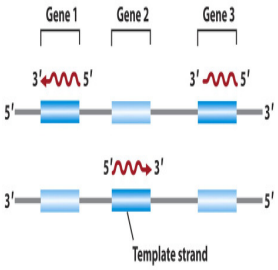
Bioinformatics

Seven

Gene Prediction

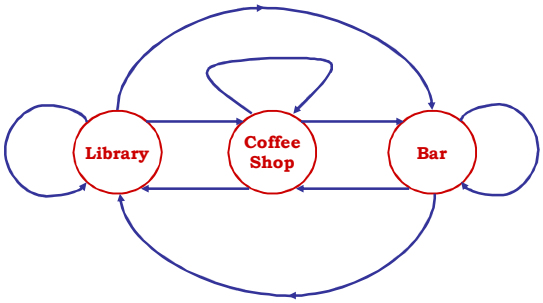
Sami Khuri
Department of Computer Science
San José State University
San José, California, USA
sami.khuri@sjsu.edu
www.cs.sjsu.edu/faculty/khuri

Gene Prediction

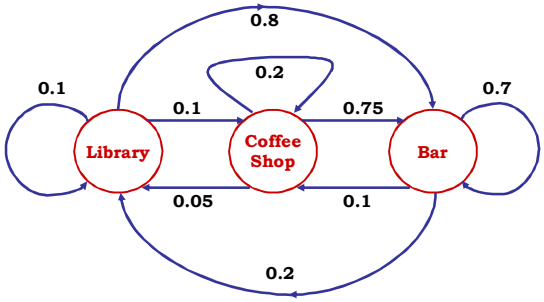


- Markov Models
- Viterbi
- Evaluation
- Decoding
- Learning
- VEIL
- GenScan

States and Transitions



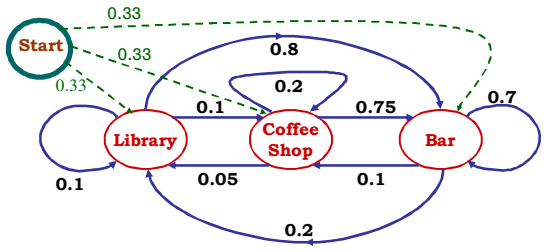
The Distressed Student Model



Evaluating Observations

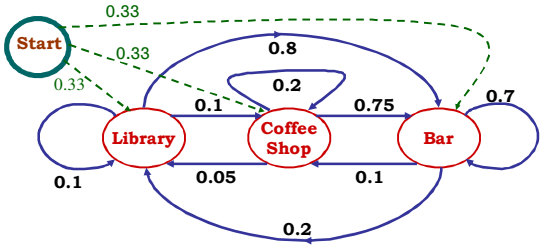
- The probability of observing a given sequence is equal to the product of all observed **transition probabilities**.
- Suppose that:
  - **L**: student is in state Library
  - **C**: student is in state Coffee Shop
  - **B**: student is in state Bar

Starting State of the Student



The Model has a Start State with transition probabilities of going to L, C, or B of 1/3.

Behavior of Three Students



Student 1 : LLLCBCLLBB LL
Student 2 : LCBLBBCBBB BL
Student 3 : CCCLCCCBCC CL

Computing Observed Sequences

- The probability of observing a given sequence is equal to the product of all observed transition probabilities.
- Example:
- P(LLLCBCLLBBLL)

  = 1/3 * 0.1 * 0.1 * 0.1 * 0.75 * 0.1 * 0.05
  * 0.1 * 0.8 * 0.7 * 0.2 * 0.1

  = $1.4 * 10^{-9}$

Computing Observed Sequences

- P(LCBLBBCBBBBL)

  = 1/3 * 0.1 * 0.75 * 0.2 * 0.8 * 0.7 * 0.1
  * 0.75 * 0.7 * 0.7 * 0.7 * 0.2

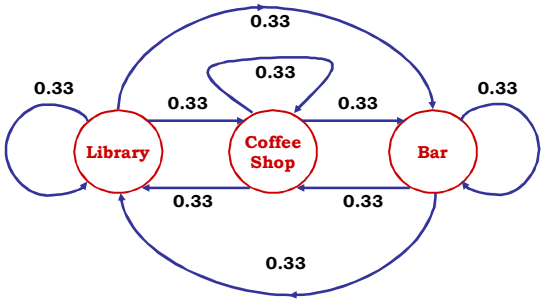
  = $1.4406 * 10^{-5}$

- P(CCCLCCCBCCCL)

