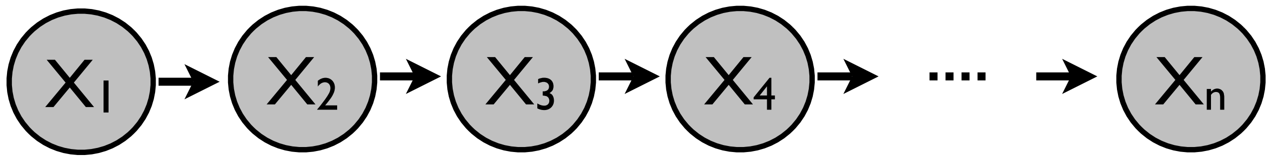
BMI/CS 776
Lecture 12
Eukaryotic Gene Finding

Colin Dewey

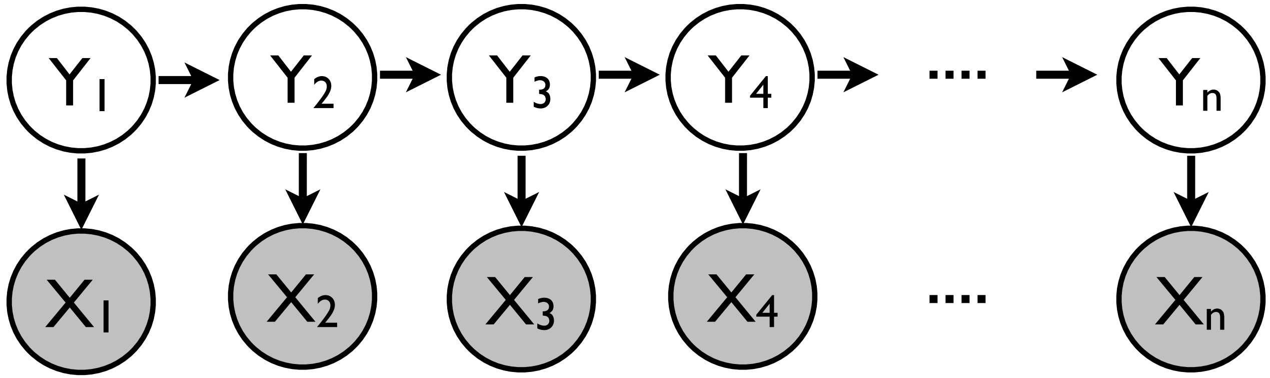
February 28, 2008

Markov models & hidden Markov models for sequences

- Markov model:
  - states are the sequence characters (observed)

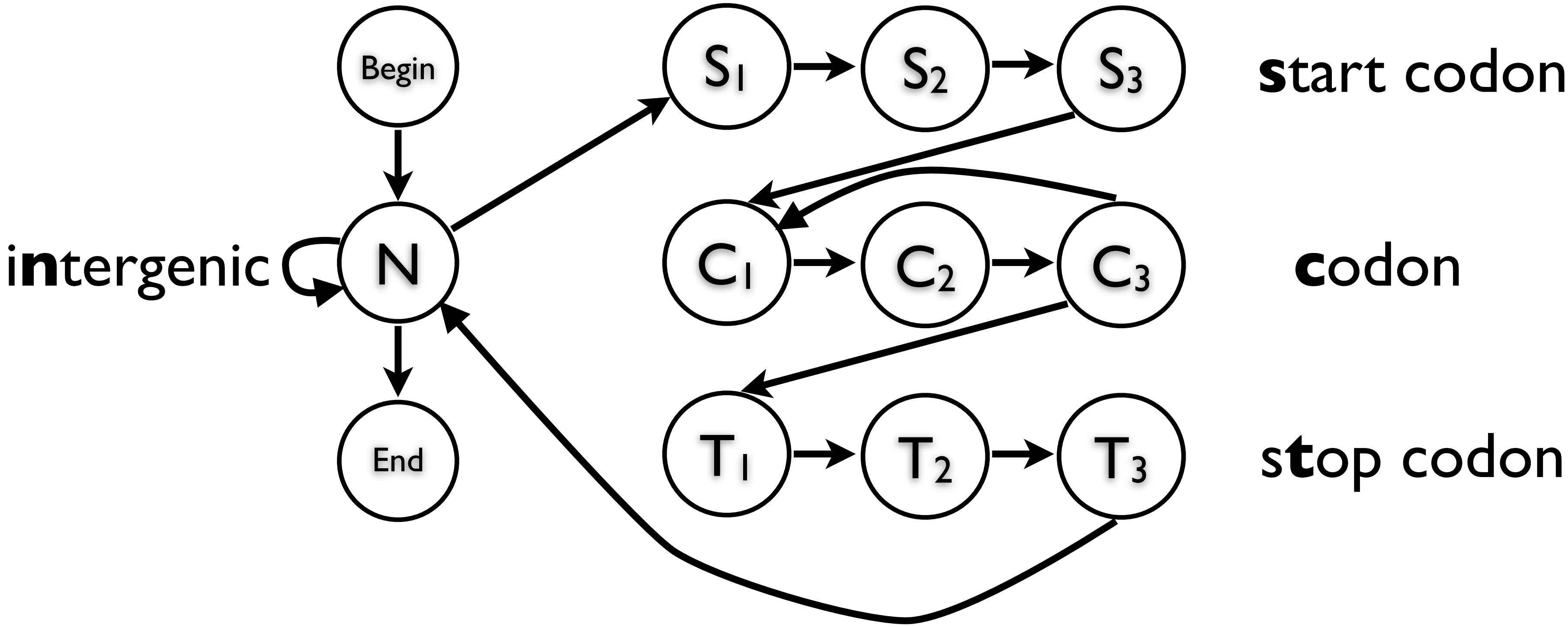


- hidden Markov model (HMM):
  - states are sequence *classes* (hidden)
  - *classes* “emit” observed sequence characters



HMMs for gene finding

- *State transition diagram* (not the graphical model!) for simple Prokaryotic gene HMM



Viterbi parses

- Use Viterbi algorithm to calculate most likely joint configuration of hidden states (a parse)

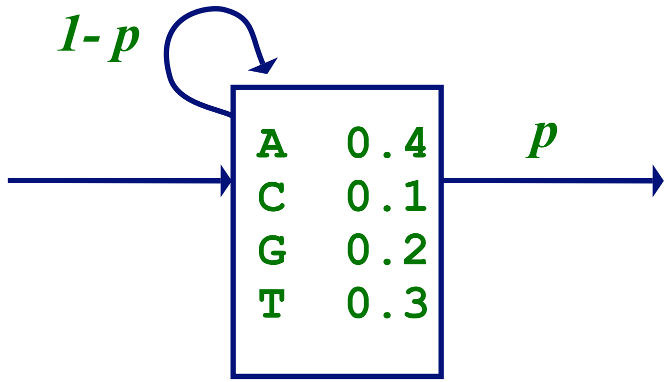
Example parse:

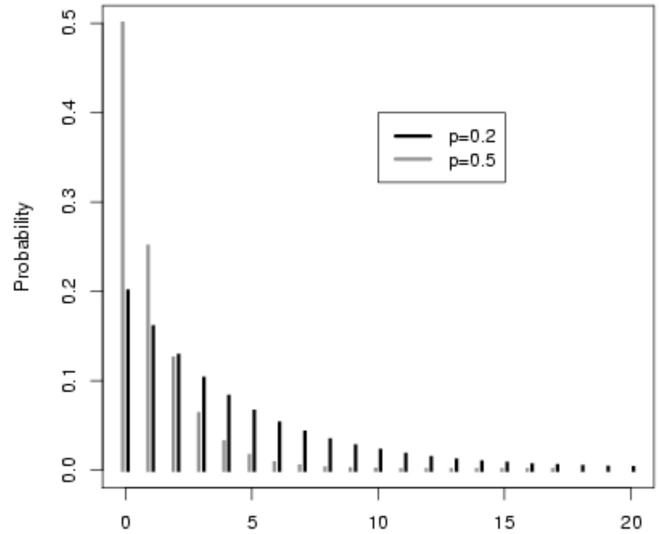
N N N N N N N N S_1 S_2 S_3 $C_1 C_2 C_3 C_1 C_2 C_3$ $C_1 C_2 C_3$ $C_1 C_2 C_3 C_1 C_2 C_3$ T_1 T_2 T_3 N N N N N N

CTGCGTAGATGCTAATGTCATCGCTATAGATCTGC

Duration Modeling in HMMs

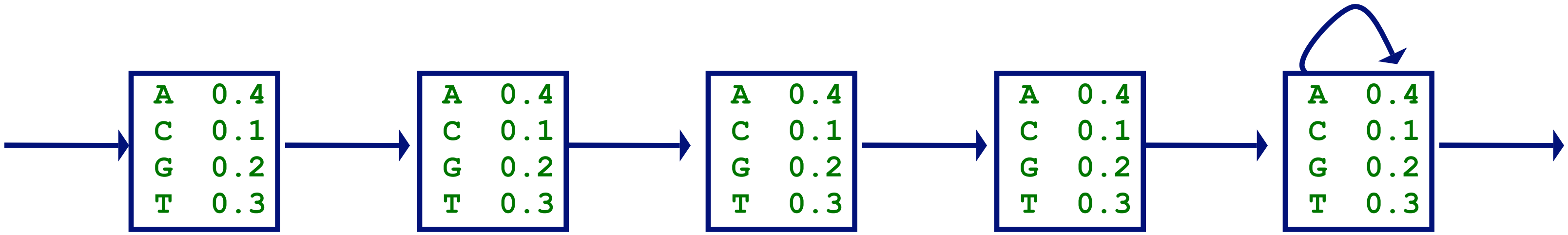
- suppose we have a type of sequence for which the base distribution is the same regardless of length
- the simplest way to model it:



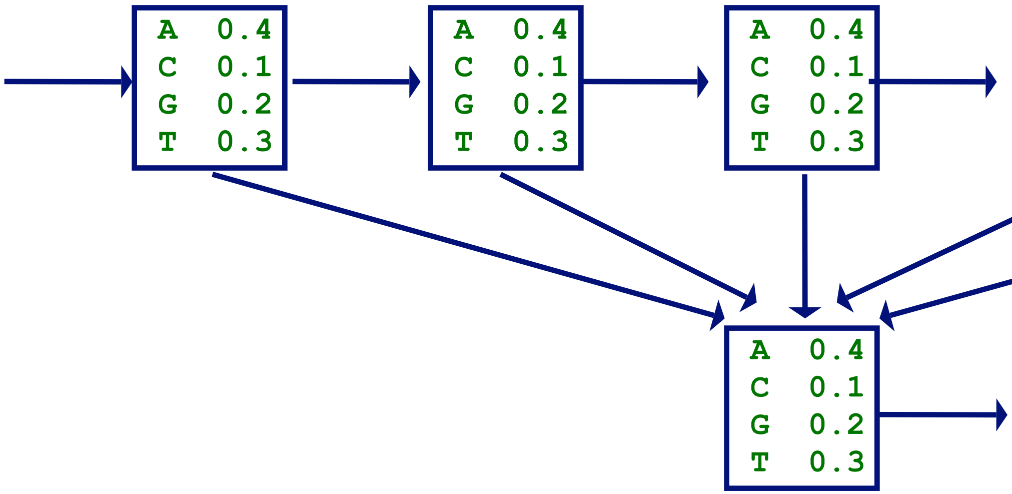


- this encodes a geometric distribution on the length of sequences

Duration Modeling in HMMs

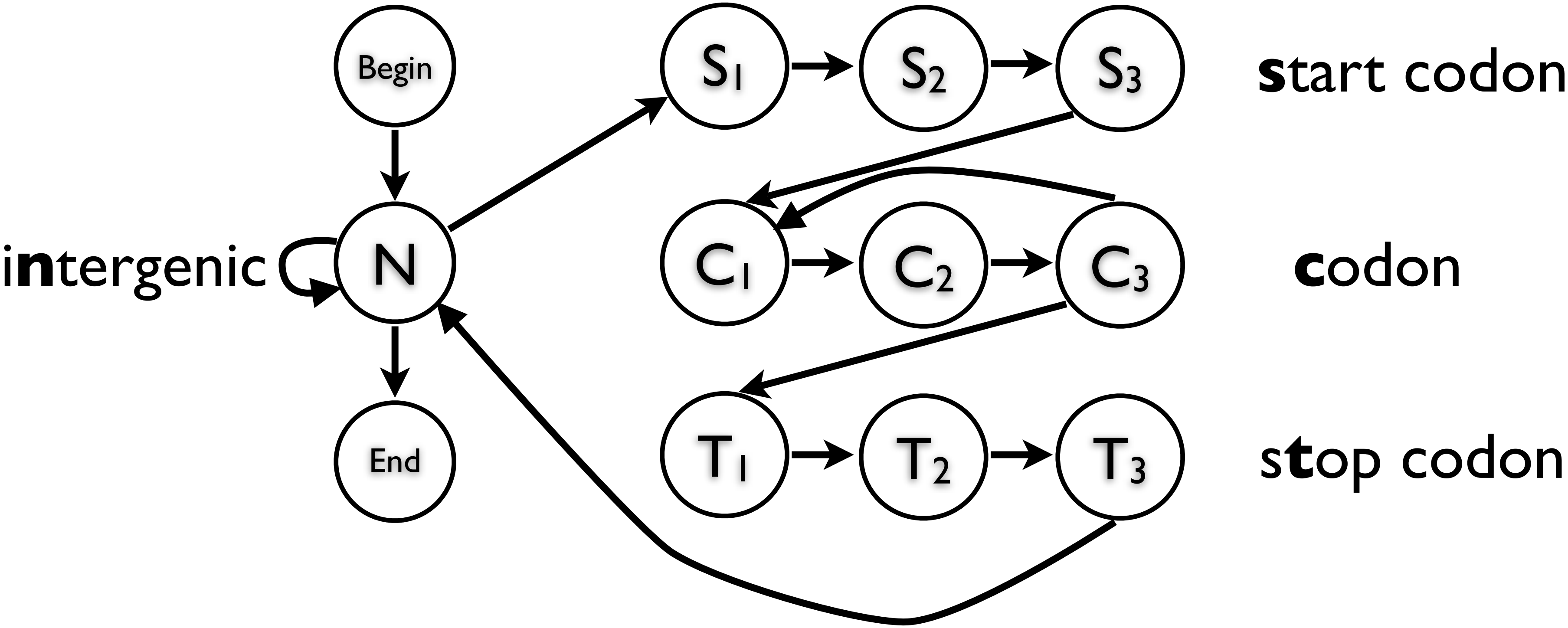


min length = 5; geometric distribution over longer sequences



arbitrary distribution over length 2 to 6

Lengths in our simple HMM



- Intergenic regions: geometric
- Number of codons in gene: geometric

Length Distributions of Introns/Exons

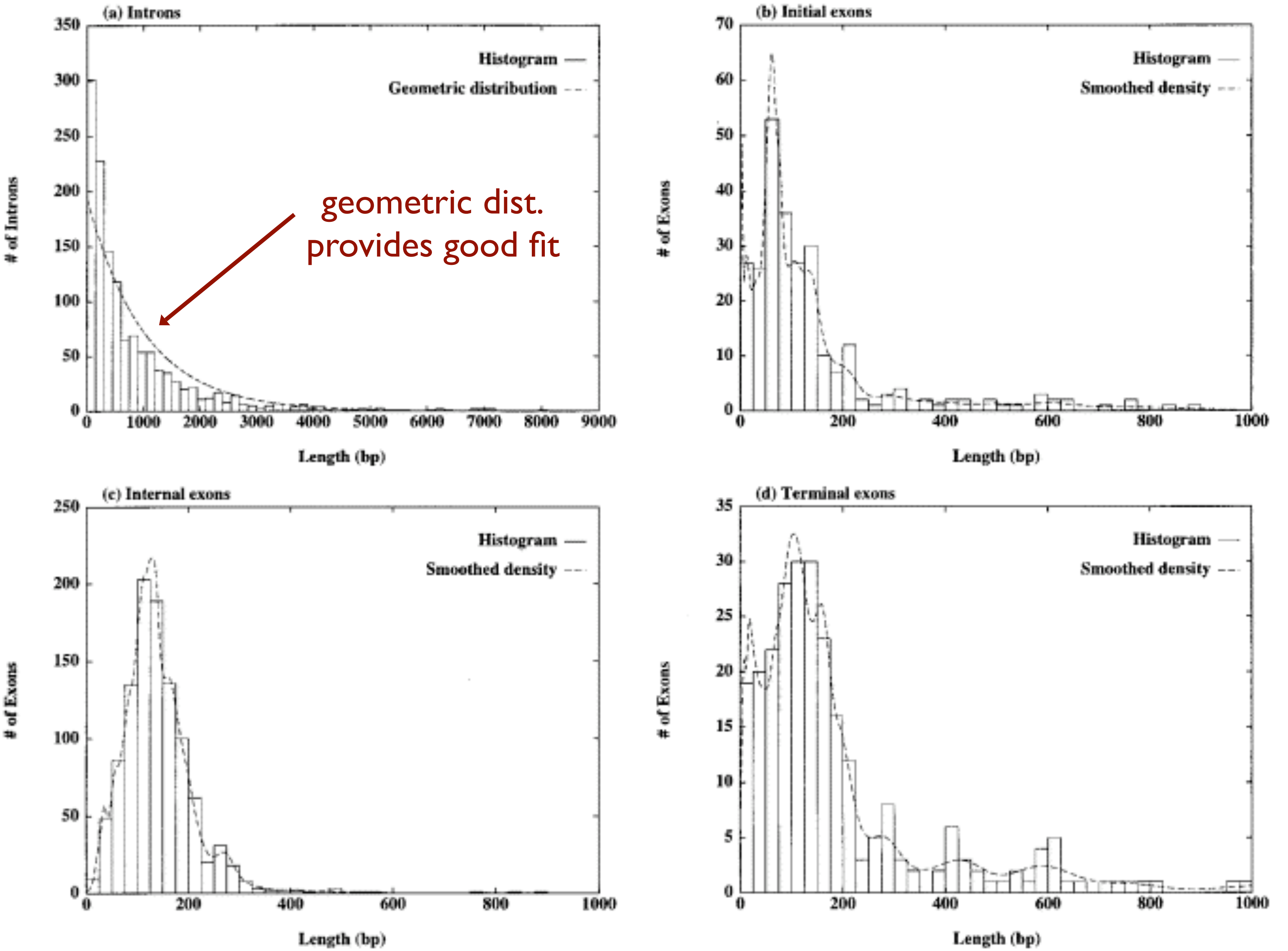


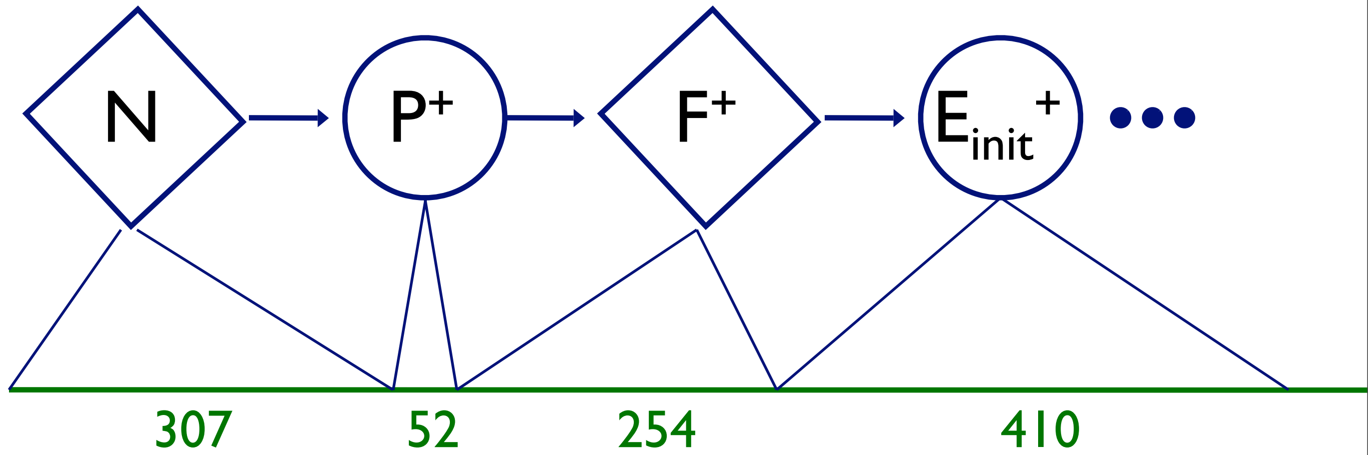
Figure from Burge & Karlin, *Journal of Molecular Biology*, 1997

