Chapter 23

Bioinformatics

Numerous large-scale data collection projects are underway in molecular biology and genetics, involving efforts to catalog the structure and function of biologically-important molecules and cellular processes, as well as the variation of biological phenomena within and across populations. Graphical models are an important part of this effort, serving as standard computational tools in data analysis pipelines, and playing important roles in attempts to integrate multiple sources of data into a coherent understanding of biological phenomena. We provide an overview of graphical models for bioinformatics in this chapter, with particular focus on models that incorporate aspects of both the molecular and genetic levels of analysis.

Many phenomena in genetics and population biology are naturally expressed in terms of graphs. Historically the development of probabilistic models of these phenomena led to specific instances of probabilistic graphical model structures and algorithms, anticipating the development of the general framework. Indeed, variants of the elimination algorithm were developed in the 1970s and 1980s by researchers working on phylogenies (graphs that express relationships among species) and pedigrees (graphs that express relationships among individuals). We will discuss these models in the current chapter, profiting from the general framework that is at our disposal to cover the basic ideas relatively quickly.

The basic entities in molecular biology of interest are sequences of various kinds, and hidden Markov models of sequential data have been embraced by many researchers as a basic data analysis tool in molecular biology. We will discuss applications of HMMs to gene finding, the modeling of gene families, and the modeling of various aspects of the structure of proteins.

The past few decades have seen an ongoing merger of molecular biology and genetics, a trend that has accelerated in the past few years with the availability of genome-scale data. This trend is reflected in the kinds of models that are studied. Given that graphical models have been used to describe phenomena at both the molecular and genetic levels, and given the natural ability of the graphical model formalism to piece together component models into larger models, it should come as no surprise that graphical models are prominent in attempts to integrate data from molecular, cellular and genetic levels of analysis.

Thus we have two principal goals in this chapter. The first is to present some specific graphical models that have played an important role in data analysis in bioinformatics. This will help to provide additional support for understanding the graphical model formalism, by grounding basic models in a concrete (and fascinating) domain, and will provide a summary of current literature. The second, and more open-ended, goal is to convey the role of graphical models as a toolbox for developing unified approaches to the study of biological phenomena.

23.1 Basic concepts in molecular biology

In this section we provide enough background on molecular biology so as to make the chapter self-contained. We gather some basic concepts here, and continue to introduce biological ideas in the context of specific models in later sections.

We cannot of course hope to do justice to molecular biology in a few pages, and it is perhaps worth being explicit on some of the kinds of injustice that we will do to the subject. First, it is the nature of biology that there are few general rules that are without exception—there will always be some organism, some cell or some process that belies any "general rule"—this is part of the fascination of the subject. Second, our limited goal is to provide enough background so as to explain models of biological phenomena, and these models themselves generally have limited scope—they require careful specification of context, and they are rarely accurate in detail. Third, we will further simplify the models so as to convey the main ideas. Thus, every statement that we make should be preceded with a silent caveat along the lines of "In many organisms, humans in particular, it is often the case that... ." In particular, one major caveat is that we focus on concepts appropriate to higher organisms (eukaryotes) such as humans. While some of these concepts are also appropriate to lower organisms (prokaryotes), many are not.

Much of molecular biology revolves around the relationships among three principal kinds of molecules—DNA, RNA and proteins. DNA is a long polymer (i.e., a chain of elementary molecules) in which the elementary molecules are the *nucleotides* A, C, G and T. The specific sequence of nucleotides—also known as *bases*—in DNA provides a blueprint for synthesizing the other complex molecules in the cell (i.e., RNA and proteins), and the compilation of this sequence for the DNA in various model organisms has been the goal of the major "genome projects." RNA is also a polymer based on an alphabet of four letters; in particular, A, C, G and U. RNA plays several roles in cells, with its principal role being that of "messenger RNA" (mRNA)—an intermediate between DNA and proteins (see below). Finally, proteins are polymers based on a larger alphabet; namely, an alphabet of 20 amino acids. Proteins have a wide variety of functions—they act as structural elements, as enzymes for catalyzing reactions, as receptors and channels that allow various signals and molecules to pass through membranes in the cell, and as transcription factors that trigger the transcription ("read-out") of the information coded in DNA. This functional variety arises in large part from the fact that proteins fold into a variety of shapes depending on the particular sequence of amino acids that they are composed of, creating particular surface geometries and exposing particular amino acids on these surfaces. RNA also folds into various patterns, and these patterns have a functional role, but the functional repertoire of RNA is small compared to that of proteins. As for DNA, to a first order of approximation, it is only the linear sequence that matters.

All cells contain the same DNA,[1] but all cells do not contain the same pools of mRNA or

[1] Of course this is false; gametes—eggs and sperm—contain half the DNA of other cells. Moreover, some cells may

protein. Indeed, it is exactly the differing pools of protein that are contained ("expressed") in different cells that makes cells differ—this is what makes cells specialize into skin cells, liver cells or blood cells. As we will see below, some proteins play a role in triggering which parts of the DNA are transcribed ("read-out"), and thus which new proteins are made. Thus once a cell has an individualized pool of protein, it can continue to diversify and maintain its specialization. What makes a cell begin to diversify? Roughly speaking, diversification is due to "external signals" that arise from various geometrical, physical, and chemical inhomogeneities in the environment of the cell.

In eukaryotes (e.g., yeast, flies, mice and humans), DNA is contained inside bodies known as *nuclei*, which are separated from the rest of the cell—the *cytosol*—by a membrane. Proteins are manufactured in the cytosol, and it is the job of mRNA to shuttle the coded information from the DNA across the nuclear membrane to the site in the cytosol where it is used in making a protein. Our first goal is to describe this process of making a protein molecule from the information stored in the DNA. To do so, we need to introduce the concept of a *gene*. Our first definition of a gene will be molecular, and we will relate it to the (perhaps more familiar) Mendelian notion of a gene later.

For present purposes it suffices to view a DNA molecule as a long sequence of nucleotides. What we are ignoring in taking this view is the fact that a strand of DNA is associated with a second strand to which it is *complementary*—the nucleotide A in the first strand is replaced by a T in the second strand (and T is replaced by A), and C in the first strand is replaced by a G in the second strand (and G is replaced by C). These two strands wind around each other, forming a double helix (with chemical bonds between the A-T pairs and the C-G pairs stabilizing the structure). A single strand contains all of the information content of the molecule, however—given the complementation rule—and this both suggests how DNA can be replicated (separate the two strands and use the complementation rule to make two new strands), and justifies our focus on a single strand.

Each protein is associated with a *gene*—a region in the DNA that encodes the protein. There are on the order of tens of thousands of genes in higher organisms. To make a given protein, the cell needs to "read out" the information encoded in the corresponding gene. The details are somewhat complex, and the reader may wish to begin to consult Figure 23.1, which summarizes some of the basic structures that are involved (and which will be discussed in the course of our presentation). Glossing over the details for now, however, the basic procedure is as follows. Certain proteins (*transcription factors*) diffuse to the physical location of the gene and bind to a region in the gene known as the *promoter region*. This triggers a process in which another protein (known as *RNA polymerase*) walks along the rest of gene, creating an mRNA molecule which is complementary to the sequence in the DNA (and where U is used in place of T). This process is known as *transcription*. The result is an mRNA molecule that is in essence a copy of the sequence of symbols in the DNA. This mRNA molecule then diffuses through the nuclear membrane and docks onto a molecule in the cytosol known as a *ribosome*. It is the responsibility of the ribosome to convert the mRNA sequence into a sequence of amino acids—a protein. This latter process is known as *translation*. The protein then folds into its distinct shape and goes about its business.

The key problem addressed by the translation process is to map from the alphabet of mRNA

---

accrue individual mutations in their DNA during the lifetime of the organism. Recall our "silent caveat."

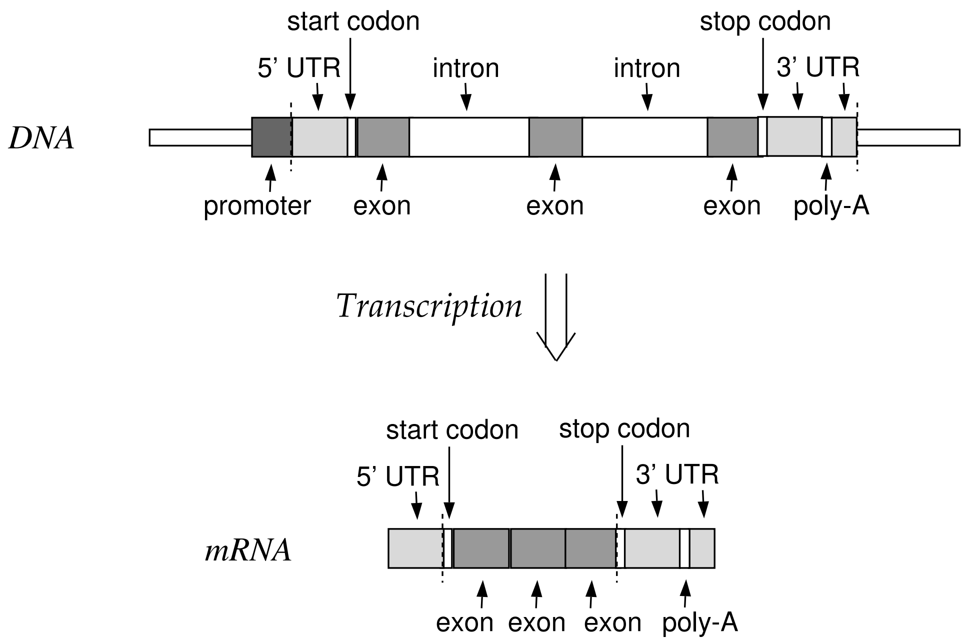

Figure 23.1: The structure of a eukaryotic gene and a resulting post-transcriptional messenger RNA.

(of size 4) to the alphabet of proteins (of size 20). The way that this is achieved is relatively straightforward. A sequence of three nucleotides is bundled up into a unit known as a *codon*. There are $4^3 = 64$ possible codons. These 64 codons are mapped onto the 20 amino acids in a many-to-one manner (i.e., some amino acids are coded for by several different codons). This mapping from codons to amino acids is known as the "genetic code," and is the same for (almost all) organisms.

Returning to the DNA, we now know that the protein-encoding regions of the DNA are organized into triplets of nucleotides. One can expect that such triplets would have different statistical structure than a region of DNA that does not code for protein, and indeed we will see later in the chapter that gene finding methods are based in part on this fact.

It is also important to note that DNA has a natural "left-to-right" orientation. In particular, each individual nucleotide is asymmetric, with one side referred to as the $5'$ side and the other side referred to as the $3'$ side. A $5'$ side links to a $3'$ side, and thus a chain of nucleotides retains the asymmetry. We generally draw a DNA molecule with the $5'$ end on the left and the $3'$ end on the right, and refer to the $5'$ side as "upstream" and the $3'$ side as "downstream." Transcription proceeds from $5'$ to $3'$—left to right on the page. Moreover, the mRNA molecule retains the $5'$–$3'$ asymmetry, and translation proceeds from $5'$ to $3'$ as well.

