# Statistical machine learning for computational biology

**Collection Editor:**
Devika Subramanian

# Statistical machine learning for computational biology

**Collection Editor:**
Devika Subramanian

**Authors:**
Andrew Hughes
Devika Subramanian

**Online:**
< http://cnx.org/content/col10455/1.2/ >

**C O N N E X I O N S**

**Rice University, Houston, Texas**

Collection structure revised: October 14, 2007
PDF generated: February 4, 2011
For copyright and attribution information for the modules contained in this collection, see p. 33.

# Table of Contents

# Chapter 1

# Introduction

## 1.1 The inspiration for the design of the course module[1]

### 1.1.1 Inspiration for the Module

The design of this course module is inspired by the following quote from Nobelist Stanley Fields in Science:February 16, 2001:1221-1224:

> *Deciphering how a mere $10^7$ nucleotides result in a yeast cell, let alone how $3 \times 10^9$ nucleotides result in a human – cannot begin until the genes have been annotated. This step includes figuring out the proteins these genes encode and what they do for a living. But understanding how all of these proteins collaborate to carry out cellular processes is the real enterprise at end.*

The quest, indeed, is for the wiring diagrams of life; particularly how they are altered in diseased cells. A fine example is the well-known cancer subway map by William C. Hahn and Robert A. Weinberg published in Nature Reviews Cancer in 2002.

[1] This content is available online at <http://cnx.org/content/m15082/1.1/>.

**The Cancer Subway Map**

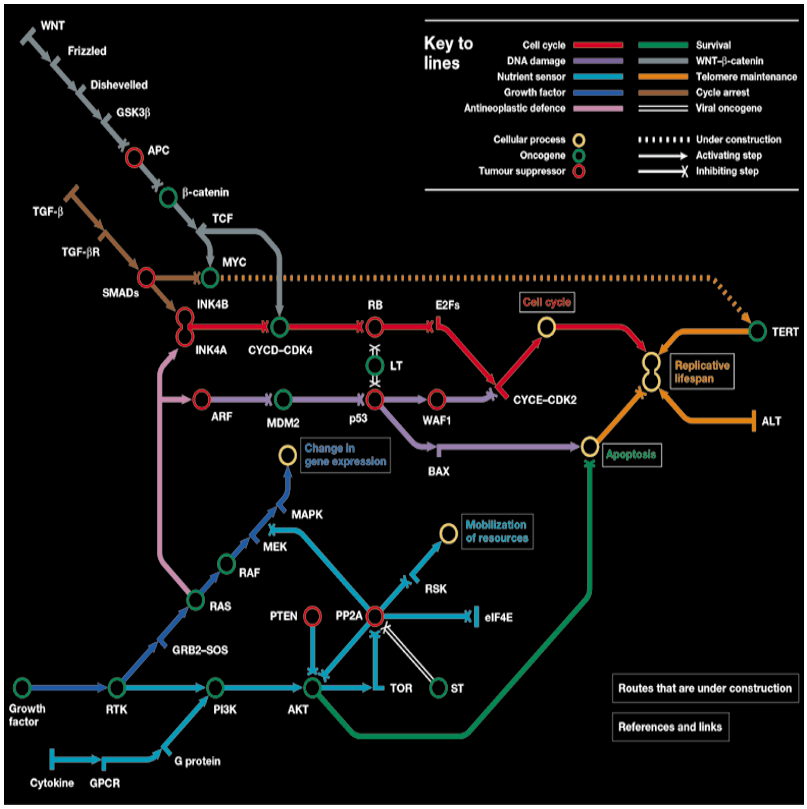


**Figure 1.1:** The Cancer Subway Map (from http://www.nature.com/nrc/poster/subpathways/).

### 1.1.2 Biology in the 21st century

Biology is awash in data today. We have ready access to the sequences of the human and other genomes, structures for several thousand proteins, sequences for over 1.5 million proteins, and information on thousands of protein-protein interactions (with over a billion interactions predicted). High-throughput assays such as microarrays, flow cytometry, SELDI-TOF spectra, cell-level imaging, and array CGH give us unprecedented access to the functioning of cells. All of this data needs to be interpreted to reveal models of cellular function so that we can understand the molecular basis of disease, and design appropriate therapeutic interventions. Data driven exploration of theories is the standard scientific paradigm in modern biology. However, these new technologies have accelerated the volume and pace at which data can be gathered, requiring significant use of computation. Further, it has allowed the focus of research to shift from individual components of a cell to system-level analysis. The situation is depicted nicely in the following cartoon taken from the Fields article in Science:February 16, 2001:1221-1224.

**Fishing from the molecular biology pond.**

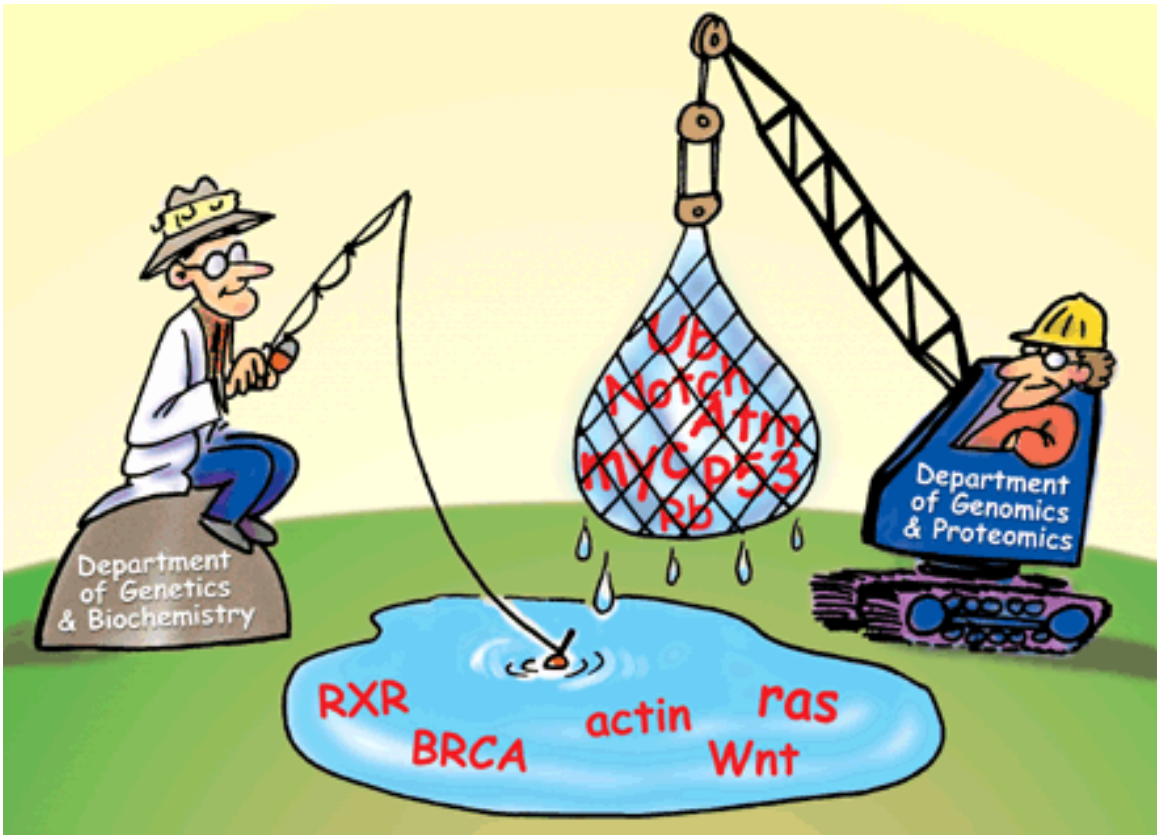


**Figure 1.2:** The new world of computational biology (from http://www.sciencemag.org/cgi/content/full/291/5507/1221/F1)

### 1.1.3 Statistical Machine Learning

What is statistical machine learning? It is the science of understanding complex systems (such as cells) by actively gathering data from them, synthesizing the data using prior knowledge (if any) of the systems to form models, and using the models to predict responses of the system to interventions. The fundamental questions in machine learning are:

- Feature Selection: What aspects of the system should be observed?
- Model Selection: What class of models need to be built from observed data and prior knowledge?
- Model validation: How do we evaluate the efficacy of the learned models?

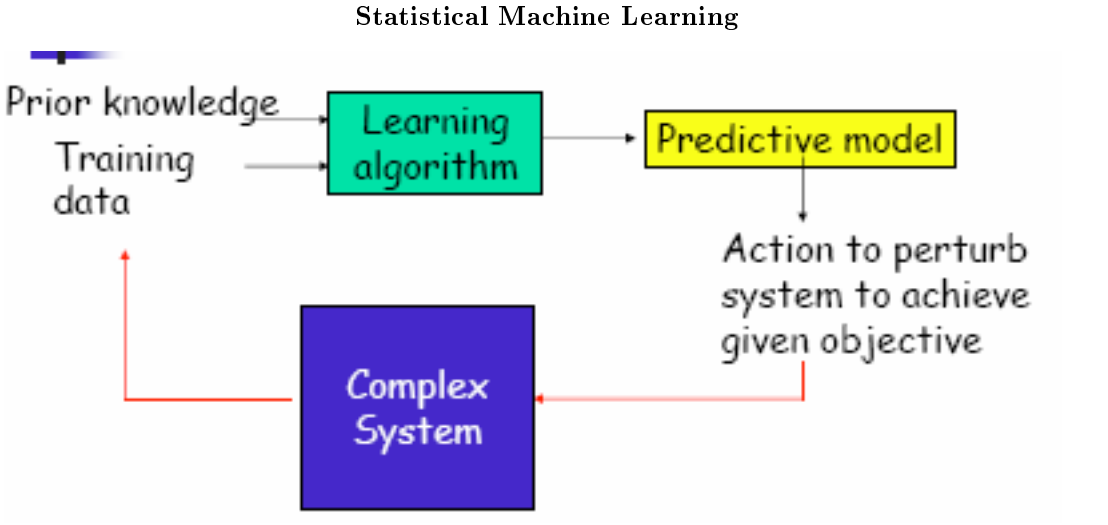


**Figure 1.3:** Observe complex systems and build models to predict their response to interventions.

### 1.1.4 Three problems from computational biology

The short course focuses on three central problems in computational biology that are at three different levels of abstraction.

- Computational Genefinding: Given a DNA sequence, find and annotate genes in it.
- Molecular fingerprinting of disease: Given micro-RNA expression levels in normal and diseased cells, find biologically significant genes that are differentially expressed.
- Learning signaling and regulatory networks: Given flow cytometry data from normal and diseased cells, learn signaling networks from them.

Here is the computational genefinding problem cast in the framework of statistical machine learning. The observational data are annotated stretches of a genome, and the models learned are Hidden Markov models which are sequential models for labeling new DNA segments.

