CHAPTER

7

Additional Bioinformatic Analyses Involving Nucleic-Acid Sequences*

OUTLINE

7.1 GENOME SEQUENCING

The traditional sequencing method involves the following steps: the DNA fragment to be sequenced is cloned into a vector that provides known primer-binding sites flanking the cloned sequence. The first set of sequencing primers is designed based on these known primer-binding sites. The sequencing runs on both strands produce two sequencing reads. New primers are designed from the 3′-end of the newly obtained sequences (Figure 7.1A). In this process, the sequence reads generated in one direction have sequence overlaps. Using the sequence overlaps, these contiguous sequence reads are assembled into a larger sequence, called a **contig**[a] (from contiguous) (Figure 7.1B; upper and lower panels). The sequencing method described above involves sequential designing of primers followed by new sequencing; hence, this sequencing method is called **primer walking**. Primer walking works well for sequencing a complementary DNA (cDNA) or a large DNA fragment of finite size. However, primer walking is costly and slow, and it involves cloning of the fragment. Although it can be scaled up, primer walking is still not a high-throughput strategy for sequencing a genome.

Primer walking is an example of **directed sequencing** because the primer is designed from a known region of DNA to guide the sequencing in a specific direction. In contrast to directed sequencing, **shotgun sequencing** of DNA is a more rapid sequencing strategy. As the name suggests, shotgun sequencing involves random fragmentation of the DNA into small pieces followed by sequencing of these small fragments. Shotgun sequencing can adopt either a **hierarchical shotgun sequencing** (top-down) approach, or a **whole-genome shotgun (WGS) sequencing** (bottom-up) approach. In the hierarchical shotgun sequencing approach, the chromosomes are sorted, broken into

*The opinions expressed in this chapter are the author's own and they do not necessarily reflect the opinions of the FDA, the DHHS, or the Federal Government.

[a]A sequence read should not be confused with a sequence contig. In theory, at least two overlapping sequence reads are needed to construct one sequence contig. In reality, a sequence contig is constructed from many sequence reads.

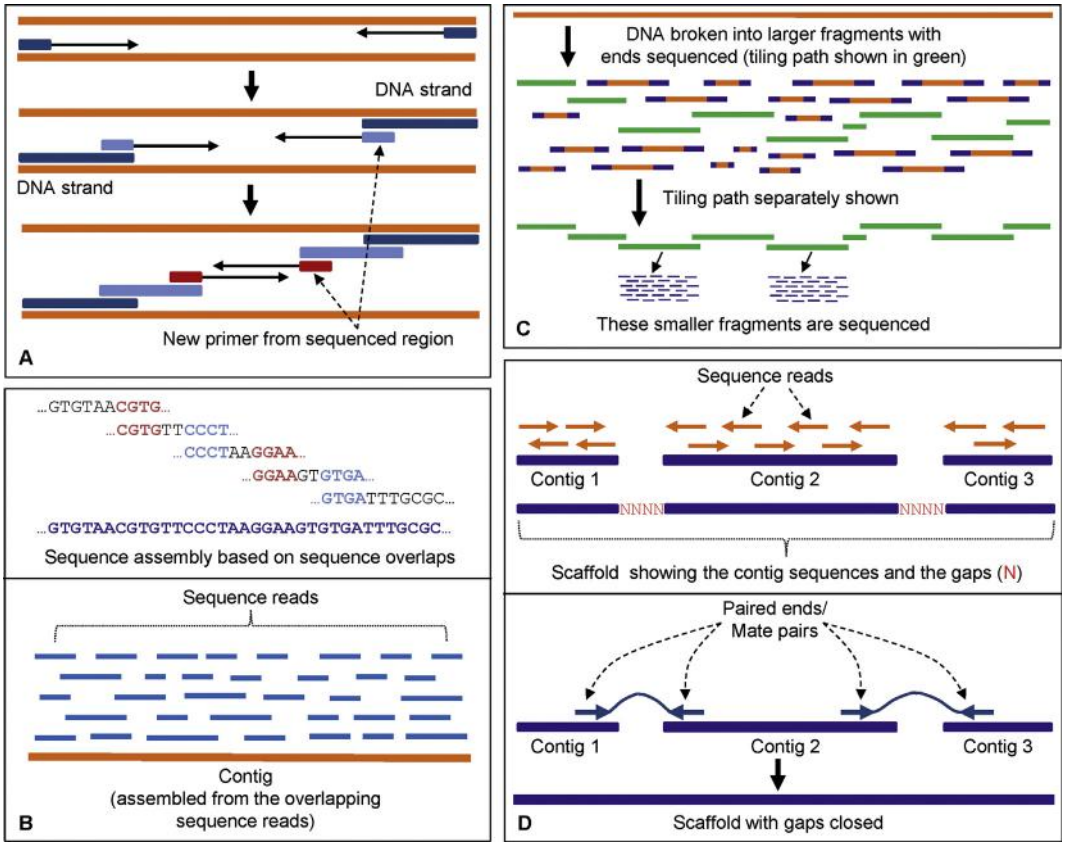


FIGURE 7.1 **Sequencing strategy.** (A) Directed DNA sequencing by primer walking. This involves sequential designing of primers from a known region. The first set of sequencing primers are designed based on the primer-binding sites flanking the cloned DNA. New primers are designed from the 3′-end of the newly obtained sequences. (B) The sequence reads have sequence overlaps that help put the contiguous sequences together in proper order (upper panel). Many such sequence reads are assembled to obtain a sequence contig (lower panel). (C) In the hierarchical shotgun sequencing approach, the chromosomes are sorted and broken into large fragments. Both ends of each clone are sequenced and the tiling path is determined based on sequence overlaps. The tiling path (shown as green fragments) is the smallest set of overlapping clones that covers the entire chromosome or contig. Once the clones in the tiling path are identified, the larger fragments in these clones are broken down into smaller fragments, which are then sequenced using a shotgun sequencing strategy. The sequence is put together by a sequence assembler. (D) A scaffold, or supercontig, is a portion of the chromosome (or genome) sequence that is composed of contigs put together in correct order. Scaffolds have gaps (upper panel); once the gaps are identified, the goal becomes sequencing those regions and closing the gaps. The lower panel shows that the scaffold of these three contigs is held together by mate pairs. The thin lines connect the paired ends.

large fragments and cloned into vectors that can hold large DNA fragments, such as bacterial artificial chromosomes (BACs) or yeast artificial chromosomes (YACs)[b]. Both ends of each clone are sequenced, producing an approximately 500–800-bp read each, together called **paired ends** or **mate pairs**, and the tiling path is determined based on sequence overlaps. This is part of the physical mapping process[c]. The **tiling path** is the smallest set of overlapping clones (i.e. clones with overlapping DNA fragments) that covers the entire chromosome or contig (Figure 7.1C). Therefore, the clones that produce the tiling path

[b]BACs can hold DNA fragments up to 300 kbp, whereas YACs can hold fragments up to 3000 kbp.

