Chapter 9

Gene Prediction

9.1 Introduction

In the 1960s, Charles Yanofsky, Sydney Brenner, and their collaborators showed that a gene and its protein product are colinear structures with direct correlation between triplets of nucleotides in the gene and amino acids in the protein. However, the concept of the gene as a synthetic string of nucleotides did not live long. Overlapping genes and genes-within-genes were discovered in the late 1960s. These studies demonstrated that the computational problem of gene prediction is far from simple. Finally, the discovery of split human genes in 1977 created a computational gene prediction puzzle.

Eukaryotic genomes are larger and more complex than prokaryotic genomes. This does not come as a surprise since one would expect to find more genes in humans than in bacteria. However, the genome size of many eukaryotes does not appear to be related to genetic complexity; for example, the salamander genome is 10 times larger than the human genome. This paradox was resolved by the discovery that eukaryotes contain not only genes but also large amounts of DNA that do not code for any proteins ("junk" DNA). Moreover, most human genes are interrupted by junk DNA and are broken into pieces called exons. The difference in the sizes of the salamander and human genomes thus reflects larger amounts of junk DNA and repeats in the genome of salamander.

Split genes were first discovered in 1977 independently by the laboratories of Phillip Sharp and Richard Roberts during studies of the adenovirus (Berget et al., 1977 [32], Chow et al., 1977 [67]). The discovery was such a surprise that the paper by Richard Roberts' group had an unusually catchy title for the academic *Cell* magazine: "An amazing sequence arrangement at the 5' end of adenovirus 2 messenger RNA." Berget et al., 1977 [32] focused their experiments on an mRNA that encodes a viral protein known as the hexon. To map the hexon mRNA on viral genome, mRNA was hybridized to adenovirus DNA and the hybrid molecules were analyzed by electron microscopy. Strikingly, the mRNA-DNA hybrids formed in this experiment displayed three loop structures, rather than the continuous duplex segment suggested by the classical "continuous gene" model. Further hybridization experiments revealed that the hexon mRNA is built from four separate fragments of the adenovirus genome. These four exons in the adenovirus genome are separated by three "junk" fragments called *introns*. The discovery of split genes (*splicing*) in the adenovirus was quickly followed by evidence that mammalian genes also have split structures (Tilghman et al., 1978 [337]). These experimental studies raised a computational gene prediction problem that is still unsolved: human genes comprise only 3% of the human genome, and no existing *in silico* gene recognition algorithm provides reliable gene recognition.

After a new DNA fragment is sequenced, biologists try to find genes in this fragment. The traditional statistical way to attack this problem has been to look for features that appear frequently in genes and infrequently elsewhere. Many researchers have used a more biologically oriented approach and attempted to recognize the locations of splicing signals at exon-intron junctions. The goal of such an approach is characterization of sites on RNA where proteins and ribonucleoproteins involved in splicing apparatus bind/interact. For example, the dinucleotides *AG* and *GT* on the left and right sides of exons are highly conserved. The simplest way to represent a signal is to give a consensus pattern consisting of the most frequent nucleotide at each position of an alignment of specific signals. Although catalogs of splice sites were compiled in the early 1980s, the consensus patterns are not very reliable for discriminating true sites from pseudosites since they contain no information about nucleotide frequencies at different positions. Ed Trifonov invented an example showing another potential pitfall of consensus:

MELON
MANGO
HONEY
SWEET
COOKY
—
MONEY

The frequency information is captured by *profiles* (or *Position Weight Matrices*) that assign frequency-based scores to each possible nucleotide at each position of the signal. Unfortunately, using profiles for splice site prediction has had limited success, probably due to cooperation between multiple binding molecules. Attempts to improve the accuracy of gene prediction led to applications of neural networks and Hidden Markov Models for gene finding.

Large-scale sequencing projects have motivated the need for a new generation of algorithms for gene recognition. The similarity-based approach to gene prediction is based on the observation that a newly sequenced gene has a good chance of having an already known relative in the database (Bork and Gibson, 1996 [41]). The flood of new sequencing data will soon make this chance even greater. As a result, the trend in gene prediction in the late 1990s shifted from statistics-based approaches to similarity-based and EST-based algorithms. In particular, Gelfand et al., 1996 [125] proposed a combinatorial approach to gene prediction, that uses related proteins to derive the exon-intron structure. Instead of employing statistical properties of exons, this method attempts to solve a combinatorial puzzle: to find a set of substrings in a genomic sequence whose concatenation (splicing) fits one of the known proteins.

After predictions are made, biologists attempt to experimentally verify them. This verification usually amounts to full-length mRNA sequencing. Since this process is rather time-consuming, *in silico* predictions find their way into databases and frequently lead to annotation errors. We can only guess the amount of incorrectly annotated sequences in GenBank, but it is clear that the number of genes that have been annotated without full-length mRNA data (and therefore are potentially erroneous) may be large. The problems of developing an "annotation-ready" gene prediction algorithm and correcting these errors remain open.

9.2 Statistical Approach to Gene Prediction

The simplest way to detect potential coding regions is to look at *Open Reading Frames (ORFs)*. An ORF is a sequence of codons in DNA that starts with a Start codon, ends with a Stop codon, and has no other Stop codons inside. One expects to find frequent Stop codons in non-coding DNA simply because 3 of 64 possible codons are translation terminators. The average distance between Stop codons in "random" DNA is $\frac{64}{3} \approx 21$, much smaller than the number of codons in an average protein (roughly 300). Therefore, long ORFs point out potential genes (Fickett, 1996 [105]), although they fail to detect short genes or genes with short exons.