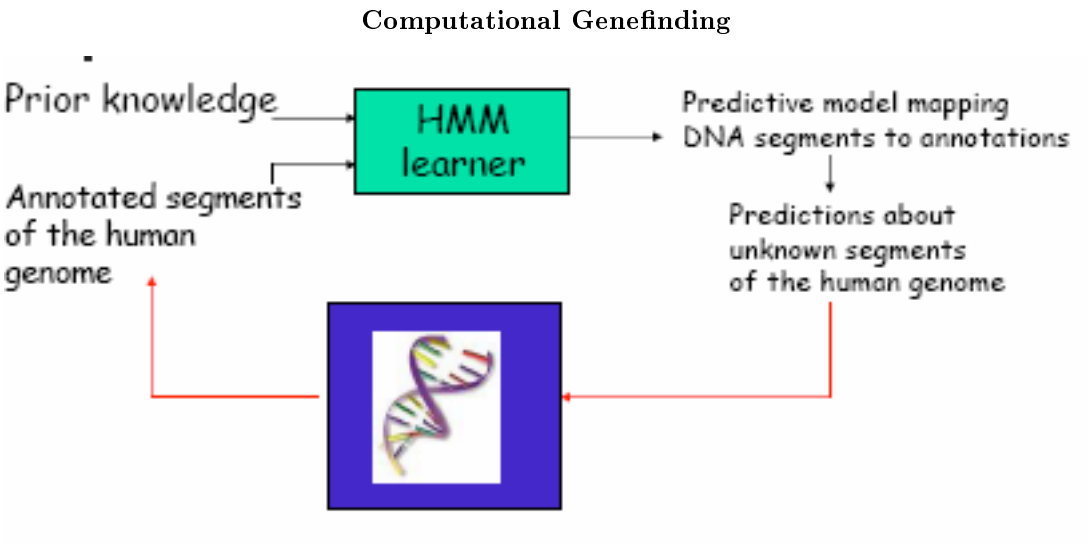


**Figure 1.4:** Computational genefinding as a statistical machine learning problem.

Here is the molecular fingerprinting or the biomarker discovery problem cast in the framework of statistical machine learning. The observational data are the mRNA or proteomic expression data from diseased and normal cells, and the model is a classifier that discriminates between them, and identifies the key genes or proteins that are key to classification.

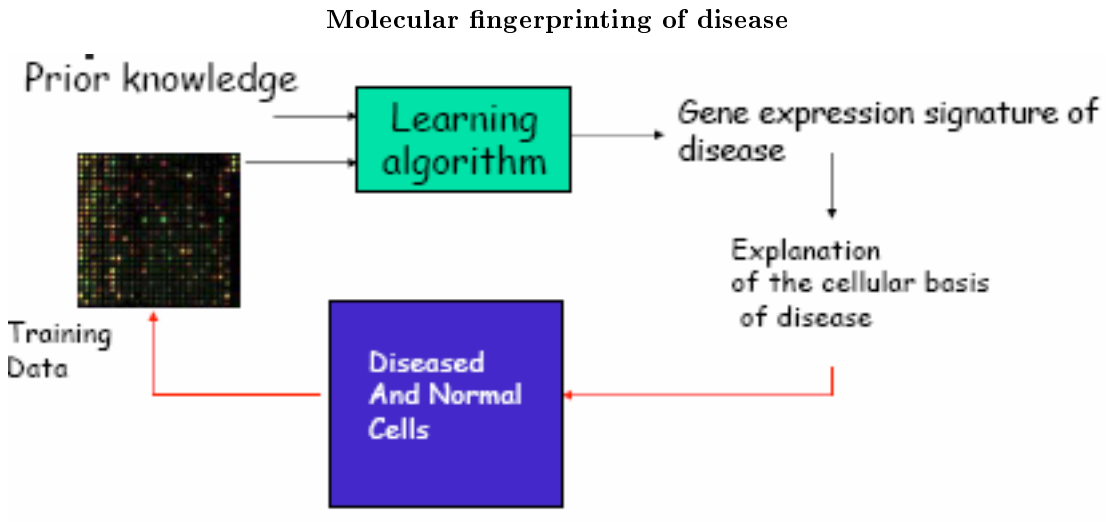


**Figure 1.5:** Molecular fingerprinting of disease as a statistical machine learning problem.

Here is the problem of learning cell signaling networks from flow cytometry data cast as a statistical machine learning problem. The observational data are the measurements at a given time of the levels of signaling molecules in a cell, prior knowledge could be known interactions between the molecules, and the

predictive model learned in a signaling network that is the best fit to the data. Therapeutic interventions can then be planned on the learned model. Since there are tremendous individual variations in cells with cancer, an ab-initio technique for inferring signaling pathways directly from observational data can pave the way for the era of personalized cancer therapy.

**Learning signaling networks from data.**

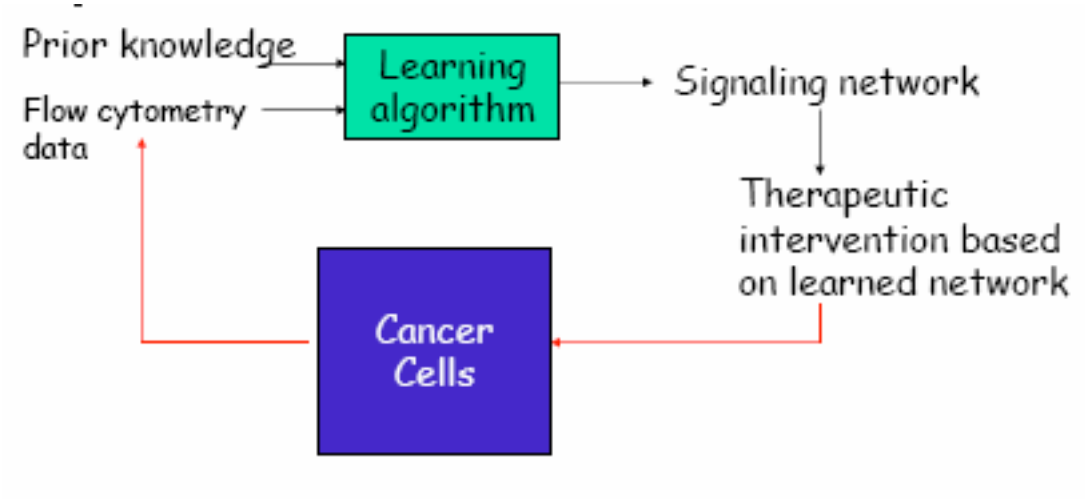


**Figure 1.6:** Learning cell signaling networks from data.

Three statistical machine learning algorithms will be introduced in the context of these three problems.

- Hidden Markov models and variants.
- k-NN classifiers and support vector machines.
- Bayesian networks: learning parameters and structure.

### 1.1.5 Module Objectives

The module objectives are:

- To show how to handle heterogeneous biological data, how to formulate biological problems in the statistical machine learning framework, and how to choose appropriate algorithms for these problems.
- To cover the basics of supervised and sequential machine learning algorithms with particular focus on Hidden Markov Models, k-NN and SVM classifiers, and Bayesian networks.
- To provide opportunities to apply these methods in the context of real data (human chromosome 22, prostate cancer gene expression data, and flow cytometry data from T-cell signaling).

## 1.2 The structure of the module[2]

### 1.2.1 Computational genefinding using Hidden Markov Models

The first problem we will study is the annotation of DNA sequences into regions of interest. Our focus is on finding genes that code for proteins. Our approach is to use available annotated DNA sequences to train

[2]This content is available online at <http://cnx.org/content/m15081/1.1/>.

sequential models, and then to use the trained model to label new DNA segments. We will cover ab initio methods, as well as comparative methods which differ on whether information from other genomes is used as prior information.

Hidden Markov models (HMMs) are the technique of choice for the problem of finding genes in DNA sequences. We will cover the structure of HMMs, the Viterbi algorithm for annotation, the Baum-Welch algorithm for learning models, and pair HMMs to model genefinding in a comparative context. GENSCAN[3] , the paradigmatic example of an ab initio gene finder, will be presented. If time permits, SLAM[4] , a comparative gene finder based on pair HMMs will be introduced.

Since gene finding is complex, our exercise for this portion of the course will be to detect CpG islands on chromosome 22[5] . We will compare the performance of a global decoding methods (Viterbi decoding) against that of a local method (posterior decoding or smoothing).

### 1.2.2 Biomarker discovery and supervised learning

The next problem we will cover is that of molecular fingerprinting of disease. This is also known as the biomarker discovery problem. Given mRNA expression levels, or protein levels from normal and diseased cells, the computational problem is of determining biologically significant genes or proteins that are differentially expressed. This is usually the first step in generating causal models of disease. This part of the course draws on the pioneering work of Golub et. al.[6] We will model the problem in the supervised learning framework and introduce k-nearest neighbours and support vector machine classifiers.

In this section of the course, we will analyse the prostate cancer microarray data set from the Broad institute to build classifiers which discriminate between cancer and normal cells. We will also experiment with various feature selection techniques to identify biologically significant genes that are differentially expressed in diseased cells. We will compare the relative effectiveness of global hyperplane classifiers like support vector machines against local methods as exemplified by k-nearest neighbor classifiers.

### 1.2.3 Systems biology and learning Bayesian networks from data

The availability of high-throughput data which reveals the levels of mRNA and proteins in cells and their change over time, makes it possible to construct system level models of cellular activity. The inspiration for this part of the course comes from the 2005 study by Sachs et. al. [7] which uses multi-parameter flow cytometry to reconstruct the T-cell signaling network in humans. The mathematical foundations of Bayesian networks will be covered, as well as the sparse candidate algorithm for learning Bayesian networks from high-throughput data. The use of interventional data to determine causal edges in the network will also be discussed.

The experiment for this section of the course is to use Bayesian network algorithms and the flow cytometry data available from the Science website to recreate the Sachs et. al. derivation of the T-cell signaling network.

### 1.2.4 Summary of course objectives

This course will teach you

- how to use the underlying biology to constrain feature and model selection.
- how to choose and adapt machine learning algorithms for biological problems.
- how to design learning protocols to deal with incomplete, noisy data.
- how to interpret the results of machine learning algorithms.

[3] http://genes.mit.edu/GENSCAN.html
[4] http://www.genome.org/cgi/content/abstract/13/3/496
[5] http://www.sanger.ac.uk/HGP/Chr22/
[6] http://www.sciencemag.org/cgi/content/full/286/5439/531
[7] http://www.sciencemag.org/cgi/content/full/308/5721/523

# Chapter 2

# Computational Genefinding

## 2.1 Basics of eukaryotic genomes[1]

The most naive picture of the eukaryotic genome is a long string of linear DNA balled up somewhere inside the cell. This formulation fails on several important grounds: first, although DNA is a linear molecule, it is not necessarily accessed in a linear fashion; second, DNA has a very significant secondary structure, it is not simply balled up at random; and third because DNA does not act in isolation, it is immersed in the context of the cell's nucleus where numerous proteins and epigenetic processes interact with the DNA to regulate gene expression.

Let's begin by discussing the in vivo structure of DNA in a typical eukaryotic cell. A molecule of DNA is composed of two antiparallel and complimentary strands of deoxyribonucleicacid. Antiparallel means that the two strands have opposite chemical polarity, or, stated another way, their sugar-phosphate backbones run in opposite directions. Direction in nucleic acids is specified by referring to the carbons of the ribose ring (ribose is a sugar) in the sugar-phosphate backbone of DNA. 5' specifies the the 5th carbon in the ribose ring, counting clockwise from the oxygen molecule, and 3' specifies the 3rd carbon in the ring. Direction of, and in reference to, DNA molecules is then specified relative to these carbons. For example, transcription, the act of transcribing DNA to RNA for eventual expression, always occurs in the 5' to 3' direction. Nucleic acid polymerization cannot occur in the opposite direction, 3' to 5', because of the difference in chemical properties between the 5' methyl group and the 3' ring-carbon with an attached hydroxyl group.

