

## LECTURE 11: GENE RECOGNITION February 8, 2005

**Lecturer:** Serafim Batzoglou

**Scribe:** William Lu

All figures are from the Lecture 11 PowerPoint slides.

### Administrative

First, a few words about the first problem set. All of you should have your grades by now. All the problem sets for problem set 1 except for a few special cases have been corrected. Everyone seemed to have done pretty well. Since several people have asked about grades, here is the breakdown.

75 or above	Really good
50 or above	Ok
below 50	See the course staff if you need help

### Introduction

Today we talk about genes, which is a popular topic. We will focus on the technical details of identifying genes within a DNA sequence with computational methods.

In a typical human, there are several trillion cells. Our close relatives have a similar number. The nucleus of the cell contains the entire DNA. Our DNA is copied each time the cell duplicates because all the cells need the entire DNA to function. [Actually, this is not true since some of the genes are silenced. For example, one copy of the two X chromosomes in females is silenced.]

In the nucleus, DNA is packaged into physical structures called **chromosomes**, which look like a tight package of yarn. There are 23 pairs of chromosomes in humans. In each chromosome, DNA is wound around proteins called **histones**. When DNA is expressed, it has to be unwound.

Genomics covers many areas and questions, which we have studied or will study.

- **DNA sequencing** determines the letters that make up DNA.
- **Alignment** compares the DNA between related organisms.
- **Gene identification** is the first step in analyzing DNA. It gives all the locations in DNA that encode for proteins.
- **Gene expression** measures the levels and conditions under which genes are expressed.
- **Genome evolution** studies how genes have changed from generation to generation and how they are related.

### Outline of the Next Few Topics

- **Gene Recognition:** Today and the next lecture will be spent on Gene Recognition, which deals with how to find genes in DNA.
- **Large-scale alignment & multiple alignment:** Next week we will talk about Large-scale alignment & multiple alignment, which compares genomes and gene families.
- **Gene Expression and Regulation:** Later we will talk about how to measure the level of expression of genes using microarray technology and how genes are regulated.

### Reading

There is no standard textbook that covers all of the topics discussed today since they are relatively new techniques developed in the past 3 to 4 years. Thus, the readings come from the GENSCAN, EasyGene, SLAM, Twinscan papers linked on the course website. An optional reading is Chris Burge's Thesis (also on the website).

### Gene expression

Up to now, we have talked about gene expression according to the central dogma:

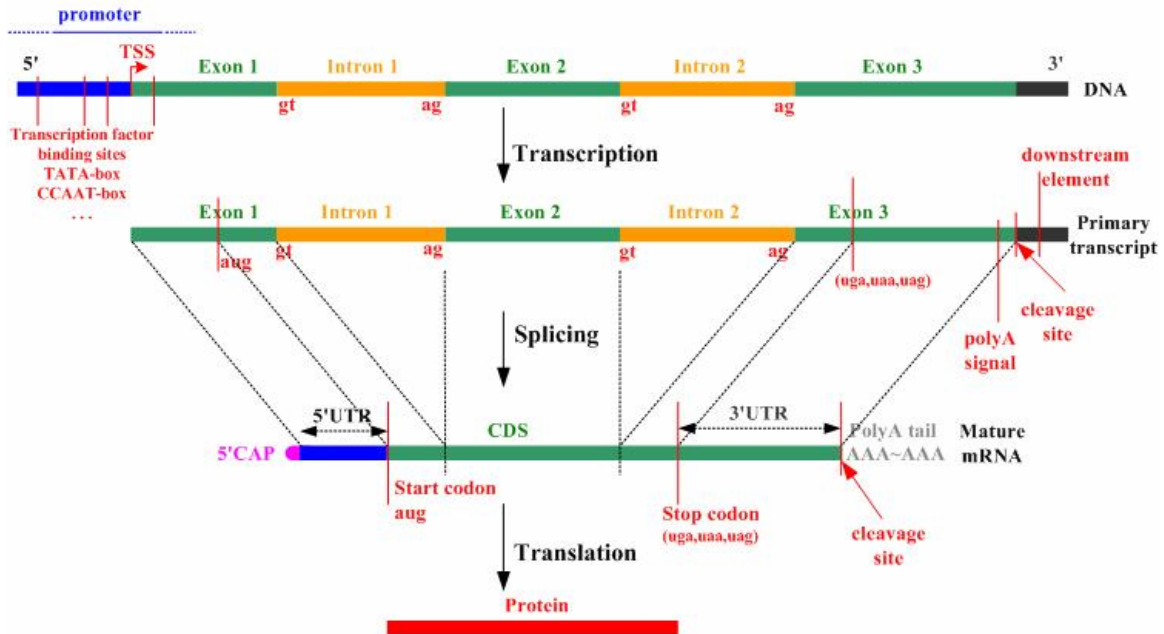


A **protein** is a chain of amino acids that, due to their interactions with the environment, folds to form a 3-dimensional structure. (If you are interested in 3-d structures you can take CS 273 next quarter which will be taught by Serafim Batzoglou and Jean-Claude Latombe.)

Unfortunately, this is not exactly how a gene is expressed. There is an intervening step called **splicing** where the coding parts of the DNA are combined into the mature mRNA, which is translated into protein. The introns are spliced out in this step and thrown away.

**Figure 1** gives an overview of this entire process. In **Figure 1**, the **promoter** is a large regulation region of DNA upstream to the gene. It contains the **co-promoter**, which is where the RNA polymerase sits to start transcription. **TSS (transcription start site)** is the specific position where transcription starts. In a gene, **exons** code for protein (expressed) whereas **introns** are non-coding (*intervening*). A gene must have at least one exon, but it may have zero or more introns.

The DNA transcript is spliced to produce the mature mRNA. The mature mRNA contains **codons**, which are triplets of nucleotides that are converted to amino acids. Protein translation starts at the start codon (AUG) and continues until it reaches a stop codon (UAA, UAG, UGA), which is not translated into an amino acid.



**Figure 1: Gene expression.**

(Just when this seems simple, there is a complication. The stop codon is found in the last exon, but it can also be found in some introns because introns can also exist in the 5' UTR (untranslated region) or the 3' UTR that are found before and after the gene in the mature mRNA transcript, respectively; however, this is a digression and is not necessary to the lecture.)

The best way to find exons is to sequence the mRNA and then align it with the original DNA so they fit perfectly. Unfortunately, this is expensive and only successful for genes that are highly expressed.

In humans, there are approximately 22,000 genes, which make up less than 1.5% of the human DNA, which makes it a very small signal. The number of genes was once larger. Older molecular biology textbooks estimated the number of genes to be around 100,000. (If you happen to have one of these textbooks you may verify this yourself.)

## Finding Genes

Knowing that genes only make up a small proportion of the genome, how do we find these genes? There are five features we can exploit.

1. **Exploit the regular gene structure**
2. **Recognize “coding bias”**

Use the different coding frequencies of the codons. Assuming a random model, we would expect 1/20 of the codons to be the stop codon, but if this does not occur, we probably have an exon.

**3. Recognize splice sites**

Use signatures (GT donor and AG acceptor sites) to determine places where the end of one exon is joined to the beginning of another. Unfortunately, these signals are not specific. Just because they occur at “every” exon-intron boundary does not mean that if we encounter them, it is an exon-intron boundary.

**4. Model the duration of regions**

In mammals, introns tend to be much longer than exons. Also, exons are biased to have a minimum length.