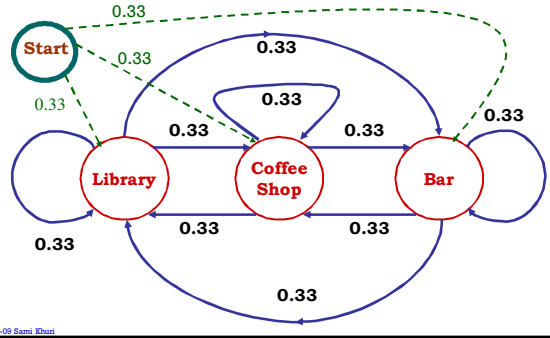
  = 1/3 * 0.2 * 0.2 * 0.05 * 0.1 * 0.2 * 0.2
  * 0.75 * 0.1 * 0.2 * 0.2 * 0.05

  = $4 * 10^{-10}$

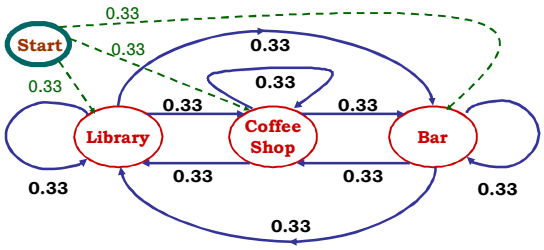
The Random Model
The Null Hypothesis



Start State with Random Model



Students with Random Model



Student 1 : LLLCBCLLBB LL = 1.8817×10^{-6}
Student 2 : LCBLBBCBBB BL = 1.8817×10^{-6}
Student 3 : CCCLCCCBCC CL = 1.8817×10^{-6}

Odds and Log Ratios

- To determine the significance of the results obtained with the 3 students, compare them to the null model (random model)
- Odds Ratio =

  P(x | Distressed Model) / P(x | Null Model)
- Log Odds =

  Log [P(x | Distressed Model) / P(x | Null Model)]

Likelihood Ratios: Distressed

- **Likelihood ratios**:

  Student 1 : $= 1.4 \times 10^{-9} / 1.8817 \times 10^{-6} = x$

  Student 2 : $= 1.4406 \times 10^{-5} / 1.8817 \times 10^{-6} = y$

  Student 3 : $= 4 \times 10^{-10} / 1.8817 \times 10^{-6} = z$

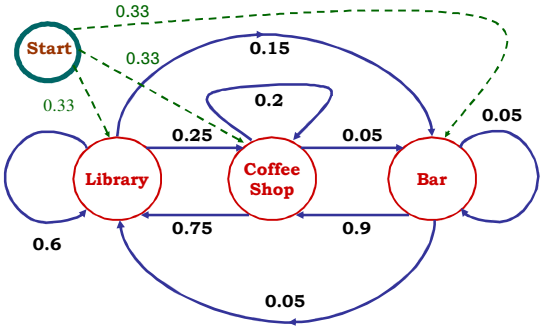
- **Log likelihood ratios**:

  Student 1 = log x = - 10.39

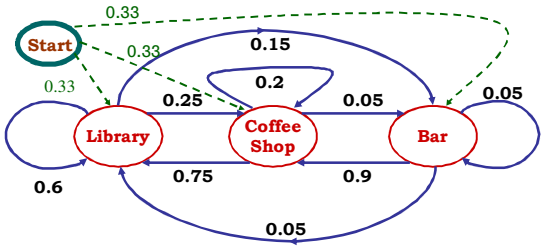
  Student 2 = log y = 2.94

  Student 3 = log z = - 12.20

The Successful Student Model



Students with Successful Model



Student 1 : LLLCBCLLBB LL

Student 2 : LCBLBBCBBB BL

Student 3 : CCCLCCCBCC CL

Outcomes with Successful Model

- P(LLLCBCLLBBLL)

  = 1/3 * 0.6 * 0.6 * 0.25 * 0.05 * 0.9 * 0.75 * 0.6 * 0.15 * 0.05 * 0.05 * 0.6 = $1.3669 * 10^{-7}$
- P(LCBLBBCBBBBL)

  = 1/3 * 0.25 * 0.05 * 0.05 * 0.15 * 0.05 * 0.9 * 0.05 * 0.05 * 0.05 * 0.05 * 0.05 = $4.3945 * 10^{-13}$
- P(CCCLCCCBCCCL)

  = 1/3 * 0.2 * 0.2 * 0.75 * 0.25 * 0.2 * 0.2 * 0.05 * 0.9 * 0.2 * 0.2 * 0.75 = $1.35 * 10^{-7}$

Likelihood Ratios: Successful

- **Likelihood ratios**:

  Student1 : $= 1.3669 \times 10^{-7} / 1.8817 \times 10^{-6} = x$

  Student2 : $= 4.3945 \times 10^{-13} / 1.8817 \times 10^{-6} = y$

  Student3 : $= 1.35 \times 10^{-7} / 1.8817 \times 10^{-6} = z$

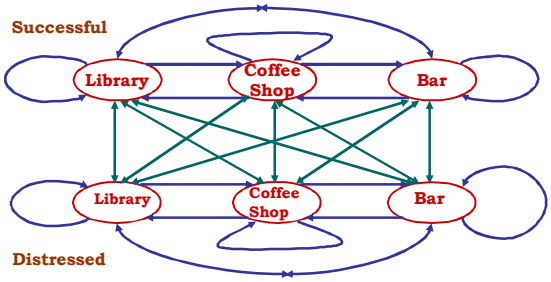
- **Log likelihood ratios**:

  Student 1 = log x = - 3.78

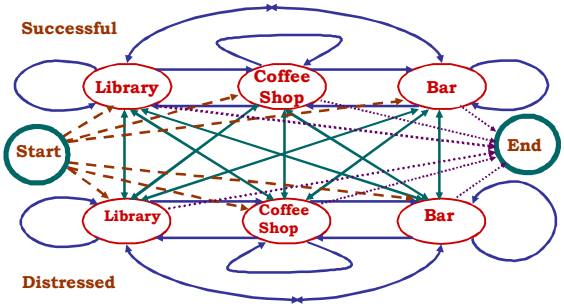
  Student 2 = log y = - 22.03

  Student 3 = log z = - 3.8

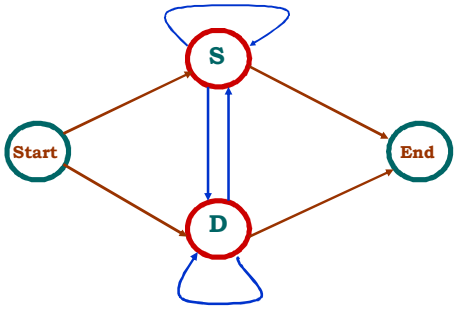
HMM – Combined Model



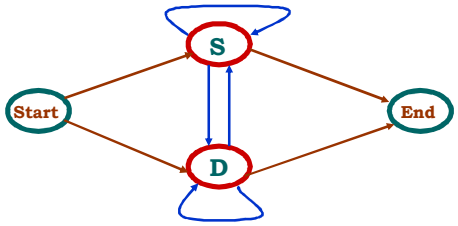
HMM – Combined Model



Hidden Markov Model



Evaluating Hidden States



Given an observation: LLLCBCLBBCL, find the sequence of states which is the most likely to have produced the observation.

Models of Sequences

- Consists of states (boxes) and transitions (arcs) labelled with probabilities.
- States have probabilities of "emitting" an element of a sequence (or nothing).
- Arcs have transitional probabilities of moving from one state to another.
  - Sum of probabilities of arcs out of a state must be 1
  - Self-loops are allowed.

Markov Chain

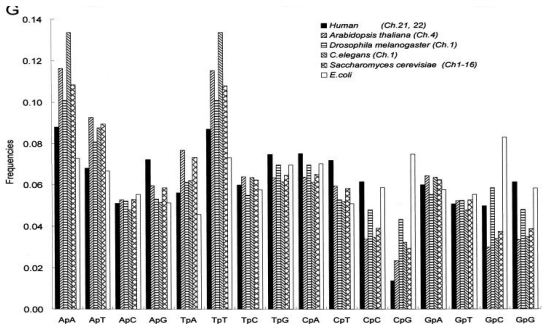
- A sequence is said to be **Markovian** if the probability of the occurrence of an element in a particular position depends only on the **previous** elements in the sequence.
- Order of a Markov chain depends on how many previous elements influence the probability:
  - **0^{th} order**: uniform probability at every position
  - **1^{st} order**: probability depends only on the immediately previous position.

Looking for CpG Islands

Example:

- A CpG island in humans refers to the dinucleotide CG and not the basepair CG.
- The C of CpG is generally methylated to inactivate genes hence CpG is found around "start" regions of many genes more often than elsewhere.
- Methylated C is easily mutated into T.

The Rarity of CpG Islands



CpG Island Criteria

- According to Gardiner-Garden and Fromer, CpG islands are commonly defined as regions of DNA
  - of at least 200 bp in length,
  - that have a G+C content above 50%
  - that have a ratio of observed vs. expected CpGs close to or above 0.6.
- Sets of CG repeat elements, usually found upstream of transcribed regions of the genome.