[1] This content is available online at <http://cnx.org/content/m11320/1.2/>.

**DNA Helix**

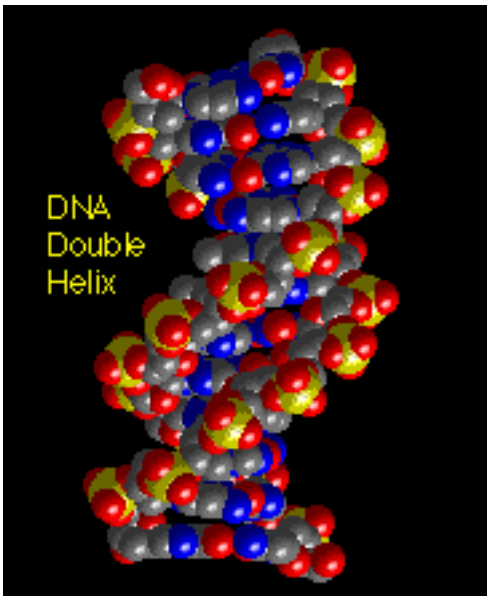


**Figure 2.1**

The basic structure of DNA can be divided into two portions: the external sugar-phosphate backbone, and the internal bases. The sugar-phosphate backbone, as its name implies, is the major structural component of the DNA molecule. It is the external portion of the DNA molecule because it is highly polar, and thus hydrophillic (meaning it likes to be immersed in water). Correspondingly, the interior bases of the DNA molecule are non-polar and hydrophobic. This duality has a very stabilizing effect on the overall structure of the DNA double helix: the hydrophobic core of the DNA molecule 'wants' to be hidden inside the sugar-phosphate backbone which acts to isolate it from the polar water molecules; thus there is a strong hydrophobic pressure gluing two molecules of DNA together.

There are four bases in DNA: adenine (A), guanine (G), thymine (T), and cytosine (C). In RNA uracil (U) is found in place of thymine (T). Inside a DNA molecule these bases pair up, A to T and C to G, forming hydrogen bonds that further serve to stabilize the DNA molecule. Because the interior bases pair up in this manner, we say the DNA double helix is complimentary. It is the sequence of these bases inside the DNA molecule that we refer to as the genetic code.

**DNA Structure**

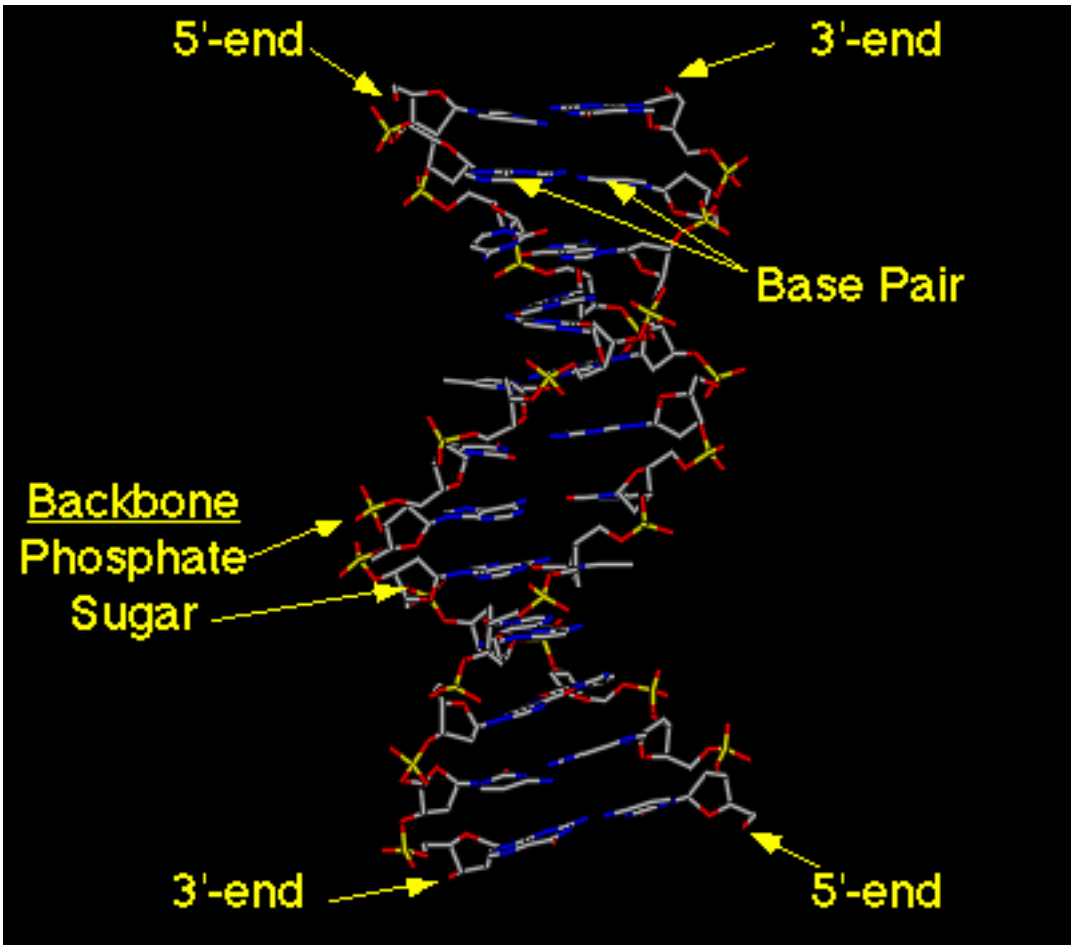


**Figure 2.2**

At this point we now have a good picture of the chemical structure of the DNA molecule, now we need to begin placing it in the context of the cell. A typical eukaryotic chromosome contains from 1 to 20 cm of DNA. However, during metaphase of mitosis and meiosis, this DNA is packaged in a chromosome with a length of only 1 to 10 um. How is this amazing density achieved inside the cell?

DNA in the cell exists packed into a dense and regular structure called chromatin. Chromatin is composed of DNA, proteins, and a small amount of RNA. The proteins found in chromatin largely consist of histones, a basic protein which is positively charged at neutral pH, and nonhistone chromosomal proteins which are largely acidic at neutral pH. Histones have been highly conserved in all eukaryotes. There are five major histone types, called H1, H2a, H2b, H3, and H4, and which exist in specific molar ratios within the chromatin. Histones bind together with the DNA to form the basic structural subunit of chromatic, small ellipsoidal beads called nucleosomes which are around 11nm in diameter and 6nm high. Each nucleosome contains 146 nucleotide pairs which wrap around the histon protein complex 1 and 3/4 turns. The nucleosome complexes give the DNA molecula a packaging ratio of 6.

**Histones**

***Image not finished***

**Figure 2.3**

Beyond the nucleosome, there are two more levels of structural packaging. The second level of packing is the coiling of the nucleosome beads into a helical structure called the 30 nm fiber that is found in both interphase chromatin and mitotic chromosomes. This structure increases the packing ratio to about 40. The final packaging occurs when the fiber is organized in loops, scaffolds and domains that give a final packing ratio of about 1000 in interphase chromosomes and about 10,000 in mitotic chromosomes.

One important note is that DNA is not always packed into the super-dense chromosome structures evident during mitotic and meiotic replication. During interphase, or the general not-currently-reproducing phase of the cell where most of a cell's work is done, the chromatin, while still highly dense, is about 1/10 as dense as during cellular replication. This is important because it is believed that the highly-dense chromatic structure of DNA sterically inhibits transcription and thus gene expression. In order for genes to be expressed the chromatin structure must be relaxed so that the transcriptional proteins can gain access to the DNA molecule.

Now that we have a good grasp on the basic structure of DNA as a molecule, as well as in vivo, lets move on to the mechanisms of gene expression. The Central Dogma of genetics is: DNA is transcribed to RNA which is translated to protein. Protein is never back-translated to RNA or DNA, and except for retroviruses, DNA is never created from RNA. Furthermore, DNA is never directly translated to protein. DNA to RNA to protein.

DNA is the long term, stable, hard-copy of the genetic material; by way of analogy it is similar to the information on a computers hard-disk drive. RNA is a temporary intermediary between the DNA and the protein making factories, the ribosomes. To further extend our computer analogy, RNA could be compared to information in a cache, in that the lifetime of RNA is much shorter than that of either DNA or the average protein, and that RNA serves to carry information from the genome, located in the nucleus of the cell, to the ribosomes, which are located outside of the nucleus either in the cytosol or on the endoplasmic reticulum (which is a large set of folded membranes proximal to the nucleus that help manufacture proteins for extra-cellular export). To complete our analogy, proteins could be viewed as the programs of the cell. They are the physical representation of the abstract information contained within the genome. However, one caveat is that RNA does have some enzymatic activity and has other functions besides ferrying messages between the DNA and the ribosomes.

Transcription is the process of creating RNA from DNA. Transcription is also the point at which most of the regulation of gene expression occurs and because of this it is a very complex process, especially with regard to its initiation. To say that DNA is transcribed to RNA is a nice (over)simplification, but we need to delve a little deeper into the details to really appreciate what is going on during transcription. A more complete view of transcription includes five steps: 1) transcription of DNA to pre-mRNA, 2) a 7-methyl guanosine cap is added to the 5' end of the transcript, 3) a poly(A) tail is added to the 3' end of the transcript, 4) the introns are spliced out of the pre-mRNA, which finally yields, 5) the mRNA transcript proper.

Because the first step, the initial transcription of DNA to pre-mRNA, is the most involved, I am going to hold off on discussing it for a moment and expand on steps 2-5 first. (2) The addition of the 5' 7-MG cap is important for two reasons: the 5' caps are recognized by protein factors that initiate translation, and it also helps protect the transcript from nucleases. Nucleases are very common in the cell and because of this unprotected RNA has a very short half-life inside the cell. Nucleases are actually so common that working with RNA in the laboratory can be quite difficult because the samples have a tendency to disintegrate

into useless bits. (3) The poly(A) tails are formed in a two step process: an endonulcease cleaves around 1000-2000 non-coding bases from the 3' end of the pre-mRNA transcript and then poly(A) polymerase adds 20-200 AMP molecules to the 3' end of the transcript. The poly(A) tail is important in the cellular transport of the mRNA transcript and, like the 5' cap, also helps to stabilize the mRNA transcript.

Once the 5' cap and the poly(A) tail have been added, only one step remains for the pre-mRNA transcript to be complete and graduate to mRNA status: splicing. Eukaryotic genes contain two types of transcribed regions: introns and exons. Exons are the regions of the genome that contain actual coding information. Introns are non-coding, meaning that intronic sequences are never translated to protein, in fact they are never included in the final processed mRNA transcript. Splicing is the process of removing introns from the pre-mRNA transcript to produce an exon-only mRNA molecule, which is then shipped off for translation. Generally, eukaryotic mRNAs are considered to monogenic. However, up to one fourth of the transcripts in C. elegans have been show to be multi-genic (i.e. they contain exons from multiple genes).

A further complication of the splicing process is that mRNA can undergo alternative splicing. To illustrate this let's imagine a gene that has 3 exons and two introns. From this gene, three different final transcripts are possible. In all transcripts the two introns are going to be removed, however, the cell can combine the exons however it wants as long as the original order is maintained. This means that for this example the possible mRNA transcripts include: Exon1-Exon2, Exon1-Exon3, and Exon1-Exon2-Exon3; however, Exon3-Exon1 is not possible because the exons are out of order.