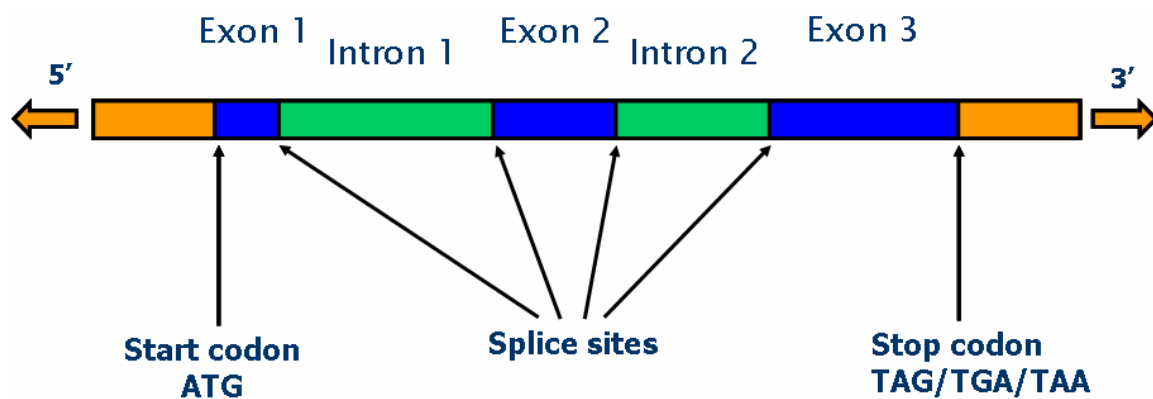
**5. Use cross-species comparison**

Take advantage of the fact that the gene structure is conserved in mammals and that exons tend to be more similar than introns.

In the 1990s, gene finding was one of the hottest research areas and led to a number of systems. **Homology** systems (BLAST and Procrustes) use comparisons with a gene library to determine if a sequence is a gene. **Ab Initio** systems (Genscan, Genie, and GeneID) have no libraries and simply identify genes using the structure of the sequence [hence *ab initio*, meaning from the beginning]. Finally, **Hybrid** systems (GenomeScan, GenieEST, Twinscan, SGP, ROSETTA, CEM, TBLASTX, and SLAM) combine both approaches.

**Exploit the regular gene structure**

In order to find genes, we can take advantage of the gene structure. The first exon must start with the **start codon (ATG)** and the last exon must end with one of the three **stop codons (TAA, TAG, TGA)**. Between the first and last exons, exons and introns alternate, creating a regular pattern as seen in **Figure 2**.



**Figure 2: Gene structure.**

The stretch of nucleotides up to the stop codon is called an **open reading frame (ORF)**. No stop codon can appear in-frame until the end of the gene. (In bacteria, ORFs are really good for identifying genes since they have very little junk DNA.)

Exons can be in one of three frames. The **frame** of the next exon refers to how many nucleotides are left over in the triplet from the previous exon.

- **Frame 0:** Previous exon ends in a triplet (e.g. *gat tac*)
- **Frame 1:** Previous exon ends with 1 letter in the next triplet (e.g. *gat tac a*)
- **Frame 2:** Previous exon ends with 2 letters in the next triplet (e.g. *gat tac ag*)

These frames can be used to match exons. It is important to get everything precise since being off by one letter shifts the entire reading frame resulting in a completely different protein.

## Recognize “coding bias”

**Coding bias** refers to the fact that the different codons appear with different frequencies because the 64 possible codons are degenerate and map to only 20 amino acids and the stop codons.

Amino Acid	SLC	DNA codons
Isoleucine	I	ATT, ATC, ATA
Leucine	L	CTT, CTC, CTA, CTG, TTA, TTG
Valine	V	GTT, GTC, GTA, GTG
Phenylalanine	F	TTT, TTC
Methionine (Start codon)	M	ATG
Cysteine	C	TGT, TGC
Alanine	A	GCT, GCC, GCA, GCG
Glycine	G	GGT, GGC, GGA, GGG
Proline	P	CCT, CCC, CCA, CCG
Threonine	T	ACT, ACC, ACA, ACG
Serine	S	TCT, TCC, TCA, TCG, AGT, AGC
Tyrosine	Y	TAT, TAC
Tryptophan	W	TGG
Glutamine	Q	CAA, CAG
Asparagine	N	AAT, AAC
Histidine	H	CAT, CAC
Glutamic acid	E	GAA, GAG
Aspartic acid	D	GAT, GAC
Lysine	K	AAA, AAG
Arginine	R	CGT, CGC, CGA, CGG, AGA, AGG
Stop codons		TAA, TAG, TGA

**Table I: Coding frequencies of the 20 amino acids and the stop codons.**

We can map the 61 non-stop codons to frequencies. These can be converted to log-odds ratios by dividing the frequency that the codon occurs by the random frequency for all the codons (1/61) and taking the logarithm.