Looking for CpG Islands

- CpG islands are therefore rare in other locations
- CpG islands are generally a few hundred base pairs long

Questions:

1. Given a short DNA fragment, does it come from a **CpG island** or not?
2. Given a long unannotated sequence of DNA, how do we find the **CpG islands**?

Building an HMM for CpG Islands

- A set of human sequences were considered and 48 CpG islands were tabulated.
- Two Markov chain models were built:
  - One for the regions labeled as CpG islands (the '+' model or Model 1)
  - One for the remainder of the sequences (the '-' model or Model 2).

Transition Probabilities

The transition probabilities of each model were computed by:

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}} \qquad a_{st}^{-} = \frac{c_{st}^{-}}{\sum_{t'} c_{st'}^{-}}$$

c_{st}^{+} is the number of times letter t followed letter s in the plus model.

The Two Transition Tables

Markov chain Model 1

Transition Frequencies within 48 putative CpG islands in humans

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

Markov chain Model 2

Transition frequencies in Non CpG island DNA

-	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	0.078	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

Log-Odds Ratio

- Given any sequence, we compute the **log-odds ratio** to discriminate between the two models :

$$S(x) = \log \frac{P(x \mid the + \text{model})}{P(x \mid the - \text{model})} = \sum_{i=1}^{L} \log \frac{a^{+}_{x_{i-1}x_i}}{a^{-}_{x_{i-1}x_i}}$$

- S(x)>0 means x is likely to be a CpG island.
- The ratio is also called the **log likelihood ratio** of transition probabilities.

Log Likelihood Ratios

$$S(x) = \log \frac{P(x \mid the + \text{model})}{P(x \mid the - \text{model})} = \sum_{i=1}^{L} \log \frac{a^{+}_{x_{i-1}x_i}}{a^{-}_{x_{i-1}x_i}}$$

Log models	A	C	G	T
A	-0.740	0.417	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-1.169	0.573	0.393	-0.679

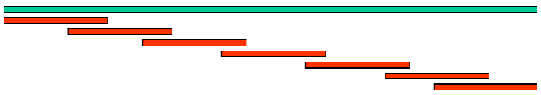
The table's unit is the bit since base 2 is used for the computation of the individual entries of the table.

Looking for CpG Islands

- Given a long unannotated sequence of DNA, how do we find the **CpG islands**?

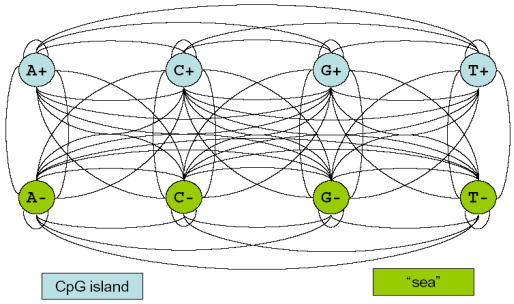
- We can use a sliding window of size 100, for example, around each nucleotide in the sequence and use the previous table to score the log-odds. **CpG islands** would stand out with positive values.

Sliding Window Size

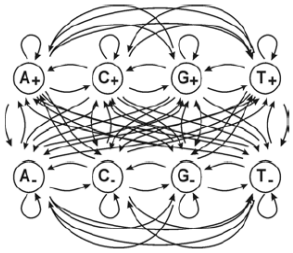


- How do we determine the window size? CpG islands are of variable lengths and might have sharp boundaries.
- A better approach is to build an **HMM** that combines both models.

The Island and the Sea

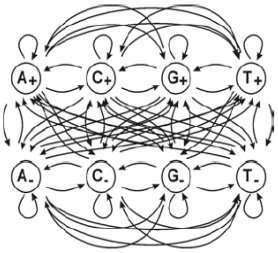


An HMM for CpG Islands (I)



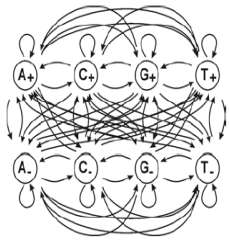
There are 8 states, one for each nucleotide in a **CpG island** (+), and one for each nucleotide not in a **CpG island** (-).

An HMM for CpG Islands (II)



There two states for each output symbol.
Example: "T" is recognized or generated by T+ or T-.
Within each group of states, the group has the same behavior as the original **Markov Model**.