An interesting side note is that some introns are capable of self-splicing, that is they can politely remove themselves without the intervention of any proteins. This is significant mainly because it is a significant counter example to the idea that RNA is an inert transcript and action is soley the domain of proteins. RNAs should really be viewed as having both enzymatic properties and abstract information-carrying ability. Because of this many people believe that RNA was the original genetic molecule and that DNA and proteins evolved later in the game.

Alternative splicing is a very important and powerful tool. To understand the benefit alternative splicing gives the cell we need to understand something about proteins. Proteins can be understood as containing modularized functional units. These functional units can be active sites on enzymes, large structural motifs such as beta-sheets or alpha-helices, or motifs that direct the eventual destination of expressed proteins. A good example of an alternatively spliced pre-mRNA transcript is the mouse IgM immuoglobulin transcript. IgM exists in two forms: excreted and membrane bound. These two forms of the protein differ in the only in the C-terminus: the secreted protein has a secreted terminus motif while the membrane-bound protein has a C-terminal membrane anchor region. Both products come from the same pre-mRNA, but alternative splicing includes either the terminal exon that creates the excreted form of IgM or the membrane-bound form of IgM.

This is a good time to take a step back from our discussion, take a deep breath, and summarize what we have covered so far. (1) DNA exists as a double stranded helix that is both complimentary and antiparrallel. (2) DNA in vivo exists in a very compact and regular structure of nucleosomes, 30nm fibers of braided nucleosomes, and loops of fibers. (3) The central dogma of genetics: DNA is transcribed to RNA, which is then translated to proteins. (4) DNA is the stable, long-term form of genetic information. (5) RNA is (mostly) an intermediary between DNA and the protein-making-factories, ribosomes. (6) RNA transcription is not nearly as simple as the central dogma might lead you to believe. Which leads us to the point I put off earlier: how is transcription initiated in the eukaryotic genome?

## 2.2 Transcription in eukaryotic genes[2]

### 2.2.1 Properties of Eukaryotic Transcription

- Complex!
- RNA polymerase is the main protein that creates the RNA transcript
- Proteins called transcription factors help RNA polymerase and regulate it's function.

[2]This content is available online at <http://cnx.org/content/m11416/1.2/>.

- 2 types of transcription factors: basal tfs and modulatory tfs.
- Basal transcription factors - required for all transcription events.
- Modulatory transcription factors - regulate expression of genes. Both positive and negative regulation.
- Transcription factors can act from relatively far away. Up to 10kb!
- pre-mRNA transcripts are spliced to remove exons.
- Alternative splicing allows the cell to combine exons differently in the final mRNA transcript from the same pre-mRNA transcript.

### 2.2.2 Transcription In Five Easy Steps

1. Transcription of DNA to pre-mRNA. DNA, RNA polymerase and transcription factors.
2. Addition of 5' methyl guanosine cap to pre-mRNA transcript.
3. Splicing of pre-mRNA transcript (yields mRNA proper)
4. Addition of 3' poly(A) tail to mRNA transcript.
5. mRNA is transported to the cytoplasm.

**5 Steps of mRNA preparation**

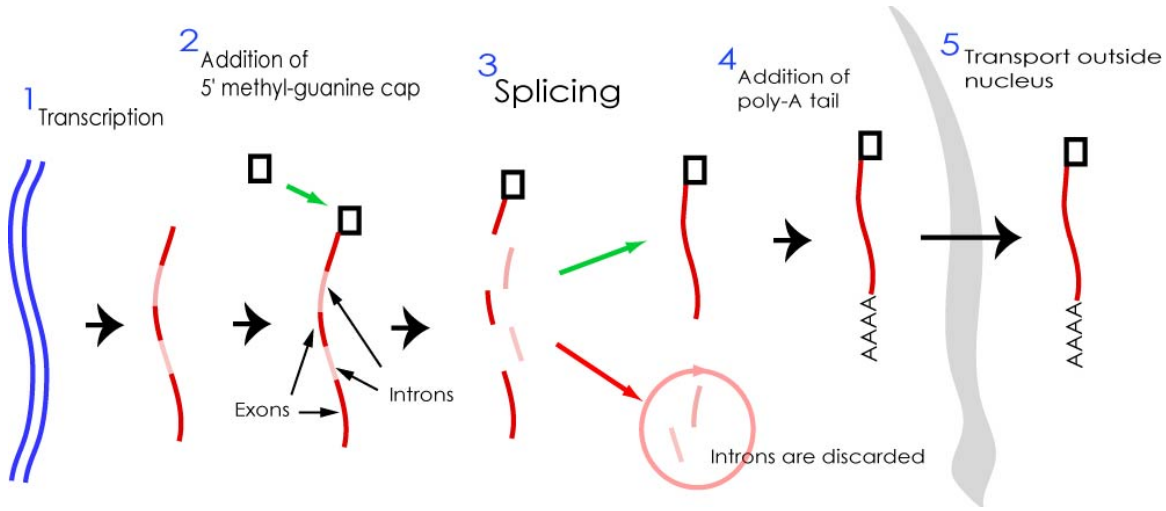


**Figure 2.4**

### 2.2.3 RNA Polymerase - main protein that creates RNA transcript.

RNA polynmerase is the protein whose job it is to 'read' the genetic code and create a complimentary RNA transcript from that code. Eukaryotes have three different types of RNA polymerases: RNA polymerase I, II, and III. RNA polymerase II is the from of polymerase that transcribes most genes and is the form of polymerase with which we need to concern ourselves.

### 2.2.4 Initiation of Transcription

Transcription initiation in eukaryotes is complicated, and the details are not entirely understood (although we do have a good grasp on the basic mechanism). The fully assembled eukaryotic transcription inititation complex contains more than 50 polypeptides. RNA polymerase II has more than 10 polypeptide subunits by itself. Keep in mind that the process outlined below is a generalization, and any given, specific transcription event will probably vary some in the details. For example, although the TATA box is the most strongly conserved promoter sequence, it is by no means present in every eukaryotic promoter. Also, there is some debate as the importance of the order of the binding of the proteins in the initiation complex: it was thought that a specific order was vital, now, however, new evidence suggests that reaching the end binding state may be what is really important, whatever the order.

**Transcription Factors - help/regulate RNA polymerase's function**

Transcription factors (tfs) are proteins involved with transcription - except for RNA polymerase. Transcription factors can be broken down into two groups: basal tfs and modulatory tfs. Eukaryotic RNA polymerases are not capable of initiating transcription alone, they require the assistance of a set of basal transcription factors. Basal tfs assist RNA polymerase in the recognition of promoter sequences and unwiding the DNA double helix, among other functions. Basal tfs are necessary for every transcription event. Modulatory transcription factors regulate the expression of a gene, or a set of gene. These tfs are important because they allow the body to differentially express genes at different times and different places in the body. Modulatory tfs are vital for multicellular life. They allow the body to create different cells, tissues, and organs.

- Learning the specific order of tf binding is not vital
- Understand that transcription initiation involves MANY polypeptides, each with a specific function.
- Functions include: recognizing promoter sequences, modulating amount of transcript production, modulating time/space of transcript production, and unwinding DNA
- Each tf/protein represents a functional unit

Transcription factors are denoted TFIIX, for Transcription Factor for polymerase II. X is an identifying letter.

**Transcription Initiation**

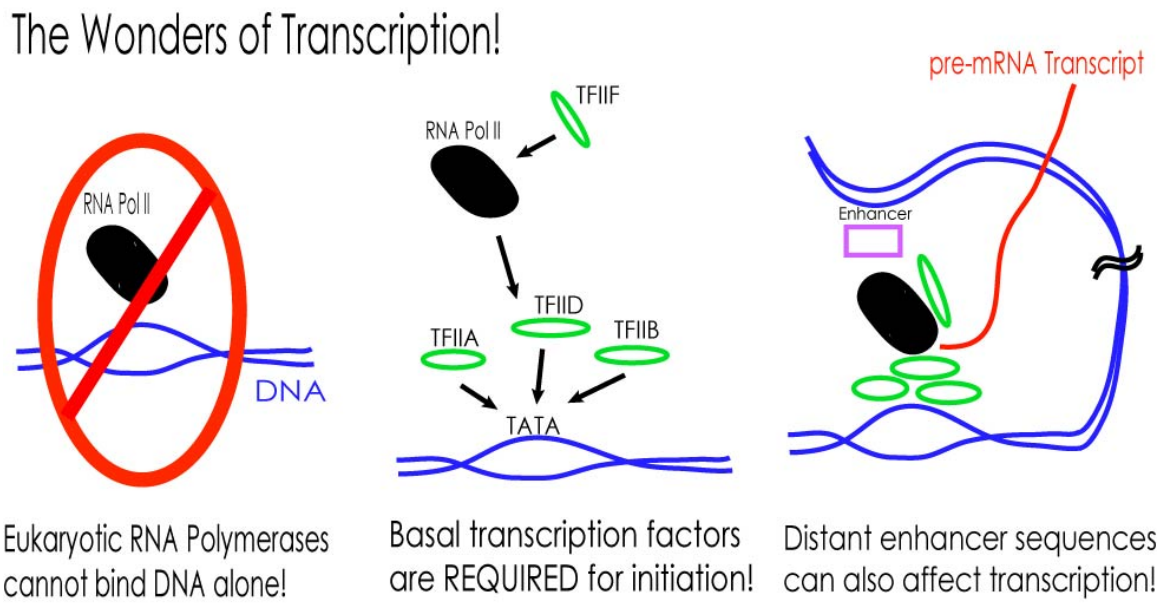


**Figure 2.5**

**Basal Transcription Factors - necessary for transcription**

The first step in transcription initiation is TFIIA binds to the TATA box in the promoter region. The TATA box is the most strongly conserved consensus sequence in the eukaryotic promoter; it has the consensus sequence TATAAAA and is usually located around position -30 (counting backwards from the transcriptional start site). After TFIID binds to the TATA box, TFIIA joins the initiation complex, followed by TFIIB. Separate from the initiation complex, TFIIF binds to RNA polymerase II. One subunit of TFIIF has DNA-unwinding properties that presumably help RNA polymerase II unwind the DNA during transcription. Once the TFIIF-RNA polymerase complex has been formed, the two proteins can then join the initiation complex. After the addition of several more proteins to the initiation complex, transcripton is finally ready to begin.

**Steps of Transcription Initiation - Basal Transcription Factors**

1. TFIID binds to the TATA box in the promoter region.
2. TFIIA joins initiation complex.
3. TFIIB joins initiation complex.
4. TFIIF binds RNA polymerase (separate from initiation complex).
5. RNA poly - THIIF complex binds initiation complex on DNA.
6. Transcription begins.

**Modulatory Transcription Factors - regulate transcription**

Regulatory transcription factors are non-intrinsic transcription factors that modulate the expression of a particular transcript. These are the transcription factors that regulate time/space differrential expression in the organism. These factors can be enhancers or silencers, meaning that they can both increase and decrease the expression of a given gene. Also, these transcription factors can act from several thousand base-pairs in either direction from the promoter. Here it is important to remember that the eukaryotic genome has significant secondary structure; DNA wrapping around nucleosomes and 30nm fiber looping can bring

distant sequence motifs proximal to promoter sites where they can have significant action. It is important to remember that an enhancer/silencer sequence motif may be present without having an effect, because the specific transcription factor protein responsible for binding to that domain may not have been expressed, ie. may not be present. The sequence on its own has no effect, the presence of the modulatory transcription factor is required. This can complicate sequence analysis w/r/t expression because in some cases we will only have half of the contextual information (ie we are missing the protein context of the cell).