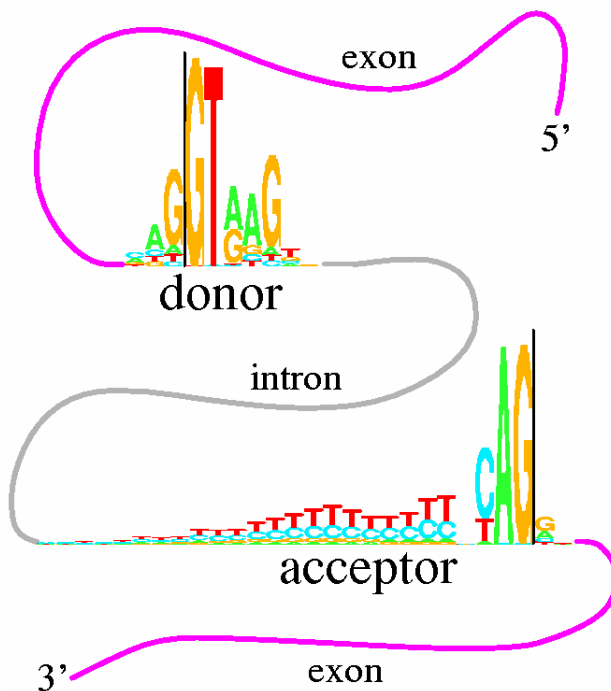
$$score = \log \frac{frequency}{random}$$

If  $score > 0$ , it means the codon frequency is larger than the random frequency, so the sequence is probably in a protein.

If  $score < 0$ , it means the random frequency is larger than the codon frequency, so the sequence is probably not in a protein.

## Recognize splice sites

We want to find the boundary between the beginning of the non-coding region and the end of the coding region. Biologically, in order to splice DNA, enzymes and other factors must be recruited to specific areas of the DNA, known as **splice sites**, where the intron is folded into a looped and cut so that the end of one exon is joined with the beginning of another exon. Detailed images of how this might work can be found at <http://genes.mit.edu/chris/>. Specific signals mark the splice sites. One set of signals that almost always occurs is the GT donor and the AG acceptor sites.



	Position										
%	-8	...	-2	-1	0	1	2	...	17		
A	26	...	60	9	0	1	54	...	21		
C	26	...	15	5	0	1	2	...	27		
G	25	...	12	78	99	0	41	...	27		
T	23	...	13	8	1	98	3	...	25		

**Figure 3: Letter frequencies as a function of position at the donor site.** The 0 position is almost always a G followed by a T.

**Figure 4: Splice sites (Stephens & Schneider, 1996).** The heights of the letters indicate bits of information with the number of bits ranging from 0 to 2. 2 bits indicate that you know the letter exactly or that the letter is almost certain to appear.

Adding together the bits at the donor and acceptor sites yield:

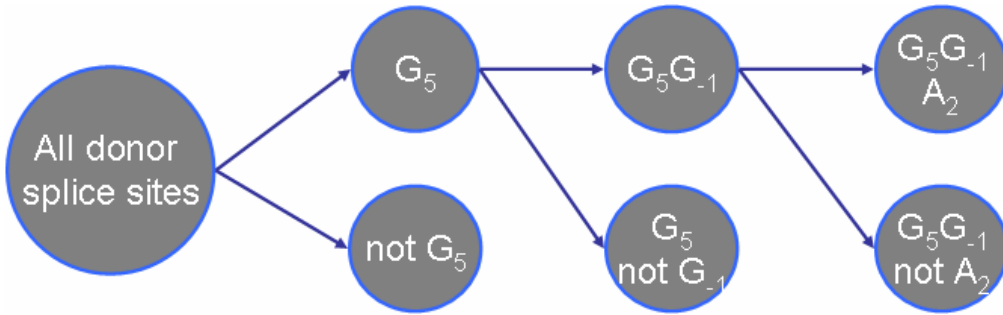
Donor: 7.9 bits  $\approx 1/2^8 = 1/256$  probability of appearing

Acceptor: 9.4 bits  $\approx 1/2^{10} = 1/1024$  probability of appearing

Due to the high probability of appearing, we will end up predicting a lot of donor and acceptor sites even though they are not really donor and acceptor sites. Nonetheless, they are a helpful signal.