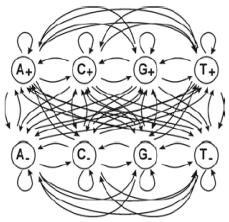
An HMM for CpG Islands (III)



Position i:	Position i+1: C+	G+	C-	G-
C+	0.37	0.27	small	small
G+	0.34	0.38	small	small
C-	small	small	0.3	0.08
G-	small	small	0.25	0.3

Assume the transitions from a (+) nucleotide to a (-) are small. And transitions from (-) nucleotides to (+) are also small.

The Two Paths with CGCG



	C	G	C	G
B	1	0	0	0
A+	0	0	0	0
C+	0	0.13	0	0.012
G+	0	0	0.034	0	0.0032
T+	0	0	0	0
A-	0	0	0	0
C-	0	0.13	0	0.0026
G-	0	0	0.01	0	0.00021
T-	0	0	0	0

Position i:	Position i+1: C+	G+	C-	G-
C+	0.37	0.27	small	small
G+	0.34	0.38	small	small
C-	small	small	0.3	0.08
G-	small	small	0.25	0.3

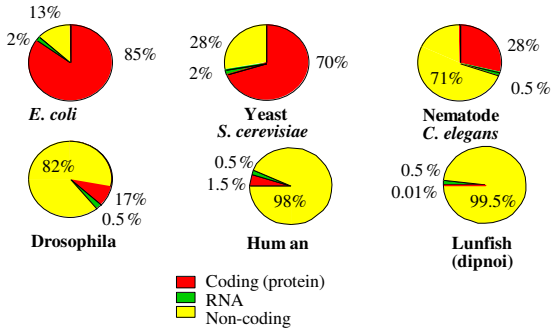
Switching between '+' & '-' States

- The maximum scoring path receives a score of 0.0032.
- The most likely state path is found to be C+G+C+G+.
- Given a much longer sequence, the derived optimal path will switch between the **CpG** and **non-CpG** states.

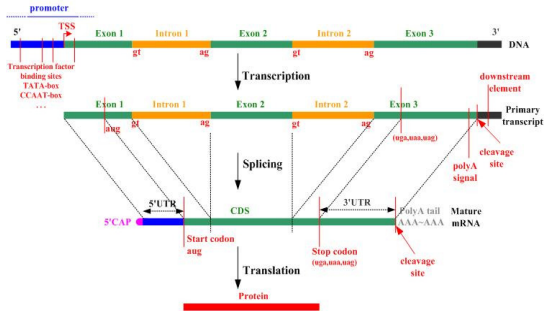
HMM for Gene Prediction

- Hidden Markov Models can be applied to predict signals in the gene structure.
- More precisely, HMM can recognize
  - Start codons
  - Stop codons
  - Donor splice sites
  - Acceptor splice sites
  - 5' Poly A site