### 2.2.5 Transcript Modifications

#### 2.2.5.1 5' Methyl Guanosine Cap

- Protects transcript from degredation.
- Helps with initiation of translation.

After the initial pre-mRNA transcript has been created the first major modification is the addition of a 5' methyl guanosine cap. The addition of the 5' 7-MG cap is important for two reasons: the 5' caps are recognized by protein factors that initiate translation, and it also helps protect the transcript from nucleases. Nucleases are very common in the cell and because of this unprotected RNA has a very short half-life inside the cell. Nucleases are actually so common that working with RNA in the laboratory can be quite difficult because the samples have a tendency to disintegrate into useless bits.

#### 2.2.5.2 Splicing

- Splicing removes non-coding exons from transcript.
- Alternative splicing allows for different combinations of exons from same pre-mRNA transcript (gene).
- Some RNAs can self-splice.

Eukaryotic genes contain two types of transcribed regions: introns and exons. Exons are the regions of the genome that contain actual coding information. Introns are non-coding, meaning that intronic sequences are never translated to protein. Introns are never included in the final processed mRNA transcript. Splicing is the process of removing introns from the pre-mRNA transcript to produce an exon-only mRNA molecule, which is then shipped off for translation. Generally, eukaryotic mRNAs are considered to monogenic. Monogenic means that an RNA transcript contains exons from only one gene. However, up to one fourth of the transcripts in C. elegans have been show to be multi-genic (i.e. they contain exons from multiple genes).

A further complication of the splicing process is that mRNA can undergo alternative splicing. To illustrate this let's imagine a gene that has 3 exons and two introns. From this gene, three different final transcripts are possible. In all transcripts the two introns are going to be removed, but the cell can combine the exons however it wants as long as the original order is maintained. This means that for this example the possible mRNA transcripts include: Exon1-Exon2, Exon1-Exon3, and Exon1-Exon2-Exon3; however, Exon3-Exon1 is not possible because the exons are out of order.

An interesting side note is that some introns are capable of self-splicing, that is they can politely remove themselves without the intervention of any proteins. This is significant mainly because it is a significant counter example to the idea that RNA is an inert transcript and action is soley the domain of proteins. RNAs should really be viewed as having both enzymatic properties and abstract information-carrying ability. Because of this many people believe that RNA was the original genetic molecule and that DNA and proteins evolved later in the game.

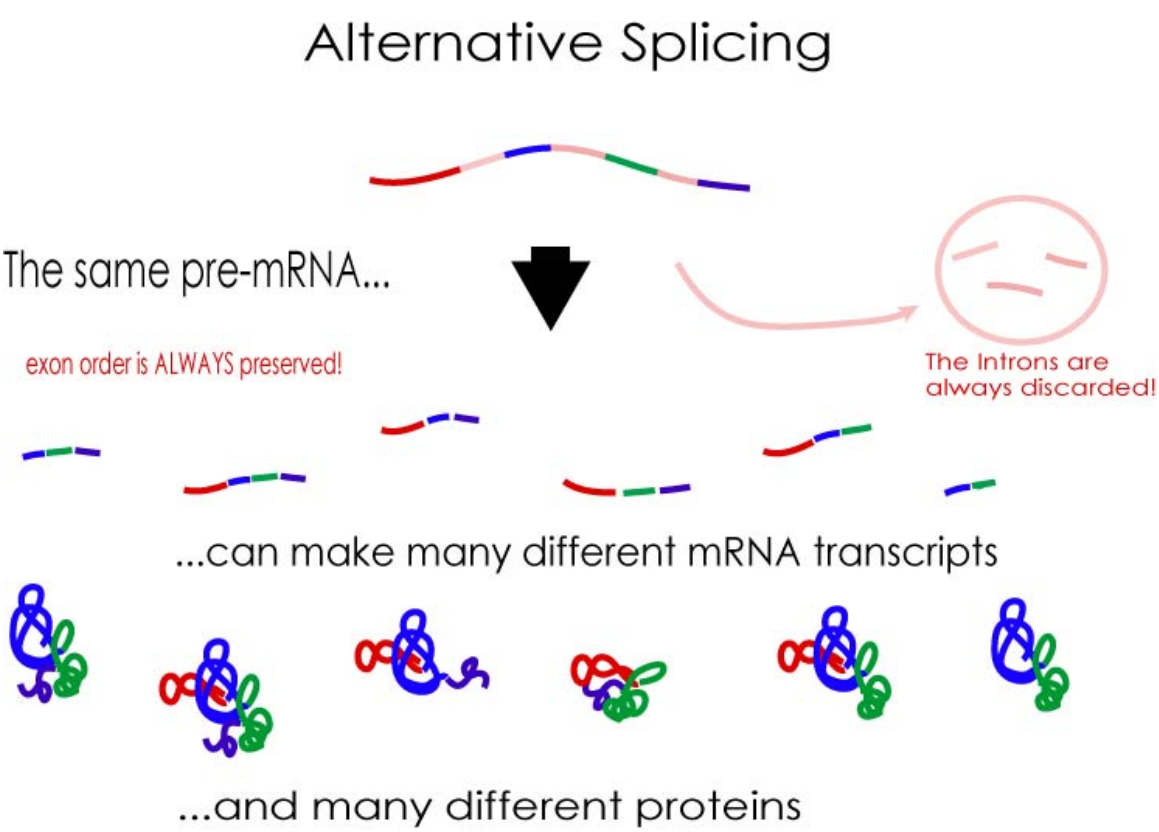


**Figure 2.6**

Alternative splicing is a very important and powerful tool. To understand the benefit alternative splicing gives the cell we need to understand something about proteins. Proteins can be understood as containing modularized functional units. These functional units can be active sites on enzymes, large structural motifs such as beta-sheets or alpha-helices, or motifs that direct the eventual destination of expressed proteins. A good example of an alternatively spliced pre-mRNA transcript is the mouse IgM immuoglobulin transcript. IgM exists in two forms: excreted and membrane bound. These two forms of the protein differ in the only in the C-terminus: the secreted protein has a secreted terminus motif while the membrane-bound protein has a C-terminal membrane anchor region. Both products come from the same pre-mRNA, but alternative splicing includes either the terminal exon that creates the excreted form of IgM or the membrane-bound form of IgM.

#### 2.2.5.3 3' poly-Adenylation

- Important for cellular transport.
- Helps stabilize the transcript

The poly(A) tails are formed in a two step process: an endonulcease cleaves around 1000-2000 non-coding bases from the 3' end of the pre-mRNA transcript and then poly(A) polymerase adds 20-200 AMP molecules to the 3' end of the transcript. The poly(A) tail is important in the cellular transport of the mRNA transcript and, like the 5' cap, also helps to stabilize the mRNA transcript.

## 2.3 Computational approaches to gene finding[3]

### 2.3.1 The human genome: chromosomes and genes

In 1953 Watson and Crick unlocked the structure of the DNA molecule[4] and set into motion the modern study of genetics. This advance allowed our study of life to transcend the wet realm of proteins, cells, organelles, ions, and lipids, and move up into more abstract methods of analysis. By discovering the basic structure of DNA we had received our first glance into the information-based realm locked inside the genetic code.

The human genome contains 3 billion chemical nucleotide bases (A,C, T and G). About 30,000 genes are estimated to be in the human genome. The human genome has physical three-dimensional structure. The genome is 6 feet (2 meters) in length and is packed in the nucleus of our cells into a structure which is only 0.0004 inches across (the head of a pin). The genome is divided among 24 chromosomes (22 pairs of autosomes and one pair of sex chromosomes (X and Y)), and that genes lie on specific chromosomes. Human chromosomes are arranged according to size with Chromosome 1 being the largest, and the Y chromosome being the smallest. Matt Ridley's fascinating book Genome[5] gives a great introduction to our chromosomes and the genes they contain. Chromosome 1 is believed to have 2968 genes, while the Y chromosome has 231 genes. To learn more about chromosomes, visit GeneMap99[6] , a site maintained by the NCBI. Here is a diagrammatic representation of the 24 chromosomes.

[3] This content is available online at <http://cnx.org/content/m15083/1.1/>.
[4] http://www.nature.com/nature/dna50/watsoncrick.pdf
[5] http://www.amazon.com/Genome-Matt-Ridley/dp/0060932902
[6] http://www.ncbi.nlm.nih.gov/projects/genome/genemap99/

**Human chromosomes**

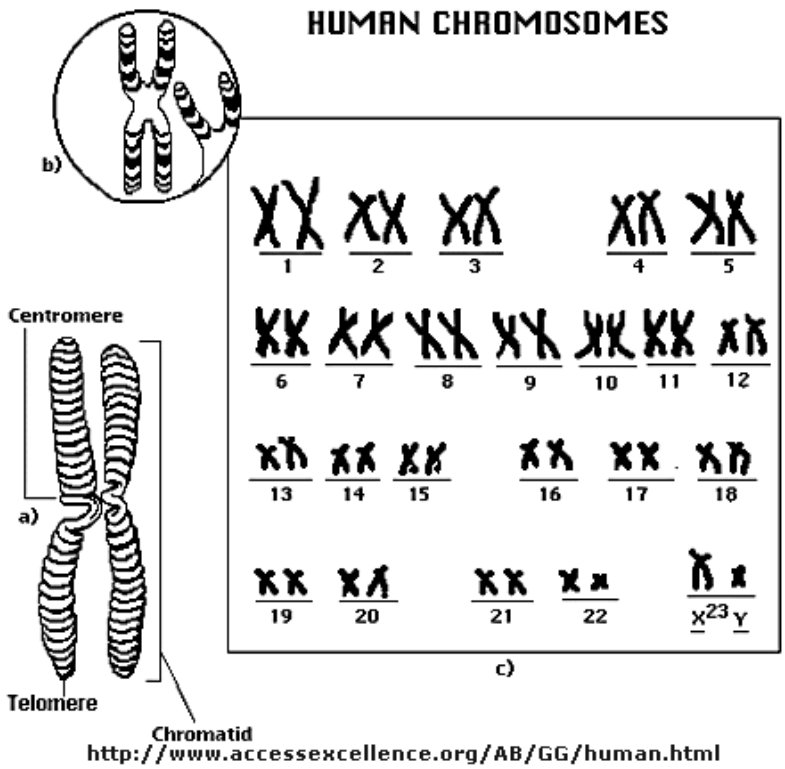


**Figure 2.7:** Representation of the 24 chromosomes of the human male.

Here is what a chromosome looks like under an electron microscope.