Obviously, there have been many machine learning attempts to try to distinguish true donor and acceptor sites from the false sites.

- WMM (weight matrix model) uses a probability scoring matrix PSSM that multiplies the probability of every position independently (Staden 1984).
- WAM (weight array model) uses a 1<sup>st</sup> order Markov model (Zhang & Marr 1993).
- MDD (maximal dependence decomposition) is one of the most accurate models still in use and uses a decision-tree algorithm to account for dependencies (Burge & Karlin 1997). It is used by GENSCAN.



**Figure 5: MDD Algorithm.** The MDD algorithm partitions all donor splice sites into groups based on correlations.

The MDD algorithm is

Take a database of all donor splice sites

For each position  $i$ , calculate  $S_i = \sum_{j \neq i} \chi^2(C_i, X_j)$

(That is, calculate the correlation of each position with all other positions.)

Choose  $i'$  such that  $S_{i'}$  is maximal and partition into two subsets

Repeat until no significant dependencies left

or

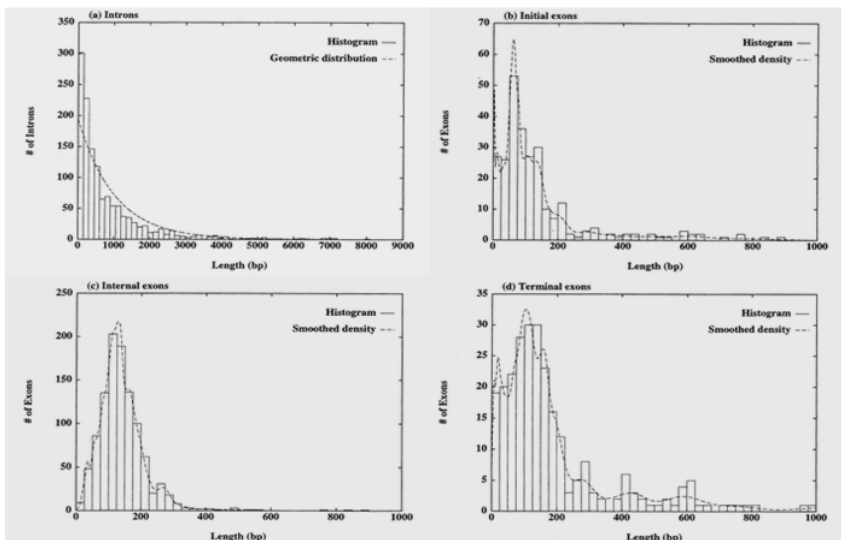
Repeat until not enough sequence in subset

Train separate WMM models for each of these subsets

(That is, assume each position independent and multiply by the probability of being in a splice site versus the probability of not being in a splice site.)

This works because the nucleotides in a splice site recruit enzymes and factors, thus, they are not independent of one another. Performing the algorithm separates these positions into groups statistically.