Many gene prediction algorithms rely on recognizing the diffuse regularities in protein coding regions, such as bias in *codon usage*. Codon usage is a 64-mer vector giving the frequencies of each of 64 possible *codons* (triples of nucleotides) in a window. Codon usage vectors differ between coding and non-coding windows, thus enabling one to use this measure for gene prediction (Fickett, 1982 [104], Staden and McLachlan, 1982 [327]). Gribskov et al., 1984 [138] use a likelihood ratio approach to compute the conditional probabilities of the DNA sequence in a window under a coding and under a non-coding random sequence hypothesis. When the window slides along DNA, genes are often revealed as peaks of the likelihood ratio plots. A better coding sensor is the *in-frame hexamer count*, which is similar to three fifth-order Markov models (Borodovsky and McIninch, 1993 [42]). Fickett and Tung, 1992 [106] evaluated many such coding measures and came to the conclusion that they give a rather low-resolution picture of coding-region boundaries, with many false positive and false negative assignments. Moreover, application of these techniques to eukaryotes is complicated by the exon-intron structure. The average length of exons in vertebrates is 130 bp, and thus exons are often too short to produce peaks in the sliding window plot.

Codon usage, amino acid usage, periodicities in coding regions and other statistical parameters (see Gelfand, 1995 [123] for a review) probably have nothing in common with the way the splicing machinery recognizes exons. Many researchers have used a more biologically oriented approach and attempted to recognize the locations of splicing signals at exon-intron junctions (Brunak et al., 1991 [50]). There exists a (weakly) conserved sequence of eight nucleotides at the boundary of an exon and an intron (5' or *donor* splice site) and a sequence of four nucleotides at the boundary of intron and exon (3' or *acceptor* splice site). Unfortunately, profiles for splice site prediction have had limited success, probably due to cooperation between multiple binding molecules. Profiles are equivalent to a simple type of neural network called perceptron. More complicated neural networks (Uberbacher and Mural, 1991 [339]) and Hidden Markov Models (Krogh et al., 1994 [209], Burge and Karlin, 1997 [54]) capture the statistical dependencies between sites and improve the quality of predictions.

Many researchers have attempted to combine coding region and splicing signal predictions into a signal framework. For example, a splice site prediction is more believable if signs of a coding region appear on one side of the site but not the other. Because of the limitations of individual statistics, several groups have developed gene prediction algorithms that combine multiple pieces of evidence into a single framework (Nakata et al., 1985 [249], Gelfand, 1990 [121], Guigo et al., 1992 [142], Snyder and Stormo, 1993 [321]). Practically all of the existing statistics are used in the Hidden Markov Model framework of GENSCAN (Burge and Karlin, 1997 [54]). This algorithm not only merges splicing site, promoter, polyadenylation site, and coding region statistics, but also takes into account their non-homogeneity. This has allowed the authors to exceed the milestone of 90% accuracy for statistical gene predictions. However, the accuracy decreases significantly for genes with many short exons or with unusual codon usage.

9.3 Similarity-Based Approach to Gene Prediction

The idea of a similarity-based approach to gene detection was first stated in Gish and States, 1993 [129]. Although similarity search was in use for gene *detection* (i.e., answering the question of whether a gene is present in a given DNA fragment) for a long time, the potential of similarity search for gene *prediction* (i.e., not only for detection but for detailed prediction of the exon-intron structure as well) remained largely unexplored until the mid-1990s. Snyder and Stormo, 1995 [322] and Searls and Murphy, 1995 [313] made the first attempts to incorporate similarity analysis into gene prediction algorithms. However, the computational complexity of exploring all exon assemblies on top of sequence alignment algorithms is rather high.

Gelfand et al., 1996 [125] proposed a spliced alignment approach to the exon assembly problem, that uses related proteins to derive the exon-intron structure. Figure 9.1a illustrates the spliced alignment problem for the "genomic" sequence

It was brilliant thrilling morning and the slimy hellish lithe doves

gyrated and gambled nimbly in the waves

whose different blocks make up the famous Lewis Carroll line:

't was brillig, and the slithy toves did gyre and gimble in the wabe

The Gelfand et al., 1996 [125] approach is based on the following idea (illustrated by Oksana Khleborodova). Given a genomic sequence (Figure 9.2), they first find a set of *candidate blocks* that contains all *true* exons (Figure 9.3). This can be done by selecting all blocks between potential *acceptor* and *donor* sites (i.e., between AG and GT dinucleotides) with further *filtering* of this set (in a way that does not lose the actual exons). The resulting set of blocks can contain many false exons, of course, and currently it is impossible to distinguish all actual exons from this set by a statistical procedure. Instead of trying to find the actual exons, Gelfand et al., 1996 [125] select a related *target* protein in GenBank (Figure 9.4) and explore all possible block assemblies with the goal of finding an assembly with the highest similarity score to the target protein (Figure 9.5). The number of different block assemblies is huge (Figures 9.6, 9.7, and 9.8), but the *spliced alignment* algorithm, which is the key ingredient of the method, scans all of them in polynomial time (Figure 9.9).