**Human chromosomes**

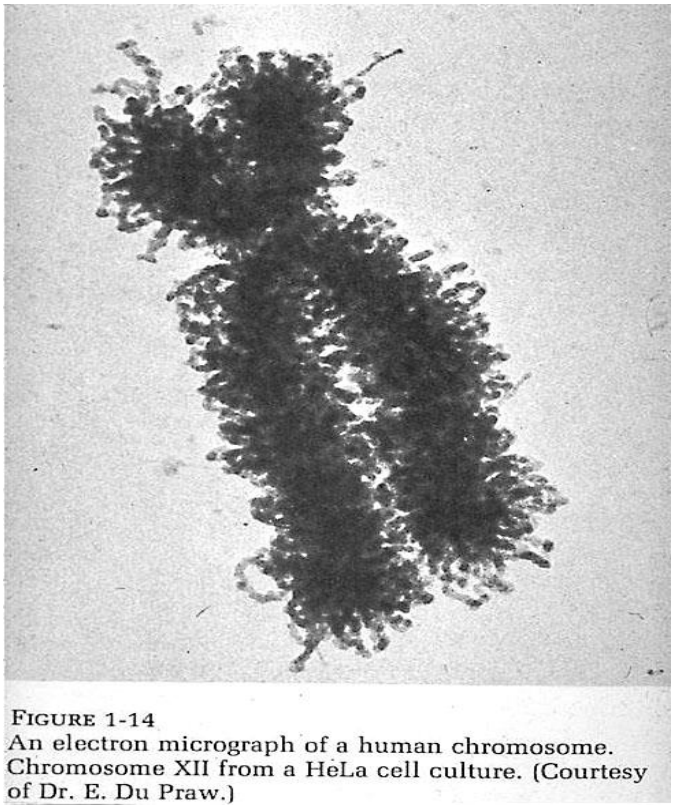

FIGURE 1-14
An electron micrograph of a human chromosome. Chromosome XII from a HeLa cell culture. (Courtesy of Dr. E. Du Praw.)

**Figure 2.8:** Human Chromosome 12 under an electron microsocope.

The average human gene contains about 3000 bases. Sizes of human genes vary greatly. The largest known human gene is dystrophin (a muscle protein) at 2.4 million bases. The smallest genes are a little over a hundred base pairs long. Less than 2% of the genome codes for proteins. Repeated sequences that are not involved in coding for proteins (sometimes called "junk DNA") make up at least 50% of the human genome. These repetitive sequences play an important role in chromosome structure and dynamics. Over time, these repeats are believed to reshape the genome by rearranging it, creating entirely new genes, and modifying and reshuffling genes. Surprisingly, genes are not distributed uniformly through the human genome. Genes appear to be concentrated in sections of the genome with high GC content, with vast areas of non-coding DNA in between. There are long stretches of C and G repeats adjacent to gene-rich areas. These CpG islands are believed to regulate gene activity, and they serve as markers for gene-rich locations on the genome. We do not yet know the function of over 50% of the discovered genes. A great site to learn more about DNA is the DNAi site [7] maintained by the HHMI.

### 2.3.2 Computational Genefinding

What is computational genefinding? Simply put, it is the development of computational procedures to locate protein coding regions in unprocessed genomic DNA sequence data. In reality however, pinpointing the mere location of a gene is part of a much larger challenge. The eukaryotic gene is a complicated and highly studied beast composed of a multitude of small coding regions and regulatory elements hidden amidst tens

[7] http://www.dnai.org

of thousands of base pairs of intronic and non-signal DNA. In order to accurately predict gene locations we must first understand how the different functional components interact to create the dynamic and complex phenomena we have come to understand as 'a gene'.

Thus genefinding is a little bit of a misnomer: in order to find genes we must first understand the content and structure of the signal the genes present to the cell's genetic machinery, and in doing this we must answer much broader questions than the seemingly facile question, "Where are the genes?" The goal of genefinding then is not simple gene prediction, but accurate modeling of the signal genes present to the cell. Furthermore, because such information does not exist in a vacuum separate from it's interpretation, implicit in the assumption of the ability to model the genetic signal is a furthering of our capacity to understand the deciphering of the genetic signal and our understanding of the inner workings of the cell itself.

### 2.3.3 Genefinding in prokaryotic genomes

There are two basic approaches to gene finding: ab-initio and comparative. Ab-initio methods use statistical properties of the given genome, while comparative methods use annotations from previously analyzed genomes as an additional input. We will begin our discussion of gene finding with ab-initio methods as applied to simpler prokaryotic genomes. Examples of such genomes include H. influenzae (the influenza virus). Over 70% of H. influenzae codes for proteins. Genes in prokaryota are contiguous stretches of base pairs with no intronic breaks. There are untranslated regions (UTRs) that flank both ends of a gene: the 5' (5-prime) and 3' (3-prime end). Genes are directional – they are read from the 5' to the 3' end. There are genes on both strands of the DNA double helix. Each gene starts with the amino acid methionine, specified by the three letter codon ATG. ATG is called the start codon. The end of a gene is signalled by one of three stop codons (TAA, TAG, TGA). The start codon signals the ribosomal machinery to start translating the bases in composites of three into amino acids until the stop codon is reached. Gene finding in prokaryota reduces to the problem of finding stretches of the genome with a start codon and a stop codons with no intervening stop codons. Such a stretch is called an open reading frame or ORF.

Given a sequence from {A,C,G,T}*, an open reading frame (ORF) is any subsequence that starts with the codon ATG and ends with a stop codon (TAA, TAG, TGA) with no stop codons in between. ORF finding algorithms are based on the following simple idea. Since coding regions are terminated by stop codons, one needs to to look for long stretches of bases without a stop codon. Once a stop codon is found, we work backward to find the start codon corresponding to the gene. Why do we look for long stretches without stop codons? If nucleotide bases were drawn uniformly at random, then a stop codon is expected once every 64/3 (about 21)codons, or about 63 base pairs. By selecting an appropriate length threshold t (typically greater than 210 bases or 70 amino acids), we reduce the likelihood of picking a random sequence with a stop codon rather than an actual coding region. Modifications to this basic algorithm to handle very short genes and overlapping genes have been developed. The most successful method for finding coding regions in prokaryotic genomes is one based on interpolated Markov models emboded in the program GLIMMER. It is available here[8] . A 2007 Bioinformatics paper details how to use this tool.

Here is a 1151 base pair segment of the Influenza B virus taken from the Entrez database. This segment has two genes: base pair 4 to 750, and 750 to 1079. The start and stop codons for the two genes are marked with capital letters below.

```
1    aaaATGtcgc tgtttggaga cacaattgcc tacctgcttt cattgacaga agatggagaa
61   ggcaaagcag aactagcaga aaaattacac tgttggttcg gtgggaaaga atttgaccta
121  gactctgcct tggaatggat aaaaaacaaa agatgcttaa ctgatataca aaaagcacta
181  attggtgcct ctatctgctt tttaaaaccc aaagaccagg aaaggaaaag aagattcatc
241  acagagcctc tatcaggaat gggaacaaca gcaacaaaaa agaaaggcct gattctagct
301  gagagaaaaa tgagaagatg tgtgagcttt catgaagcat ttgaaatagc agaaggccat
361  gaaagctcag cgctactata ttgtctcatg gtcatgtacc tgaatcctgg aaattattca
421  atgcaagtaa aactaggaac gctctgtgct ttgtgcgaga aacaagcatc acattcacac
```

[8] http://cbcb.umd.edu/software/glimmer

```
481  agggctcata gcagagcagc gagatcttca gtgcccggag tgagacgaga aatgcagatg
541  gtctcagcta tgaacacagc aaaaacaatg aatggaatgg gaaaaggaga agacgtccaa
601  aagctggcag aagagctgca aagcaacatt ggagtattga gatctcttgg agcaagtcaa
661  aagaatgggg aaggaattgc aaaggatgta atggaagtgc taaagcagag ctctatggga
721  aattcagctc ttgtgaagaa atatctaTAA TGctcgaacc atttcagatt ctttcaattt
781  gttcttttat cttatcagct ctccatttca tggcttggac aatagggcat ttgaatcaaa
841  taaaaagagg agtaaacatg aaaatacgaa taaaaggtcc aaacaaagag acaataaaca
901  gagaggtatc aattttgaga cacagttacc aaaaagaaat ccaggccaaa gaaacaatga
961  aggaagtact ctctgacaac atggaggtat tgagtgacca catagtgatt gaggggcttt
1021 ctgccgaaga gataataaaa atgggtgaaa cagttttgga gatagaagaa ttgcatTAAa
1081 ttcaattttt tactgtattt cttattatgc atttaagcaa attgtaatca atgtcagcaa
1141 ataaactgga a
```

To get a feel for the process of finding ORFs, go to Artemis[9] at the Sanger Institute. If you have Java Web start, you can launch Artemis by clicking here[10] . Load the clbot.fsa[11] fasta file onto your local machine. Follow the excellent tutorial here [12] to find ORFs in a simple prokaryotic genome.

### 2.3.4 Genefinding in eukaryotic genomes

We now turn to ab initio approaches to gene finding in eukaryotic genomes. They rely on **signals** which are specific DNA sequences that indicate the presence of a nearby gene, or **content** which are statistical properties of the gene itself. Examples of signals include promotor and regulatory sequences that precede a gene, binding sites for the polyA tail at the end of a gene, as well as CpG islands (stretches of DNA with high GC content) that occur before the start of a gene.

[9] http://www.sanger.ac.uk/Software/Artemis
[10] http://www.sanger.ac.uk/Software/Artemis/v9/v9_5/Artemis.jnlp
[11] http://www.nematodes.org/teaching/genomics/Tech4_Gene_finding/clbot.fsa
[12] http://www.nematodes.org/teaching/genomics/Tech4_Gene_finding/techsession_genefin.html

**The structure of an eukaryotic gene**

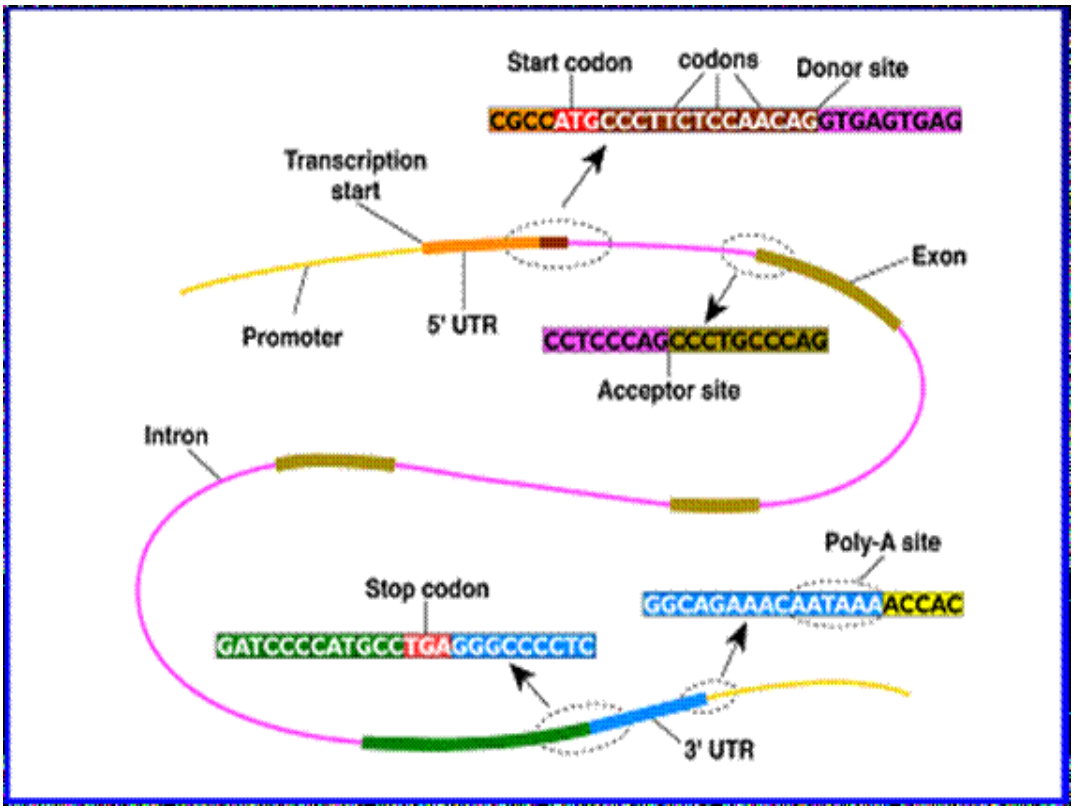


**Figure 2.9:** The structure of an eukaryotic gene (source: unknown).

Eukaryotic genes are considerably more complicated than their prokaryotic counterparts. First, a gene is no longer a contiguous stretch of bases between the start codon and a stop codon. It is broken or spliced into coding regions called exons with intervening non-coding sections called introns. Splicing mechanisms in the eukaryotic cell stitch the exons together before translation. Alternative splicing mechanisms allow the exons to be put together in a variety of ways — thus a single gene can code for a variety of proteins. The one gene - one protein mapping that is characteristic of prokaryotes is lost. Second, most DNA in eukaryotes is non-coding; only about 3% codes for proteins. Finding the exons in a sea of introns and intergenic material is very difficult. In addition, many of the regulatory signals may be quite far from the start codon. These factors make eukaryotic gene finders much less successful than their prokaryotic counterparts like GLIMMER with prediction accuracies of 98%.