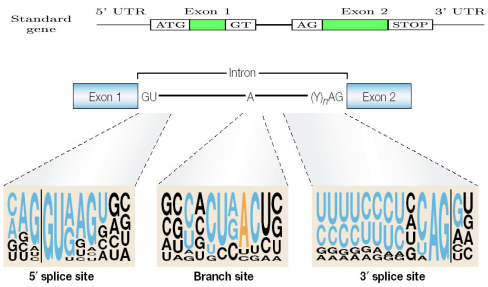
Gene Prediction

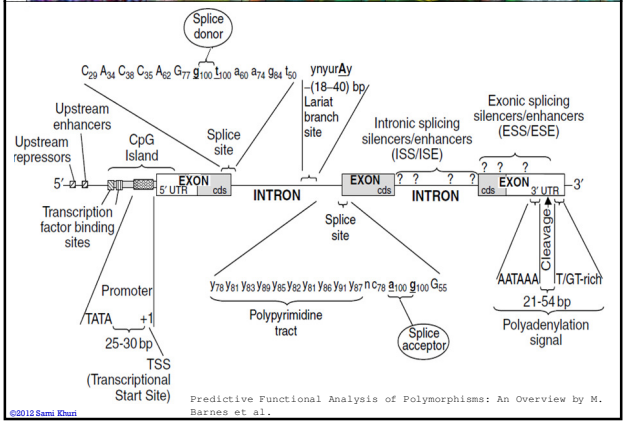


Gene: From DNA to Protein



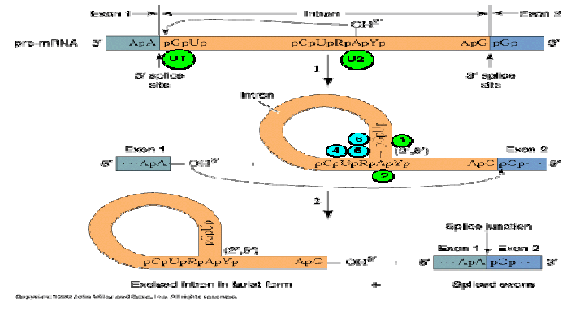
Anatomy of an Intron



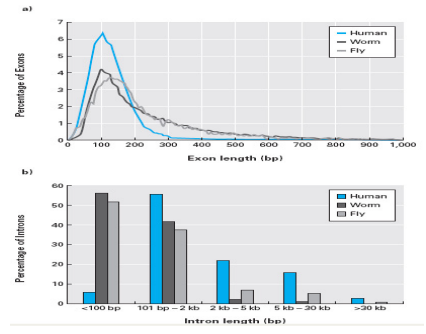


Predictive Functional Analysis of Polymorphisms: An Overview by M. Barnes et al.

The Splicesome



Exons and Introns Lengths



Species comparison of **exon** and **intron** sizes in:
- Homo Spaniens
- C.elegans
- Drosophila

The Position Weight Matrix

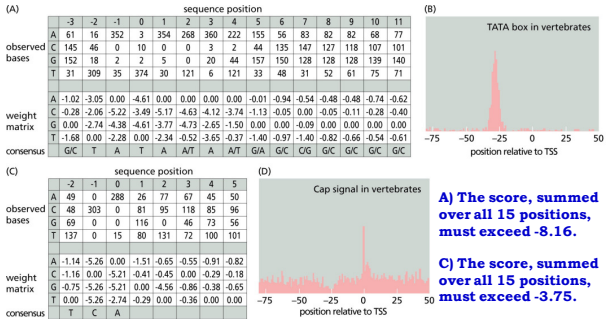
- Due to the high variability of the promoters, exact methods cannot be used for identifying promoter regions.
- Instead we use a pattern search method based on frequencies called the **position weight matrix** method.
- Use the known promoter regions to construct a table of statistics, where an entry is the frequency of a certain base at a given position.

The TATA Box of E.Coli

	A	C	G	T
1	0.04	0.09	0.07	**0.80**
2	**0.88**	0.03	0.01	0.08
3	0.26	0.11	0.12	**0.51**
4	**0.59**	0.13	0.16	0.13
5	**0.49**	0.22	0.12	0.18
6	0.03	0.05	0.02	**0.89**

The frequency of each base found in the -10 position in known E.coli **promoters**.

Weight Matrices for TATA and Cap



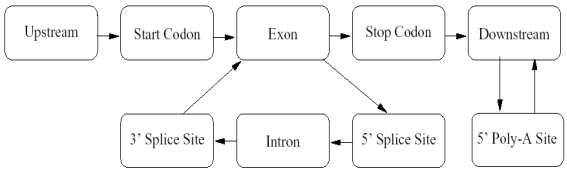
A) The score, summed over all 15 positions, must exceed -8.16.

C) The score, summed over all 15 positions, must exceed -3.75.

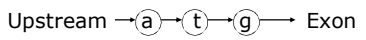
VEIL

- **VEIL:** the **V**iterbi **E**xon-**I**ntron **L**ocator was developed by Henderson, et al. at Johns Hopkins University.
- **VEIL** has a modular structure:
  - It uses a HMM made up of sub-HMMs to describe different parts of the sequence:
    - exon, intron, start, stop, splice, upstream, etc..
- **VEIL** assumes test data starts and ends with noncoding DNA and contains exactly one gene.

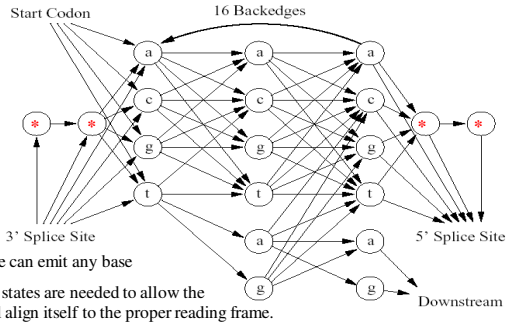
VEIL: The Combined Model



The start codon model is very simple:



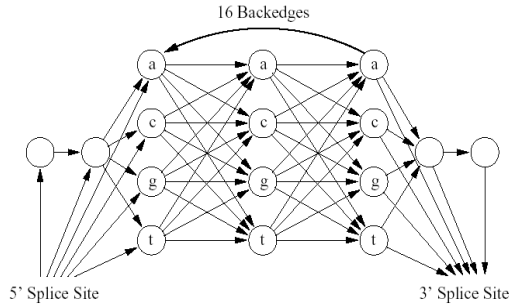
VEIL: Exon Model



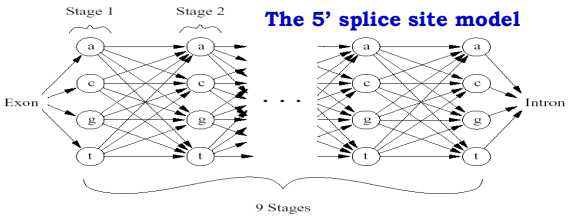
* State can emit any base

The 4 states are needed to allow the model align itself to the proper reading frame.