Semi-Markov HMMs (a.k.a. Generalized HMMs)

- key idea: decouple length from composition
- represent a parse π, as a sequence of states and associated lengths (durations)

$\vec{q} = \{q_1, q_2, \ldots, q_n\}$

$\vec{d} = \{d_1, d_2, \ldots, d_n\}$



Semi-Markov Models

- representing a parse π, as a sequence of states and associated lengths (durations)

  $\vec{q} = \{q_1, q_2, \ldots, q_n\}$ $\vec{d} = \{d_1, d_2, \ldots, d_n\}$

- the joint probability of generating parse π and sequence x

$$\Pr(x, \pi) = a_{start,q_1} \Pr(d_1 | q_1) \Pr(x_1 | q_1, d_1) \times$$

$$\prod_{i=2}^{n} a_{q_{i-1},q_i} \Pr(d_i | q_i) \Pr(x_i | q_i, d_i)$$

transition probabilities

the i^{th} segment of the sequence

DP with Semi-Markov Models

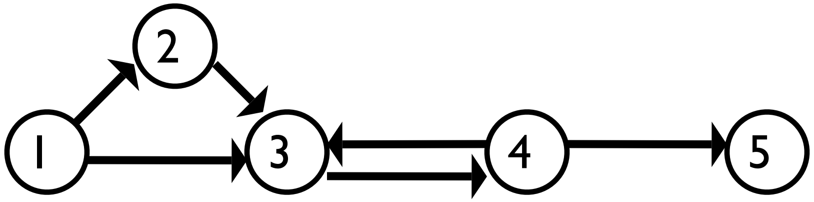
- review: Forward algorithm recurrence for HMMs

$$f_l(i) = \sum_k \left[f_k(i-1) \underbrace{a_{kl}}_{\substack{\text{transition} \\ \text{from } k \text{ to } l}} \underbrace{\Pr(x_i|q_l)}_{\substack{\text{prob. of emitting} \\ x_i \text{ from } l}} \right]$$

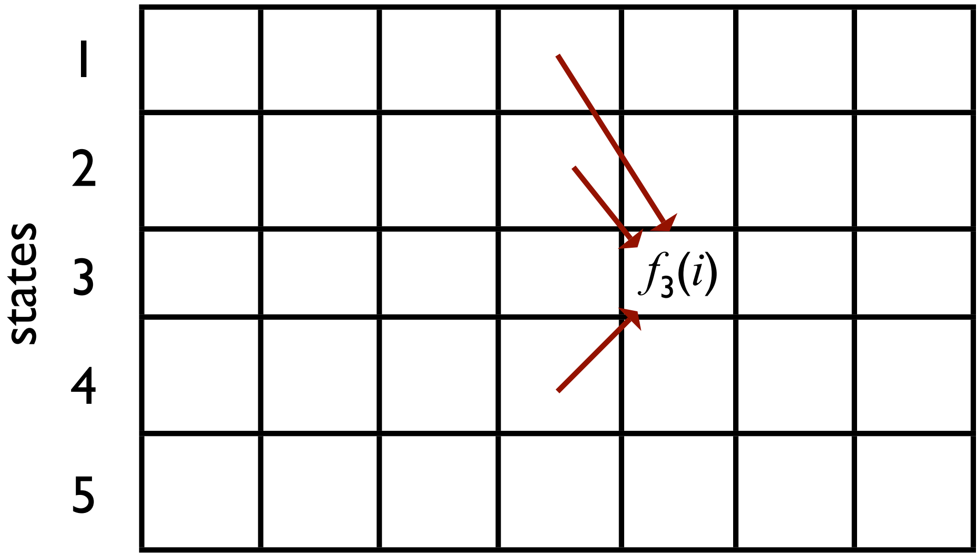
- for semi-Markov models: each Forward value assumes we're ending a segment in the given state

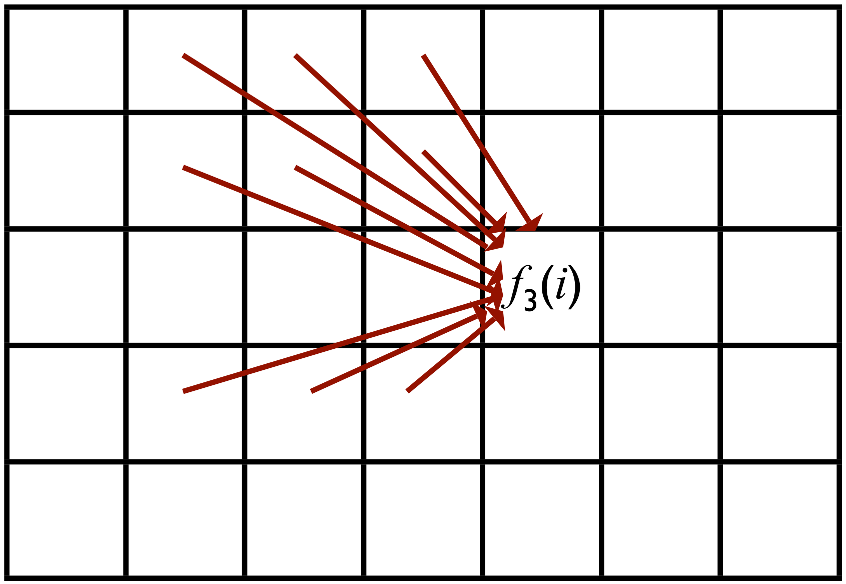
$$f_l(i) = \sum_k \sum_{d=1}^{D} \left[f_k(i-d) a_{kl} \underbrace{\Pr(d|q_l)}_{\substack{\text{prob. of length} \\ d \text{ segment from } l}} \underbrace{\Pr(x_{i-d+1} \ldots x_i|q_l)}_{\substack{\text{prob. of emitting} \\ x_{i-d+1} \ldots x_i \text{ from } l}} \right]$$

DP with Semi-Markov Models



sequence positions





complexity of Viterbi/Forward/Backward in standard HMMs is $O(S^2L)$ where S = number of states, L = sequence length

complexity in semi-Markov HMMs is $O(S^2LD)$ where D = maximum length of a segment

The GENSCAN HMM for Eukaryotic Gene Finding [Burge & Karlin '97]

Each shape represents a structural unit of a gene or genomic region

Pairs of intron/exon units represent the different ways an intron can interrupt a coding sequence (after 1st base in codon, after 2nd base or after 3rd base)