The picture that we have presented thus far is over-simplified, and we now consider a fuller picture of the structure of a eukaryotic gene. The first additional complexity to be aware of is that the sequence of protein-encoding nucleotides is not in general a contiguous sequence, but is broken into segments known as *exons*. Intervening between the exons are segments known as *introns*. The transcription process is not merely a simple "read-out" of the DNA sequence; rather, it also involves a *splicing* process which retains the exons and removes the introns. This happens in two phases—the RNA polymerase first walks along the DNA and transcribes the entire sequence, including the introns. Then the resulting mRNA molecule docks onto a molecule known as the *spliceosome* whose job it is to remove the introns. The result is an mRNA molecule consisting solely of exons, which then migrates to the ribosome to make the protein. Actually there is an additional complication known as *alternative splicing.* In particular, the splicing process may choose to omit certain exons as well as the introns. Thus, a given gene can yield several distinct kinds of mRNA molecules, and thus several distinct (but related) protein species. The situation is not "one gene/one protein" but rather "one gene/several proteins."

How does the spliceosome know where to splice the mRNA sequence? That is, how does it find the *splice sites*—the boundaries between exons and introns? This is achieved at least in part via the presence of specific subsequences of nucleotides in the DNA surrounding the splice sites. These subsequences are not deterministic, but they do have statistical fingerprints that are meaningful to the spliceosome (and can be exploited by gene finding algorithms as well).

Now that we have a somewhat better idea of how DNA is transcribed into mRNA, let us return to the ribosome and consider the translation process in more detail. As we have discussed, the process basically involves walking along the mRNA sequence and converting triplets of nucleotides into amino acids. This process does not start with the first nucleotide in the mRNA however; rather it skips along the sequence until it reaches the triplet AUG, a triplet known as the *start codon.* This sequence is translated into an amino acid (AUG codes for the amino acid methionine), and the process thereafter continues along the mRNA molecule, translating codons into amino acids. The process stops when it reaches a *stop codon*—one of the three sequences TAA, TAA or TAA (none of which codes for an amino acid, thus the last translated amino acid is that coded for by the codon preceding the stop codon).

Thus there are three possible *frames* in the mRNA sequence, which correspond to the three different possible starting points of codons in the sequence. Shifting by one or two nucleotides in one or the other direction yields a different frame, and an entirely different sequence of amino acids. The location of the start codon defines the correct frame for translation. Moreover, a stop codon ends the translation process only if it is in the correct frame.

The segment of nucleotides upstream of the start codon is referred to as the $5'$ *untranslated region* (the $5'$ UTR). Similarly the segment of nucleotides downstream from the stop codon is referred to as the $3'$ *UTR.* While these nucleotides are skipped over in the translation process, they do play a role in translation—subsequences in the UTR are bound by proteins that help the ribosome to recognize the start and stop codons.

Returning once again to the DNA molecule in Figure 23.1, we see the $5'$ UTR and the $3'$ UTR at the left and right ends of the molecule, respectively. Note also that parsing into exons and introns is unrelated to the parsing into ($5'$ UTR, coding region, $3'$ UTR). That is, the first exon might contain the start codon or it might not. The former case is depicted in the figure. In the latter case (which is not depicted in the figure), the $5'$ UTR extends throughout the first exon and into the second exon, with an intron intervening.

It is also worth noting that the parsing into codons is unrelated into the parsing into exons and introns. That is, a codon can span a splice site, with part of the codon in one exon and part in the following exon. Once the intron is removed by splicing, the codon becomes contiguous, but it is not contiguous in the DNA sequence.

Although we have been very specific about where the translation process starts and ends, we have not been very specific about where the transcription process starts and ends. On the left side of the figure, we have indicated a *transcription start site* (TSS). Although the cellular machinery knows where the TSS is, the TSS has been more difficult to pin down using computational methods than the other structures in the gene, and most current gene finders do not attempt to locate it. Similarly, on the right side of the figure, we see that the gene contains a *poly-adenylation* site near the end of the final exon. Again, the cell knows where to stop transcription, but it has proved difficult in general to locate this site using computational methods. Biologically, the $3'$ end of an mRNA molecule is somewhat indeterminate, with the poly-adenylation site essentially being a signal that triggers a process in which long, variable-length sequences of A's are added to the molecule. These sequences provide protection for the mRNA molecule from degradation by enzymes.

To summarize, a gene is turned into a protein via a pair of processes. *Transcription* locates the sequence of exons in the gene and removes the intervening introns. *Translation* locates the start and stop codon, removes the UTRs upstream of the start codon and downstream of the stop codon, and converts the resulting sequence of codons into a sequence of amino acids.

23.2 Hidden Markov models for gene finding

Our focus in this section is the problem of *de novo gene finding*. Assume that we are given a stretch of DNA. Does it contain any genes? Where are the exons, the start codon, the stop codon, etc? Indeed, suppose that we are given the entire DNA sequence from a given organism. Where are the genes?

The problem of gene finding can be viewed as a parsing problem, and the workhorse parser of the graphical model framework—the hidden Markov model—is a natural tool. The advantages of the HMM include not only the ability to assign probabilities to gene occurrences and to gene structures, but also the ability to estimate parameters from either labeled or unlabeled sequences, or from partially-labeled sequences. A number of HMM-based gene finders have been deployed in the bioinformatics literature in recent years. The most successful models have been those based on the most elaborate state spaces, capturing all or most of the features of genes discussed in Section 23.1. The discussion in this section is based on the HMM-based gene finders known as GENSCAN Burge and Karlin (1997) and GENIE Kulp et al. (1996).

The data for a gene finder are sequences of bases, and thus the emissions of an HMM-based gene finder are multinomial random variables ranging over the set $\{A, C, G, T\}$. The states are multinomial random variables ranging over symbols such as "exon," "intron," and "intergenic." In particular, for the HMM that we will consider, the states and the non-zero entries of the state transition matrix are displayed as a stochastic automaton in Figure 23.2.

A first point to note is that there are no self-loops on the exon states. Recall that the number of steps that an HMM remains in a given state—and thus the length of the corresponding subsequence—is distributed according to a geometric distribution (see Exercise **??**). While the

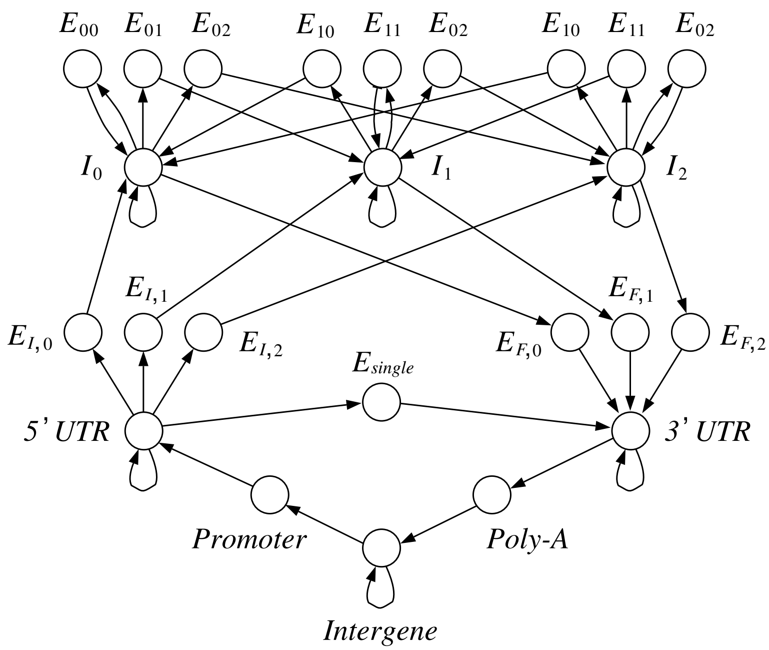

Figure 23.2: The state space for a generalized HMM for gene finding.

lengths of introns and intergenic regions can be reasonably modeled by geometric distributions, it turns out that exon lengths cannot be modeled as geometric variables (Burge and Karlin, 1997). In particular, short exons are rare—there appear to be biological constraints on the minimum size of viable exons. Thus exons states are *generalized states*, and the HMM in Figure 23.2 is a *generalized HMM*. In particular, upon entering an exon state, a length d is chosen from a given (non-geometric) distribution, and a sequence of length d is emitted. We provide more details about this emission distribution below.

The second point to note is that the biology requires that the total length of the set of exons generated is a multiple of three. To implement this constraint, we use an augmented state space that counts the number of exonic bases generated thus far modulo three. In particular, rather than using a single intron state, there are three intron states. Entering the state I_0 means that the number of exonic bases that the system has generated thus far is a multiple of three (i.e., 3, 6, 9, ...). Similarly, entering the state I_1 means that the system has generated a number of bases that is equal to one (mod 3) (i.e., 4, 7, 10, ...). Transitions out from I_0 are constrained to enter the exon states E_{0i}, where the first index retains the memory that zero (mod 3) bases have been generated thus far, and the second index indicates the number of additional bases generated (mod 3). In general, entering E_{ij} means that i (mod 3) bases have been generated thus far, and j (mod 3) bases will have been generated once the system leaves the exon state E_{ij}, at which point the system necessarily jumps to intron state I_j. Note in particular that this implies a constraint on the length distributions; the length associated with state E_{ij} must be equal to $j - i$ (mod 3).

The role of the initial exon states $E_{I,i}$ and final exon states $E_{F,i}$ should be clear. In particular the latter directly enforce the constraint that the total number of exon bases generated is a multiple of three.

Let us now consider the emission probability distribution for exons. From the point of view of the generalized HMM formalism, this distribution simply provides a likelihood term for a graphical model inference algorithm, and can be essentially anything. That is, given a sequence y of length d, the probability $p(y \mid q_t = E_{ij}, d)$ is simply a number that is evaluated and plugged in to an inference algorithm. From the point of view of biological modeling, however, it is essential to make good choices about how to represent these probabilities. This is done as follows. First, by convention splice sites are generally modeled as part of the exon. Thus, upon entering a state E_{ij}, the system generates the bases of an intron-exon splice site (a so-called *acceptor site*). Such sites are generally modeled as fixed-width subsequences, with different implementations making different choices for the particular form of probability distribution in this window. Most implementations use various kinds of graphical models for this distribution, the simplest case being a factorized model in which each base in the window is generated independently. Next, the model chooses a length from a state-specific length distribution and generates a sequence of the appropriate number of exonic bases. The length distribution is generally nonparametric (i.e., a table), and the probability distribution for the bases conditional on the length is given by a fifth-order Markov chain. This choice of order makes it possible to capture correlations between neighboring codons (for example, the choice of base in the third position of a codon depends on the previous two bases in the codon, and all three of the bases in the preceding codon). Note that the Markov chain is nonhomogeneous (different transition matrices are used for the different sequence positions) and three-periodic (the same distribution is used for a given codon position). Finally, the model generates an exon-intron splice site (a *donor site*). Again, such sites are generally modeled using simple graphical models for the random variables in a fixed-width window.

A similar structure is used for the exon models associated with $E_{I,i}$ and $E_{F,i}$, omitting the acceptor splice site model and the donor splice site model, respectively, and replacing them with start-of-translation and end-of-translation models. In the simplest case, the former is simply a deterministic distribution that places all of its mass on the start codon ATG, and the latter distributes mass on the three possible stop codons. More generally, it is common to augment these models with simple fixed-width models around start codon and stop codon cores. Finally, for the single exon model associated with E_{single}, no splice sites are needed, and it suffices to open with a start codon, generate an integer number of codons, and close with a stop codon.