**The exon-intron structure of the Human PSA gene**

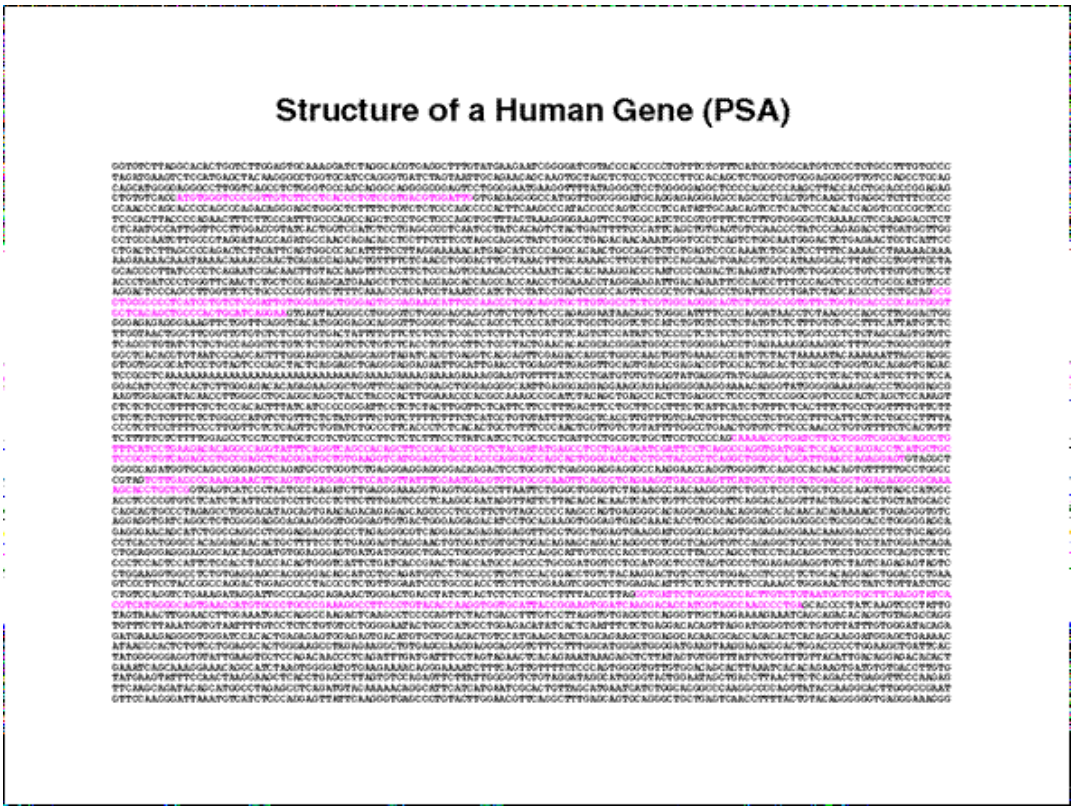


**Figure 2.10:** The exon-intron structure of the Human PSA (source: unknown). The magenta sections are exons embedded in a sea of black introns.

Ab initio methods use information embedded in the genomic sequence exclusively to predict gene structure in eukaryotic genes. The problem of gene finding is usually posed as a probabilistic inference problem: find the structure G representing gene boundaries and internal gene structure which maximizes the probability of G given the genomic sequence. Hidden Markov models are the predominant generative method for the problem. Ab-initio methods allow for the prediction of novel genes, genes that are unlike any that are known. However, ab initio techniques are generally not effective in detecting alternately spliced forms, interleaved or overlapping genes. They also have difficulty in accurate identification of exon/intron boundaries. Almost all ab-initio gene finders generate large numbers of false positive predictions arising from learnign overfitted models on small training sets. With these caveats in mind, we embark on the study of Hidden Markov models for finding genes in complex eukaryotic genes.

### 2.3.5 A simple example: finding CpG islands

Normally a C (cytosine) followed immediately by a G (guanine) (a CpG) is rare in eukaryotic DNA because the Cs in such an arrangement tend to be methylated [Wikipedia][13] . This methylation helps distinguish the newly synthesized DNA strand from the parent strand, which aids in the final stages of DNA proofreading after duplication. However, over evolutionary time methylated Cs tend to turn into Ts because of spontaneous deamination. The result is that CpGs are relatively rare unless there is selective pressure to preserve them.

[13] http://en.wikipedia.org/wiki/CpG_island

In bulk human DNA CpG dinucleotides occur about five times less frequently than expected (Bird 1980, Jones et al 1992). CpG islands are thus unmethylated regions of the genome that are associated with the 5' ends of most house-keeping genes and many regulated genes. The absence of methylation slows CpG decay, and so CpG islands can be detected in DNA sequence as regions in which CpG pairs occur at close to the expected frequency. The fact that CpG islands can be detected in this way indicates that the corresponding germline DNA has been substantially hypomethylated for an extended period of time, and in fact about 80% of CpG islands are common to man and mouse (Antequera and Bird 1993). About 56% of human genes and 47% of mouse genes are associated with CpG islands (Antequera and Bird 1993). Often CpG islands overlap the promoter and extend about 1000 base pairs downstream into the transcription unit. Identification of potential CpG islands during sequence analysis helps to define the extreme 5' ends of genes, something that is notoriously difficult with cDNA based approaches.

**CpG Islands**

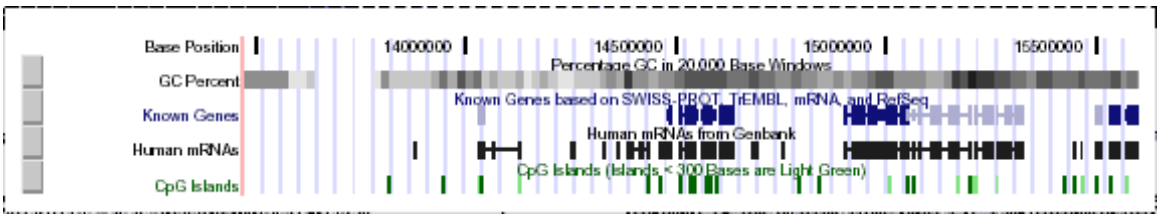


**Figure 2.11:** CpG islands in the human genome (Chromosome 22, Entrez browser)

We follow the presentation of the excellent text[14] by Eddy and Durbin, and pose two problems in this context.

- Given a short DNA sequence, does it come from a CpG island or not?
- Given a long DNA sequence, identify the CpG islands in that sequence.

How do we model the problem of recognizing CpG islands? If we look at examples of CpG islands in the human genome (say on Chromosome 22 available from Genbank), you will see that we are unlikely to come with a deterministic set of rules for classifying sequences as being parts of CpG islands or not. We are going to build a probabilistic recognizer; one that takes a sequence and returns a probability that it is part of a CpG island. We will delve into the theory of Markov chains to set up such a model. But before that, here is a brief interlude.

Steve Skiena of SUNY Stony Brook has a very interesting viewpoint on the cultural differences between computer scienctists and biologists. It is a bit of a caricature, but there is a lot of truth in it. Here is his list of contrasts.

- Almost nothing is ever completely true or false in biology. Everything is either true or false in computer science.
- Biologists strive to understand the complicated messy natural world. Computer scientists seek to build their own clean and organized virtual worlds.
- Biologists are data driven. Computer scientists are more algorithm driven. Once consequence is that CS web pages have fancier graphics, while biology web pages have more content.
- Biologists are obsessed with being the first to discover something. Computer scientists are obsessed with being the first to invent or prove something.
- Biologists are comfortable with the idea that all data has errors. Computer scientists are not.
- Computer scientists get high-paid jobs after graduation. Biologists have to complete one or more post-docs before getting a permanent job.

[14] http://www.amazon.com/Biological-Sequence-Analysis-Probabilistic-Proteins/dp/0521629713

## 2.4 Modeling the genefinding problem[15]

Our task is to find coding regions in eukaryotic genomes. We have already studied the complexity of the structure of eukaryotic genes. if you need a refresher, check out this self-paced tutorial[16] from the University of Glasgow.

### 2.4.1 Early approaches to genefinding

One fundamental approach to finding genes is to detect functional sites in genomic DNA. Fixed length sites like splice sites, start and stop codons, polyA sites, ribosomal binding sites, and transcription factor binding sites are called signals, and algorithms that detect them are called **signal sensors**. Variable length regions like exons and introns in eukaryotic DNA are recognized by another family of methods called **content sensors**.

---

**A consensus sequence**

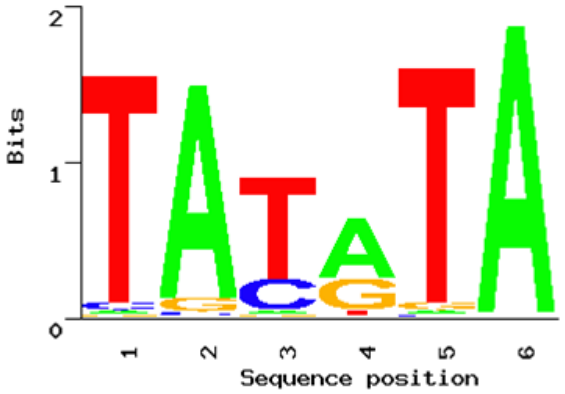


**Figure 2.12**

---

Show above is a sequence motif of size 6. The letters in each position are drawn in proportion to the probability of having that letter in that position. This probability information is summarized in a weight matrix shown below. Weight matrices are the simplest form of signal sensors.

**Weight matrix**

| Position | A | C | G | T |
|---|---|---|---|---|
| 1 | 0.028 | 0.034 | 0.026 | 0.912 |
| 2 | 0.805 | 0.031 | 0.123 | 0.041 |
| 3 | 0.046 | 0.158 | 0.022 | 0.774 |
| 4 | 0.669 | 0.019 | 0.253 | 0.059 |
| 5 | 0.024 | 0.044 | 0.028 | 0.904 |
| 6 | 0.962 | 0.012 | 0.014 | 0.012 |

[15] This content is available online at <http://cnx.org/content/m15084/1.2/>.
[16] http://www.gla.ac.uk/faculties/vet/teaching/CAL/biomolecular/module1.htm

**Table 2.1**

The weight matrix is an example of a probabilistic sequence model. We treat each sequence as a string in the alphabet {A,C,G,T}. Each entry (i,j) in the matrix represents the probability of a base i in position j of the string. Given a DNA sequence or string s of length 6, we can evaluate its likelihood with respect to this model. That is, we can calculate the probability that the sequence is generated by the weight matrix above.