[c]A physical map of a chromosome is a set of cloned DNA fragments whose position relative to each other in the chromosome is known. In physical mapping, a large number of clones from the recombinant library of each chromosome are end sequenced to obtain a fingerprint for each clone. A fingerprint is a unique sequence signature that identifies a specific clone. The information about such signatures can be obtained by random sequencing or by examining sequence information already existing in the database. For example, the sequence of a known unique gene in the chromosome will provide the fingerprint for a clone that contains this sequence. This type of short DNA sequence (usually less than 500 bp) that occurs only once in the chromosome (or genome) is known as a sequence tagged site (STS). Appropriate overlaps between clones are determined based on such clone-specific fingerprints. Fingerprinting the clone contigs generates many genomic landmarks along the length of the chromosome. These landmarks help in the process of accurate sequence assembly, particularly if the genome is rich in repetitive sequences.

constitute a set of **clone contigs** (contiguous clones). Once the clones in the tiling path are identified, the larger fragments in these clones are broken down into smaller fragments, which are then sequenced using a shotgun sequencing strategy. The sequence is put together by a sequence assembler. During assembly, the contigs are assembled in correct order to produce longer **supercontigs**, also called **scaffolds**. Scaffolds usually have gaps (Figure 7.1D; upper panel). Once the gaps are identified, special care is taken to sequence the gapped regions; this is part of the finishing process for genome sequencing and assembly (Figure 7.1D; lower panel).

In the bottom-up WGS sequencing approach, the DNA is randomly sheared into small pieces, fragments are size selected and subcloned into a "universal" cloning vector containing "universal" priming sites. Clones are sequenced. Numerous sequence reads are generated from numerous small fragments. The sequence is put together by a sequence assembler with very high computing capacity. In 1988, Eric Lander and Michael Waterman published a paper in which they demonstrated mathematically that at least 8–10-fold sequencing coverage is needed for the successful assembly of most of the genome, assuming an even distribution of sequence reads.[1]

Both hierarchical shotgun sequencing and WGS sequencing have advantages and disadvantages. Hierarchical shotgun sequencing creates a physical map of the genome; hence, it produces genomic landmarks that can be helpful in sequence assembly if the genome is rich in repetitive sequences (like the human genome). However, hierarchical sequencing is slow because it proceeds through many steps. The WGS sequencing approach is rapid and direct, but the assembly of sequences may run into problems if the genome is rich in repetitive sequences. The number of sequencing reads generated in WGS sequencing is very high; therefore, the computing power needed for WGS sequence assembly is very high. Currently, the computing power is less of an issue, but it was an issue in early days of genome sequencing. Current genome-sequencing efforts adopt a combination of both strategies for speed and accuracy. Use of the next-generation (next-gen) sequencing technique has further added to the speed because it does not need cloning of the fragments.

7.2 SEQUENCE ASSEMBLY

Genome assembly from sequence reads is an algorithm-driven automated process. DNA-sequence-assembly programs have utilized sequence overlaps for sequence assembly in correct order. The computational aspect of assembly algorithms is beyond the scope of this book. Nevertheless, a few terms will be discussed in plain language for the sake of familiarity. Sequence assembly can be done using one of three approaches: (1) **greedy**, (2) **overlap-layout-consensus (OLC) and Hamiltonian path**, and (3) **de Bruijn graph and Eulerian path**[d].

Greedy is a rapid-assembly algorithm, which joins together the sequence reads that are the most similar to each other based on as much sequence overlap as possible. In doing so, the greedy algorithm first compares all fragments in a pairwise fashion to identify sequences that have overlaps; next, the sequences that have the best overlaps are merged; this merging process continues (iterative process) until all the sequences with overlaps have been merged. In this process, some reads may not be assembled, which are shown as gaps. Paired-end sequencing is used to close the gaps. Many early assemblers were based on the greedy algorithm and were extremely useful, such as **Phrap, TIGR assembler**, and **CAP**. The **Phred–Phrap–Consed** suite of programs has been widely used. Phred and Phrap were developed by Drs Phil Green and Brent Ewing at the University of Washington, Seattle, in 1998 for the Human Genome Sequencing project. Phred is base-calling software that assigns a quality score to each base called. Phrap is de novo shotgun sequence-assembly software. Consed is the sequence-assembly editor companion to Phrap, and it is a tool for viewing, editing, and finishing sequence assemblies created with Phrap. Many such assembly suites also include sequence-alignment tools.

The overlap-layout-consensus (OLC) algorithm is based on all pairwise comparisons, and it generates a directed graph using reads and overlaps[e]. In the graph, each sequence is created as a node and an edge is created between any two nodes whose sequences overlap. The algorithm then tries to find the Hamiltonian traversal path of the graph, which contains all the nodes (sequences) exactly once, and combines the overlapping sequences in the nodes into the sequence of the genome. Some assemblers that utilize

[d]If the reader is interested to learn more about the computational aspects behind the key methods in simple terms, a good source to consult is *Bioinformatics for Biologists*.[2]

[e]A graph is represented by a set of nodes (vertices) and a set of edges (arcs) between the nodes; hence, it can be conceptualized as balls (nodes) in space with arrows (edges) connecting them. If the edges can be traversed in only one direction, the graph is known as a directed graph. Each directed edge represents a connection from one "source node" to one "sink node"; the sink node of one edge forms the source node for any subsequent nodes. The assembly process is like finding the path through the graph in a way that the path visits every node only once.[3]

the OLC algorithm are **Arachne**, **CABOG** (**Celera Assembler**), **Newbler**, **Minimus**, **Edena**, and **MIRA**. Overlap-based approaches have been mostly used for longer reads (>200 bp). However, overlap-based assemblers for short reads have also been developed.[4]

The de Bruijn–graph-based approach has been successfully employed in assembling short reads (<100 bp). However, de Bruijn graph assemblers have also been successfully used with longer reads.[4] Some assemblers that utilize the de Bruijn–graph algorithm are **Euler-SR**, **Oases**, **Velvet**, **ALLPATH**, **ABySS**, and **SOAPdenovo**. Sequence assembly based on significant sequence overlap, as done using the standard Sanger method, works well when there are a finite number of sequence reads to be assembled. However, next-gen sequencing generates hundreds of millions of sequence reads. The assembly of such a large number of sequence reads cannot be done easily using this traditional method. The problem of scalability is solved by using the de Bruijn graph. The de Bruijn graph does not use the actual sequence reads for assembly, but breaks each sequence read down to smaller sequences called *k*-mers. These *k*-mers are aligned using $(k-1)$ sequence overlaps. The actual size of k depends on sequence coverage, read length, etc., but usually is not less than half of the actual read length. For example, a 106-base read can be divided into 49 overlapping 58-mers (sequence read length − *k*-mer length + 1 = # of *k*-mers; hence, $106-58+1=49$). Because breaking one sequence read into *k*-mers increases the number of short sequence reads (e.g. just one 106-base read generates 49 *k*-mers, each one 58 bases long), it is likely that the resulting *k*-mers generated from all sequence reads will represent nearly all *k*-mers from the genome for sufficiently small k. This process seemingly compensates for missing sequence reads—that is, the sequence reads that could not be generated through sequencing for a variety of technical reasons.[5] Therefore, computational application of the de Bruijn graph helps alleviate many problems of de novo sequence assembly, but it is still not a fool-proof process.