High-order Markov models are also often used for the emission distributions associated with intron states. (Recall that it is always possible to condition a given output on previous outputs in an HMM; this does not induce any additional coupling among the state variables, and thus does not complicate the inference procedure). In this case the Markov models are homogeneous (there is no codon structure to capture). Such emission distributions are also generally used for the intergenic state and for the states corresponding to the untranslated regions (UTR's).

For simplicity, we have modeled UTR's using single HMM states. This is inaccurate biologically. As we mentioned in the previous section, UTR's can be broken up by introns, and a more accurate model must contain machinery that allows splice sites and intervening introns within UTR's. We ask the reader to design such a model in Exercise **??**. Note that although UTR's are exons, they are non-coding exons; i.e., they do not contain codons. Thus a simpler emission model suffices.

The poly-adenylation state is actually a short subsequence of six states, providing an emission distribution that assigns most of its mass to the sequence $AATAAA$. (We can choose to view this as a single generalized state, with a degenerate length distribution).

The modeling of promoter regions is a whole modeling enterprise of its own; as we discuss in Section 23.8.1, there is a substantial amount of biology surrounding the organization of promoter regions. In the context of gene-finding, however, it is often the case that implementations make use of a small set of states to provide a very minimal notion of promoter. We refer the reader to Burge and Karlin (1997) for an example.

Finally, it is worth noting that while we have restricted our discussion of gene finding to a single strand of DNA, in reality genes occur on both strands of DNA. Making the simplifying assumption that a gene on one strand cannot overlap with a gene on the reverse strand, we can build a gene finder that searches on both strands by simply duplicating all nodes in the state space other than the intergenic node, transposing the duplicate graph (i.e., reversing the directions of the arrows), and replacing all nucleotides with the complements (A with T, etc).

We will not discuss the problem of parameter estimation for gene finders, instead referring the reader to the primary literature. In brief, however, it is often the case that labeled training data are available (genomic sequences in which humans have labeled the gene structures). Thus, we are in the setting of completely observed models, and simple maximum likelihood or Bayesian estimates are obtained for individual elements of the model.

How well do HMM-based gene finders perform? Performance evaluation of gene finders is well beyond our scope, but a few comments are in order. For well-curated data sets, where each sequence is relatively short, and known to contain a single gene, HMMs perform reasonably well; in particular, Burge and Karlin (1997) report sensitivities and specificities over 90%. For longer, genome-scale sequences, however, performance is significantly poorer. One reasonable reaction to this fact is to consider more realistic, complex models. After all, the biological machinery can find the genes and splice them accurately, so the signals must exist in the data. Another reaction is to bring other kinds of information to bear on the problem. For example, practical gene finders are often augmented with information from RNA or protein libraries; these libraries provide constraints on putative exons. As another example, cross-species data can provide constraint on gene structure; a true gene is likely to also be found in a nearby species. It is this latter source of constraint that we explore in the next several sections. It will lead us into the study of new kinds of graphical models—phylogenies—and graphical models that combine phylogenies and HMMs.

23.3 Hidden Markov models for pairs of genes

In the previous section we have discussed gene finders that rely solely on information contained within the DNA sequence itself. The evidence that a particular site in the genome is a gene is accrued from the match of the site to the model of gene structure provided by the HMM. Another source of evidence that a site may be a gene is the presence of a similar sequence in a different organism. The logic is as follows: the presence of such a sequence suggests that the sequence has been conserved by evolution, which suggests that the sequence may be useful to the organism, which

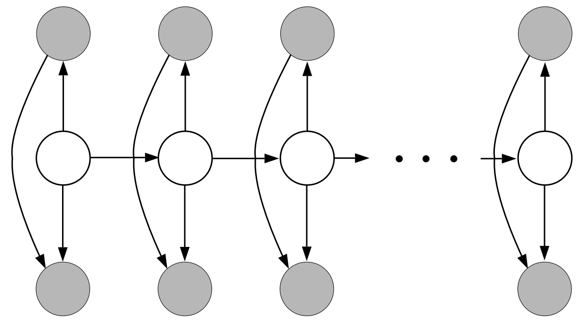

Figure 23.3: A pair HMM.

suggests that the sequence may be a gene. Each of these inferences may be wrong, but overall they provide useful source of probabilistic constraint that should be combined with the evidence from gene structure.[2]

There are two aspects of the problem of combining evidence from pairs of sequences. The first is the problem of *alignment*: Given a sequence element in one sequence, which is the corresponding element in the other sequence? Do we want to insist that each element can be so aligned? The second is the problem of relating aligned sequences to an underlying structure, namely that of a gene.

Let us begin by considering the second problem. In particular, let us assume (for now) that an alignment has already been provided, perhaps by one of a variety of dynamic-programming-based programs that are able to align biological sequences (such as the popular BLAST program Altschul et al. (1997)). The output of these programs include the gap symbol "$-$," thus we augment our vocabulary to include this symbol.

We now consider a generative model for an aligned pair of output sequences. The model is known as a *pair HMM*, and it is simply a hidden Markov model in which the output produced in each state is a pair of symbols rather than a single symbol. The model is drawn as a graphical model in Figure 23.3.

The state space of the pair HMM captures our assumptions about the structure of a gene just as in the single-sequence case. Indeed we can essentially borrow the state space used for the single-sequence gene finder (see Figure 23.2) for the pair HMM. The only difference is that a pair of symbols is emitted in each state. Thus, if we are an in intron state, then we emit a pair of bases, using a joint distribution that reflects the assumption that the same nucleotide is most likely to appear in both of the sequences, and allows "mutations" with appropriate probabilities. If we are allowing generalized states, then we must be prepared to generate a pair of lengths and a pair of sequences of the appropriate lengths. Note, however, that exons provide the key example of the

[2]Genes in two different organisms that derive from the same gene in an underlying ancestral organism are said to be *homologous*. Homologous genes can either be *orthologous* or *paralogous*: Orthologous genes are direct descendants from the underlying gene, diverging after a speciation event, whereas paralogous genes derive from a process that copies a gene from one place in a genome to another place. Paralogous genes tend to diverge more quickly that orthologous genes, given that the function provided by the gene can be provided by either one of the paralogous copies. Thus the methods that we discuss in this section tend to be most useful for orthologous genes.

need for generalized states in the gene-finding setting, and exons are highly conserved. Thus it may suffice to consider simplified models in which a single length is generated for the exon states, and the gap symbol is used to handle small perturbations around that length for the two exon sequences.

We have depicted the joint emission distribution using an directed edge in Figure 23.3. This reflects the fact that we may wish to make use of biological knowledge to specify these parameters; in particular, evolutionary biologists have provided tables specifying conditional probabilities of mutations of codons or nucleotides at given evolutionary distances, and these may be used directly to parameterize the model, or may be useful as priors. But it also may be convenient to parameterize this distribution using an undirected edge, capturing the basic symmetry in the problem.

It should be clear that although there are details to be worked out, the pair HMM approach is relatively straightforward if we can assume that the data have already been aligned. Can we be more ambitious and try to solve the alignment problem itself within the pair HMM framework? Is it reasonable to try to do this?

Sequence alignment is a field of its own (see, e.g., Gusfield, 1997). Alignment algorithms are often based on models in which various transformations on sequences are considered, such as insertion, deletion and replacement. Costs are associated with these transformations, and optimal sets of transformations can be found via dynamic programming. By using costs that reflect biological knowledge, we obtain efficient algorithms for aligning biological sequences—BLAST is an example. This approach makes somewhat limited use of biological knowledge, however. For example, in the gene-finding setting, one sequence may differ from another by the insertion of a long intronic segment. Such sequences will be appear to be quite different from the point of view of classical sequence alignment algorithms. From the biological point of view, however, such sequences would yield identical proteins and should be considered to be essentially the same. As another example, the cost metric for the replacement of one base by another should be different in introns and exons; in the latter case we should use a codon-based cost.

These considerations suggest that the notion of alignment between genomic sequences is best defined in terms of a model of gene structure. That is, to decide whether a subsequence is similar to another subsequence, we need to know whether these subsequences appear in an exon, an intron, or another part of the genome. This suggests that we need an integrated model that solves the alignment problem and the gene finding problem together.

Although our presentation of the pair HMM assumed that an alignment was given, it turns out that the pair HMM does not actually require an alignment, and indeed the pair HMM provides the integrated approach to alignment and gene finding that we are seeking. This has been pointed out by several authors, including Meyer and Durbin (2002) and Alexandersson et al. (2003) whose work provide the basis for the discussion in the remainder of this section.

Our problem is illustrated by the pair of sequences shown in Figure 23.4. As suggested by the illustration, we would like to align certain subsequences in pairs of sequences, and be free to leave other subsequences unaligned. For example, it may be the case that an intronic segment is present in one sequence and not in the other, and the bases in this intronic segment do not have any correspondence with any bases in the other sequence. Biologically, this is a common scenario—insertion of genetic material in an intron does not change the protein coded by the gene. Another

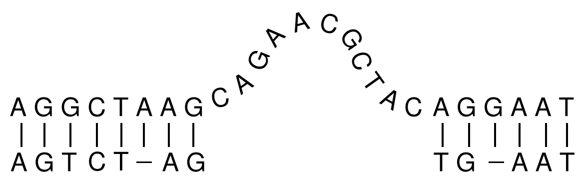

Figure 23.4: A depiction of an alignment. The vertical bars denote positions that are aligned, and the dashes denote deleted bases. The bases in the curved subsequence (CAGA...) in the upper sequence are not aligned to any bases in the lower sequence.

common biological phenomenon is deletion—homologous genes often differ due to a codon or base being deleted relative to an ancestral sequence in one but not the other gene. While both insertions and deletions can be denoted by using a gap symbol, these phenomena are different biologically, and the underlying model should treat these cases differently.

The pair HMM provides an elegant treatment of these issues. The first point to note is that not all states need to emit pairs of symbols; some states can emit single symbols. Thus we can have an intron state I_X that emits only symbols belonging to the first sequence, and an intron state I_Y that emits only symbols belonging to the second sequence. The second point is that we can make use of the gap symbol as an explicit member of the output alphabet, and interpret the generation of these symbols as deletions.

Augmenting the state space to allow states such as I_X and I_Y is a relatively straightforward exercise, which we ask the reader to explore in Exercise **??**. On the other hand, drawing the graphical model corresponding to this elaborated pair HMM presents something of a problem—how do we handle the fact that some states emit pairs of symbols and other states emit single symbols? In fact, this model is *not* a graphical model; it is an example of a generalized graphical model as discussed in Section **??**. We can draw a representation of this model using a plate for each observed variable, in which the number of replicates is random variable ranging over $\{0, 1\}$. Of course, we can also allow these variables to range over the nonnegative integers, to handle cases of generalized states that emit more than one symbol.

Despite these diagrammatic complications, it is not difficult to write out variants of the forward-backward algorithm and Viterbi algorithm for these pair HMMs (see Exercise **??**), and to perform all of the inference and parameter estimation procedures that are available for standard HMMs. Note in particular that the Viterbi algorithm yields a path through the state space, which in general will include states that generate pairs of symbols and states that generate single symbols. Thus, we obtain exactly the kind of partial alignment that we asked for in the illustration in Figure 23.4. Moreover, each symbol in each sequence is annotated with an underlying genomic structure. We solve the problem of simultaneously annotating and aligning the two sequences.

