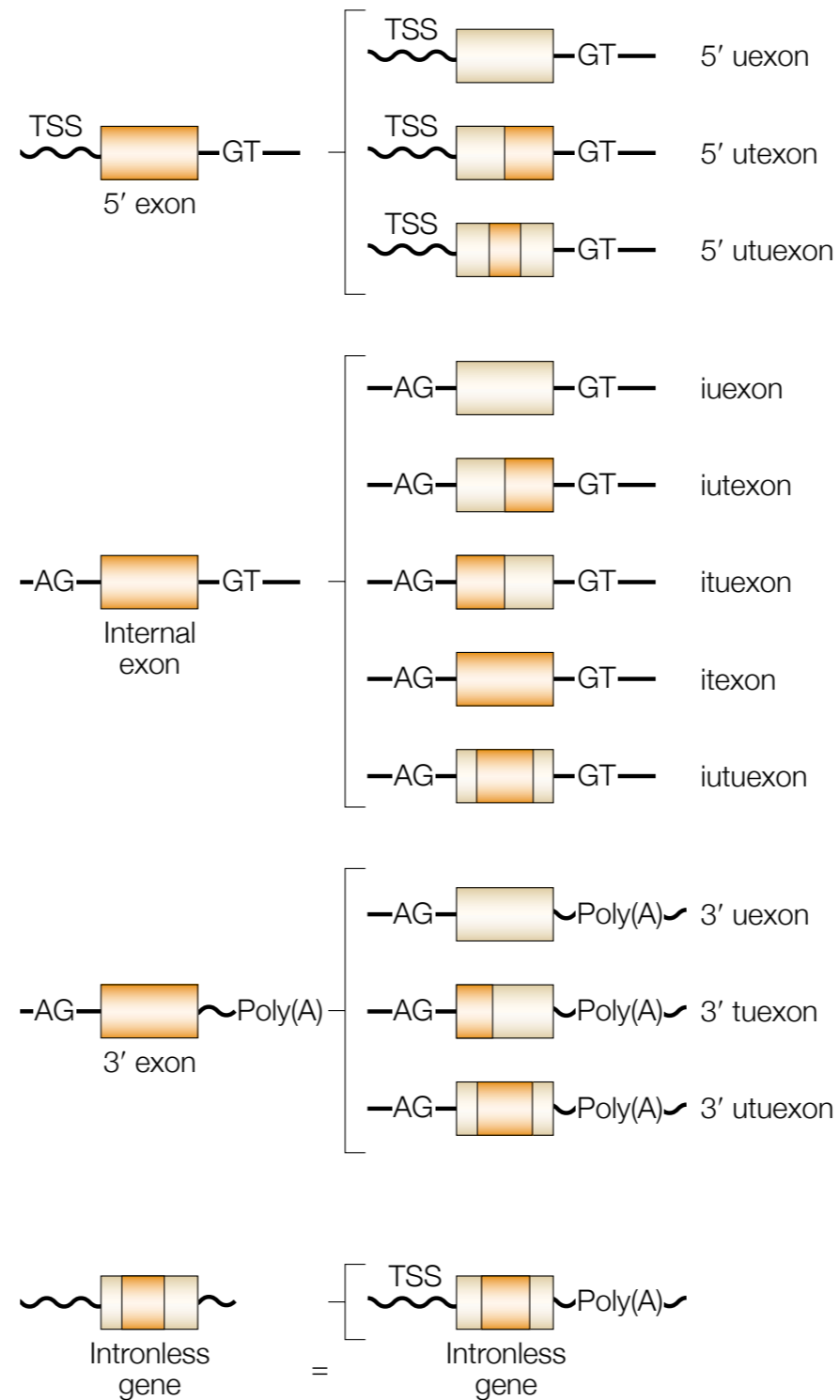
# Gene Expression

Cytoplasm

Nucleus

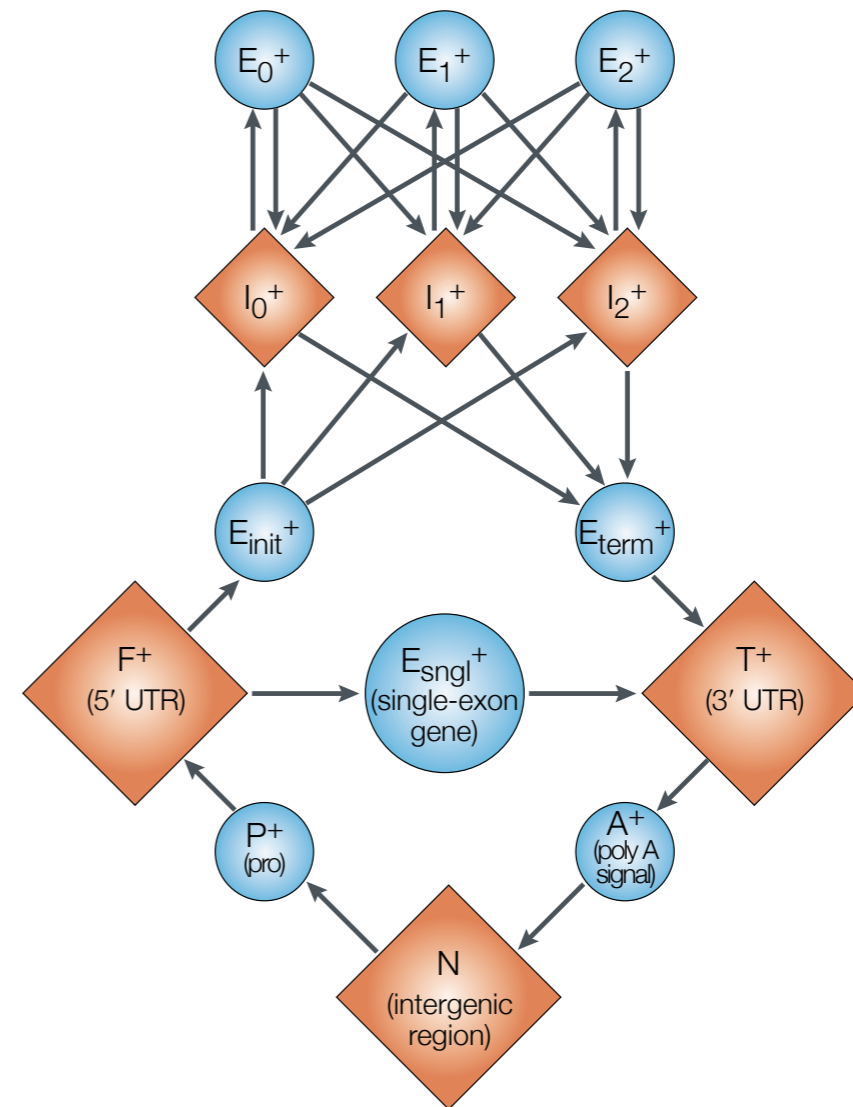


# Exon types



# GENSCAN HMM

- Semi-Markov model: each state generates an observation sequence
- $P(\text{seq}|\text{state})$  includes state-length distribution
- Transitions are taken on *signals* indicating change of state (eg. exon to intron)