With the improvement of sequence coverage and computing power, software is being constantly being developed or improved based on newer algorithms. Sequence reads can now be accurately assembled based on overlaps as small as 15 bp.[6]

A genome sequence assembly can be performed in two ways: **mapping and assembly**, or **de novo assembly**. If the genome has been sequenced before and a **reference genome** sequence already exists, then the newly obtained resequence reads are first mapped to the reference genome through alignment and then assembled in proper order; this mode of assembly is called "mapping and assembly." Bowtie is an ultrafast, memory-efficient short-read aligner that helps in mapping and assembly. It rapidly aligns large sets of short sequencing reads to a reference sequence, at a rate of over 25 million 35-bp reads per hour. For reads longer than about 50 bp, **Bowtie 2** is generally faster, more sensitive, and uses less memory than the original Bowtie (http://bowtie-bio.sourceforge.net/index.shtml).

In contrast, if there is no reference genome sequence then the assembly is called "de novo assembly." For de novo assembly, paired reads work better than single reads because paired reads help generate scaffolds. Therefore, genome assembly is a hierarchical process; it is performed in steps beginning from the assembly of the sequence reads into contigs, assembly of the contigs into scaffolds (supercontigs), and assembly of the scaffolds into chromosomes. Many genome assemblies remain restricted to scaffold level for a long time because the gaps can not be easily sequenced. Some scaffolds can be placed within a chromosome, while the chromosomal assignment of other scaffolds may remain difficult.

The de novo genome assembly can be assessed based on a number of parameters, such as the number of contigs and scaffolds available and their size, and the fraction of reads that can be assembled. One widely used metric to evaluate the quality of assembly is the contig and scaffold **N50** value (see Box 7.1). An N50 contig is the size of the shortest contig such that the sum of contigs of that size or longer constitutes at least 50% of the total size of the assembled contigs. For example, an N50 contig of 100 kb means that when contigs of 100 kb or longer are added up, the resulting size represents at least 50% of the total size of all assembled contigs. Likewise, an N50 scaffold size is the length of the shortest scaffold such that the sum of the scaffolds of that size or longer constitutes at least 50% of the total size of all assembled scaffolds.

Although genome sequencing has become high throughput and very cheap, and the computational power in genome-sequence assembly has tremendously increased, the current methods have many problems, partly owing to the nature of the genome sequence itself and partly owing to problems inherent in the sequencing method. Consequently, de novo sequence assembly is still a major challenge and can be fraught with errors and missing sequence.[7] This makes finishing a genome sequence and assembly a continuous and long-drawn-out process.

7.3 GENOME ANNOTATION

Genome annotation is the process by which biological information is assigned to the genome sequence. It involves the prediction of exons, introns, regulatory elements, various signal sequences, alternatively

BOX 7.1

The N50 contig value can be determined by first sorting all contigs in decreasing order of size, then adding the contigs until the total added size reaches at least half of the total size of all assembled contigs. The size of the smallest contig used in this addition process represents the N50. The scaffold N50 is calculated in the same fashion using the scaffold size. For example, if the contigs assembled are 0.43, 0.75, 1, 0.6, 0.8, 0.55, 0.32, and 0.25 Mbp, the total assembled size of all contigs is 4.7 Mbp. Now, organizing the contigs in decreasing order of size, we get: 1, 0.8, 0.75, 0.6, 0.55, 0.43, 0.32, and 0.25 Mbp. Adding just 1, 0.8, and 0.75 yields 2.55 Mbp, which is 54% of the total assembled size of all contigs. The smallest contig used in this addition process is 0.75 Mbp. Therefore, the N50 contig is 0.75 Mbp. The larger the N50 value, the better is the assembly. Using the same concept, higher values of N are also used, such as N60 and N80. If the N50 scaffold length is too short, additional rounds of shotgun sequencing are recommended.

spliced variants, noncoding RNAs, etc., that ultimately reflects the function and sheds light on molecular (sequence) evolution. Therefore, annotation has a structural aspect and a functional aspect. Annotation can be done computationally or manually; the latter requires human expertise. In reality, both computational and manual annotations are used to optimize the annotation process. Expectedly, the existence of similar annotated genomes greatly facilitates the annotation of newly sequenced genome. *The median gene lengths are roughly proportional to genome size; hence, bigger genomes have bigger genes*. Thus, accurate annotation of a larger genome requires a more contiguous genome assembly in order to avoid splitting genes across scaffolds.[8]

In brief, at the beginning of genome annotation, repeats are identified and masked computationally (e.g. using **RepeatMasker**; created by Smit, A.F.A., Hubley, R., and Green, P.; http://www.repeatmasker.org) because repeats, if not removed, can produce false evidence of gene annotations through spurious BLAST alignments. Repeats include low-complexity sequences (homopolymeric runs of nucleotides) and transposable elements, including long interspersed nuclear elements (LINEs) and short interspersed nuclear elements (SINEs). Computational masking of repeat sequence frequently involves replacing the sequence with "N".

After repeat masking, the genome assembly is aligned to known expressed sequence tag (EST), RNA, and protein sequences; these sequences may include previously identified transcripts and proteins from the same organism whose genome is being annotated, or they may be from other organisms. When sequences from other organisms are used, evolutionarily conserved proteins provide useful information. The alignment process uses BLAST and BLAT (discussed in Chapters 2, 5, and 6) in order to rapidly identify approximate regions of homology. BLAT can also map these sequences to the genome. The alignment data are filtered to eliminate marginal alignments as revealed by low % identity or % similarity. The filtered alignment data are then inspected for the presence of redundant sequences, which would be removed. Further alignment is performed to obtain greater precision of exon boundaries using splice-site detecting alignment algorithms, such as **Splign** (http://www.ncbi.nlm.nih.gov/sutils/splign/splign.cgi) and **Spidey** (http://www.ncbi.nlm.nih.gov/spidey/spideydoc.html). Both Splign and Spidey compute mRNA/cDNA-to-genome alignments, including spliced sequence alignments. Splign was developed by Kapustin et al.[9] and Spidey was developed by Wheelan et al.[10] Figure 7.2 shows how Splign can be used online. The example used is mouse *Slco1a6* mRNA (cDNA) (RefSeq NM_023718.3), which was mapped to and aligned with the mouse genome to find the genomic location of the exons and splice-junction sites. Figure 7.3 shows partial information of Splign output.