Given this elegant solution to the problem of annotation based on pairs of sequences, can we generalize to more than two organisms? Different regions in the genome presumably show a variety of degrees of conservation, and constraints from organisms at a variety of evolutionary distances would surely be useful. This problem can indeed be addressed within the graphical model formalism, as we will see in Section 23.5. Before presenting methods for gene finding based

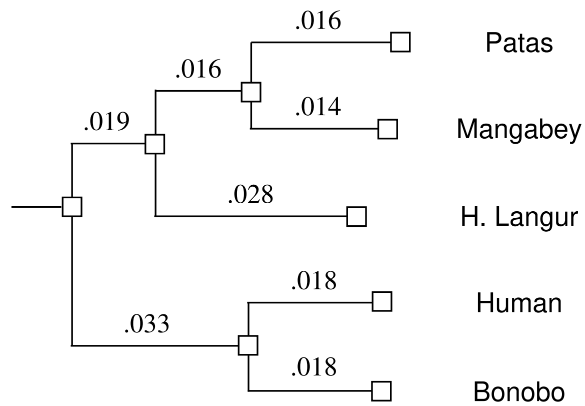

Figure 23.5: A traditional representation of a phylogenetic tree. The leaves represent a set of extant species, and the nonterminal nodes represent putative ancestral species. The numbers above the branches are the branch lengths, which represent a measure of evolutionary time.

on multiple organisms, however, we need to discuss phylogenies, which are of major importance in and of themselves.

23.4 Phylogenetic trees

Consider the diagram shown in Figure 23.5. This diagram is a *phylogenetic tree*, sometimes called an *evolutonary tree*. It is intended to capture relationships among a set of extant species, by relating them to a set of putative ancestral species. Where do these diagrams come from? How are they to be interpreted?

Phylogenetic trees can be interpreted as graphical models, and the problem of estimating the structure and parameters of the trees from data can be treated via the tools discussed in this book. In this section we explain this interpretation, and in particular we will discuss the probablistic assumptions that underlie the interpretation. As will be seen, some of these assumptions are only rough approximations to biological reality. Graphical models also provide a useful framework for weakening these assumptions and for considering more complex and realistic models.[3]

A phylogenetic tree aims to describe the evolution of a "phylogenetic character" (e.g., a DNA sequence, a protein, or a phenotypic trait) as it is passed down through a set of species over (evolutionary) time. Speciation events are assumed to be binary, with an ancestral species giving rise to two sub-species. Using a binary directed tree to represent these speciation events, the basic data structure of interest is a directed tree with M leaves representing extant species, and $M-1$ non-terminal nodes representing ancestral species.

For phylogenetic analysis to make sense, the M characters that we consider need to be *homol-*

[3]It should be noted that while the statistical approach to phylogenetic analysis that we focus on here is a popular one, it is by no means the only approach. Phylogeny is a large field and there are many competing approaches. See, e.g., Felsenstein (2003) for a fuller perspective.

ogous—they need to have arisen from some underlying ancestral character. To decide conclusively that characters are homologous generally requires expert biological knowledge. In practice, rougher methods are often used; e.g., given a DNA sequence or a protein, one uses sequence comparison algorithms such as BLAST (or the profile HMMs that we discuss in Section 23.6) to find other sequences that are "sufficiently close." Moreover, in the case of sequential data, we need to decide which elements in each of the sequences are homologous to each other. That is, we need to find a *multiple alignment*. In practice, phylogenetic analysis is generally based on the assumption that a multiple alignment has already been found, perhaps via expert knowledge or via one of a number of heuristic algorithms (Felsenstein, 2003). This is suboptimal, because phylogenetic analysis and multiple alignment are inherently related, and ideally would be performed together.[4] For reasons of computational cost, however, they are generally separated in practice.

We return to this issue in Section 23.4.2. In the meantime, in Section 23.4.1, we avoid multiple alignment issues by considering single genomic sites—"sequences" of length one. We focus on DNA for concreteness, and thus our interest is in phylogenetic models for single nucleotides. Note again that although this avoids the multiple alignment problem, it doesn't avoid alignment issues altogether. To decide that one site in each of a set of genomes are homologous is in essence to align those genomes to each other at that site. We assume that this has already been done, via expert knowledge or some other method.

23.4.1 Single sites

We begin with the simple case in which the phylogenetic characters of interest are single nucleotides. Each such character is treated as a multinomial random variable, ranging over the set $\{A, C, G, T\}$. As shown in Figure 23.6, we treat the phylogenetic tree on these characters as a graphical model, with one such multinomial random variable at each node. The leaves of the tree correspond to the observed nucleotides in the extant species, while the non-terminal nodes are latent.

The fact that the graph is a tree captures various assumptions of conditional independence. In particular, given the value of the base in an ancestral species, the value of the base is independent in the resulting sub-species. Given that the definition of a species generally includes the lack of propensity to mate with another species, and thus to exchange DNA, this conditional independence assumption seems reasonable. It should be noted, however, that mating is not the only way that DNA can be exchanged; in bacterial species, DNA can be shuttled from one organism to another via mechanisms of "lateral gene transfer." In these organisms, DNA does not necessarily evolve independently in different branches, and one would want to consider a richer family of graphical models that allows lateral edges in the graph.

We now consider the parameterization of a phylogenetic model. For the root node, X_r, we require a marginal distribution, which we denote as $(\pi_A, \pi_C, \pi_G, \pi_T)$. For the non-root nodes, we need to specify the conditional probability of the node X_i given its parent X_{π_i}—a 4×4 table of probabilities. These can be viewed as mutation probabilities for nucleotides. We will choose a

[4]Indeed, heuristic multiple alignment programs are often based on notions such as "guide trees," which allow closely related species to have more impact on the alignment. These guide trees are in essence quick-and-dirty phylogenetic trees.

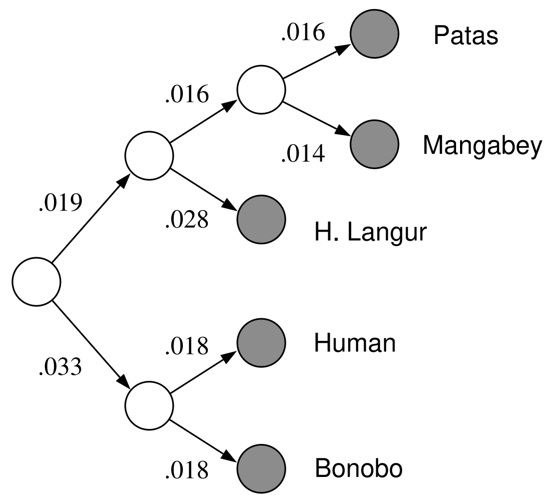

Figure 23.6: A graphical model representation of a phylogenetic tree. The leaves represent a set of extant species, and the nonterminal nodes represent putative ancestral species. The numbers above the branches are the branch lengths $\{t_j\}$, which are parameters of the conditional probability distributions associated with the edges of the graph.

particular functional form for these mutation probabilities, and parameterize them in terms of an "evolutionary time" parameter t.

A variety of models of nucleotide mutation are available in the literature. These models are generally expressed as continuous-time Markov chains. Recall that a continuous-time Markov chain is specified in terms of an instantaneous transition matrix Q, with the transition matrix over a finite time interval t obtained via the matrix exponential $\exp(tQ)$. That is, letting $X(t)$ denote the state of the Markov chain at time t, and denoting the probability distribution at time t as $p(x(t))$, we have:

$$p(x(t)) = e^{tQ} p(x(0)) = M e^{t\Lambda} M^{-1} p(x(0)), \tag{23.1}$$

where M is the matrix whose columns are the eigenvectors of Q and Λ is the diagonal matrix whose elements are the eigenvalues of Q. Ranging over initial state distributions that place unit mass on each of the four possible states, and thus picking out the four columns of $M\exp(t\Lambda)M^{-1}$, we obtain the conditional probabilities $p(x(t) \,|\, x(0))$. Making an assumption of stationarity in time, these conditional probabilities are the parameterized expressions that we require. We associate one such conditional probability with each edge in the tree, and thus associate a parameter t_j with the jth edge in the tree. The set of parameters $\{t_j\}$ are known as the *branch lengths* of the tree.

To provide an example of the procedure, let us consider the simplest possible evolutionary model. This model, known as the Jukes-Cantor model (Jukes and Cantor, 1969), embodies the assumption that all transitions are equally probable:

$$Q_{JC} = \begin{pmatrix} -1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & -1 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & -1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & -1 \end{pmatrix}. \tag{23.2}$$

Note that the rows sum to zero; this is a requirement for any continuous-time instantaneous transition matrix. We have also normalized the matrix such that the off-diagonal elements sum to one within each row; this corresponds to a choice of scale for t.

Computing the eigenvectors and eigenvalues of Q_{JC}, and taking the first column of $\exp(tQ)$ we obtain the transitions from the state $X(0) = A$:

$$p(X(t) = C \,|\, X(0) = A) = \frac{1}{4}(1 - e^{-\frac{4}{3}t}), \tag{23.3}$$

and identical values for $p(X(t) = G \,|\, X(0) = A)$ and $p(X(t) = T \,|\, X(0) = A)$. We also obtain:

$$p(X(t) = A \,|\, X(0) = A) = \frac{1}{4}(1 + 3e^{-\frac{4}{3}t}), \tag{23.4}$$

for the probability that the nucleotide is found in the initial state at time t (it may have left that state and returned). Note that for small values of t, the state is likely to be found in the initial state, and for large values of t, all bases become equally probable.

This last fact indicates a weakness of the Jukes-Cantor model. The unconditional distribution $(\pi_A, \pi_C, \pi_G, \pi_T)$ at the root presumably embodies some sort of equilibrium distribution, and for consistency we would like this same distribution to arise as the limiting distribution of the conditional probabilities. This can be achieved with more complex mutation models. Consider in particular the following model due to Hasegawa et al. (1985):

$$Q_{HKY} = \begin{pmatrix} -1 & \alpha\pi_G + \beta\pi_G & \alpha\pi_C & \alpha\pi_T \\ \alpha\pi_A + \beta\pi_A & -1 & \alpha\pi_C & \alpha\pi_T \\ \alpha\pi_A & \alpha\pi_G & -1 & \alpha\pi_T + \beta\pi_T \\ \alpha\pi_A & \alpha\pi_G & \alpha\pi_C + \beta\pi_C & -1 \end{pmatrix}, \tag{23.5}$$

where α and β are parameters. It can be verified that the equilibrium distribution associated with this matrix is precisely the unconditional distribution $(\pi_A, \pi_C, \pi_G, \pi_T)$. Moreover, the model provides an additional dose of biological realism in providing separate control over the rate of "transitions" (mutations between the purines A and G or between the pyrimidines C and T) and "transversions" (mutations from a purine to a pyrimidine or vice versa). Again, the eigenstructure of this matrix can be computed, and one obtains an explicit formula for the conditional probability of each base as a function of t.