## Model the duration of regions

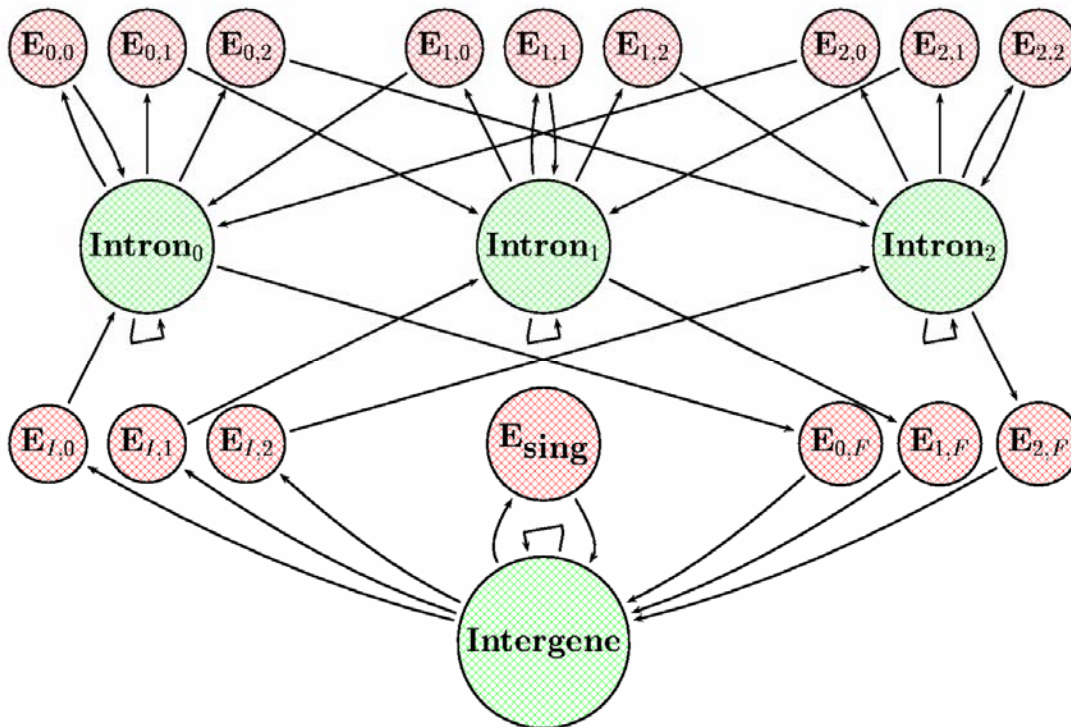


**Figure 6: Clockwise from the top: distribution of intron lengths, initial exon lengths, terminal exon lengths, and internal exon lengths.**

Looking at the distribution of the lengths of introns, they appear to follow an exponential distribution and is very close to a geometric distribution. As a result, they can be easily modeled with a HMM. Unfortunately, the distributions for the exons suggest that they tend to follow a negative binomial distribution. So they require a duration model.

There have been many attempts to use HMMs to capture genes and the best method was the subject of much debate from 1995 to 1997 until GENSCAN beat them all. These HMMs typically consisted of Intergene states to model the region between genes, Exon states, and Intron states. HMM-based gene finders include:

- GENSCAN (Burge 1997)
- FGENESH (Solovyev 1997) – a commercial package, currently one of the best
- HMMgene (Krogh 1997)
- GENIE (Kulp 1996)
- GENMARK (Borodovsky & McIninch 1993)
- VEIL (Henderson, Salzberg, & Fasman 1997)



**Figure 7: Generalized Duration HMM.** The Intergene state can transition to a single exon state. Alternatively, it can go to an initial exon state.

The problem with a HMM is that a gene is translated by coding triplets, or codons, but the HMM is memoryless. Thus, having a simple three state HMM meant that an exon in frame 1 would not know that it is in frame 1 rather than frame 0. The solution is to create three exons states for each case corresponding to frame 0, frame 1, and frame 2 as used in **Figure 7**. Using this, we can perpetuate the frame information onto the intron state,

which is also duplicated to propagate the frame information to the next exon, where the frame information is needed.

Since introns follow a geometric distribution, they can simply be modeled by regular HMM states. Exons, on the other hand, require a special duration state model, where one spends a certain duration  $D$  in the state.

How do we perform algorithms in such a HMM? If we were to run Viterbi, we know at  $V_{E0,0}(i)$ , we have just finished processing  $i$  letters and are in state  $E0,0$ . What are all the ways we could have processed the  $i$  letters? Assuming we emitted the last  $d$  letters in the exon state  $E0,0$ , the probability of emitting those letters is  $\prod_{j=i-d+1..i} e_{E0,0}(x_j)$  times the probability of staying in state  $E0,0$   $d$  times. The remaining  $i-d$  letters must have been emitted earlier, so we use the Viterbi of the Intron0 state to get the probability. Finally, we include the transition probability to go from the Intron0 state to the  $E0,0$  state. If we maximize this probability over all  $d$ , we would get the appropriate function for Viterbi.

$$V_{E0,0}(i) = \max_{d=1..D} \{ \text{Prob}[\text{duration}(E0,0) = d] \times a_{\text{Intron0},E0,0} V_{\text{Intron0}}(i-d) \times \prod_{j=i-d+1..i} e_{E0,0}(x_j) \}$$

where  $i$  is an admissible exon-ending state and  $D$  is a limit based on the longest ORF or a reasonable upper bound.