The final stage of annotation is best done manually but is being increasingly done computationally. Although manual annotation is high quality, it is time consuming, expensive, and labor intensive. In the age of massive genomic data generation, available genomic information, and increased computational power, genome annotation projects are increasingly utilizing automated annotation. The ultimate goal of annotation is to obtain a synthesis of alignment-based evidence with gene predictions to obtain a final set of gene annotations. Annotation of a genome undergoes repeated quality-control checks and it is a long ongoing process. The target for annotation is to generate a "high-quality draft" assembly that is at least 90% complete.[8] RNA sequencing (RNA-seq) data can be used to greatly improve the accuracy of gene annotations because such data provide strong evidence for exons, splice sites, and alternatively spliced exons. The interested reader is urged to read an excellent overview of eukaryotic genome annotation by Yandell and Ence.[8]

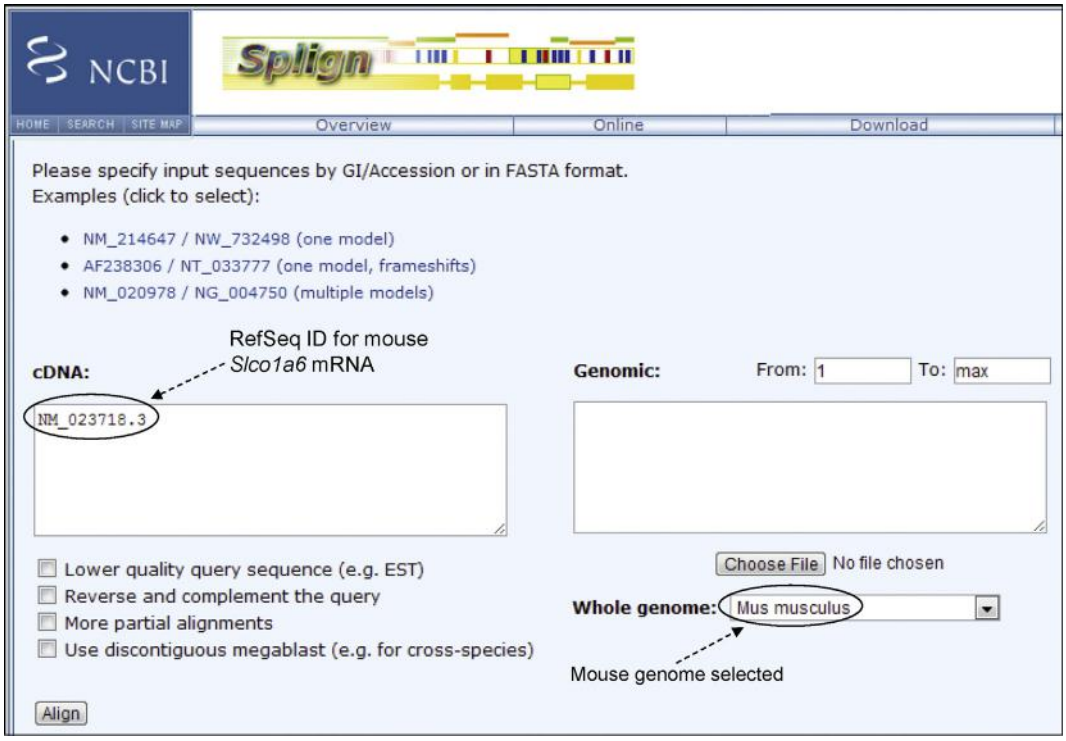


FIGURE 7.2 **The use of Splign online.** In the box for cDNA, either the sequence or the accession number/GI number can be entered. The sequence has to be entered in FASTA format. The example used is mouse *Slco1a6* mRNA (cDNA) (RefSeq NM_023718.3). The goal is to map the sequence to and align it with the mouse genome to find the genomic location of the exons and splice-junction sites. The default settings were maintained.

7.3.1 Gene Prediction

Gene prediction, which is part of genome annotation, involves the identification of putative coding exons in an unannotated DNA sequence. In other words, gene prediction attempts to predict putative coding sequences. The process is probabilistic and the putative exons are scored for the probability of being a true exon.

Gene prediction in prokaryotes (Bacteria and Archaea) involves fewer confounding factors than in eukaryotes because in prokaryotes the genome size is small and gene density is high, with ~88% of the genome containing coding sequences.[11] Bacteria do not have introns (Archaea have introns in rRNA and tRNA genes[12]), and the genomes have fewer repeat sequences. This is in contrast to eukaryotic genomes that are very large and full of repeat sequences; the majority of the eukaryotic genome is non-protein-coding, and the protein-coding genes contain large introns. Bacterial genes also have **Shine–Dalgarno sequence** (consensus AGGAGGT), which is the ribosomal binding site that lies upstream of the translational initiation codon (ATG) but downstream of the transcription start site. The end of the transcriptional unit (operon) has a terminator sequence that can form a stem–loop structure followed by a string of "T"s. The frequency of certain codons is much higher because of known codon preferences. These telltale signals, coupled with high gene density and fewer repeat sequences in the genomes, tend to make gene prediction in prokaryotes easier than in higher eukaryotes.

Gene prediction in an unannotated genome can be performed by **intrinsic** or ***ab initio*** prediction, **extrinsic** or **evidence-based** prediction, and **homology-based** prediction.

In the absence of any reference sequence (genome, EST, protein) from a related organism, gene prediction relies on **intrinsic** or ***ab initio*** prediction—that is, prediction based on the identification and analysis of telltale signals of protein-coding genes. In other words, the prediction is based on the information contained in the genomic sequence itself. Some of these signals are: start and stop codons, known codon preferences, intron splice signals, poly(A) signal sequence, TATA boxes, cap sites, transcription-factor-binding sites, Kozak sequence, and termination signals. In addition, the nucleotide composition differences known to exist between coding and noncoding regions as well as many essential features of gene structure are also taken into account, such as gene density, typical number of exons/gene, typical exon length, and open reading frame (ORF)-specific hexamer composition versus