9.4 Spliced Alignment

Let $G = g_1 \ldots g_n$ be a string, and let $B = g_i \ldots g_j$ and $B' = g_{i'} \ldots g_{j'}$ be substrings of G. We write $B \prec B'$ if $j < i'$, i.e., if B ends before B' starts. A sequence $\Gamma = (B_1, \ldots, B_p)$ of substrings of G is a *chain* if $B_1 \prec B_2 \prec \cdots \prec B_p$. We denote the *concatenation* of strings from the chain Γ as $\Gamma^* = B_1 * B_2 * \ldots * B_p$. Given two strings G and T, $s(G, T)$ denotes the score of the *optimal alignment* between G and T.

Let $G = g_1 \ldots g_n$ be a string called *genomic sequence*, $T = t_1 \ldots t_m$ be a string called *target sequence*, and $\mathcal{B} = \{B_1, \ldots B_b\}$ be a set of substrings of G called *blocks*. Given G, T, and $\mathcal{B}$, the *spliced alignment problem* is to find a chain Γ of strings from $\mathcal{B}$ such that the score $s(\Gamma^*, T)$ of the alignment between

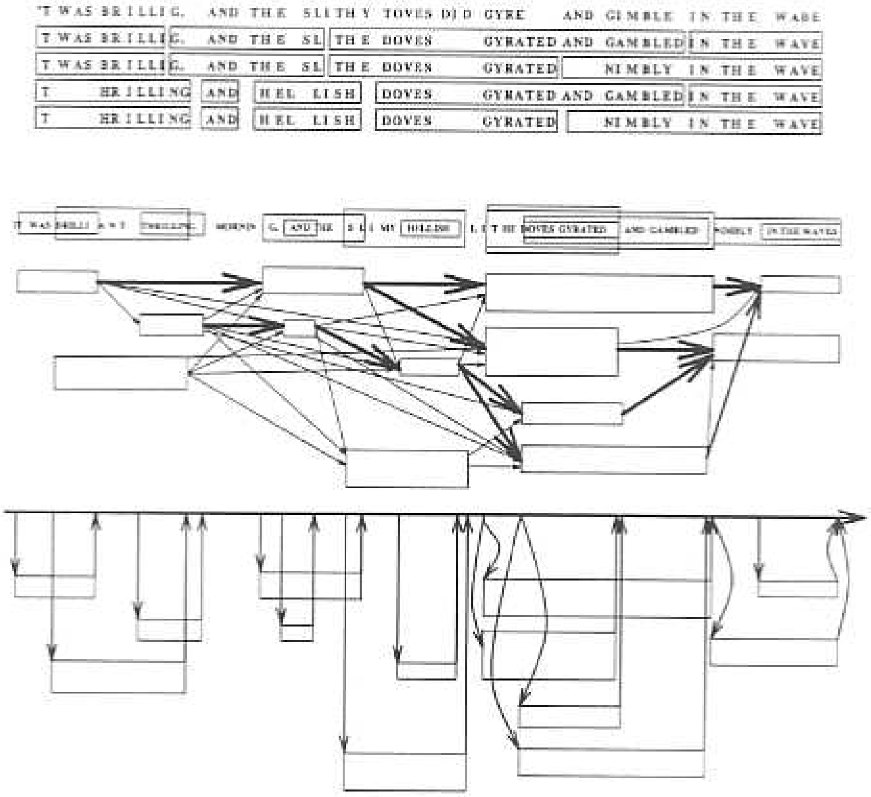
'T WAS BRILLIG, AND THE SLITHY TOVES DID GYRE AND GIMBLE IN THE WABE
T WAS BRILLIG, AND THE SL THE DOVES GYRATED AND GAMBLED IN THE WAVE
T WAS BRILLIG, AND THE SL THE DOVES GYRATED NIMBLY IN THE WAVE
T HRILLING AND HEL LISH DOVES GYRATED AND GAMBLED IN THE WAVE
T HRILLING AND HEL LISH DOVES GYRATED NIMBLY IN THE WAVE

Figure 9.1: Spliced alignment problem: a) block assemblies with the best fit to Lewis Carroll's line, b) corresponding alignment network, and c) equivalent transformation of the alignment network.

the concatenation of these strings and the target sequence is maximum among all chains of blocks from $\mathcal{B}$.

A naive approach to the spliced alignment problem is to detect all relatively high similarities between each block and the target sequence and to assemble these similarities into an optimal subset of compatible similar fragments. The shortcoming of this approach is that the number of blocks is typically very large and the endpoints of the similarity domains are not well defined.

Gelfand et al., 1996 [125] reduced the exon assembly problem to the search of a path in a directed graph (Figure 9.1b). Vertices in this graph correspond to the blocks, edges correspond to potential transitions between blocks, and the path weight is defined as the weight of the optimal alignment between the concatenated

Figure 9.2: Studying genomic sequence.

blocks of this path and the target sequence. Note that the exon assembly problem is different from the standard minimum path problem (the weights of vertices and edges in the graph are not even defined).