Complementary submodel (not shown) detects genes on opposite DNA strand

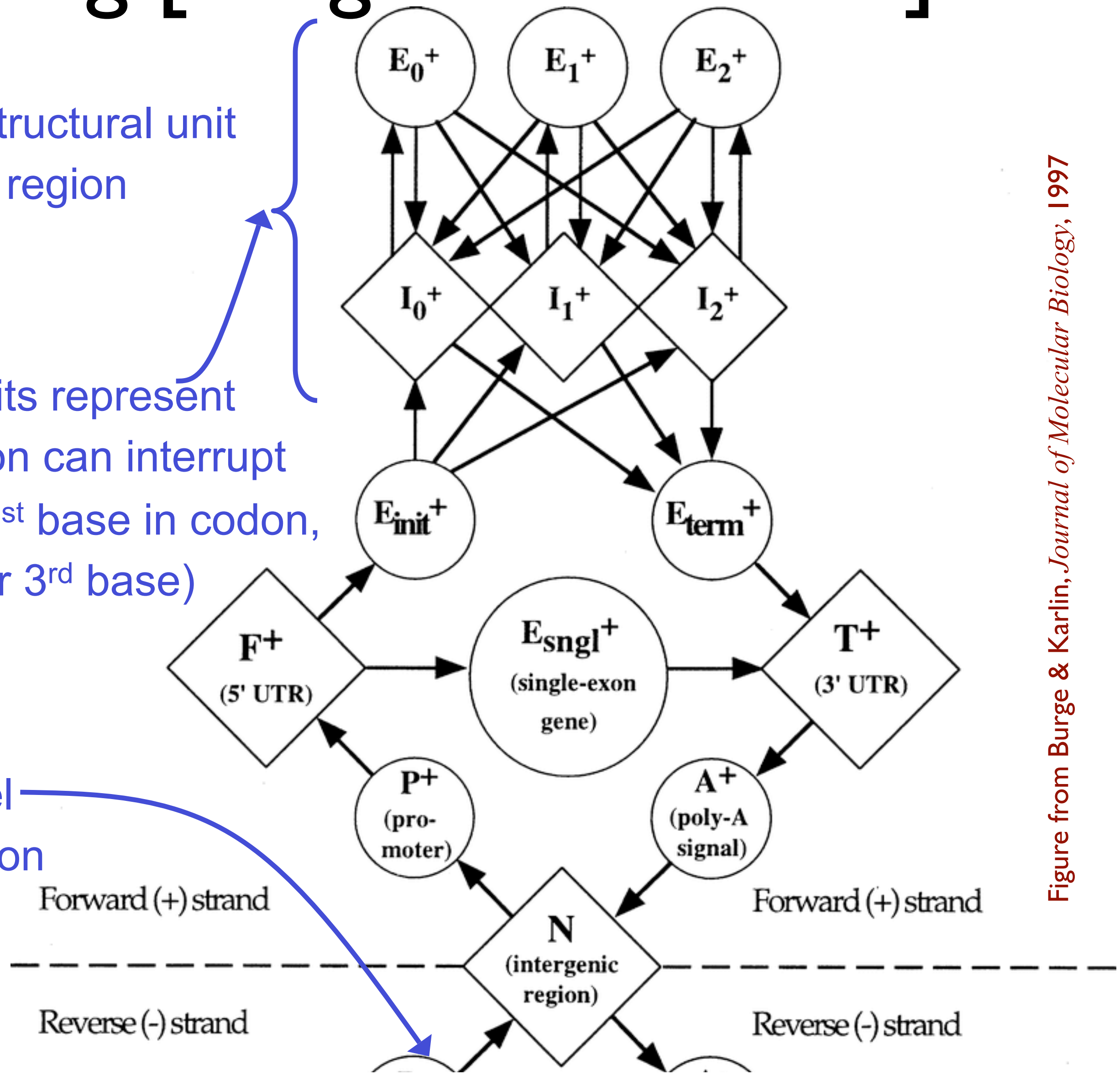
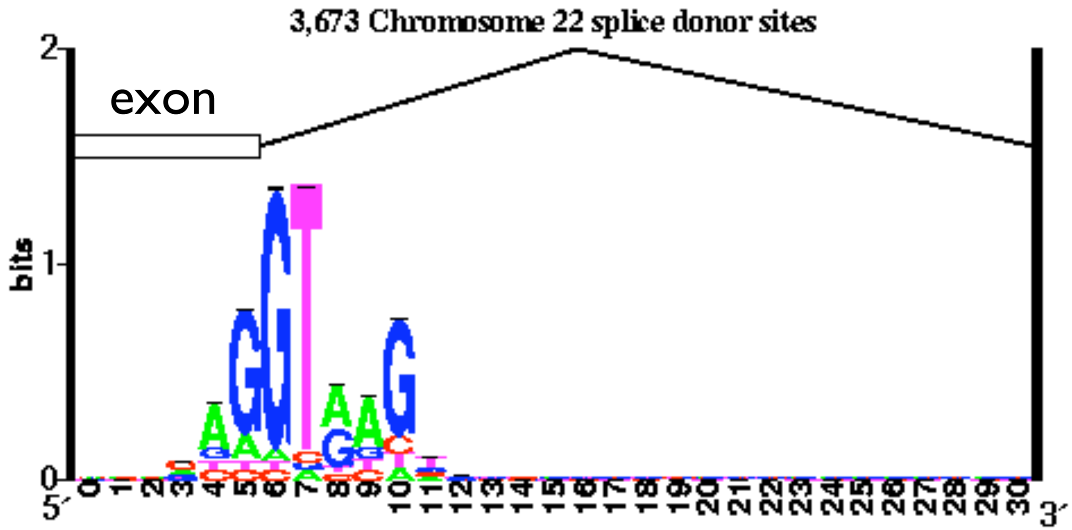


Figure from Burge & Karlin, *Journal of Molecular Biology*, 1997

The GENSCAN HMM

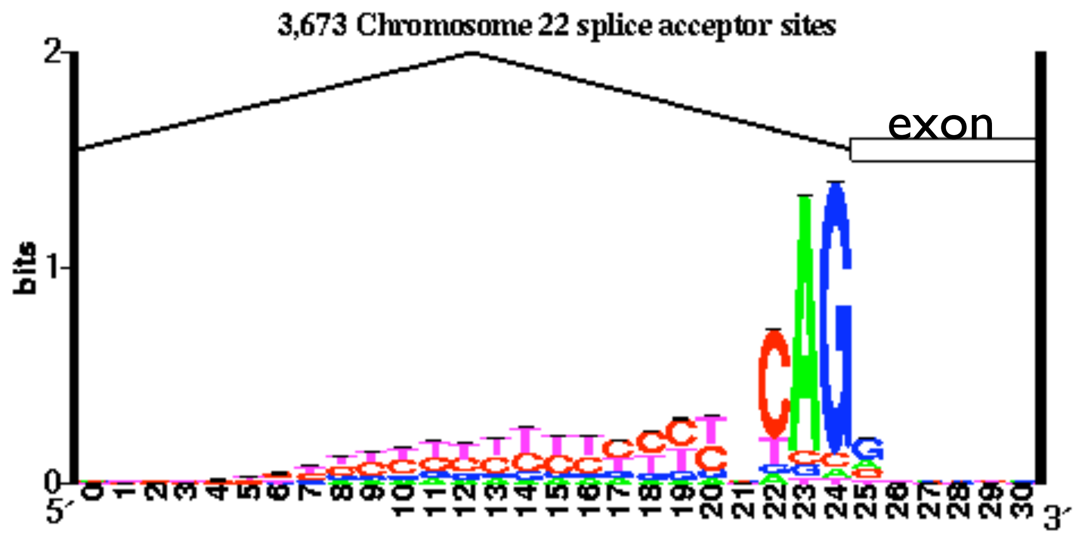
- for each sequence type, GENSCAN models
  - the length distribution
  - the sequence composition
- length distribution models vary depending on sequence type
  * nonparametric (using histograms)
  - parametric (using geometric distributions)
  - fixed-length
- sequence composition models vary depending on type
  - 5^{th}-order, inhomogeneous (exons)
  - 5^{th}-order homogenous (noncoding classes)
  - 1^{st}-order inhomogeneous (acceptor splice site)
  * tree-structured variable memory (MDD) (donor splice site)