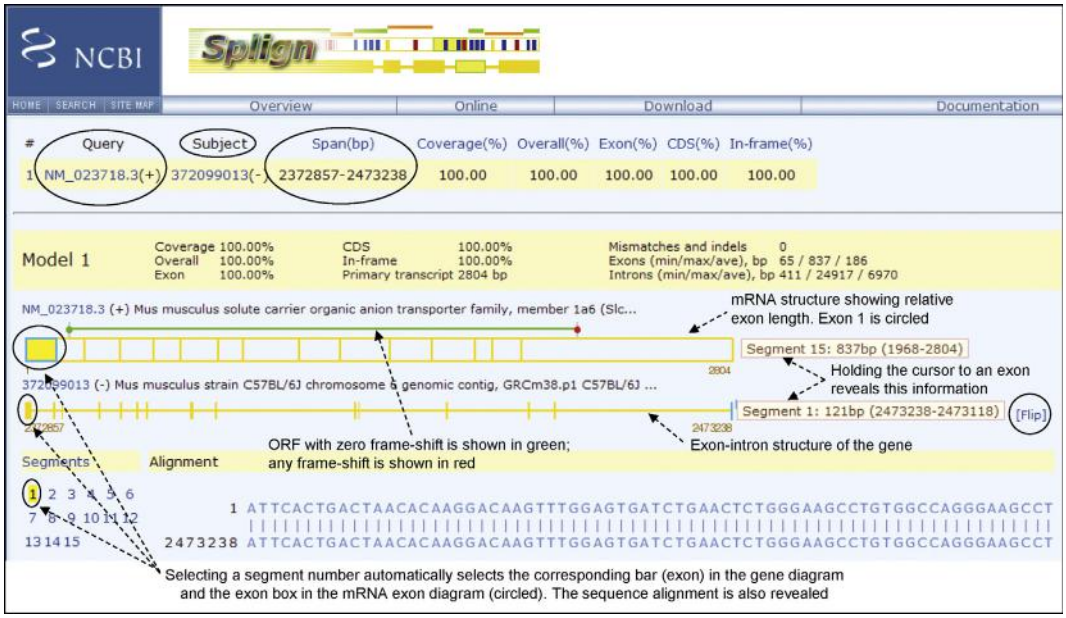


FIGURE 7.3 **Partial Splign output.** Splign has aligned the input sequence to the mouse genome, and has created 15 segments, displayed under "Segments" link on the left-hand side. In this example, each segment corresponds to one exon. Above the "Segments" link is the exon−intron organization of the gene, in which each exon is represented by a vertical line. Above the gene diagram is the mRNA diagram, in which each exon is represented by a box and the length of each box is proportional to the length of the exon. So, exon 15 (the last exon) is the longest. Above the mRNA, the open reading frame (ORF) is represented by a line. The green line here shows that there is no frameshift in the input sequence. Any frameshift would be represented by a partial red line. The green dot at the beginning and the red dot at the end of the ORF denote the start and the stop codon, respectively. Although not shown here, mismatches are denoted by vertical red lines and insertions/deletions (indels) are denoted by vertical blue lines inside the rectangular boxes representing exons. If the cursor is held close to an exon in the gene (vertical line), its genomic location appears as long as the cursor is held in place (segment 1 in this example); similarly, if the cursor is held close to an exon in the mRNA (rectangular box), its location in the mRNA appears (segment 15 in this example). Note that for the mRNA, the orientation is 5′→3′ from left to right; hence, segment 15 (exon 15) is at the right, whereas for the gene, the orientation is 5′→3′ from right to left; hence segment 1 (exon 1) is at the right. This is because the gene is located in the reverse orientation in the genome, which is indicated by the word "Flip" (right-hand side, circled). In the figure, the location of exon 15 (segment 15) of the mRNA and segment 1 (exon 1) in the genome are shown; one of them is copied and pasted separately in the figure. This is because only one at a time can be obtained, not both. As soon as a segment is selected, the corresponding vertical line in the gene diagram becomes blue and the corresponding rectangular box in the mRNA diagram becomes highlighted in yellow with its border becoming blue (in the figure, exon 1). Also, the alignment with the genomic sequence is displayed.

ORF-independent hexamer composition (in introns and intergenic regions).

The nucleotide composition of coding versus noncoding regions is analyzed using probabilistic statistics, such as various versions of Markov models. For example, the wobble base (third position in a codon) tends to be higher in G + C content in a coding region. Thus, if the local G + C content in a genomic region is significantly higher than the background, it suggests the likelihood of an ORF in that region. The sequence can be translated in all six frames (three sense, three antisense). Because there are 3 stop codons plus 61 amino-acid codons, a random unbiased distribution of bases should produce approximately 1 stop codon for every 20 codons in an ORF search. If the region is rich in A + T, a stop codon is expected even before 20 codons because the stop codons (TAA, TAG, TGA) are A + T rich (7 A + T out of 9 bases). These features and generalizations are expected for noncoding regions, but not for coding regions. Therefore, if an ORF search of a genomic region produces a translated ORF that shows a significantly high number of codons, such as >50 or so, before a stop codon appears, it suggests the likelihood of a legitimate ORF. With some exceptions, the number of codons in most ORFs is far greater than 60; in fact, proteins containing <200 amino acids are still considered to be small proteins and are known to play important roles in development.[13] Therefore, the *ab initio* approach combines statistical analyses along with other gene signals for gene prediction.

AUGUSTUS (http://bioinf.uni-greifswald.de/augustus/submission) is an *ab initio* gene-prediction program that uses the hidden Markov model (HMM; see Box 7.2). The program has used a diverse training set of approximately 60 genomes belonging to four different groups of organisms: animals; Alveolata (single-celled eukaryotes); plants and algae; and fungi, and is therefore able to predict genes in a wide range of species. The original version of AUGUSTUS utilized a purely *ab initio* method and was

BOX 7.2

THE HIDDEN MARKOV MODEL

Gene-prediction algorithms have become more sophisticated with the incorporation of statistical methods, particularly the Markov model and its variants. A **Markov model** is a stochastic model—that is, a model to predict the outcome of a stochastic (random) process. The simple Markov model is a **Markov chain** that represents an ordered sequence of discrete events, moving from one "state" (event) to another with a certain probability, called the **transition probability**. In a Markov chain, at any given point in time, each current state has a previous state s_i, which has evolved into the current state s_j with a transition probability p_{ij}, and the current state s_j will evolve into a future state s_k with a transition probability p_{jk}. In this sequence of events, p_{jk} depends on s_j but not s_i. In other words, a Markov model assumes that the probability of the future state depends on the current state but NOT on the past state.

A Markov model predicts the evolution of an observable event that depends on internal factors. The observable event can be called an "output signal" and the internal factor can be called a "state." In a Markov model prediction, both the "output signal" and the "state" are observable. Markov models are used to predict many events in day-to-day life, such as stock market performance, to make weather forecasts, and so on. In contrast to Markov models, in the hidden Markov model (HMM) the "output signal" is observable but the "state" is not. Examples of HMM from biology are DNA and protein sequences. A DNA sequence is an observable output signal (from sequence determination) but the state of the sequence—that is, whether the sequence belongs to exon or intron or regulatory element or intergenic region—is not directly observable. Similarly, the sequence of amino acids in a protein is an observable output signal (from sequence determination), but the state of the sequence—that is, whether the sequence is part of a specific domain (e.g. a transmembrane domain)—is not directly observable. These hidden states can be modeled and predicted with certain probabilities by HMM. Consequently, HMMs have been used in, among other things, gene prediction, pairwise and multiple sequence alignment, base-calling, modeling DNA sequencing errors, protein secondary structure prediction, noncoding RNA (ncRNA) identification, RNA structural alignment, acceleration of RNA folding and alignment, and fast noncoding RNA annotation.[14]

Markov models can be **fixed order** or **variable order**, as well as **inhomogeneous** or **homogeneous**. In a fixed-order Markov model, the most recent state is predicted based on a fixed number of the previous state(s), and this fixed number of previous state(s) is called the **order** of the Markov model. For example, a **first-order** Markov model predicts that the state of an entity at a particular position in a sequence depends on the state of one entity at the preceding position (e.g. in various *cis*-regulatory elements in DNA and motifs in proteins). A **second-order** Markov model predicts that the state of an entity at a particular position in a sequence depends on the state of two entities at the two preceding positions (e.g. in codons in DNA). Similarly, a **fifth-order** Markov model predicts the state of the sixth entity in a sequence based on the previous five entities (e.g. in hexamers in coding sequence). It has been observed that the probability of occurrence of pairs of codons (hexamers) in a coding sequence is significantly higher than in noncoding sequence. A fifth-order Markov model calculates the probability of the sixth base based on the previous five bases in the sequence. In addition to the order, if the probability of occurrence of the state also depends on the position within the sequence, the model is called an inhomogeneous Markov model. In contrast, in a homogeneous Markov model all positions in the sequence are described by the same set of conditional probabilities.