As we have stated, the data available for phylogenetic inference problems generally involve observations of the extant species at the leaves of the tree, with the ancestral species being unobserved. In this setting there are two computations of interest: (1) computation of the posterior probability of the latent ancestral nodes, and (2) computation of the likelihood of the observed variables at the leaves. The former problem is readily solved via the sum-product algorithm, and the latter problem is solved with the sum-product algorithm or the elimination algorithm.[5]

We would also like to estimate the parameters (the evolutionary time parameters $\{t_j\}$, one for each branch of the tree), and to estimate the tree structure. The former problem can be handled via the EM algorithm, whereas the latter requires some form of search. In both cases, however, we require more data than one observation per leaf in order to have some hope of estimating these quantities reliably. We thus turn to the case of multiple sites.

23.4.2 Multiple sites

We now consider the case in which we have not a single base at each node of the tree but a sequence of bases at each node. We assume that these sequences are known to be homologous. For example, the sequences may be genes, and knowledge of the corresponding protein products and their functions in the cell may be enough to establish homology. Ideally, we would not require any additional knowledge about homology; in particular, we would not require a multiple alignment which specifies the homology of individual bases. Rather, we would find a multiple alignment as part of the problem of fitting a phylogeny. To do this, we could consider models in which entire sequences mutate along branches of the tree, with insertions, deletions and base changes allowed. Tracking these transformations throughout the tree, we would be able to construct a multiple alignment of the sequences at the leaves. There have been several proposals along these lines (Felsenstein, 2003). The resulting algorithms are computationally demanding, however, particularly in the context of phylogenetic algorithms that are searching over large sets of possible trees. In practice, phylogenetic analysis is generally based on a pre-existing multiple alignment, and this will be our assumption in this section.

Thus, our data consist of an $M \times L$ matrix X of nucleotides, where the rows index the M observed species and the columns index L aligned sites in the genome. To accommodate a (poor man's) notion of deletion or insertion, one can augment the vocabulary to include a symbol for a missing value, and marginalize over such values.

We need to consider how aligned sequences evolve along the branches of a tree. The simplest possible assumption is that each of the sites are independent and identically distribution. The independence assumption is biologically dubious—there are biochemical interactions that affect mutation probabilities at neighboring sites, and there are global constraints that alter mutation rates in conserved regions of DNA and proteins—but the assumption leads to simple models and is widely made in practice. Making this assumption the joint probability model is simply the product over individual sites, and can be represented by placing a plate around the graph shown in Figure 23.6. The phylogenetic tree on sequences factors into an independent set of phylogenetic trees on sites.

Inference in this model reduces to separate runs of the sum-product algorithm for each of the individual sites. The EM algorithm is also straightforward in this setting, involving an accumulation

[5]It is interesting to note that one of the first historical appearances of the elimination algorithm was in the calculation of the likelihood of phylogenetic trees. The algorithm was called "pruning" Felsenstein (1973).

of expected sufficient statistics across sites. Finally, one can attempt to find maximum likelihood trees, or attempt to sample from a posterior distribution on trees, using search techniques such as MCMC. The computational burden is quite high in either case, due to the large number of possible trees. Some relief can be obtained by avoiding the recalculation of expected sufficient statistics when it is not necessary to do so; e.g., for trees that differ by small changes. But for even a modest number of species, there are a large number of trees to consider, and any search technique is only able to explore a small fraction of the space.

The iid assumption can be weakened in various ways, leading to improvements in phylogenetic estimates. In particular, it has been found useful to separate out a multiplicative *mutation rate* at each site, and to allow the mutation rate to vary across sites. Define a mutation rate r by replacing t by rt in Eq. (23.3) and Eq. (23.4). Yang (1993) has shown that treating r as a gamma random variable leads to improved estimates. Another approach is to consider a finite set of mutation rates, and define a Markov chain on these rates. We will discuss this latter approach in the following section.

A final remark concerns the choice of directed trees as the modeling formalism for phylogenies. Directed trees have the virtue that they correspond to the underlying biology, where an underlying ancestral character gives rise causally to an observed set of phylogenetic characters. From a statistical point of view, however, it turns out that we are not necessarily able to infer directionality from the data. In particular, the continuous-time Markov chains described above have the property of being *reversible*. As we ask the reader to show in Exercise **??**, this implies that the likelihood is identical for any choice of root in a directed tree. Thus the data alone cannot orient the tree. Without any additional evidence regarding the location of the root, the undirected graphical model formalism should be used.[6]

23.5 Hidden Markov phylogenies

In this section we discuss the *hidden Markov phylogeny (HMP)*—a graphical model that combines HMMs and phylogenies. We will discuss applications to phylogenetic analysis, to comparative gene finding, and to the modeling of the secondary structure of proteins.

As a graphical model, the HMP is simply an HMM in which a phylogenetic tree hangs off each state variable (see Figure 23.7). Thus, the emission distribution is modeled by a phylogenetic tree, and the output is not a single symbol but rather a *set* of symbols—those at the leaves of the tree. Both the state variables of the HMM and the non-terminal nodes in the tree are latent variables. In most applications, the set of trees are identical structurally across the state variables and for each value of a given state variable, but they can also be allowed to vary.

The various inference problems of interest in the HMP are readily handled via the junction tree algorithm, in its sum-product or max-product forms, and thus our focus in the current section is not the algorithmic issues, but rather the modeling issues. What modeling leverage do we obtain by combining HMMs and phylogenies?

[6]Such additional evidence may take the form of an "outgroup" which is known to be distant from the other species being considered, and thus must be a child of the root. For example, mouse DNA might serve as an outgroup for consideration of primate DNA sequences.

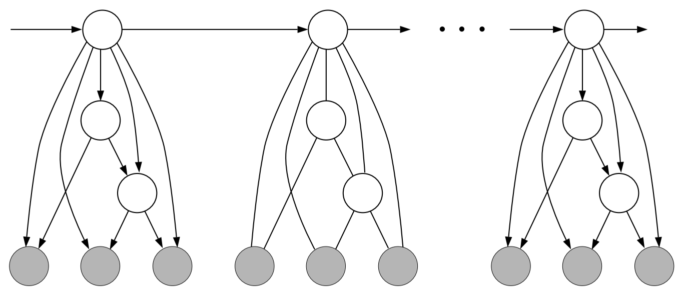

Figure 23.7: The hidden Markov phylogeny (HMP). The HMP is an HMM in which the emission distribution is modeled as a phylogenetic tree. As the shading indicates, the most common application of the HMP is in a scenario in which the leaves of the phylogenetic trees are the observed data, and both the Markov states and the ancestral nodes in the phylogenetic trees are unobserved.

23.5.1 HMPs and phylogenetic analysis

As we have discussed in the previous section, phylogenetic analysis is often carried out under the assumptions that the sites are independent and identically distributed—assumptions that are unrealistic biologically. The HMP makes it possible to weaken both assumptions and obtain improved phylogenetic estimates (Yang, 1995, Felsenstein and Churchill, 1996). Consider an HMP in which the state space of the Markov chain encodes a set of rates. Given a choice of rate at a given site, all transitions in the phylogenetic tree corresponding to that site are made at that rate. Moreover, given a choice of rate at one site, the choice of rate at the next site is made according to a Markov transition matrix. This allows the model to capture the biological fact that certain regions of a sequence may be highly conserved and other regions less so.

More formally, let Q_s denote the state of the Markov chain at the sth site. The state variable Q_s ranges over a discrete set of K values, and there is a rate parameter r_i associated with the ith value of the state. The Markov chain transition matrix defines transitions between values of the state variable, and thus effectively induces transitions between values of the rate parameter.

Now let us consider the branches within the phylogenetic tree. Whereas in our earlier presentation, the transition matrix for the jth branch in a phylogenetic tree was given by $\exp(t_j Q)$, where t_j is the branch length parameter associated with the jth branch, the transition matrix is now given by $\exp(r_i t_j Q)$, when the Markov chain is in the ith state. Note that whereas the parameter t_j varies within the tree, the parameter r_i is fixed for the entire tree.

Given a set of aligned sequences, the parameters of the HMP can be estimated via the EM algorithm. Moreover, as in the simpler independent sites model, likelihoods can be computed via the junction tree algorithm and used to compare between different choices of structures for the phylogenetic trees. The output of this procedure might be the phylogenetic tree itself, with the HMP machinery used simply to provide an improved estimate of this tree—an estimate that is not limited by the iid assumptions made in classical phylogenetic analysis. The states of the HMP model may also be of interest. In particular, in biological sequences one expects to see stretches in which the mutation rate is small, followed by stretches in which it is large. The Viterbi version of the junction tree algorithm can provide this kind of parsing for a set of sequences of interest.

Similarly, the algorithm can provide estimates of most likely values for the ancestral nodes in the phylogenetic trees. These estimates will reflect dependencies among neighboring sites, and may provide better estimates than those that are available from an iid model.

23.5.2 HMPs and gene finding

While the previous section focused on the benefits that a hidden Markov chain can provide to phylogenetic analysis, one can take a complementary point of view and consider the benefits that a phylogenetic tree can provide to hidden Markov models, in particular those that we have discussed for gene finding in Section 23.2 and Section 23.3.

The gene finders that we have discussed thus far make use of single sequences or pairs of sequences.[7] The HMP provides a methodology for gene finding when more than two sequences are available (Siepel and Haussler, 2003, Pedersen and Hein, 2003, McAuliffe et al., 2003). Evidence for gene structure can be accumulated across several organisms, and thus we would expect improvements in gene finding performance, particularly in terms of sensitivity. But the HMP should also be expected to yield improvements in the specificity of gene finding. When estimating parameters for HMM-based gene finders, one generally treats the training sequences as independent samples (i.e., the overall likelihood is a product across sequences). In reality, however, biological sequences are not independent; rather, they are related by evolution. By attempting to capture the evolutionary process that relates biological sequences, the HMP combines the evidence in a manner that takes this dependence into account. This avoids "double-counting" of evidence, and should lead to improved specificity.

As we discussed in Section 23.4, phylogenetic analysis methods generally assume that an alignment is given, and current applications of the HMP inherit this limitation. Thus, in contradistinction to our discussion of the pair HMM in Section 23.3, in this section the input is assumed to be an aligned set of sequences. This need not be a limiting assumption in practice; for species that are not highly diverged, an alignment is not generally difficult to obtain. Moreover, it can be argued that comparative gene finding is most appropriate in such a setting (organisms that are evolutionary near-neighbors should have similar sets of genes, and thus provide the strongest boost to gene-specific signals).

In applications of the HMP to gene finding, the design of the state space for the Markov chain is in principle no different than the design in the single sequence case. In particular, states correspond to gene structures such as exon, intron, untranslated regions, etc. Moreover, it is important to allow the exon states to be generalized states, yielding a *generalized hidden Markov phylogeny* (McAuliffe et al., 2003).

As in Section 23.5.1, all of the transitions in the phylogenetic trees are conditional on the state, and thus these transitions can occur with different rates in the different states. In particular, the rates for exons can be slower than the rates for intergenic and intron states, reflecting evolutionary conservation of coding sequences. As in our previous discussion, this can lead to improved phylogenetic estimation. More relevant to our current discussion, it can also lead to improved gene

[7]Actually, the pair HMM in Figure 23.3 can be viewed as a simple HMP, in which a single ancestral node is marginalized out, yielding a link between the two leaves.

finding performance. McAuliffe et al. (2003) used a generalized HMP for gene finding in human DNA sequences, making use of data from over a dozen primate species. Running the max-product algorithm on the generalized HMP to yield a parsing into exons, introns and intergenic regions, they reported significant improvements in gene finding performance relative to GENSCAN.