Splice Signals



donor sites

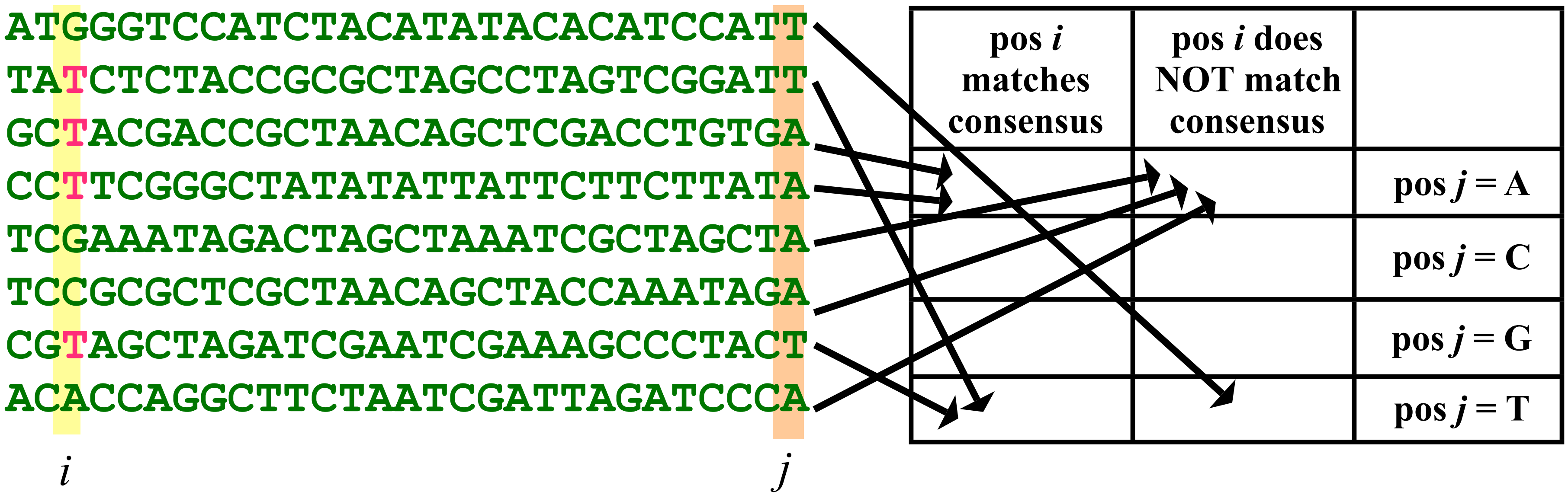
Figures from the Sanger Center.



acceptor sites

Motivation for MDD

- How can we model significant dependencies between non-adjacent positions?



compute χ^2 values using 2×4 table

alternative hypothesis: distribution for column j depends on what is in column i

null hypothesis: distribution for column j is the same in both cases

Motivation for MDD

- Table shows χ^2 values for pairs of positions around donor sites
- values marked with * show statistically significant dependency

Table 4. Dependence between positions in human donor splice sites: χ^2 – statistic for consensus indicator variable C_i *versus* nucleotide indicator X_j

i	Con	j: −3	−2	−1	+3	+4	+5	+6	Sum
−3	c/a	—	61.8*	14.9	5.8	20.2*	11.2	18.0*	131.8*
−2	A	115.6*	—	40.5*	20.3*	57.5*	59.7*	42.9*	336.5*
−1	G	15.4	82.8*	—	13.0	61.5*	41.4*	96.6*	310.8*
+3	a/g	8.6	17.5*	13.1	—	19.3*	1.8	0.1	60.5*
+4	A	21.8*	56.0*	62.1*	64.1*	—	56.8*	0.2	260.9*
+5	G	11.6	60.1*	41.9*	93.6*	146.6*	—	33.6*	387.3*
+6	t	22.2*	40.7*	103.8*	26.5*	17.8*	32.6*	—	243.6*

The Maximal Dependence Decomposition (MDD) Approach

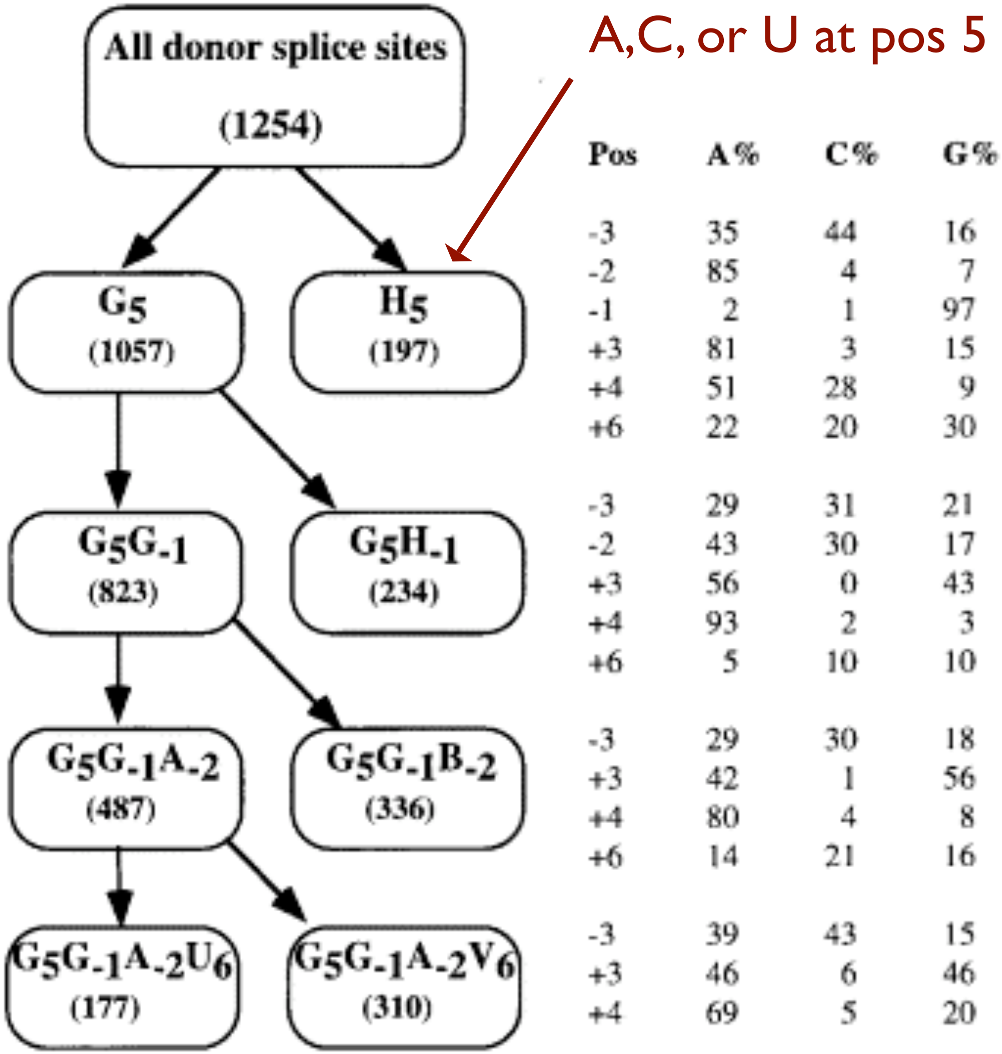
- induce a tree that represents the dependency structure apparent in the data
- induce partial position weight matrices for each node and leaf of tree