Reverse strand: mirror reflection of above

# Global constraints

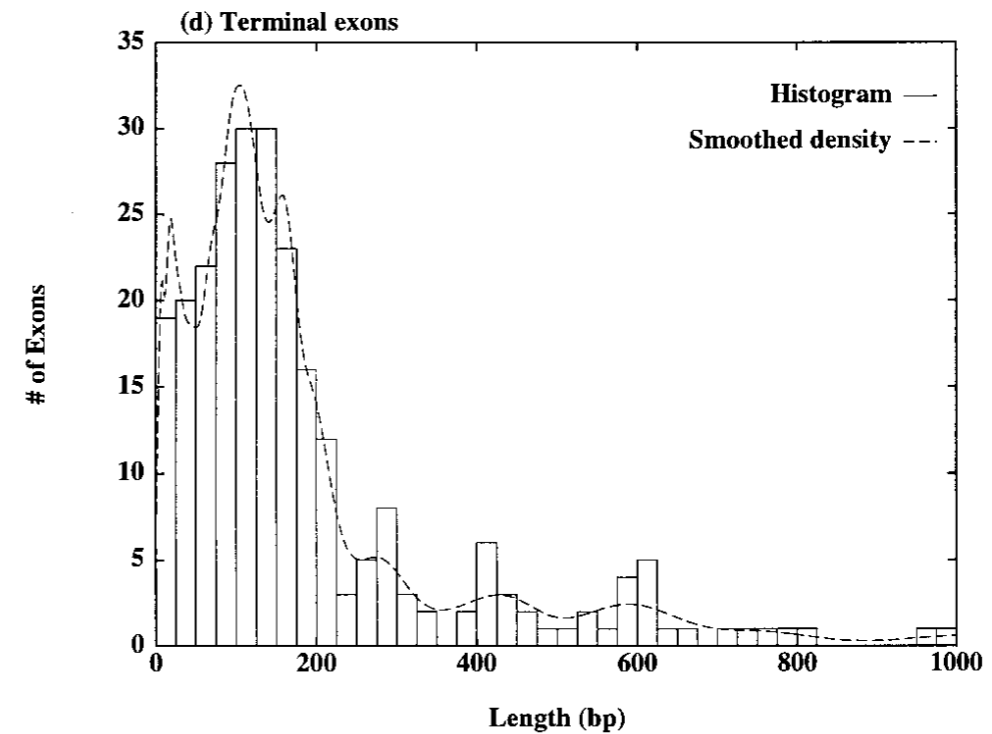
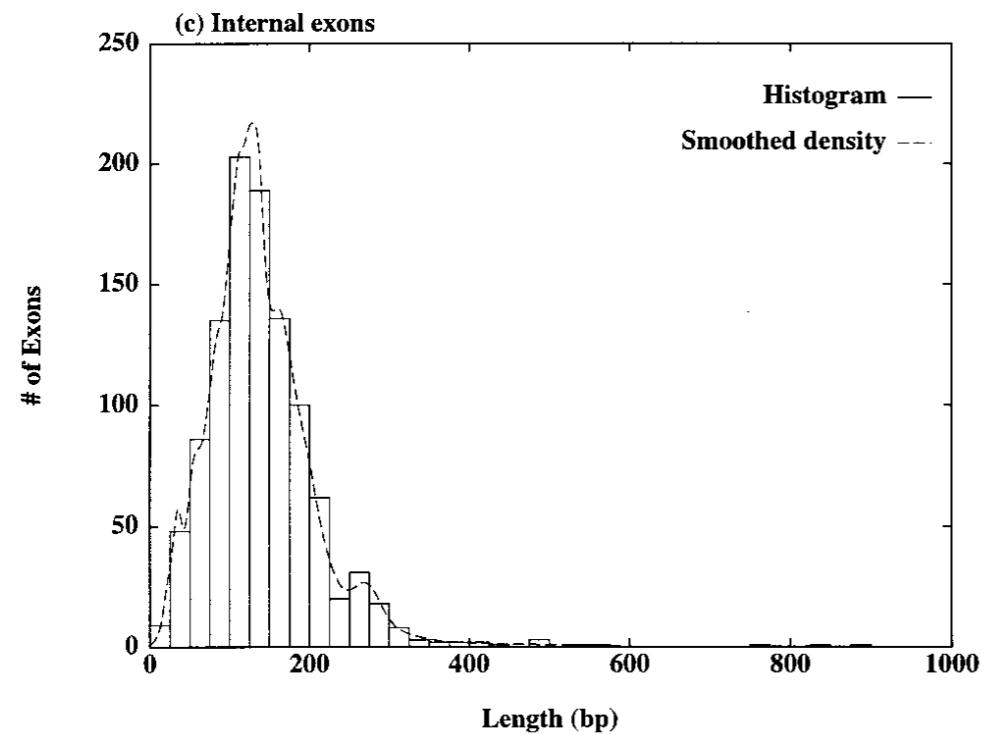
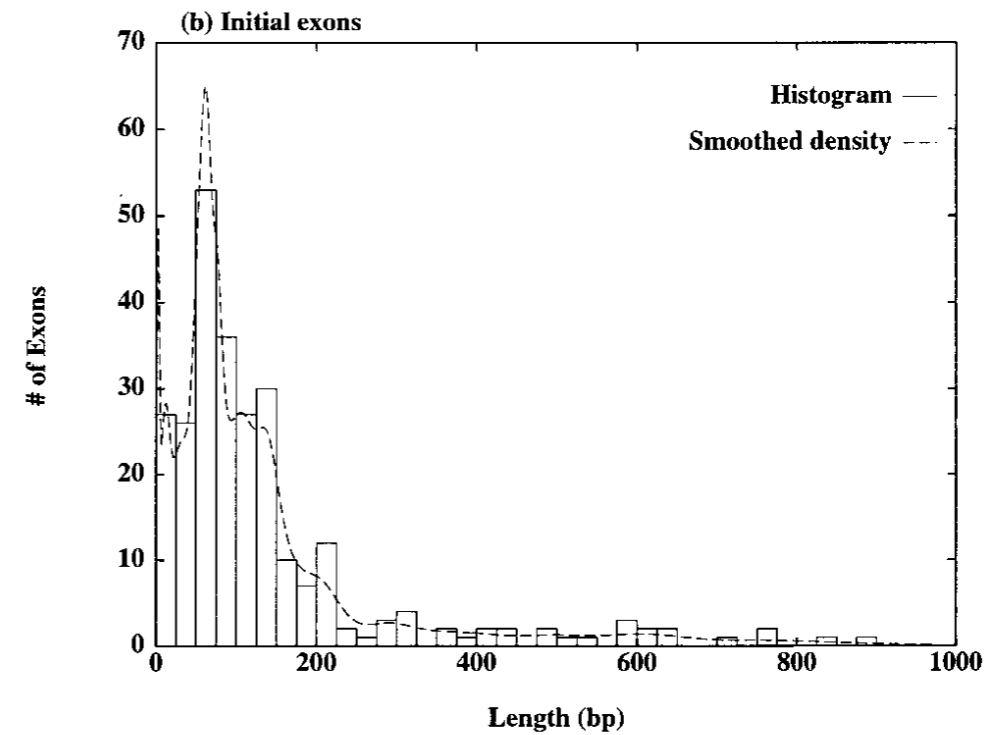
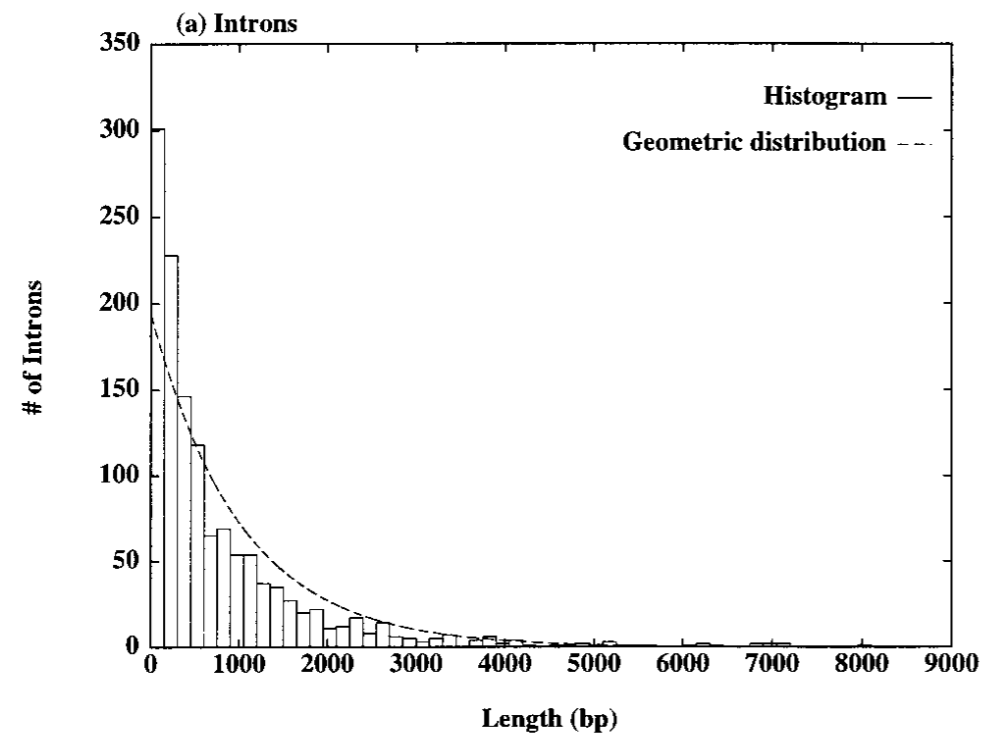
- Integrate models for both strands:
  - Forward on one strand
  - Reverse on the other (reverse of an HMM is an HMM)
  - Prevents conflicting (overlapping) predictions on the two strands
- A harder problem: global correlations
  - Exons
  - Genes in same genomic region