Let $B_k = g_m \dots g_i \dots g_l$ be a substring of G containing a position i. Define the *i-prefix* of B_k as $B_k(i) = g_m \dots g_i$. For a block $B_k = g_m \dots g_l$, let $first(k) = m$, $last(k) = l$, and $size(k) = l - m + 1$. Let $B(i) = \{k : \ last(k) < i\}$ be the set of blocks ending (strictly) before position i in G. Let $\Gamma = (B_1, \dots, B_k, \dots, B_t)$ be a chain such that some block B_k contains position i. Define $\Gamma^*(i)$ as a string

Figure 9.3: Filtering candidate exons.

$\Gamma^*(i) = B_1 * B_2 * \ldots * B_k(i)$. Let

$$S(i, j, k) = \max_{\text{all chains } \Gamma \text{ containing block } B_k} s(\Gamma^*(i), T(j)).$$

The following recurrence computes $S(i, j, k)$ for $1 \le i \le n$, $1 \le j \le m$, and $1 \le k \le b$. For the sake of simplicity we consider sequence alignment with *linear* gap penalties and define $\delta(x, y)$ as a similarity score for every pair of amino acids x and y and δ_{indel} as a penalty for insertion or deletion of amino acids.

Figure 9.4: Finding a target protein.

$$S(i,j,k) = \max \begin{cases} S(i-1,j-1,k) + \delta(g_i,t_j), & \text{if } i \neq first(k) \\ S(i-1,j,k) + \delta_{indel}, & \text{if } i \neq first(k) \\ \max_{l \in B(first(k))} S(last(l),j-1,l) + \delta(g_i,t_j), & \text{if } i = first(k) \\ \max_{l \in B(first(k))} S(last(l),j,l) + \delta_{indel}, & \text{if } i = first(k) \\ S(i,j-1,k) + \delta_{indel} \end{cases} \tag{9.1}$$

After computing the 3-dimensional table $S(i,j,k)$, the score of the optimal spliced alignment is

$$\max_k S(last(k),m,k).$$

Figure 9.5: Using the target protein as a template for exon assembly.

The spliced alignment problem also can be formulated as a *network* alignment problem (Kruskal and Sankoff, 1983 [211]). In this formulation, each block B_k corresponds to a path of length $size(k)$ between vertices $first(k)$ and $last(k)$, and paths corresponding to blocks B_k and B_t are joined by an edge $(last(k), first(t))$ if $B_k \prec B_t$ (Figure 9.1b). The network alignment problem is to find a path in the network with the best alignment to the target sequence.

Gelfand et al., 1996 [125] reduced the number of edges in the spliced alignment graph by making equivalent transformations of the described network, leading to a reduction in time and space. Define

$$P(i,j) = \max_{l \in B(i)} S(last(l), j, l).$$

Figure 9.6: Assembling.

Then (9.1) can be rewritten as

$$S(i,j,k) = \max \begin{cases} S(i-1,j-1,k) + \delta(g_i,t_j), & \text{if } i \neq first(k) \\ S(i-1,j,k) + \delta_{indel}, & \text{if } i \neq first(k) \\ P(first(k),j-1) + \delta(g_i,t_j), & \text{if } i = first(k) \\ P(first(k),j) + \delta_{indel}, & \text{if } i = first(k) \\ S(i,j-1,k) + \delta_{indel} & \end{cases} \tag{9.2}$$

where

$$P(i,j) = \max \begin{cases} P(i-1,j) \\ \max_{k:\, last(k)=i-1} S(i-1,j,k) \end{cases} \tag{9.3}$$

Figure 9.7: And assembling...

The network corresponding to (9.2) and (9.3) has a significantly smaller number of edges (Figure 9.1c), thus leading to a practical implementation of the spliced alignment algorithm.

The simplest approach to the construction of blocks B is to generate all fragments between potential splicing sites represented by AG (acceptor site) and GT (donor site), with the exception of blocks with stop codons in all three frames. However, this approach creates a problem since it generates many short blocks. Experiments with the spliced alignment algorithm have revealed that incorrect predictions for distant targets are frequently associated with the *mosaic effect* caused by very short potential exons. The problem is that these short exons can be easily combined together to fit any target protein. It is easier to "make up" a given sen-

Figure 9.8: And assembling...............

tence from a thousand random short strings than from the same number of longer strings. For example, with high probability, the phrase "filtration of candidate exons" can be made up from a sample of a thousand random two-letter strings ("fi," "lt," "ra," etc. are likely to be present in this sample). The probability that the same phrase can be made up from a sample of the same number of random five-letter strings is close to zero (even finding a string "filtr" in this sample is unlikely). This observation explains the mosaic effect: if the number of short blocks is high, chains of these blocks can replace actual exons in spliced alignments, thus leading to predictions with an unusually large number of short exons. To avoid the mosaic effect, the candidate exons are subjected to some (weak) filtering procedure; for example, only exons with high coding potential may be retained.