Fifth-order Markov models are often used in gene prediction. For example, **GeneMark** (http://opal.biology.gatech.edu/GeneMark/) is a family of gene-prediction programs that uses an inhomogeneous fifth-order Markov model. However, a potential problem with a higher-order (e.g. fifth-order) Markov model is having enough data for the training set. For example, a fifth-order Markov model will require 4^5 (=4096) probabilities (probable combinations) to be estimated from the training data. In order to estimate these probabilities, many occurrences of all possible *k*-mers must be present in the data. The lack of availability of such huge amount of data may limit the usefulness of a higher-order Markov model. The **interpolated Markov model** (**IMM**) overcomes this problem by combining probabilities from contexts of varying lengths to make predictions, and by only using those contexts (oligomers) for which sufficient data are available.[15] The IMM method involves sampling dimers ($k = 1$) to nine-mers ($k = 8$) and adding the probabilities of all weighted *k*-mers, placing less weight on rare *k*-mers and more weight on more abundant *k*-mers. Therefore, the probability of the model is the sum of all probabilities of all weighted *k*-mers for which sufficient data are available. **GLIMMER** (Gene Locator and Interpolated Markov ModelER) is a microbial gene prediction and genome annotation tool that uses IMM and is available to run online at the NCBI (http://www.ncbi.nlm.nih.gov/genomes/MICROBES/glimmer_3.cgi). The majority of gene-prediction software uses HMM for prediction.

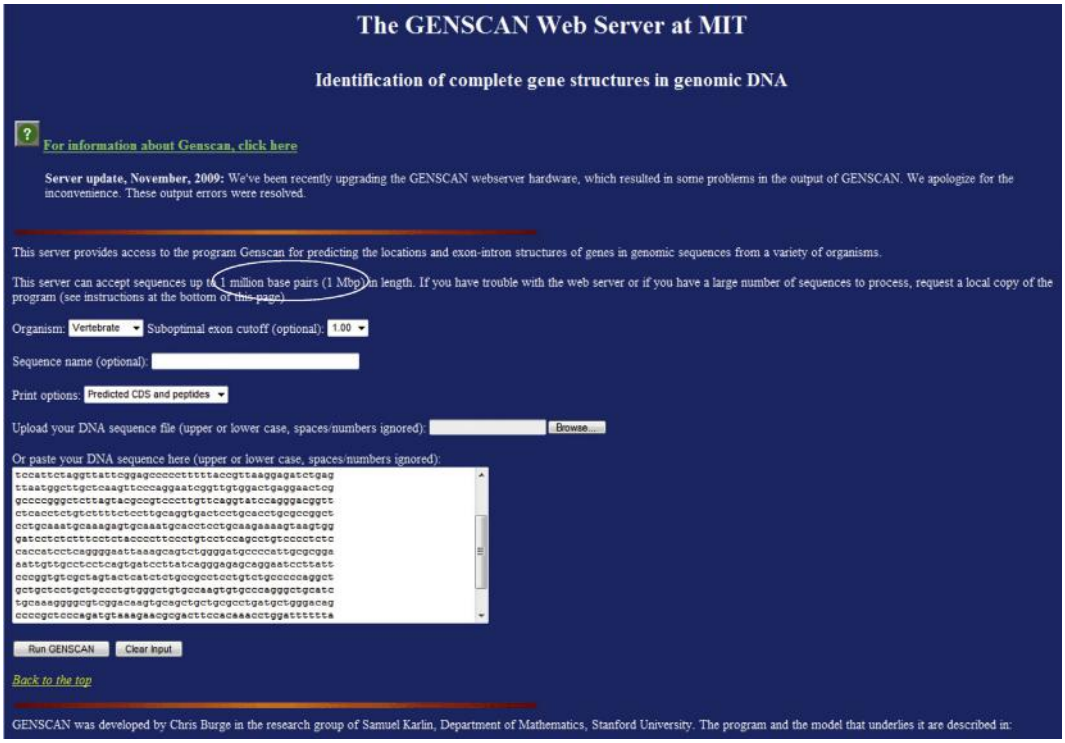


FIGURE 7.4 **GENSCAN home page.** Currently, GENSCAN can analyze an input sequence of up to 1 million bases (circled).

found to be one of the best *ab initio* algorithms for gene prediction.[16] **FGENESH** is a very fast and accurate *ab initio* gene-prediction program. The SoftBerry home page (http://linux1.softberry.com/berry.phtml) provides link to FGENESH and to a diverse set of other bioinformatics applications. **GENSCAN** (http://genes.mit.edu/GENSCAN.html) is another *ab initio* prediction tool developed early on by Dr Chris Burge in the research group of Samuel Karlin at Stanford University[17]; it also utilizes HMM. GENSCAN was trained using 570 vertebrate gene sequences.[18] When tested on standardized sets of human and vertebrate genes, GENSCAN accurately predicted 75 to 80% of exons.[17] Figure 7.4 shows the GENSCAN home page, and Figure 7.5 shows a GENSCAN analysis of a 932-bp input DNA fragment[f].[19] Based on the G + C content, the input sequence is predicted to belong to **isochore 3**[i] (circled).

Ab initio prediction algorithms fail to accurately predict alternative splicing, very long or short exons, nested and overlapping genes, any non-canonical features associated with the gene (e.g. non-ATG start codon, selenocysteine codons, split start or stop codons, etc.). Purely *ab initio* predictions are generally 50% or less accurate at the gene level.

Another approach is **extrinsic** or **evidence-based** prediction, in which some information is available, such as mRNA, EST, or protein product information. As more and more genomes have been sequenced and annotated, and more and more genomic information has become available, the pure *ab initio* prediction algorithms have been modified to incorporate genomic information and develop extrinsic prediction algorithms. For example, the newer version of AUGUSTUS combines the prediction ability of an *ab initio* algorithm with extrinsic information, such as matches to protein databases or alignments of genomic sequences, to improve the prediction accuracy. Because of this improvement, the new version of AUSGUSTUS is also able to predict splice variants, which the original algorithm could not do. **MAKER 2** (http://www.yandell-lab

[f]GenBank: NC_000016.9, Region: 56642478 − 56643409

[i]**Isochores** have been defined as >300-kb-long DNA segments in warm-blooded vertebrates (birds and mammals) with a characteristic, relatively homogeneous base composition. Based on the G + C content, isochores are classified in two "G + C-poor" types (L1 and L2) and three "G + C-rich" types (H1−H3). The average G + C content of isochore 3 (H3) is the highest (~ 54%) and it constitutes ~ 3% of the genome. In general, genes with higher G + C content belong to G + C-rich isochores (types H1−H3). The H2 and H3 isochores together have been termed the "genome core" because of their higher gene concentrations, which makes up about 12% of the genome (9% for H2 and 3% for H3). In the human genome, the H3 isochore apparently contains 25% of the genes, and the genome core (H2 + H3 combined) contains about 54% of the genes.