	1	2	3	4	5	6	7	8
A	0.1	0.3	0.1	0.2	0.2	0.4	0.3	0.1
C	0.5	0.2	0.1	0.1	0.6	0.1	0.2	0.7
G	0.2	0.2	0.6	0.5	0.1	0.2	0.2	0.1
T	0.2	0.3	0.2	0.2	0.1	0.3	0.3	0.1

- use the tree + weight matrices to calculate the probability of a given sequence

An MDD Learned Tree

Pos	A%	C%	G%	U%
-3	33	36	19	13
-2	56	15	15	15
-1	9	4	78	9
+3	44	3	51	3
+4	75	4	13	9
+6	14	18	19	49
-3	34	37	18	11
-2	59	10	15	16
+3	40	4	53	3
+4	70	4	16	10
+6	17	21	21	42
-3	37	42	18	3
+3	39	5	51	5
+4	62	5	22	11
+6	19	20	25	36
-3	32	40	23	5
+3	27	4	59	10
+4	51	5	25	19



Pos	A%	C%	G%	U%
-3	35	44	16	6
-2	85	4	7	5
-1	2	1	97	0
+3	81	3	15	2
+4	51	28	9	12
+6	22	20	30	28
-3	29	31	21	18
-2	43	30	17	11
+3	56	0	43	0
+4	93	2	3	3
+6	5	10	10	76
-3	29	30	18	23
+3	42	1	56	1
+4	80	4	8	8
+6	14	21	16	49
-3	39	43	15	2
+3	46	6	46	3
+4	69	5	20	7

Figure from Burge & Karlin, *Journal of Molecular Biology*, 1997

The MDD Algorithm: Learning the Tree

Given: a set of aligned training sequences T

positions $P = \{1, \ldots, k\}$

tree = find_MDD_subtree(T, P)

find_MDD_subtree(T, P)
for each position i in P
    determine the consensus base C_i
    calculate dependence between C_i, other positions

$$S_i = \sum_{j \neq i} \chi^2(C_i, x_j)$$

if stopping criteria not met
    choose i such that S_i is maximal
    make a node with C_i as the test
    D_i^+ = sequences in T with base C_i at position i
    D_i^- = other sequences
    left subtree = find_MDD_subtree(D_i^+ , $P - \{i\}$)
    right subtree = find_MDD_subtree(D_i^- , $P - \{i\}$)

Stopping Criteria for MDD Tree Learning

1. the $(k\text{-}1)^{th}$ level is reached; no further positions to split on

2. no significant dependencies between positions are detected

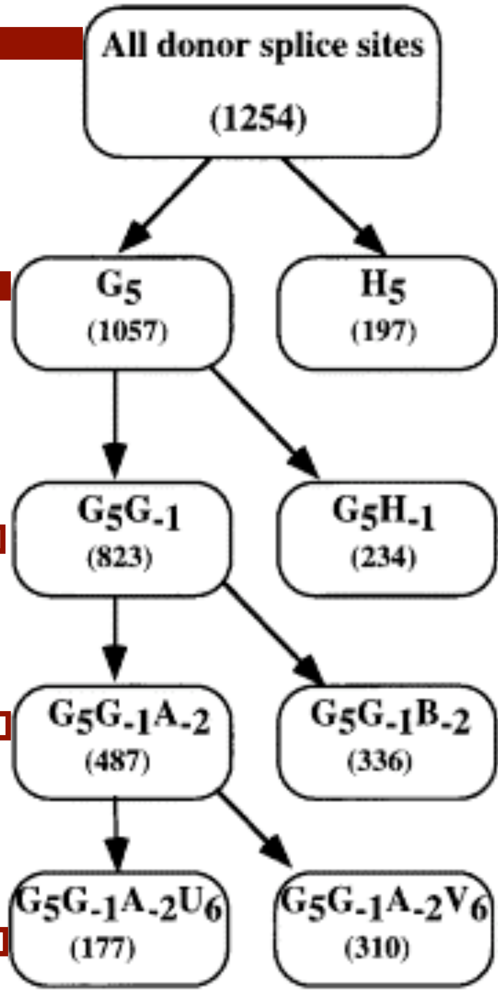
3. number of sequences in given subset is sufficiently small

Explaining a Sequence with an MDD Tree

- shown are selected position weight matrices for the tree

	5
A	0.1
C	0.1
G	0.7
U	0.1

	-1
A	0.1
C	0.1
G	0.7
U	0.1



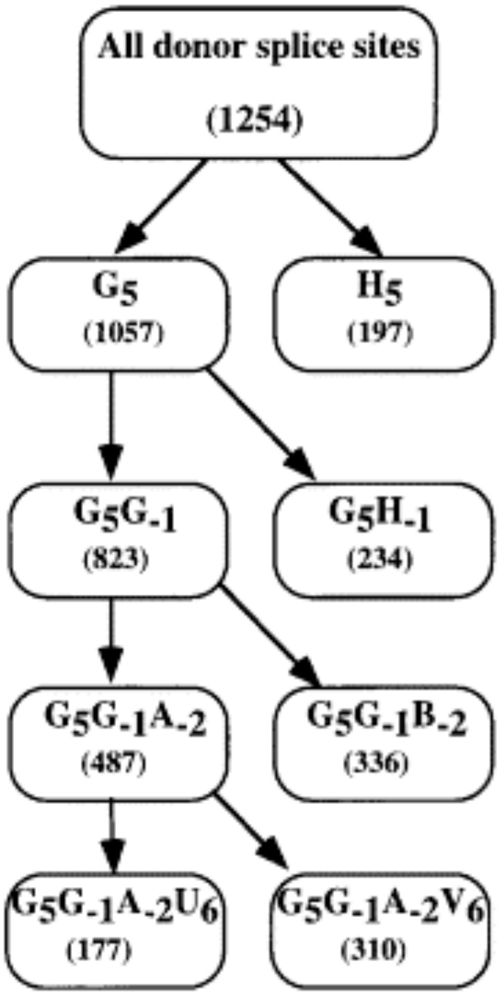
	-2
A	0.5
C	0.3
G	0.1
U	0.1

	6
A	0.1
C	0.2
G	0.1
U	0.6

	-3	-2	-1	3	4	5	6
A	0.3	0.4		0.2	0.5		0.1
C	0.4	0.3		0.1	0.1		0.1
G	0.2	0.2		0.6	0.2		0.1
U	0.1	0.1		0.1	0.2		0.7

	-3	-2	-1	3	4	5	6
A	0.3			0.2	0.5		
C	0.4			0.1	0.1		
G	0.2			0.6	0.2		
U	0.1			0.1	0.2		

Explaining a Sequence with an MDD Tree



calculate Pr(x_5)