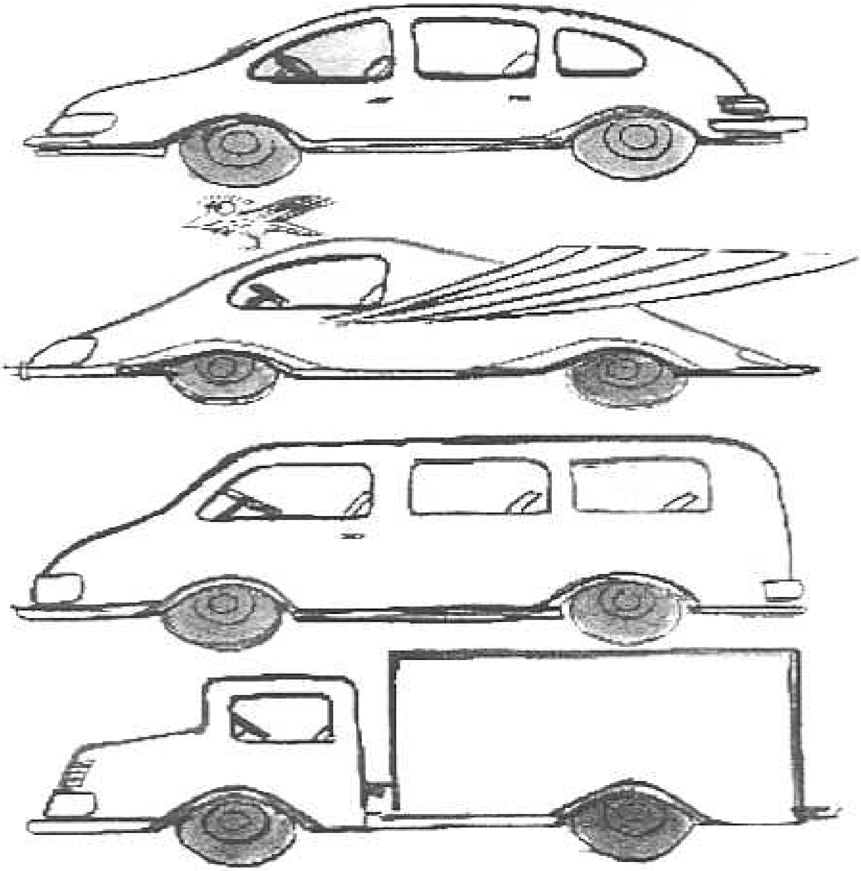

Figure 9.9: Selecting the best exon assembly.

After the optimal block assembly is found, the hope is that it represents the correct exon-intron structure. This is almost guaranteed if a protein sufficiently similar to the one encoded in the analyzed fragment is available: 99% correlation with the actual genes can be obtained from targets with distances of up to 100 PAM (40% similarity). The spliced alignment algorithm provides very accurate predictions if even a distantly related protein is available: predictions at 160 PAM (25% similarity) are still reliable (95% correlation). If a related mammalian protein for an analyzed human gene is known, the accuracy of gene predictions in this fragment is as high as 97% – 99%, and it is 95%, 93%, and 91% for related plant, fungal, and prokaryotic proteins, respectively (Mironov et al., 1998 [242]). Further progress in gene prediction has been achieved by using EST data for similarity-based gene

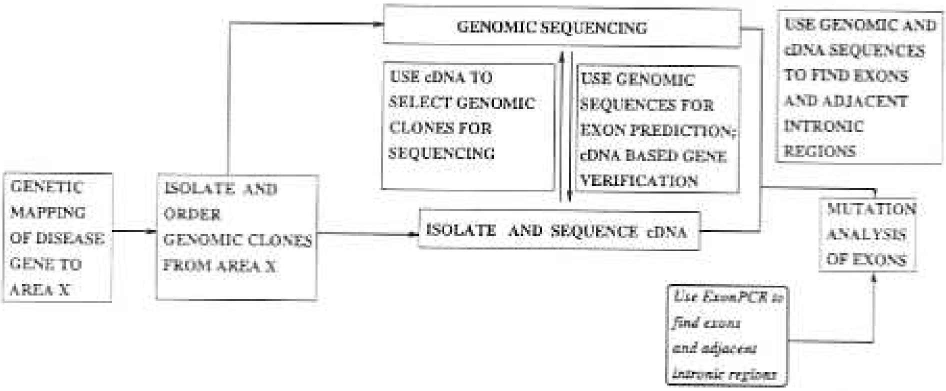
GENOMIC SEQUENCING
USE cDNA TO SELECT GENOMIC CLONES FOR SEQUENCING
USE GENOMIC SEQUENCES FOR EXON PREDICTION; cDNA BASED GENE VERIFICATION
USE GENOMIC AND cDNA SEQUENCES TO FIND EXONS AND ADJACENT INTRONIC REGIONS
GENETIC MAPPING OF DISEASE GENE TO AREA X
ISOLATE AND ORDER GENOMIC CLONES FROM AREA X
ISOLATE AND SEQUENCE cDNA
MUTATION ANALYSIS OF EXONS
Use ExonPCR to find exons and adjacent intronic regions

Figure 9.10: Positional cloning and ExonPCR.

prediction. In particular, using EST assemblies, Mironov et al., 1999 [240] found a large number of alternatively spliced genes.

9.5 Reverse Gene Finding and Locating Exons in cDNA

Gene finding often follows the *positional cloning* paradigm:

genomic DNA sequencing → exon structure → mRNA → protein

In positional cloning projects, genomic sequences are the primary sources of information for gene prediction, mutation detection, and further search for disease-causing genes. The shift from positional cloning to the *candidate gene library* paradigm is reversing the traditional gene-finding pathway into the following:

protein/mRNA → exon structure → limited genomic DNA sequencing

Consequently, modern gene discovery efforts are shifting from single-gene positional cloning to analysis of polygenic diseases with candidate gene libraries of hundred(s) of genes. The genes forming a candidate gene library may come from different sources, e.g., expression analysis, antibody screening, proteomics, etc. Of course, hundred(s) of positional cloning efforts are too costly to be practical.