23.5.3 HMPs and secondary structure of proteins

Goldman et al. (1996) have described an HMP that can be used for parsing proteins into structural components. This is again a very natural application of combined structural and phylogenetic analysis. Evolutionary rates differ in different structures of proteins, and this constraint should be taken into account in parsing amino acid sequences. Also, in the analysis of protein sequences the fact that the available sequences are not iid is again a problem, and the HMP provides a natural solution to this problem.

Proteins can be described at a number of different levels of granularity. The linear sequence of amino acids is known as the *primary structure* of the protein. The full three-dimensional structure of a folded protein is known as *tertiary structure*. Between these two descriptions is a level of description known as *secondary structure*. This level refers to various structural patterns that arise in amino acid sequences and provide a basic vocabulary out of which larger-scale structures in proteins are constructed. In particular, certain sequences of amino acids tend to fold into helical structures known as *alpha helices*, and other sequences tend to fold into flat structures known as *beta sheets*. Other secondary structures include *loop regions* and *coils*. Proteins are often built out of collections of alpha helices and/or beta sheets, connected by less well-defined loop regions. The latter can often be of very different lengths in homologous proteins.

Recognizing secondary structure elements from primary sequence data is an annotation problem that is analogous to gene finding, and indeed, HMMs in the spirit of Section 23.2 have been developed to solve this problem.

The HMP presented by (Goldman et al., 1996) attempts to annotate secondary structure based on data from multiple species, taking into account the phylogenetic relationships among the species. The model is based on a simple set of structural states, namely states for alpha helices, beta sheets, loop regions and coil regions. The model also considers states that encode the solubility of amino acids (their tendency to prefer to be in contact with water or to prefer to avoid contact with water), treating solubility as a binary variable. The basic idea is that evolutionary rates should be expected to differ for amino acids that are buried inside the protein versus amino acids that are exposed to external solute. The overall state space considered by (Goldman et al., 1996) is a Cartesian product of the states for secondary structure and the binary solubility state.

The phylogenetic trees differ from those considered in previous sections only in their use of a different set of characters, namely amino acids rather than nucleotides. This requires a 20x20 transition matrix instead of the 4x4 transition matrix for nucleotides. Standard matrices are available in the molecular evolution literature.

23.6 Protein families and profile hidden Markov models

In previous sections, we have discussed hidden Markov models for finding genes in DNA sequences. These models are based on very general assumptions about the structure of genes, for example that they consist of sequences of exons and introns. In the current section, we consider hidden Markov models that aim at a more fine-grained set of distinctions. We consider proteins rather than DNA sequences, and describe models that aim to capture characteristics of particular subsets or "families" of proteins.

Given the wide variety of functions that proteins can subserve, and given the corresponding complexity of the three-dimensional structures of proteins, it is surely an ambitious goal to identify and model "families" of proteins, if by "family" we mean something closely related to function or structure. Moreover, the models that we discuss will be based entirely on primary sequence data—the sequence of amino acids making up the protein—and will not make (explicit) use of the biochemistry or biophysics of amino acid sequences. Given this large gap between goals and methods, what might we reasonably attempt to achieve, and why should we expect that relatively simple statistical models should be useful?

A major constraint operating in our favor is the fact that proteins are the product of evolution. Two proteins that have similar function in two different organisms may well have arisen from a common ancestral protein. If so, the proteins may more similar at the level of primary sequence than would be expected from the functional constraint alone. Moreover, even the amino acids that are incidental to the function of the protein may be partially conserved (although less so than those that are critical to the function), and their presence can be exploited as a weak statistical constraint indicating that two proteins are "similar."

Thus, a useful intermediate goal might be to identify "families" with "homologies," and to develop models that can detect homologies. Such models would be useful in understanding function, to the extent that homology is correlated with function, and would also be of intrinsic interest (in evolutionary biology, for example).

It should be noted that there are many examples of homologous proteins that have very little direct similarity at the level of primary sequence. One reason for this is that amino acids can often be substituted by other amino acids that have similar biochemical properties and thus leave the structure and function of the protein intact. A related reason is that some parts of the protein may be irrelevant to function and thus highly variable between members of a family. These kinds of variability are not, however, beyond the scope of statistical models; indeed, they are the target of the profile HMMs that we discuss below.

Another constraint operating in our favor is the fact that proteins have intermediate levels of structure between the primary sequence and the three-dimensional structure. The models that we discuss in this section do not attempt to identify such secondary structures explicitly, but they do exploit the fact that secondary structures impose statistical constraint upon observed protein sequences.

A final source of constraint is the nature of the data that are available in protein modeling problems. We can often expect data in which families are defined explicitly via biological experiments or via expert labeling based on analysis of tertiary structure. That is, rather than expecting

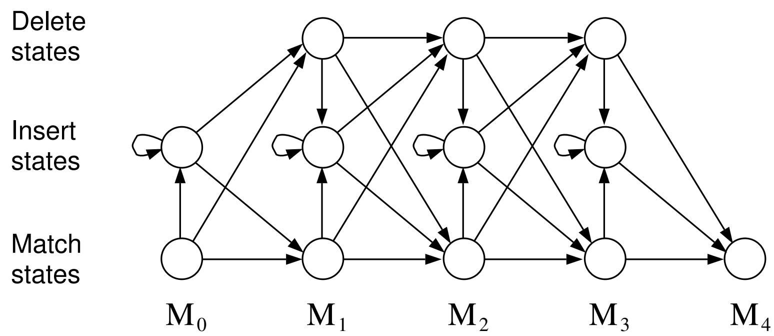

Figure 23.8: The state space of a profile HMM.

a statistical method to identify families of proteins *de novo* from an unstructured set of proteins (i.e., to solve a clustering problem), the goal is often that of recognizing new members of a family based on known instances of the family (i.e., to solve a classification problem).

Despite these constraints, the problem remains difficult. For the kinds of protein families that are of interest in practice, there are generally significant divergences among the sequences. In particular, sequences can differ due to long insertions or deletions. Thus, we are far from the setting of fixed-length feature vectors generally required by classification algorithms. Given a set of sequences belonging to a family, we must solve a multiple alignment problem as part of the modeling problem, and in this section we (finally) face this problem head-on. The profile HMMs that are our focus are essentially a minimal probabilistic model of sequence families that treats the multiple alignment problem.

23.6.1 Profile HMMs

A *profile hidden Markov model* is an HMM with three kinds of states, known as *match states*, *insert states* and *delete states* (Durbin et al., 1998). The state space of a generic profile HMM is shown as a stochastic automaton in Figure 23.8.

Ignoring the insert states and the delete states for the moment, we see that the match states form a linear sequence (note the absence of self-loops). We can view these states as representing the most highly conserved amino acids in the protein family. The emission distributions associated with these states should be expected to concentrate on subsets of amino acids that are able to fulfill similar biochemical roles within the protein. To the extent that alpha helices and beta sheets are conserved within a protein family, we would hope to be able to partition the match states into segments associated with these structures.

The delete states allow the HMM to skip over one or more of the match states. There is no output symbol associated with a delete state. The insert states are responsible for insertions of sequences. Noting the self-loop, we see that inserted sequences are implicitly assumed to have a geometric length distribution.

We define alignments solely in terms of the match states and the delete states. Let M denote the number of match states. Note that any legal path through the profile HMM goes through exactly M match states or delete states. Thus, if we generate N sequences, we can create an $N \times M$ alignment matrix in which the nth row contains the sequence of symbols generated for the nth sequence in the match states and delete states, where a dash is used to indicate a delete state.

It is important to note that subsequences generated from insert states are not considered to be part of the alignment. This assumption is a good fit to the underlying biology. In particular, insert states will often be used to denote unconserved loop regions in proteins, where individual proteins may show long idiosyncratic insertions relative to their homologs. These insertions should not be viewed as present in an ancestral protein and then deleted; thus, they should not be aligned.

Standard HMM technology can be used for the inference and parameter estimation problems associated with the profile HMM. In particular, the Viterbi algorithm takes on an important role in this setting. By returning the most probable path associated with a given observed sequence, the Viterbi algorithm allows that sequence to be aligned to the model. By finding the paths associated with each of a set of sequences, we can form an alignment matrix.

If the input sequences are unaligned, then parameter estimation is achieved via the EM algorithm. The output of the algorithm is a model, and, running the Viterbi algorithm on the resulting model, a multiple alignment. If the sequences have already been pre-aligned—by some other multiple alignment method—then the parameter estimation problem is simplified; we in essence obtain a completely observed model and the EM algorithm is not needed.

The model selection can as usual be approached with various forms of generic local search algorithms. Given the special form of the state space of profile HMMs, however, there are heuristics available that are specifically geared to the problem. In particular, the main model selection problem for a profile HMM is to choose the number of match states, and this can be approached by computing Viterbi paths for training sequences and counting the number of insert and delete states that are used (Durbin et al., 1998). Overusage or underusage of insert and delete states suggests that the number of match states should be adjusted.

While we have emphasized the ability of the profile HMM to form alignments, it is also important to note that the HMM machinery allows sequences to be compared without being aligned. This is a natural approach to sequence comparison—rather than picking a specific alignment and then comparing aligned sequences, one would like to average over all possible alignments. This can be readily achieved by fitting a profile HMM based on one sequence, and then computing the likelihood of other sequences under the estimated model.

This has important applications to the representation and search of protein databases. In particular, a library of profile HMMs known as Pfam has been built to represent protein families (Bateman et al., 2000).[8] Given a new protein, this database can be searched by computing likelihoods of the new protein under all of the available models and taking the maximum. This approach is slower than competing methods such as BLAST, but it is more sensitive—it can find homologies that are more remote.

[8] Pfam actually focuses on *protein domains*, which are smaller units that are often associated with well-determined functional roles. The same protein domain can be contained in multiple proteins, and any given protein may contain many domains.

23.7 Pedigrees

While phylogenies attempt to model relationships among the instances of a single gene as found in different species across evolutionary time, *pedigrees* are aimed at a finer level of granularity. A pedigree displays the parent-child relationships within a group of organisms in a single species, and attempts to account for the presence of variants of a gene as they flow through the population. A *multilocus pedigree* is a pedigree which accounts for the flow of multiple genes. Multilocus pedigrees turn out to be a special case of the factorial hidden Markov models that we discussed in Section **??**.

23.7.1 Basic concepts in genetics

We once again need to review some basic biology. In particular, we need to take into account the fact that there are two copies of each gene in a genome, and to discuss the way that genes are transmitted from parents to children.

The genome of an organism is broken into a set of *chromosomes*. In the case of the human genome there are 46 chromosomes. Arrayed along each *chromosome* are a set of *loci*, which correspond to *genes* or other markers. Chromosomes occur in pairs (for all but the X and Y chromosomes, which we will not discuss), and the loci in a given pair of chromosomes can be matched up. Thus a given individual has two copies of each gene. These two copies are not entirely identical, however. In particular, each gene occurs in one of several variant forms—*alleles*—in the population, and any given individual may possess any two of these alleles. If the two alleles happen to be identical, the individual is said to be *homozygous* at that locus. Otherwise, the individual is *heterozygous* at that locus.