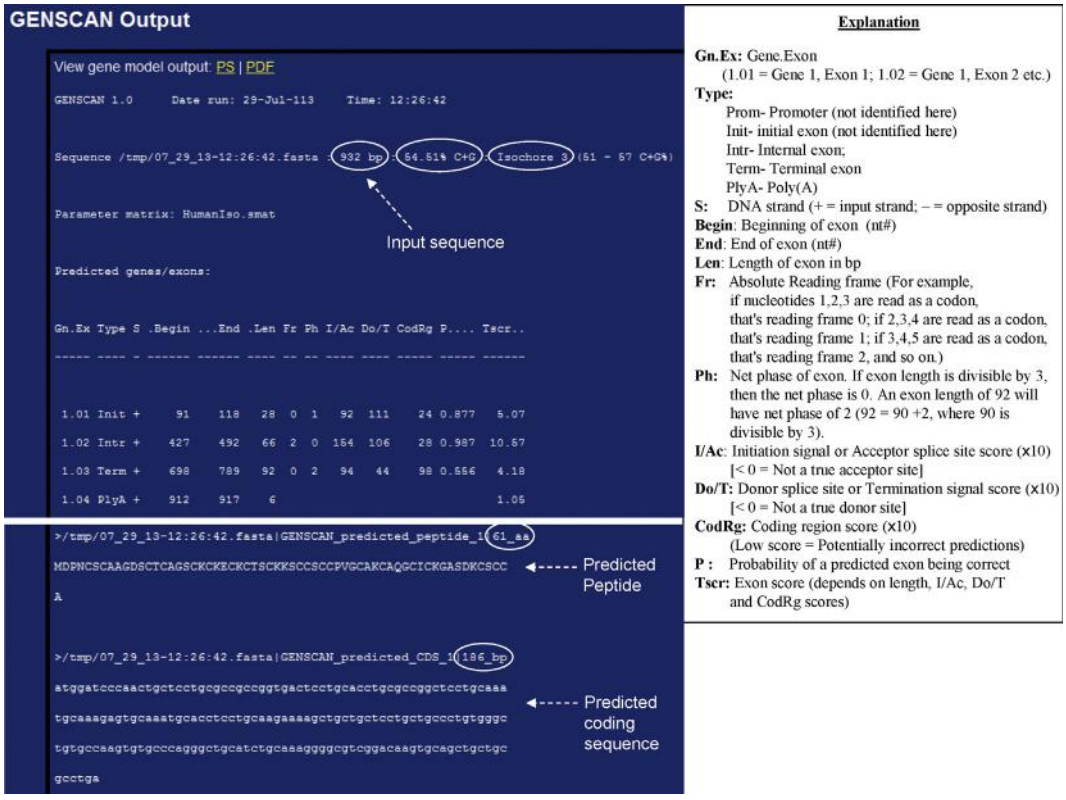


FIGURE 7.5 **GENSCAN analysis of a input DNA sequence fragment.** The upper left panel shows the analysis output and also the length of the input sequence (932 bp) and its G + C content (54.51%) (circled). Based on the G + C content, the input sequence is predicted to belong to isochore 3 (circled). The lower left panel shows a 186-bp predicted ORF and a 61-amino-acid predicted protein. The abbreviations are explained in the right-hand panel.

.org/software/maker.html) is another gene-prediction and genome-annotation program that combines *ab initio* and extrinsic approaches to produce gene annotations having evidence-based quality values. **GenomeScan** (http://genes.mit.edu/genomescan.html) is the successor of GENSCAN and it performs gene prediction in humans and other vertebrates. The algorithm utilizes two principal sources of information: (1) models of exon−intron and splice-signal composition; and (2) sequence similarity information, such as BLASTX hits. The probabilistic model used by GenomeScan is based on that used by GENSCAN.

Homology-based prediction relies on identifying significant matches of the query sequence with sequences in known and annotated genome sequences from related species. Thus, homology-based prediction relies on comparative genomics, and has been made possible because the genomes of many organisms have been sequenced. Homology-based prediction is based on the molecular evolutionary principle that functionally important parts of the genome evolve at a slower rate compared to the rest of the genome; therefore, many gene sequences, particularly in related species, should be highly conserved and therefore be recognizable by the prediction algorithm. Consequently, homology-based prediction has a high level of accuracy, and the greater the number of available genomes of related species, the greater the accuracy and completeness of prediction. The homology-based gene prediction tools align syntenic regions of unannotated genomes, and utilize a probabilistic framework for gene structure prediction. Several programs have been developed for homology-based prediction, such as **SLAM** (http://baboon.math.berkeley.edu/~syntenic/slam.html), **CEM**, and **Twinscan/N-SCAN** (http://mblab.wustl.edu/software.html), and **EuGene'Hom** (http://tata.toulouse.inra.fr/apps/eugene/EuGeneHom/cgi-bin/EuGeneHom.pl); for plant genomes. Comparative-genomics-based gene-finding programs outperform *ab initio* gene-finding programs.[20,21]

Many of these software programs can be downloaded for noncommercial and research purposes to carry out sequence analysis and gene prediction. A list of many gene-prediction software programs is available at the geneprediction.org website (http://www.geneprediction.org/software.html). Many of these can be accessed and run online by simply entering the input sequence either in plain text format or in FASTA format. The reader can try these links using known

genomic sequence (containing a known gene) and learn firsthand how each algorithm performs gene prediction and what the different outputs look like. A flow-chart for practice activity is given below.

Go to the NCBI home page → select "Gene" from the drop-down list of databases → enter Oatp-5 (or Slco1a6) in the "Search" space and hit enter → from the "Results" page, click "Mus musculus Slco1a6" → scroll down the Slco1a6 page → under the "NCBI Reference Sequences (RefSeq)" bar, locate the section "Reference GRCm38.p1 C57BL/6J" → under this section, locate the heading "NC_000072.6" → under this heading, click the "GenBank" link[j].

This will take the user to the RefSeq nucleotide sequence page of chromosome 6 showing VERSION NC_000072.6 and GI: 372099104. The sequence is 100,382 bases long. Copy the sequence. Now open a new web browser page → Google "Readseq" (the file conversion tool) → open Readseq from any of the sites, such as EBI, NIH, or Indiana University link → paste the sequence → from the "Output format" drop-down menu select the format as "Plain/Raw" if plain text format is desired or "Pearson/Fasta" if FASTA format is desired, and check the box for "Remove gap symbols" (or "degap" if using the EBI link). "Submit" the sequence and the desired sequence format will be returned without base numbers and gaps. Now copy this sequence and paste it in any of the gene prediction tools and run gene prediction. The Readseq link at the Indiana University site (http://iubio.bio.indiana.edu/cgi-bin/readseq.cgi) provides an option to download the sequence file, but the default is "View in browser."

Although the three approaches have been discussed here separately; in reality they are combined to increase the prediction accuracy. The sequencing and annotation of an ever-increasing number of prokaryotic and eukaryotic genomes have made it possible to successfully combine all three approaches. A common current approach for gene finding involves the following activities: several sets of gene predictions by different gene finders are compiled, and alignments from ESTs and proteins to the genome are constructed. All these data are combined to find the most plausible gene sequence, either manually or by using meta tools that combine several predictions and alignments.[16]

7.4 PREDICTION OF PROMOTERS, TRANSCRIPTION-FACTOR-BINDING SITES, TRANSLATION INITIATION SITES, AND THE ORF

Many free software packages are available online for the prediction of putative promoter sequences, transcription start sites, *cis*-regulatory elements, translation initiation sites, and the ORF.