A positional cloning approach to finding a gene responsible for a disease starts with genetic mapping and proceeds to the detection of disease-related mutations. A multitude of steps are required that include genomic cloning of large DNA fragments, screening cDNA libraries, cDNA isolation, subcloning of the large genomic clones for sequencing, etc. (Figure 9.10). In many gene-hunting efforts, the major motivation for the genomic subcloning and sequencing steps is to determine the gene's exon-intron boundaries. This step is often critical to searches for mutations (or polymorphisms) associated with a disease gene. It requires the design of intronic PCR primers flanking each exon. Traditionally, the exon boundaries are obtained by comparing the cDNA and genomic sequences. The whole process can be time-consuming and may involve multiple subcloning steps and extensive sequencing.

ExonPCR (Xu et al., 1998 [372]) is an alternative experimental protocol that explores the "reverse" gene-finding pathway and provides a fast transition from finding cDNA to mutation detection (Figure 9.10). ExonPCR finds the "hidden" exon boundaries in cDNA (rather than in genomic DNA) and does not require sequencing of genomic clones. In the first step, ExonPCR locates the approximate positions of exon boundaries in cDNA by PCR on genomic DNA using primers designed from the cDNA sequence. The second step is to carry out ligation-mediated PCR to find the flanking intronic regions. As a consequence, the DNA sequencing effort can be vastly reduced.

The computational approaches to finding exon boundaries in cDNA (Gelfand, 1992 [122], Solovyev et al., 1994 [323]) explored *splicing shadows* (i.e., parts of the splicing signals present in exons). However, since the splicing shadow signals are very weak, the corresponding predictions are unreliable. ExonPCR is an experimental approach to finding exon boundaries in cDNA that uses PCR primers in a series of adaptive rounds. Primers are designed from the cDNA sequence and used to amplify genomic DNA. Each pair of primers serves as a query asking the question whether, in the genomic DNA, there exists an intron or introns between the primer sequences. The answer to this query is provided by comparison of the length of PCR products in the cDNA and genomic DNA. If these lengths coincide, the primers belong to the same exon; otherwise, there exists an exon boundary between the corresponding primers. Each pair of primers gives a yes/no answer without revealing the exact positions of exon boundaries. The goal is to find the positions of exon boundaries and to minimize both the number of primers and the number of rounds. Different types of strategies may be used, and the problem is similar to the "Twenty Questions" game with genes. The difference with a parlor game is that genes have a "no answer" option and sometimes may give a false answer and restrict the types of possible queries. This is similar to the "Twenty Questions Game with a Liar" (Lawler and Sarkissian, 1995 [216]) but involves many additional constraints including lower and upper bounds on the length of queries (distance between PCR primers).

ExonPCR attempts to devise a strategy that minimizes the total number of PCR primers (to reduce cost) and at the same time minimizes the number of required rounds of PCR experiments (to reduce time). However, these goals conflict with each other. A minimum number of primer pairs is achieved in a sequential "dichotomy" protocol where only one primer pair is designed in every round based on the results of earlier rounds of experiments. This strategy is unrealistic since it leads to an excessive number of rounds. An alternative, "single-round" protocol designs all possible primer pairs in a single round, thus leading to an excessively large number of primers. Since these criteria are conflicting, ExonPCR searches for a trade-off between the dichotomy strategy and the single-round strategy.

9.6 The Twenty Questions Game with Genes

In its simplest form, the problem can be formulated as follows: given an (unknown) set I of integers in the interval $[1, n]$, reconstruct the set I by asking the minimum number of queries of the form "does a given interval contain an integer from I?" In this formulation, interval $[1, n]$ corresponds to cDNA, I corresponds to exon boundaries in cDNA, and the queries correspond to PCR reactions defined by a pair of primers. A non-adaptive (and trivial) approach to this problem is to generate n single-position queries: does an interval $[i, i]$ contain an integer from I? In an adaptive approach, queries are generated in rounds based on results from all previous queries (only one query is generated in every round).

For the sake of simplicity, consider the case when the number of exon boundaries k is known. For $k = 1$, the optimal algorithm for this problem requires at least $\lg n$ queries and is similar to Binary Search (Cormen et al., 1989 [75]). For $k > 1$, it is easy to derive the lower bound on the number of queries used by any algorithm for this problem, which utilizes the decision tree model. The decision tree model assumes sequential computations using one query at a time. Assume that every vertex in the decision tree is associated with all k-point sets (k-sets) that are consistent with all the queries on the path to this vertex. Since every leaf in the decision tree contains only one k-set, the number of leaves is $\binom{n}{k}$. Since every tree of height h has at most 2^h leaves, the lower bound on the height of the (binary) decision tree is $h \geq \lg \binom{n}{k}$. In the biologically relevant case $k << n$, the minimum number of queries is approximately $k \lg n - k \lg k$. If a biologist tolerates an error Δ in the positions of exon boundaries, the lower bound on the number of queries is approximately $k \lg \frac{n}{\Delta} - k \lg k$. The computational and experimental tests of ExonPCR have demonstrated that it comes close to the theoretical lower bound and that about 30 primers and 3 rounds are required for finding exon boundaries in a typical cDNA sequence.