VEIL: Intron Model

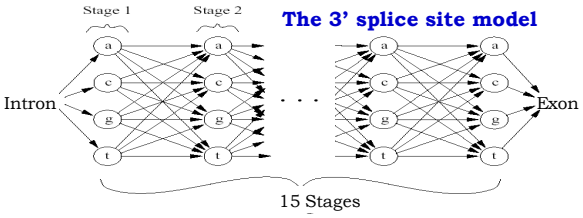


VEIL: Donor Model



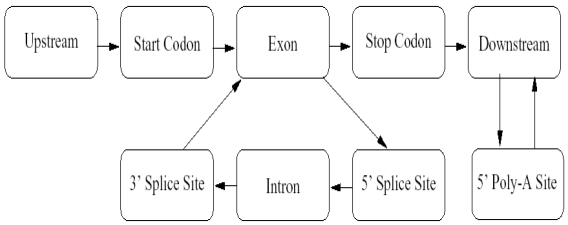
The **donor model** (5' splice site) has 9 stages because it is based on the length of the consensus sequence for donors which is of length 9.
Stages 1 to 3 are part of the exon and 4 to 9 of the intron.

VEIL: Acceptor Model



The **acceptor model** (3' splice site) has 15 stages because it is based on the length of the consensus sequence for acceptors which is of length 15.
Stages 1 to 14 are part of the intron and 15 of the exon.

VEIL: Transitions and States



The **VEIL** hidden Markov model has 1003 edges and 241 states

VEIL: Preparing the Training Data

- They used a set of 570 vertebrate sequences from different species each containing exactly one gene.
- Quality control step: filter the data by removing pseudogenes, entries with no introns (from cDNA), entries with non-standard splice junctions (introns that do not begin with GT and end with AG).
- The 570 vertebrate sequences contain a total of 2649 exons and 2079 introns.

VEIL: Five-Fold Cross Validation

- A five-fold cross validation experiment was performed to estimate how well the system would perform when tested on data that was not in the training set.
  - The 570 sequences were randomly partitioned into five sets of 114 sequences each.
  - For each partition, the system is trained on 4 sets and tested on the fifth.
  - Combine the results from the five test sets.

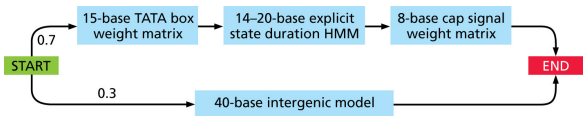
Training and testing VEIL

- The testing involved calculating sensitivity and specificity of both nucleotide labelling (coding/ noncoding) and exons exactly found.
- The nucleotide sensitivity was roughly 80%, and specificity ~ 70%
- Getting both ends of the exon right is a much harder test:
  - sensitivity ~ 50%
  - specificity ~ 50%.

GenScan

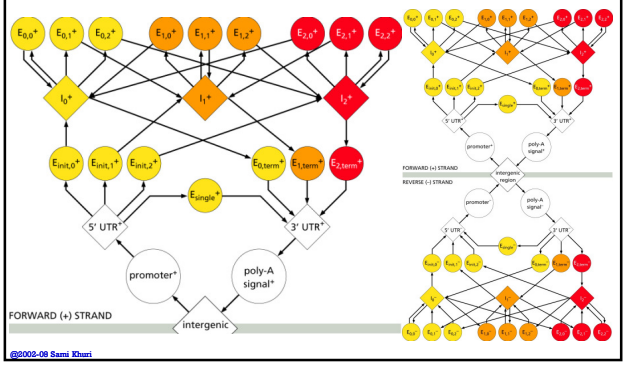
- **GenScan** was Created by Chris Burge
- It uses a Hidden Markov Model to predict whole gene structures.
- GenScan is available online at:
  http://genes.mit.edu/GENSCAN
- GenScan predicts where genes are for vertebrates, maize, and for Arabidopsis.