if $x_5 \neq G$, use the weight matrix for H_5 subset

else

calculate Pr(x_{-1}) from from G_5 subset

if $x_{-1} \neq G$, use the WM for G_5H_{-1} subset

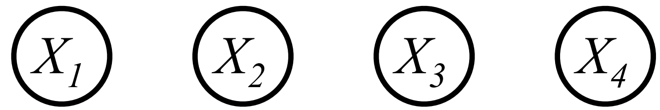
else

⋮

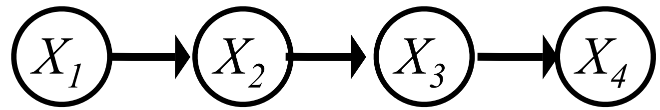
calculate Pr(x_{-2}) from G_5G_{-1} subset

Graphical Model View

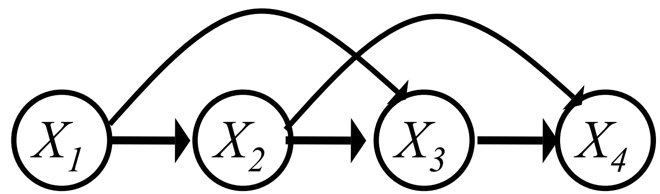
- 0th order Markov model (e.g., Profile matrix)



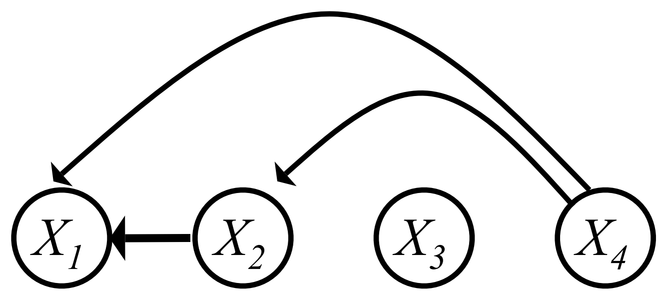
- 1st order Markov model



- 2nd order Markov model

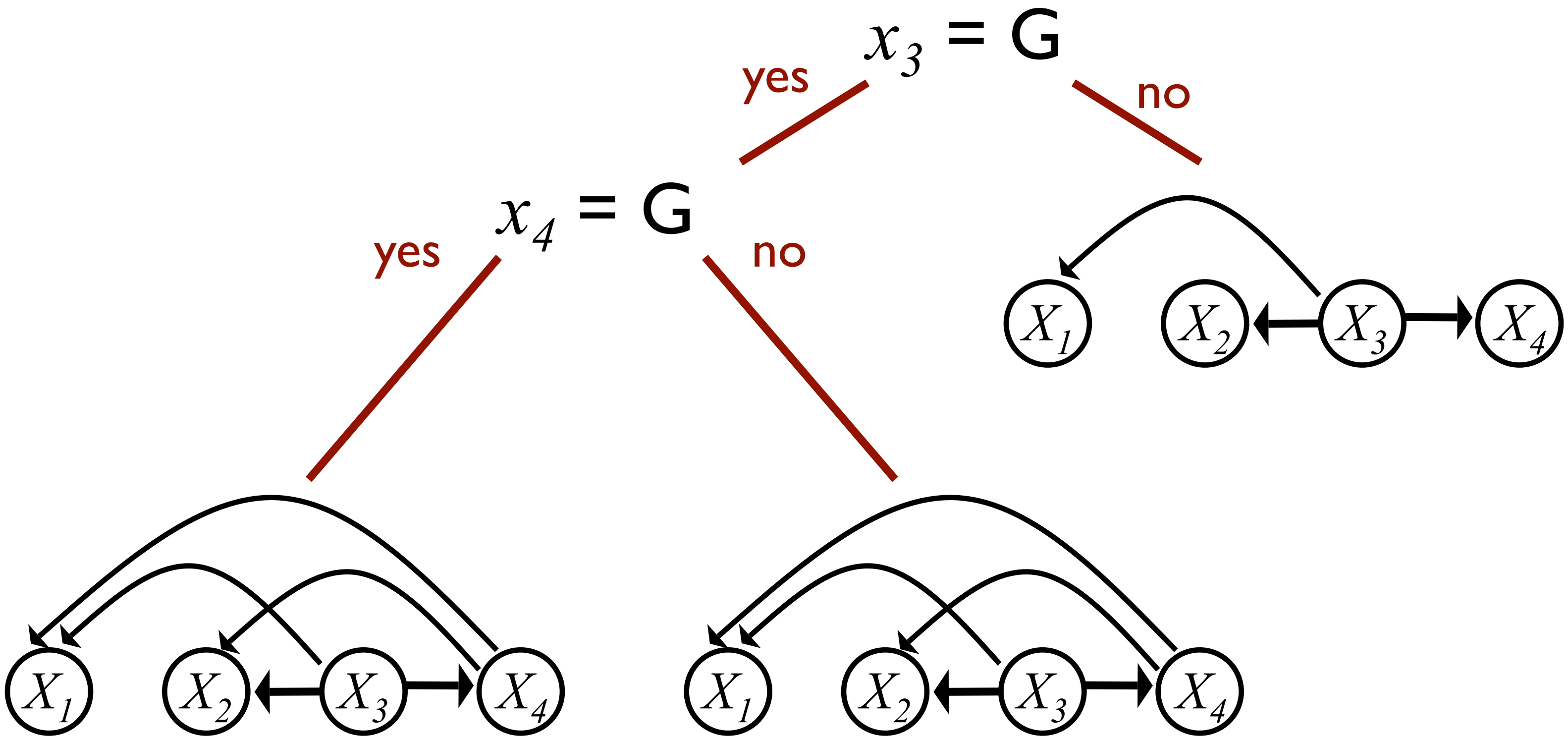


- for a fixed-length model, we could consider arbitrary dependencies



Graphical Models from a MDD tree

- MDD allows arbitrary dependencies conditioned on *realization* of certain variables



GENSCAN Conclusions

- HMMs readily enable background knowledge to be incorporated into the model
  - state topology
  - length distributions
  - order of Markov chains
- key technical ideas
  - semi-Markov models (old): can represent arbitrary length distributions
  - MDD (new): can represent context-specific dependencies