# Semi-Markov models

- Model generates a *parse* of a genomic sequence
  - state sequence
  - sequence of corresponding lengths
  - nucleotide subsequence for each state

$$P(\mathbf{s}, \mathbf{d}, \mathbf{x}) = \prod_i P(s_i | s_{i-1}) P(d_i | s_i) P(x_{k_i} \cdots x_{k_i + d_i - 1} | s_i, d_i)$$
$$k_i = k_{i-1} + d_{i-1}$$

# Length distributions



# State model

- How to model the probability of a nucleotide sequence given a state and length?

- States: Markov models

$$P(x_{k_i} \cdots x_{k_i+d_i-1} | s_i, d_i) \approx \prod_{j=k_{i-1}+1}^{k_i} P(x_j | x_{j-1}, \dots, x_{j-m}, s_i, d_i)$$

- Signals: weight matrices for positions adjacent to transition

# Generalized HMMs

- Focus on the probability of a parse *given* the sequence
- Leaving *generative model* allows much richer dependencies on nucleotide sequence

$$P(\mathbf{s}, \mathbf{d} | \mathbf{x}) \propto \prod_i T(s_i | s_{i-1}, d_{i-1}, \mathbf{x}) S(d_i | s_i, \mathbf{x})$$

- Transition ( $T$ ) and state ( $S$ ) scoring functions may use any suitable model (weight matrix, Markov, neural network, non-parametric)

# GHMM training

- Transition scores:
  - Parametric functions of signal scores
- Train separate signal and state scoring functions
- Adjust transition parameters to maximize likelihood of correct parse
- *Potentially better*: integrated model