P(TATATA) = 0.912 * 0.805 * 0.774 * 0.669 * 0.904 * 0.962 = 0.33

P(ATATAT) = 0.028 * 0.041 * 0.046 * 0.059 * 0.024 * 0.012 = 8.9 * 10^(-10)

We can see that with respect to the sequence model described by the weight matrix, the string TATATA is overwhelmingly more likely than the string ATATAT. The weight matrix model assumes that the bases at each position are independent of each other. We will study more sophisticated models, called Markov models, which take dependencies between bases at different positions into account. The advantage of weight matrix models is their simplicity which allows them to be estimated with very little data. The disadvantage is that the models are quite rigid and do not accommodate the kind of variability seen in real biological sequences.

Content sensors include detectors of CpG islands, an example we will consider in detail later in this module. Exon and intron detectors are some of the most widely studied in the literature. The GRAIL [17] system detects exons, polyAs,and CpG islands. You can submit a DNA sequence on the form linked above to check a content sensor program out.

Signal and content sensors alone cannot solve the genefinding problem. The statistical signals they are trying to recognize are too weak, and there are dependencies between signals and content that they cannot capture. Since the late nineties, attempts have been made to develop probabilistic systems that combine signal and content sensors to try to identify complete gene structure. One of the best known of these systems is Genscan, developed by Chris Burge and his advisor Samuel Karlin at Stanford University in 1997. Genscan is based on hidden Markov models.

### 2.4.2 An example: finding CpG islands

This example is taken from the excellent textbook Biological Sequence Analysis: probabilistic models of proteins and nucleic acids[18] by Durbin, Eddy, Krogh and Mitchison. CpG islands are regions of the genome with a higher than normal percentage of C and G bases adjacent to each other. The usual percentage of adjacent CG bases in the genome is about 1%, but in CpG islands that percentage is over 6%. The reason that C followed by G is relatively rare in The "p" in "CpG" refers to the phosphodiester bond between the cytosine and the guanine, and serves to distinguish it from the C and G pairing on the double stranded DNA helix. CpG islands are bioogically intersting because they are in or near 40% of the promoters in mammalian genes and 70% in human promoter genes. CpG islands vary in length between 300 and 3000 basepairs. Thus fixed-length consensus sequence based approaches do not work well for detecting them. Effective identification of of CpG islands can aid in localizing genes in eukaryotes. CpG island detection also serves as an excellent problem to illustrate the power of Markov models.

We will consider two problems.

- Given a short DNA sequence, does it come from a CpG island or not?
- Given a long DNA sequence, find all the CpG islands on it, if any.

### 2.4.3 Generative models of biological sequences

We will construct generative models of CpG islands. A generative model produces strings, and the model parameters are tuned to reflect the characteristics of CpG islands.

[17] http://compbio.ornl.gov/Grail-1.3/

[18] http://www.amazon.com/Biological-Sequence-Analysis-Probabilistic-Proteins/dp/0521629713

**Generative models for CpG island detection**

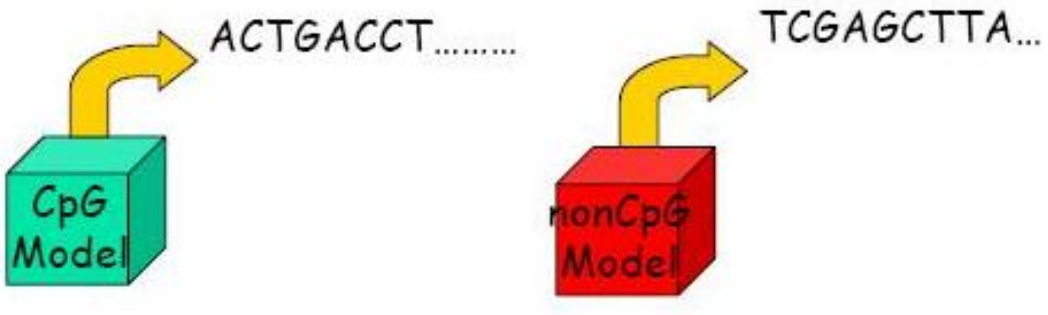


**Figure 2.13**

The simplest probabilistic generative DNA sequence model associates a probability with the occurrence of each base: P(A), P(C), P(G) and P(T) such that these probabilities all sum to 1. For H. influenzae, these probabilities are P(A) = 0.3, P(C) = 0.2, P(G) = 0.2, and P(T) = 0.3. To generate a sequence based on this model, we first choose the length L of the sequence that we wish to construct. Then we draw bases for each position based on the discrete distribution above, as shown in the code fragement below.

```
i = 1;
while i less-than-or-equal-to L do
 S[i] = a base drawn from the discrete probability distribution [0.3,0.2,0.2,0.3] (for A,C,G,T)
 i = i+1
end
```

This model does not capture interdependencies between bases. It assumes that the choice of base in each position of the generated sequence is independent of the bases surrounding it. A more complex model of DNA sequences can be constructed using the theory of Markov chains. In Markov chains, the probability of observing a base at a given position in a sequence is conditioned on the bases preceding it. Thus, Markov chains can model local correlations among the nucleotides. A Markov chain of order 1 assumes that the probability of a base at position i is dependent only on the base at position i - 1. A first order Markov chain can be specified by a probability matrix as shown below.

**A first order Markov model for generating DNA sequences**

| | **A** | **C** | **G** | **T** |
|---|---|---|---|---|
| **A** | 0.6 | 0.2 | 0.1 | 0.1 |
| **C** | 0.1 | 0.1 | 0.8 | 0.0 |
| **G** | 0.2 | 0.2 | 0.3 | 0.3 |
| **T** | 0.1 | 0.8 | 0.0 | 0.1 |

**Table 2.2**

Note that every row of the first order Markov model above sums to 1. Since the probability of T following G (row 3, column 4) is zero, no sequence generated by this model will contain the subsequence "GT". We

can use the model to generate a sequence of length L, using a small variation of the code snippet shown above.

```
i = 1; S[i] = a base chosen uniformly randomly from {A,C,T,G}.
for (i = 1; i less-than L; i++) do
   S[i+1] = a base chosen from a discrete distribution from row corresponding to base S[i] in Markov mo
end
```

We can use the model for evaluating the likelihood that a given sequence is generated from it. It represents the probability of that sequence given the model.

P(s1...sn) = P(s1) * P(s2|s1) * P(s3|s1) * ... * P(s{n-1}|sn)

We can factor the probability of the whole sequence into the probabilities of observing each transition starting from the first base. We can simplify the probabiltiy computation because of the first order Markov condition – the probability of a base at a given position depends only the base before it in the sequence. Thus, the probability of observing sequence ACGT based on the first order Markov model shown above is:

P(ACGT) = P(A) * P(C|A) * P(G|C) * P(T|G) =0.25 * 0.2 * 0.8 * 0.3

We can use the model to compare the likelihood of two sequences. Thus, given sequence "ACGT" and "TGAC", we can calculate their individual likelihoods as shown above. P(TGAC) = P(T) * P(G|T) * P(A|G) * P(C|A) = 0.25 * 0 * 0.2 * 0.2 = 0. The sequence "TGAC" can never be generated by this model!

If we have a first order Markov model of CpG islands, we can now calculate the probability of a DNA sequence with respect to that model. If that probability exceeds a given threshold (say, 0.8), then we will assert that the given DNA sequence is in fact from a CpG island. How can we acquire a first order Markov model of CpG islands? There are databases of CpG islands available from the NCBI. The CpG islands for human chromosomes can be obtained from here[19] .

Given a set of CpG island sequences, we can calculate the probability P(a|b) in the model, for a, b in {A,C,G,T} by counting the percentage of times the subsequence "ba" occurs in those sequences. We can then estimate all sixteen entries in the first order Markov model over the nucleotides. These models are extremely easy to acquire, requiring just counting operations. A first order Markov model learned from CpG islands, and another from a data set of non-CpG islands are shown below.

**A first order Markov model for CpG islands**

| | **A** | **C** | **G** | **T** |
|---|---|---|---|---|
| **A** | 0.18 | 0.27 | 0.43 | 0.12 |
| **C** | 0.17 | 0.37 | 0.28 | 0.19 |
| **G** | 0.16 | 0.34 | 0.38 | 0.13 |
| **T** | 0.08 | 0.36 | 0.38 | 0.18 |

**Table 2.3**

**A first order Markov model for non-CpG islands**

| | **A** | **C** | **G** | **T** |
|---|---|---|---|---|
| **A** | 0.30 | 0.21 | 0.29 | 0.21 |
| **C** | 0.32 | 0.30 | 0.08 | 0.30 |
| **G** | 0.25 | 0.25 | 0.30 | 0.20 |
| **T** | 0.18 | 0.24 | 0.30 | 0.30 |

[19] http://www.ncbi.nlm.nih.gov/projects/genome/guide/human/

**Table 2.4**

You can see that P(G|C) = 0.28 in the CpG island model, while P(G|C)=0.08 in the non-CpG islands model.

### 2.4.4 Using generative models to classify sequences

Given first order models of CpG and non-CpG islands and a DNA sequence s, we can determine whether the sequence comes from a CpG island or not by computing its log-odds ratio with respect to the models. If P(s|CpG) > P(s|non-CpG), then s is classified as a CpG island. The decision rule can be alternately cast as P(s|CpG)/P(s|non-CpG) > 1, or taking logarithms on both sides, we get log[P(s|CpG)/P(s|non-CpG)] > 0. The logarithm of the ratios of the two probabilities is called the **log-odds ratio**. If the log-odds ratio is greater than 0, then s is part of a CpG island.

**Histogram of log-odds ratios**

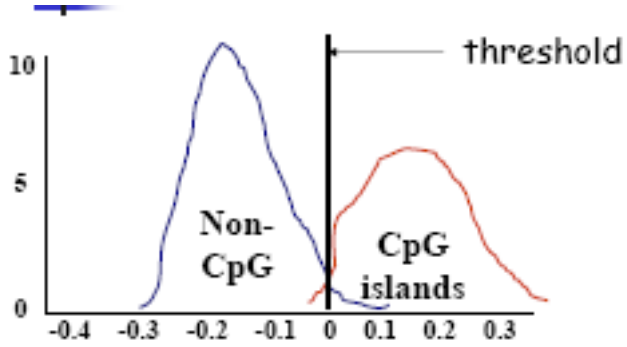


**Figure 2.14**

## Index of Keywords and Terms

**Keywords** are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

# Attributions

**Statistical machine learning for computational biology**
The course is the second module of a three module course entitled "Bioinformatics: from sequence to structure". This course focuses on learning statistical models from biological data. Three problems are covered: gene finding, classification of gene expression data, and inferring regulatory networks from mRNA and proteomic data. The computational techniques covered include: HMMs, support vector machines, and structure learning with Bayesian networks. This course is made possible by a curriculum development grant from the NSF.

**About Connexions**
Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.