9.7 Alternative Splicing and Cancer

Recent studies provide evidence that oncogenic potential in human cancer may be modulated by alternative splicing. For example, the progression of prostate cancer from an androgen-sensitive to an androgen-insensitive tumor is accompanied by a change in the alternative splicing of fibroblast growth factor receptor 2 (Carstens et al., 1997 [59]). In another study, Heuze et al., 1999 [160] characterized a prominent alternatively spliced variant for Prostate Specific Antigene, the most important marker available today for diagnosing and monitoring patients with prostate cancer. The questions of what other important alternatively spliced variants of these and other genes are implicated in cancer remains open. Moreover, the known alternative variants of genes implicated in cancer were found by chance in a case-by-case fashion.

Given a gene, how can someone find *all* alternatively spliced variants of this gene? The problem is far from simple since alternative splicing is very frequent in human genes (Mironov et al., 1999 [240]), and computational methods for alternative splicing prediction are not very reliable.

The first systematic attempt to elucidate the splicing variants of genes implicated in (ovarian) cancer was undertaken by Hu et al., 1998 [167]. They proposed long RT-PCR to amplify full-length mRNA and found a new splicing variant for the human multidrug resistance gene MDR1 and the major vault protein (MVP). This method is well suited to detecting a few prominent variants using fixed primers but will have difficulty detecting rare variants (since prominent variants are not suppressed). It also may fail to identify prominent splicing variants that do not amplify with the selected primer pair.

The computational challenges of finding all alternatively spliced variants (*an Alternative Splicing Encyclopedia* or *ASE*) can be explained with the following example. If a gene with three exons has an alternative variant that misses an intermediate exon, then some PCR products in the cDNA library will differ by the length of this intermediate exon. For example, a pair of primers, one from the middle of the first exon and another from the middle of the last exon, will give two PCR products that differ by the length of the intermediate exon. This will lead to detection of both alternatively spliced variants.

Of course, this is a simplified and naive description that is used for illustration purposes only. The complexity of the problem can be understood if one considers a gene with 10 exons with one alternative sliding splicing site per exon. In this case, the number of potential splicing variants is at least 2^{10}, and it is not clear how to find the variants that are present in the cell. The real problem is even more complicated, since some of these splicing variants may be rare and hard to detect by PCR amplification.

Figure 9.11 illustrates the problem of building an ASE for the "genomic" sequence

'twas brilliant thrilling morning and the slimy hellish lithe doves

gyrated and gambled nimbly in the waves

whose alternatively spliced variants "make up" different mRNAs that are similar to the Lewis Carroll's famous "mRNA":

't was brillig, and the slithy toves did gyre and gimble in the wabe

The "exon assembly" graph (Figure 9.11) has an exponential number of paths, each path representing a potential splicing variant. The problem is to figure out which paths correspond to real splicing variants. For example, one can check whether there exists a splicing variant that combines the potential exons X and Y represented by $\boxed{\textit{T WAS BRILLI}}$ and $\boxed{\textit{G, AND THE SL}}$ with a *spanning primer* XY that spans both X and Y (for example, *BRILLIG, AND T*). In practice, an XY-primer is constructed by concatenation of the last 10 nucleotides of exon X with first 10 nucleotides of exon Y. Pairing XY with another primer (e.g., one taken from the end of exon Y) will confirm or reject the hypothesis about the existence of a splicing variant that combines exons X and Y. Spanning primers allow one to trim the edges in the exon assembly graph that are not supported by experimental evidence. Even after some edges of the graph are trimmed, this approach faces the difficult problem of deciding which triples, quadruples, etc. of exons may appear among alternatively spliced genes. Figure 9.11 presents a relatively simple example of an already trimmed exon assembly graph with just five potential exons and five possible paths: ABCDE, ACDE, ABDE, ABCE, and ACE. The only spanning primers for the variant ACE are AC and CE. However, these spanning primers (in pairs with some other primers) do not allow one to confirm or rule out the existence of the ACE splicing variant. The reason is that the presence of a PCR product amplified by a primer pair involving, let's say, AC, does not guarantee the presence of the ACE variant, since this product may come from the ACBD alternative variant. Similarly, the CE primer may amplify an ABCE splicing variant. If we are lucky, we can observe a relatively short ACE PCR product, but this won't happen if ACE is a relatively rare variant. The solution would be given by forming a pair of spanning primers involving *both* AC and CE. This primer pair amplifies ACE but does not amplify any other splicing variants in Figure 9.11.

The pairs of primers that amplify variant X but do not amplify variant Y are called X+Y- pairs. One can use X+Y- pairs to detect some rare splicing variant X in the background of a prominent splicing variant Y. However, the problem of designing a reliable experimental and computational protocol for finding all alternative variants remains unsolved.