This is a complicated way to model durations and there is a much better way to do this. As mentioned above, the distribution of exons lengths seem to follow a negative binomial distribution. Thus, it is possible to model exons using the duration HMM based on the negative binomial distribution. Indeed, this was done by EasyGene: Prokaryotic gene-finder in a paper by Larsen & Krogh.

## GENSCAN

One of the best HMMs at identifying genes was GENSCAN, which had a big jump in accuracy for *de novo* gene finding [*de novo* meaning from the beginning]. GENSCAN was developed at Stanford by Chris Burges, who is now a professor of biology at MIT. The actual GENSCAN HMM can do forward parsing and on the reverse complement. It can also start and end the parse anywhere in the original implementation. As a result, it can predict any number of genes.

The real trick to GENSCAN, however, was the fact that it was really four models rather than one. It took advantage of the fact that the amount of C and G nucleotides, which vary biologically due to regulation and silencing of genes by methylation, correlate positively with the gene content and the mean exon length and negatively with the mean intron length. If the CG content is high, you are likely to find more genes, so the model should be more permissive (i.e., intron lengths should be shorter, exon lengths should be longer, and exons should be more frequent). As you lower the CG content, it should

predict fewer genes. The hidden weapon of GENSCAN was that it trained the parameters for the HMM in the four different CG content regions (low, medium, high, and very high). Based on the region it is in, GENSCAN sends the sequence to the correct model for the parse. This demonstrates an important point. *Careful understanding of the biological process and what the real structure looks like can make a big difference in how successful the computational method is.*