The full set of pairs of alleles for a given individual is referred to as the *genotype* of that individual. Given the genotype, there is a (generally stochastic) mapping to the *phenotype*—a set of observable traits. In the simplest case a trait is determined by a single pair of alleles, but this is not the general case, and our modeling formalism does not require this assumption.

In the process of *meiosis*, one of the alleles in the pair at each locus is selected and transmitted to the offspring. The offspring receives one allele at each locus from his or her father, and the other allele at that locus from his or her mother. In the simplest case, the way that this works is that one of the two chromosomes in each pair is selected, and all of the alleles on that chromosome are transmitted. More generally, chromosomes can undergo *recombination*, such that the transmitted chromosome contains segments from both of the two original chromosomes. Alleles that appeared on separate chromosomes in the parent can thus appear on the same chromosome in the child. However, the closer that two alleles lie along a chromosome, the less likely they are to be split up by recombination, and the more likely that they will be transmitted together.

23.7.2 A single locus

Let us now define a graphical model that captures these phenomena. We begin by considering a single locus.

The *genotype* of organism i consists in two alleles at the locus of interest. We denote these genotype variables as $G_{f,i}$ and $G_{m,i}$, where the subscript f refers to the paternal allele (the allele

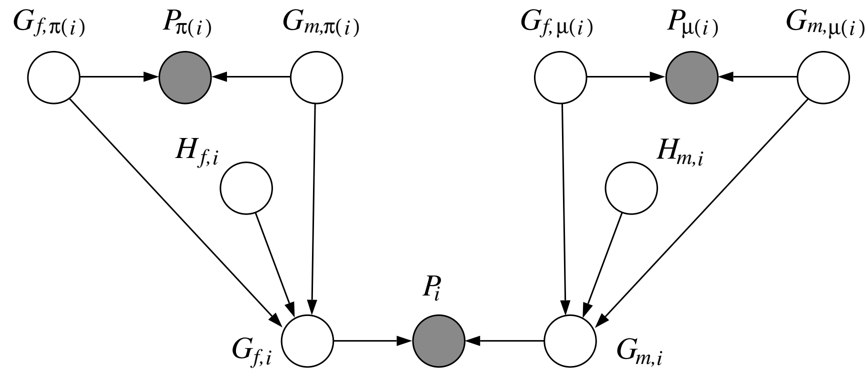

Figure 23.9: A pedigree for a given locus on three organisms—a child i, the father $\pi(i)$ and the mother $\mu(i)$. The variables G, H and P encode the values of the genotype, haplotype and phenotype, respectively, for each of the three organisms.

transmitted from the father) and where the subscript m refers to the maternal allele (the allele transmitted from the mother). Let $\pi(i)$ denote the father of i and let $\mu(i)$ denote the mother. The father has alleles $G_{f,\pi(i)}$ and $G_{m,\pi(i)}$, where the f and m refer to grandfather and grandmother, respectively. Similarly, the mother has alleles $G_{f,\mu(i)}$ and $G_{m,\mu(i)}$.

The probabilistic relationship between these variables is given by Mendel's first law. That is, the child's paternal allele $G_{f,i}$ is equal to $G_{f,\pi(i)}$ or $G_{m,\pi(i)}$ with equal probability, and its maternal allele $G_{m,i}$ is equal to $G_{f,\mu(i)}$ or $G_{m,\mu(i)}$ with equal probability. Mendel's first law derives from an assumption that during meiosis there is an equal probability of picking each of the two chromosomes,

We will find it convenient to turn the choice of chromosome into an explicit "switching" variable. Let $H_{f,i}$ denote a binary *haplotype* variable. This variable is is equal to one if organism i receives the father's paternal allele (i.e., the grandfather's allele), and is equal to zero if organism i receives the father's maternal allele (i.e., the grandmother's allele). Similarly, let $H_{m,i}$ denote the corresponding binary haplotype variable on the mother's side. These haplotype variables are given marginal distributions of $(0.5, 0.5)$ for each of their two states. Conditional on the parent genotypes and the haplotype variables, the child genotype is a deterministic function.

Finally, let P_i denote the phenotype variable corresponding to the genotype at locus i. For simplicity we assume that the phenotype is determined only by the genotype at locus i, but it will be clear how to remove this assumption when we consider multilocus models.

The relationships between these variables are summarized in the graphical model fragment in Figure 23.9. Note the lack of parent nodes for the haplotype variables, and the relationships between parental genotypes and the child genotypes. Note also that we assume that only the phenotype is known; all other variables are latent.

The graph in Figure 23.9 is a fragment of a *pedigree*. In general, pedigrees involve many organisms, and the graphical topology is not a tree.

We have already discussed most of the ingredients for the parameterization of the pedigree. Haplotype variables have no parents, and their marginal is uniform over their two states. The genotype variables are a deterministic function of their parents as discussed. Some of the genotype variables have no parents; these correspond to *founder* organisms. The marginal probabilities of the alleles for these genotypes can be set equal to values derived from population genetics theory, or can be treated as parameters to be estimated from data. Finally, the link from G variables to P variables is referred to as the *penetrance* function. Its form is gene-specific and depends on potentially complex interactions between the genome and the environment; it is beyond our scope.

The inference problems of interest are as usual the computations of marginal probabilities of various nodes in the graph and the computation of the likelihood. In particular, we might observe a disease phenotype at some node in the graph, and want to compute probabilities of disease at other nodes in the graph. The calculation of the likelihood can be solved via the elimination algorithm, and it is of historical interest to note that one of the first published appearances of the elimination algorithm for general graphs was in fact in the setting of the calculation of likelihoods on pedigrees Cannings et al. (1978). As for the computation of marginal probabilities, this problem can be solved in principal by the junction tree algorithm, but the presence of loops at many scales in pedigrees generally renders the junction tree algorithm infeasible, and approximate inference techniques are required.

23.7.3 Multiple loci

We now consider the case of multiple loci. The new issue that we must face in this setting is the possibility of recombination.

We index all variables with a superscript n denoting the locus. Thus, we have genotype variables $G_{f,i}^{(n)}$ for the paternal allele at the nth locus in the ith organism, and $G_{m,i}^{(n)}$ for the maternal allele at that locus. Let $H_{f,i}^{(n)}$ denote the paternal haplotype variable, and let $H_{m,i}^{(n)}$ denote the maternal haplotype variable. Let $P_i^{(n)}$ denote the corresponding phenotype variable.

To accommodate the possibility of recombination, we connect the haplotype variables at different loci. The basic idea is that if the value of the haplotype variable is equal at two neighboring loci, then there is no recombination between these loci.[9] If one haplotype variable is equal to one and its neighbor is a zero, then there is a recombination. Making the assumption that there is no interaction between recombination events (an assumption which is not entirely faithful to the biology), we obtain a two-state Markov chain linking the haplotype variables. The chain is non-homogeneous, and its parameters can be interpreted directly in terms of the probabilities of recombination or no recombination in the corresponding region of the chromosome. Estimating these parameters from data yields an estimate of *genetic distance* between loci, and yields *genetic maps* of the chromosomes.

A graphical model displaying this Markovian assumption is shown in Figure 23.10. In this figure, we have suppressed the detailed interactions among the genotype and phenotype variables at each locus, replacing these collections of nodes with ovals.

In general there is one Markov chain for each pair of chromosomes and for each organism. It is important to note that there are no couplings between these different Markov chains. This is

[9] Actually, haplotype variables being equal mean that there are an even number of recombination events between the corresponding loci. If the loci are reasonably close together, however, there is little likelihood of two or more recombination events.

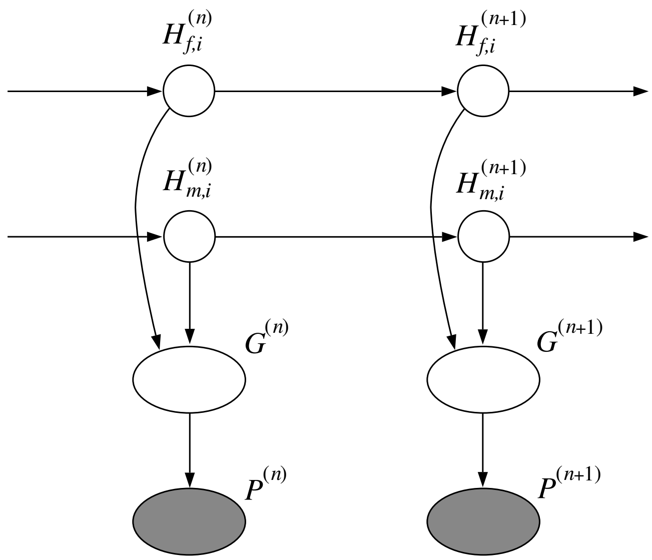

Figure 23.10: A representation of a fragment of a multilocus pedigree for one organism and two loci. The figure is obtained from the pedigree diagram in Figure 23.9 by grouping all of the genotype variables at locus n into a single oval denoted $G^{(n)}$, grouping all of the phenotype variables into a single oval denoted $P^{(n)}$, making copies of the haplotype variables $H_{m,i}$ and $H_{f,i}$ for the two loci n and $n+1$, and connecting the haplotype variables between the two loci.

not an approximation, but is a reflection of the biology: Recombination events between different chromosome pairs are unrelated, and recombination events among different organisms are unrelated.

Overall, the structure is that of a *factorial hidden Markov model.* The model can also be compared to the hidden Markov phylogeny. In that case, conditional on the state of an HMM, the emission distribution is a phylogeny. In the current model, conditional on the state configuration of a factorial HMM, the emission distribution is a pedigree.

Classical algorithms for inference on multilocus pedigrees are variants of the elimination algorithm on this factorial HMM, and correspond to different choices of elimination order (Lander and Green, 1987, Elston and Stewart, 1971). While these algorithms are viable for small problems, exact inference is intractable for general multilocus pedigrees. Indeed, focusing only on the haplotype variables, it can be verified that the treewidth is bounded below by the number of organisms, and thus the computational complexity is exponential in the number of organisms. The graphical model interpretation that we have presented here has been discussed by Thomas et al. (2000) and Fishelson and Geiger (2002), who have presented approximate inference algorithms. In particular, Thomas et al. (2000) have proposed a blocking Gibbs sampler that takes advantage of the graphical structure in Figure 23.10.

One of the most important applications of multilocus pedigrees is to the problem of *linkage analysis.* Suppose that we are interested in locating the genetic cause for a particular disease. Simplifying the issue, let us suppose that the disease is caused by a single gene. The problem is to find the gene. We can approach the problem by gathering data from a multilocus pedigree on L

loci, where the loci may correspond to genes whose location in the genome is known, or to other "markers." Along with these data we also record whether each organism has the disease or not. We then calculate $L+1$ different likelihoods, corresponding to $L+1$ models in which the disease is placed as a hypothetical locus between each of the measured loci. Selecting the model with the largest likelihood gives an indication as to the genomic location of the disease-causing gene.

23.8 Other graphical models in bioinformatics

In this section we present a brief discussion of some other areas in bioinformatics in which graphical models have proved useful.