9.8 Some Other Problems and Approaches

9.8.1 Hidden Markov Models for gene prediction

The process of breaking down a DNA sequence into genes can be compared to the process of parsing a sentence into grammatical parts. This naive parsing metaphor was pushed deeper by Searls and Dong, 1993 [312], who advocated a linguistic approach to gene finding. This concept was further developed in the Hidden Markov Models approach for gene prediction (Krogh et al., 1994 [209]) and culminated in

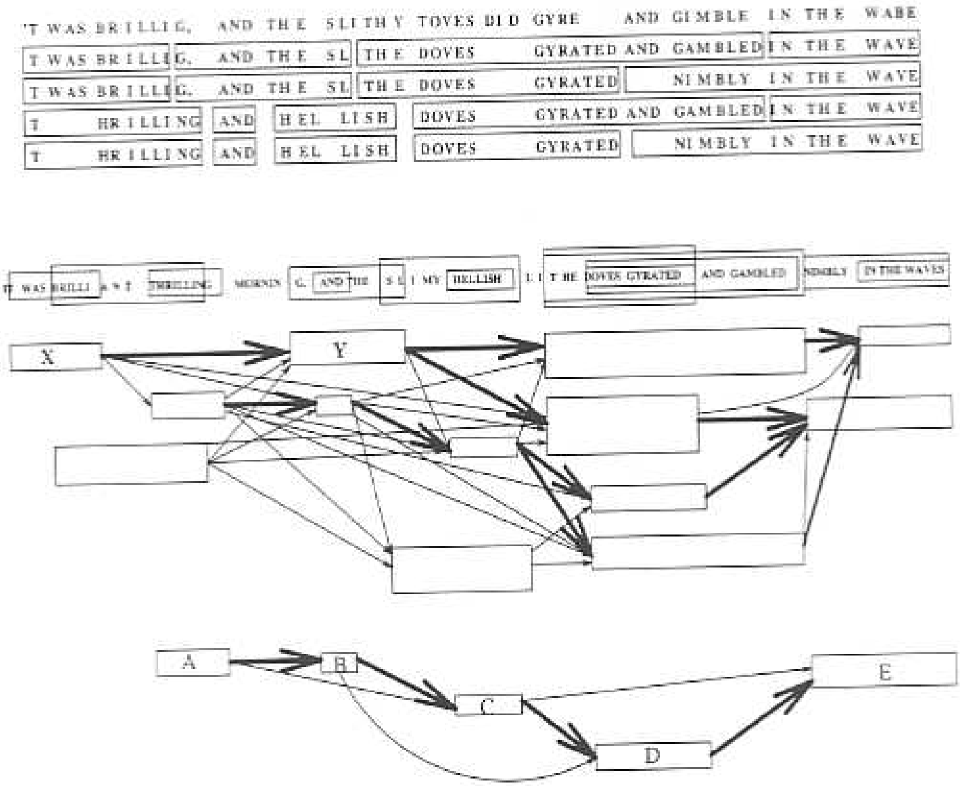
'T WAS BRILLIG, AND THE SLITHY TOVES DID GYRE AND GIMBLE IN THE WABE
T WAS BRILLIG, AND THE SL THE DOVES GYRATED AND GAMBLED IN THE WAVE
T WAS BRILLIG, AND THE SL THE DOVES GYRATED NIMBLY IN THE WAVE
T BRILLING AND HEL LISH DOVES GYRATED AND GAMBLED IN THE WAVE
T BRILLING AND HEL LISH DOVES GYRATED NIMBLY IN THE WAVE
'T WAS BRILLI A N T THRILLING MORNIN G, AND THE S L I MY HELLISH L I T HE DOVES GYRATED AND GAMBLED NIMBLY IN THE WAVES
X
Y
A
B
C
D
E

Figure 9.11: Constructing an Alternative Splicing Encyclopedia (ASE) from potential exons. Four different splicing variants (above) correspond to four paths (shown by bold edges) in the exon assembly graph. The overall number of paths in this graph is large, and the problem is how to identify paths that correspond to real splicing variants. The graph at the bottom represents the trimmed exon assembly graph with just five potential splicing variants (paths).

the program GENSCAN (Burge and Karlin, 1997 [54]). HMMs for gene finding consist of many blocks, with each block recognizing a certain statistical feature. For example, profile HMMs can be used to model acceptor and donor sites. Codon statistics can be captured by a different HMM that uses Start codons as *start* state, codons as intermediate states, and Stop codon as *end* state. These HMMs can be combined together as in the Burge and Karlin, 1997 [54] GENSCAN algorithm. In a related approach, Iseli et al., 1999 [176] developed the ESTScan algorithm for gene prediction in ESTs.

9.8.2 Bacterial gene prediction

Borodovsky et al., 1986 [43] were the first to apply Markov chains for bacterial gene prediction. Multiple bacterial sequencing projects created the new computational challenge of *in silico* gene prediction in the absence of any experimental analysis. The problem is that in the absence of experimentally verified genes, there are no positive or negative test samples from which to learn the statistical parameters for coding and non-coding regions. Frishman et al., 1998 [113] proposed the "similarity-first" approach, which first finds fragments in bacterial DNA that are closely related to fragments from a database and uses them as the initial training set for the algorithm. After the statistical parameters for genes that have related sequences are found, they are used for prediction of other genes in an iterative fashion. Currently, GenMark (Hayes and Borodovsky, 1998 [157]), Glimmer (Salzberg et al., 1998 [295]), and Orpheus (Frishman et al., 1998 [113]) combine the similarity-based and statistics-based approaches.