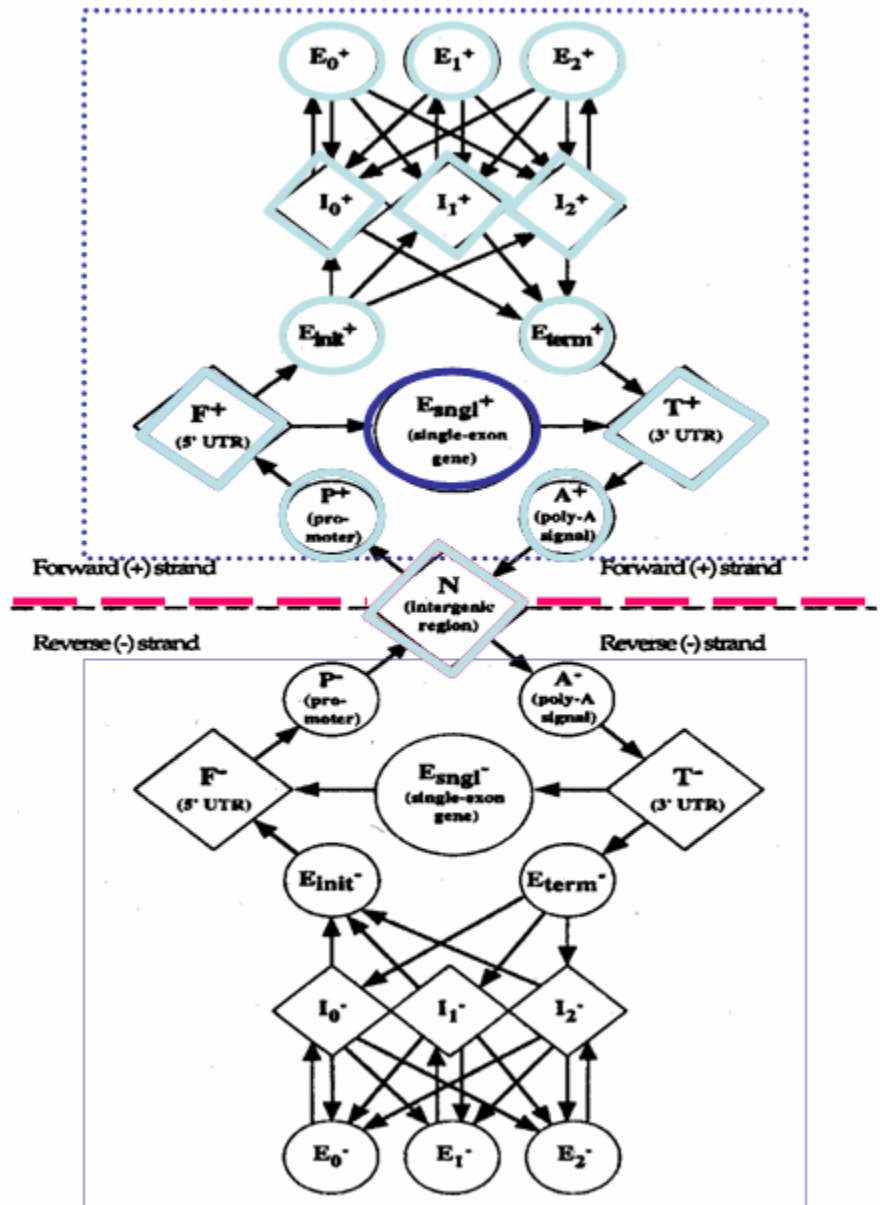


Figure 8: GENSCAN HMM.

GENSCAN had 0.95 sensitivity and 0.90 specificity. It achieved 80% exact exon detection with 10% partial exons and 10% wrong exons. In general, while it has been the best in *de novo* prediction, it overestimated the number of human genes by approximately twofold. It predicted more than 60,000 genes and was promiscuous in gene content regions.

## Evaluation of Accuracy

Understand the results of GENSCAN require a little background on testing.

- **true positive (TP)** refers to the number of actual exons that were predicted to be exons. That is, they were truly, indeed, positive.
- **false negative (FN)** refers to the number of actual exons that were predicted not to be exons. That is, they were wrongly predicted to be negative.
- **false positive (FP)** refers to the number of non-exons that were predicted to be exons. That is, they were wrongly predicted to be positive.
- **true negative (TN)** refers to the number of non-exons that were predicted not to be exons. That is, they were truly, indeed, negative.



Figure 9: Summary of true positive (TP), false negative (FN), false positive (FP), and true negative (TN).

The **sensitivity (Sn)** refers to the fraction of all the exons that were predicted to be exons. It measures how many the test got right. The sensitivity is defined as the true positive divided by the sum of the true positive and the false negative.  $S_n = TP / (TP + FN)$

The **specificity (Sp)** refers to the fraction of all the predicted exons that were in actuality exons. It measure how many the test got true. The specificity is defined as the true positive divided by the sum of the true positive and the false positive.  $S_p = TP / (TP + FP)$

The **correlation coefficient (CC)** combines sensitivity and specificity into one measurement ranging from - 1 (always wrong) to + 1 (always right). Of course, you would never be at - 1 since you can just invert all predictions and be always right.

$$CC = \frac{(TP * TN) - (FN * FP)}{((TP + FN) * (TN + FP) * (TP + FP) * (TN + FN))^{1/2}}$$

## Conclusion

Everything said in this entire lecture is completely false. [This may lead the reader to question whether this statement is true or false.] Throughout this lecture, we have been assuming that one gene gives rise to one spliced mature mRNA, which gives rise to one mature protein; however, there is a process called **alternative splicing**, among other processes, that interferes with our assumption. Alternative splicing affects about half of the genes. In alternative splicing, the same premature mRNA can be spliced in more than one possible way. The other forms may be less abundant than the original form or occur in special cells, but they still exist nonetheless. Even though there are 22,000 genes, there may be more than 1 million different proteins.

## Preview

The next lecture will be about comparison-based gene finding methods. By looking at what is conserved across organisms, we can determine the functions of those regions. These gene finders look at DNA across related organisms and take advantage of the fact that exons will be very similar in those organisms to identify genes.