[This section is not yet complete].

23.8.1 Motif models

We first turn to the problem of modeling *DNA motifs*. This topic is a large and important one, and will require some additional biological background.

Our emphasis thus far has been on the structure of various molecules such as DNA and proteins, and we have not yet discussed the dynamical processes that link these molecules. To understand the concept of a DNA motif, and why it is important, we need to discuss some of these dynamical processes.

As we mentioned in Section 23.1, the transcription of a gene is triggered by the presence of certain *transcription factors*. Transcription factors are proteins, and their role is to bind to the DNA in the vicinity of the transcription start site of a gene and thereby trigger the RNA polymerase to transcribe the DNA into messenger RNA. The *promoter region* is a region upstream of the transcription start site that contains the binding sites of certain essential transcription factors, in whose absence the gene cannot be transcribed. More generally, there are *regulatory regions* upstream, downstream, and even in the introns of the gene, where transcription factors and regulatory proteins can bind, and thereby enhance or diminish the rate of transcription. The overall process is a complex one, as suggested by the drawing shown in Figure 23.11. We see that the DNA bends into a three-dimensional shape due to the presence of the transcription factors, a shape which may enhance the access of the RNA polymerase to the gene.

Transcription factors are proteins and are thus the product of the transcription and translation of other genes. This implies that genes are linked in a dynamical network in which the some genes regulate the transcription of other genes. This network also interacts with the outside world, via mechanisms that influence the numbers or the binding affinity of transcription factors. Substances such as hormones or small molecules may diffuse across the cell membrane, bind to transcription factors and enable or disable them. These substances may also bind directly to the regulatory regions and prevent transcription factors from binding.

A full understanding of the regulatory circuitry of a cell is one of the major goals of molecular biology, requiring sustained research in biochemistry, cellular biology and genetics. A more modest goal is to begin to understand the set of transcription factors that influence the transcription of any given gene, and to determine where these transcription factors bind. We arrive at the problem

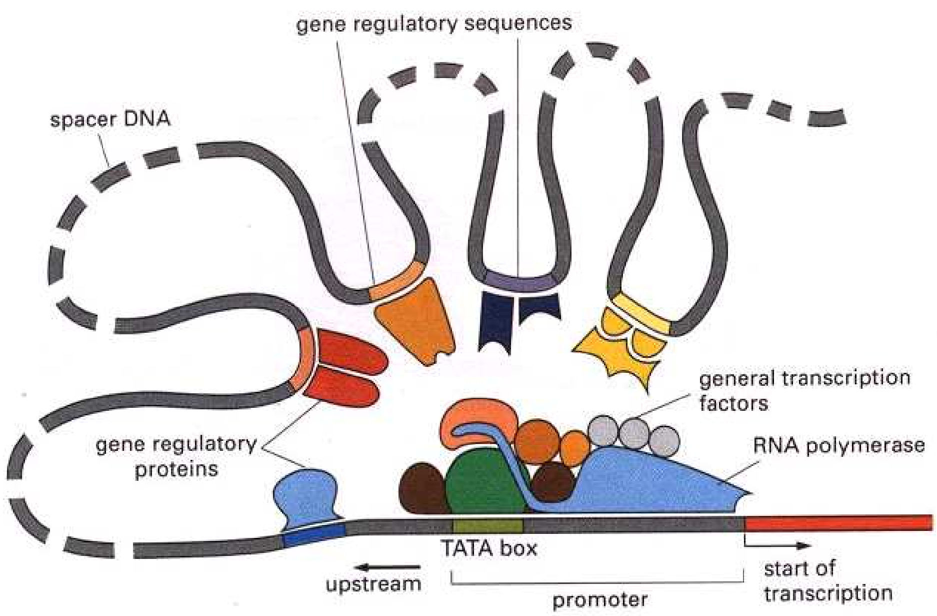

Figure 23.11: A depiction of the binding of transcription factors and other regulatory proteins to the regulatory regions associated with a gene. DNA motifs are referred to as “gene regulatory sequences” in the diagram.

of modeling DNA motifs. DNA motifs are short segments of DNA that are the binding sites of transcription factors. Physically, they are the substrate upon which the transcription factor binds. Statistically, the binding sites associated with a given transcription factor form a set of short DNA patterns that recur in the regulatory regions of multiple genes.

[A short discussion of motif finders to be provided here.]

23.8.2 Haplotype clustering and phasing

In Section 23.7.3 we introduced the problem of multilocus pedigree analysis. Given a set of phenotypes, the model that we discussed allows us to infer probabilities for underlying genotypes. Moreover, we also included variables representing the *haplotype phase* associated with each individual organism. These variables assigned particular alleles to particular chromosomes, and allowed us to infer locations of recombination events.

There are many other problems associated with the inference of haplotypes in populations. One such problem involves the attempt to infer haplotype phase from a collection of unrelated individuals; thus dispensing with the pedigree. There are two facts that make this a viable enterprise. First, it is an empirical fact about human populations that alleles tend to co-occur in so-called *haplotype blocks* (Gabriel et al., 2002). These are regions of chromosomes that have not yet been broken up by recombination and presumably reflect the particular set of alleles on a chromosome of a relatively recent set of human ancestors. Second, it is possible to measure the genotypes of humans, in the sense that technology exists that can determine which alleles a given individual possesses at each of a set of loci. This technology loses the association of allele to chromosome, however; exactly the haplotype phase. That is, the technology returns an unordered set of alleles at each loci. An interesting problem is to attempt to restore the haplotype phase from a set of genotypes for a set of individuals.

Suppose first that we consider a set of loci which are relatively close together, in particular lying within a single haplotype block. In this case, the problem of inferring haplotypes can be viewed as a finite mixture problem. Let $\mathcal{H}$ denote the set of all possible haplotypes associated with a given block (a set of cardinality 2^k in the case of binary polymorphisms, where k is the number of heterozygous loci in the block), the probability of a genotype is given by:

$$p(g) = \sum_{h_1, h_2 \in \mathcal{H}} p(h_1, h_2) 1(h_1 \oplus h_2 = g), \tag{23.6}$$

where $1(h_1 \oplus h_2 = g)$ is the indicator function of the event that haplotypes h_1 and h_2 are consistent with g. Under the assumption that the birth of a new individual can be modeled as the independent selection of a pair of haplotypes from the population, the mixing proportion $p(h_1, h_2)$ factors as $p(h_1)p(h_2)$.

A number of authors have presented estimation algorithms for this mixture model, including EM (Excoffier and Slatkin, 1995) and Gibbs sampling (Stephens et al., 2001) algorithms. Xing et al. (2003) have discussed a variant in which a Dirichlet process is used as a nonparametric prior that allows the number of mixture components to vary.

It is also possible to consider models in which longer stretches of DNA are considered, and the haplotype blocks are themselves inferred. Greenspan and Geiger (2003) have presented a graphical model approach to this problem, treating the haplotype block inference problem as a model selection problem.

23.8.3 Transmembrane proteins

Finally, we mention another application of hidden Markov models in bioinformatics—the modeling of transmembrane proteins. These are proteins that are inserted in the membrane and act as gateways for various molecules to pass into and out of the cell. They can change configuration depending on various binding factors, and thus serve as context-dependent gates. Transmembrane proteins have characteristic patterns: (1) they tend to pass through the membrane multiple times, and thus the sequence of amino acids alternates between hydrophilic and hydrophobic amino acids; (2) the cell-interior portion of the protein tends to interact with proteins such as kinases that trigger signaling cascades; and (3) the cell-exterior portion of the protein tends to interact with various molecules such as hormones that arrive from other cells and from the external environment. Krogh et al. (2001) have presented a hidden Markov model (the "TMHMM") whose state space captures these and other signatures of transmembrane proteins.

23.9 Historical remarks and bibliography

Bibliography

Alexandersson, M., Cawley, S., and Pachter, L. (2003). SLAM—cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.*, 13:496–502.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402.

Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Howe, K. L., and Sonnhammer, E. L. (2000). The pfam protein families database. *Nucleic Acids Research*, 28:263–266.

Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78–94.

Cannings, C., Thompson, E. A., and Skolnick, M. H. (1978). Probability functions on complex pedigrees. *Advances in Applied Probability*, 10:26–91.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, UK.

Elston, R. C. and Stewart, J. (1971). A general model for the genetic analysis of pedigree data. *Human Heredity*, 21:523–542.

Excoffier, L. and Slatkin, M. (1995). Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–7.

Felsenstein, J. (1973). Maximum likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics*, 25:471–492.

Felsenstein, J. (2003). *Inferring Phylogenies.* Sinauer Associates, Sunderland, MA.

Felsenstein, J. and Churchill, G. A. (1996). A hidden Markov model approach to variation among sites in rate of evolution. *Mol. Biol. Evol.*, 13(1):93–104.

Fishelson, M. and Geiger, D. (2002). Exact genetic linkage computations for general pedigrees. In *Intelligent Systems for Molecular Biology*, pages 189–198.

Gabriel, S. B. et al. (2002). The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229.

Goldman, N., Thorne, J., and Jones, D. (1996). Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. *Journal of Molecular Biology*, 263:196–208.

Greenspan, D. and Geiger, D. (2003). Model-based inference of haplotype block variation. In *Proceedings of the 7th International Conference on Computational Molecular Biology (RECOMB 2003)*.

Gusfield, D. (1997). *Algorithm on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, Cambridge, UK.

Hasegawa, M., Kishino, H., and Yano, T. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 21:160–174.

Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules. In Munro, H. N., editor, *Mammalian Protein Metabolism*, pages 21–132. Academic Press, New York.

Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580.

Kulp, D., Haussler, D., Reese, M. G., and Eeckman, F. H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. In *Intelligent Systems for Molecular Biology*, pages 134–141. AAAI Press.

Lander, E. S. and Green, P. (1987). Construction of multilocus genetic maps in humans. *Proceedings of the National Academy of Sciences*, 84:2363–2367.

McAuliffe, J. D., Pachter, L., and Jordan, M. I. (2003). Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. Technical Report 647, Department of Statistics, University of California, Berkeley.

Meyer, I. M. and Durbin, R. (2002). Comparative *ab initio* prediction of gene structures using pair HMMs. *Bioinformatics*, 18(10):1309–1318.

Pedersen, J. S. and Hein, J. (2003). Gene finding with a hidden Markov model of genome structure and evolution. *Bioinformatics*, 19:219–227.

Siepel, A. and Haussler, D. (2003). Combining phylogenetic and hidden Markov models in biosequence analysis. In *Proceedings of the Seventh Annual International Conference on Computational Biology*, pages 277–286.

Stephens, M., Smith, N., and Donnelly, P. (2001). A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989.

Thomas, A., Gutin, A., Abkevich, V., and Bansal, A. (2000). Multilocus linkage analysis by blocked Gibbs sampling. *Statistics and Computing*, 10:259–269.

Xing, E. P., Sharan, R., and Jordan, M. I. (2003). Bayesian haplotype inference via the Dirichlet process. Technical Report CSD-03-1275, Division of Computer Science, University of California, Berkeley.

Yang, Z. (1993). Maximum likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular Biology and Evolution*, 10:1396–1401.

Yang, Z. (1995). A space-time process model for the evolution of DNA sequences. *Genetics*, 139:993–1005.