GenScan: The Promoter

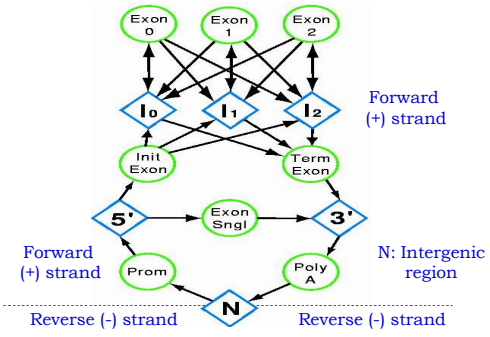


The promoter prediction in GenScan has two alternative and mutually exclusive paths:
• The **TATA box** and **cap signal** are identified using weight matrices.
A sequence of 14 to 20 bases long is between the TATA box and the cap signal.
• A 40-base long region without specific sequence signals identified

GenScan



GenScan: Forward Strand



The Challenging Factors

- The Genome (*H. sap.* & friends) is a desert
- Many false splice sites and other signals
- Very short exons (3bp), especially initial
- Many very long introns
- TATA-less promoters
- Genes within genes
- Alternative splicing

Knowledge Based Gene Prediction

Use characteristics that are known from annotated sequences to predict the genes in unannotated sequences.

- Codon Usage
- Base Composition
- Splice Sites
- CpG Distribution
- Promoters
- Poly A signals
- Transcription Signals
- Translation Signals
- Dinucleotide Frequencies
- Trinucleotide Frequencies
- Hexanucleotide Frequencies
- Branchpoint Distribution
- Enhancers
- Transcription Factors
- Kozak Sequence

Markov Model Assumptions (I)

- A set Q of N states, denoted by 1,2,…,N
- An observable sequence, O:

$$o_1, o_2, \ldots, o_t, \ldots, o_T$$

- An unobservable sequence, q:

$$q_1, q_2, \ldots, q_t, \ldots, q_T$$

- First order Markov model:

$$P(q_t = j \mid q_{t-1} = i, q_{t-2} = k, \ldots) = P(q_t = j \mid q_{t-1} = i)$$

Markov Model Assumptions (II)

- An initial probability distribution:

$$\pi_i = P(q_1 = i) \qquad 1 \le i \le N$$

where $\sum_{i=1}^{N} \pi_i = 1$

- Stationary condition:

$$P(q_t = j \mid q_{t-1} = i) = P(q_{t+l} = j \mid q_{t+l-1} = i)$$

State Transition Probabilities

State transition probability matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1j} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2j} & \ldots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \ldots & a_{ij} & \ldots & a_{iN} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{Nj} & \ldots & a_{NN} \end{bmatrix}$$

where:

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \qquad 1 \le i, j \le N$$

$$a_{ij} \ge 0, \qquad \forall i, j$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall i$$

Hidden Markov Model

- N: the number of hidden states
  A set of states $Q = \{1,2,\ldots,N\}$
- M: the number of symbols
  A set of symbols $V = \{1,2,\ldots,M\}$
- A: the state-transition probability matrix
  $a_{i,j} = P(q_{t+1} = j \mid q_t = i) \qquad 1 \le i, j \le N$
- B: Emission probability distribution; k is a symbol:
  $B_j(k) = P(o_t = k \mid q_t = j) \qquad 1 \le i, j \le M$
- The initial state distribution π:
  $\pi_i = P(q_1 = i) \qquad 1 \le i \le N$

The entire model λ: $\lambda = (A, B, \pi)$

Three Basic Questions

1. **EVALUATION** – given observation $O=(o_1, o_2,\ldots,o_T)$ and model $\lambda = (A, B, \pi)$, efficiently compute $P(O \mid \lambda)$.
   - Given two models λ and λ', this can be used to choose the better one.
     Forward Algorithm or **Backward Algorithm**
2. **DECODING** - given observation $O=(o_1, o_2,\ldots,o_T)$ and model λ find the optimal state sequence $q=(q_1, q_2,\ldots,q_T)$.
   - Optimality criterion has to be decided (e.g. maximum likelihood)
     Viterbi Algorithm
3. **LEARNING** – given $O=(o_1, o_2,\ldots,o_T)$, estimate model parameters $\lambda = (A, B, \pi)$ that maximize $P(O \mid \lambda)$.
   EM and Baum-Welch Algorithms

Conclusion (I)

- Gene finding is improving fast, but is still far from perfect.
- Exons are much better predicted than genes.
- Gene finding is genome-specific : software have to be adapted and trained for each genome.
- The best software for species A (e.g. GenScan for human) is not necessarily the best for species B.

Conclusion (II)

- The more computers are involved in automating genome annotation, the greater the need for collaboration with biologists.
- Best methods are based on detailed probabilistic models that include all known effects.
- We are still a long way from having reliable tools for deducing biological functions from sequences.