Transcription of all classes of RNA (rRNA, mRNA, tRNA) in prokaryotes is catalyzed by one RNA polymerase, which is a multi-subunit enzyme. It contains a core polymerase that is composed of five subunits (α^{I}, α^{II}, β, β', ω), and a sigma (σ) factor. The **sigma factor** is the **initiation factor** that helps position the core polymerase to the promoter. The promoter has two consensus sequences, one at the −10 position (TATAAT in *Escherichia coli*), also known as the **Pribnow** box, and the other at the −35 position (TTGACA in *E. coli*) relative to the transcription start site. Bacteria possess different types of sigma factors. In *E. coli* and other bacteria, the sigma factor that initiates transcription of housekeeping genes and many other genes has a molecular weight of 70 kDa (hence σ^{70}). In prokaryotes, a transcriptional unit (i.e. an **operon**) may contain one gene or a number of genes under the control of one promoter. The transcription of one gene produces **monocistronic** RNA, whereas the transcription of many genes produces **polycistronic** RNA. Therefore, the promoter is located upstream of the first gene in a polycistronic transcriptional unit. Wang et al.[22] predicted operons in *Staphylococcus aureus* with $>90\%$ accuracy using a scoring system to annotate the intersection between two genes. In other words, this method identified whether two adjacent genes belong to the same operon. The scoring system was based on a number of parameters, such as intergenic distance, presence/absence of a terminator, comparison with other known prokaryotic genomes, etc.

Transcription in eukaryotes is carried out by three different RNA polymerases—RNA polymerases I, II, and III—which all bind to the promoter regions of the respective genes that will be transcribed. Of these, RNA polymerase II (pol II) produces translatable mRNAs. RNA pol II binds to the promoter, and also interacts with various other proteins for transcription. The DNA-binding proteins bind to specific sequence elements, called *cis*-response elements or *cis*-regulatory elements, that are all located at variable distances upstream of the transcription start site. The eukaryotic promoter can be divided into the **core** (or basal), **proximal**, and **distal** promoter, based on function and distance from the transcription start site.

In general, the transcription start site is determined by the TATA box (consensus TATAAA) and initiator (Inr) element (consensus: Y-Y-+1-N-T/A-Y-Y, where Y = pyrimidine, +1 = transcription start site, N = any nucleotide), or by the Inr element and downstream promoter element (DPE; consensus: $(A/G)_{+28}$ G(A/T) (C/T)$(G/A/C)_{+32}$) in the case of TATA-less promoters.

[j]These commands are current as of July, 2013. They may change if the mouse genome assembly version changes.

Typically, the core promoter is about 35 bp long, and can extend either upstream or downstream of the transcription start site (−35 to +35).[23] The core promoter may contain two or more of the following sequence motifs: TATA box, Inr element, and DPE. In most higher eukaryotic genes, the TATA box is located approximately 25-nt upstream (usually between −30 and −25) from the transcription start site. In many genes, a variation of the classic Inr may be present.[24]

The proximal promoter is about 250 bp long and can extend between the −250 and +250 nt positions, relative to the transcription start site.[25] Two transcription-activating response elements found in the proximal promoter are the CAAT box (binds the transcription factor NF-I) and the GC box (binds the transcription factor Sp1). The CAAT box is located ∼75 nt upstream of the transcription start site and has a consensus sequence GG(T/C)CAATCT. The GC box is located ∼90 nt upstream of the transcription start site and has a consensus sequence GGGCGG. The CAAT box and the GC box operate as enhancer elements because they can activate transcription in an orientation-independent manner.

Distal promoter sequences are further upstream of the proximal promoter elements.[26] The majority of transcription-regulatory protein-binding sites are located within 500 bp upstream of the transcription start site. Some regulatory-protein-binding sites can also be located downstream of the transcription start site.

Prediction of the translation initiation site (TIS) in a genomic sequence is an important problem to address. TIS prediction at the genome level is still not a trivial task because of the noise in the data. Some algorithms take into account weighted signal-based translation initiation site scores as well as the coding potential of sequences flanking TISs. At the gene level, an important sequence feature relevant for translation initiation and identification of the correct ATG codon by the translation initiation complex is the **Kozak sequence**. The original functional Kozak sequence (in the sense strand of DNA) was described as 5′-GCCRCC**ATG**G-3′ (where R is a purine, which in most vertebrate mRNAs is an "A"; **ATG** is the translation initiation codon). A shorter and more effective version (5′-ACC**ATG**G-3′) of the original Kozak sequence was also described later. The translation initiation region is characterized by certain features. Many genes contain the consensus Kozak sequence while others contain some variant. Still others may not have any Kozak sequence at all. The "G" after the ATG (i.e. ATG**G**) is the most prevalent base in the vast majority of mRNAs. If there is an ATG codon before the actual start codon, the sequence context of that ATG codon—such as lack of Kozak sequence around it, lack of a "G" immediately following the ATG, etc.—can help the ribosome bypass the incorrect ATG and detect the right ATG codon through scanning (known as **leaky scanning**). The incorrect ATG is usually out of frame with respect to the true initiation codon. If translation is initiated from the incorrect ATG codon that precedes the correct ATG codon, the ribosome encounters a premature stop codon, which is in-frame with the incorrect ATG codon. In such cases, translation is initiated again (**reinitiation**) from the correct initiation codon.

The National Center for Biotechnology Information (NCBI) ORF prediction tool **ORF Finder**[k] (http://www.ncbi.nlm.nih.gov/gorf/gorf.html) is a graphical analysis tool that finds all ORFs of a selectable minimum size in the six frames (three sense; three antisense), using the standard or alternative genetic codes. The ORF translation in three frames is achieved by sliding the translational frame one base at a time. Because the genetic code is triplet, moving by three bases will find all possible frames. Figure 7.6A shows the graphics of computational translation of mouse *Slco1a6* mRNA in six frames. When the longest predicted ORF (top frame) is clicked, the sequence and other details of the sequence are displayed (Figure 7.6B). The entire sequence is not displayed in the figure. Clicking the "SixFrames" link shows the six frames (Figure 7.6C). In each of these frames, the blue vertical lines represent the in-frame ATG codons and the red lines represent in-frame STOP codons. As is evident, each of these frames except the top one is full of in-frame stop codons. The total number of entries on the right-hand side (15), each with a small blue square, corresponds to the total number of translational reading frames present in all six frames combined; hence, each entry on the right corresponds to one translational reading frame. Clicking any blue square reveals the corresponding translational reading frame (both turn red), and the sequence of the reading frame is revealed.

There are many online tools available for the prediction of promoters and *cis*-regulatory elements. These programs are not all trained on the same training data set; consequently, the prediction outputs may not be identical. Thus, it is a good idea to check the prediction using multiple programs to find out at least the common elements predicted by different programs. *It should be remembered that the bioinformatic predictions of the* cis*-regulatory elements (regulating transcription) as well as the translation initiation site (i.e. the beginning of the ORF) need to be experimentally verified. A more than 10% error rate in computationally predicted ORFs compared to experimentally derived values has been reported.* The errors are due to the variation in predicting the translation initiation site. Such error is partly due to

[k]Tatiana Tatusov and Roman Tatusov are credited on the ORF Finder home page.

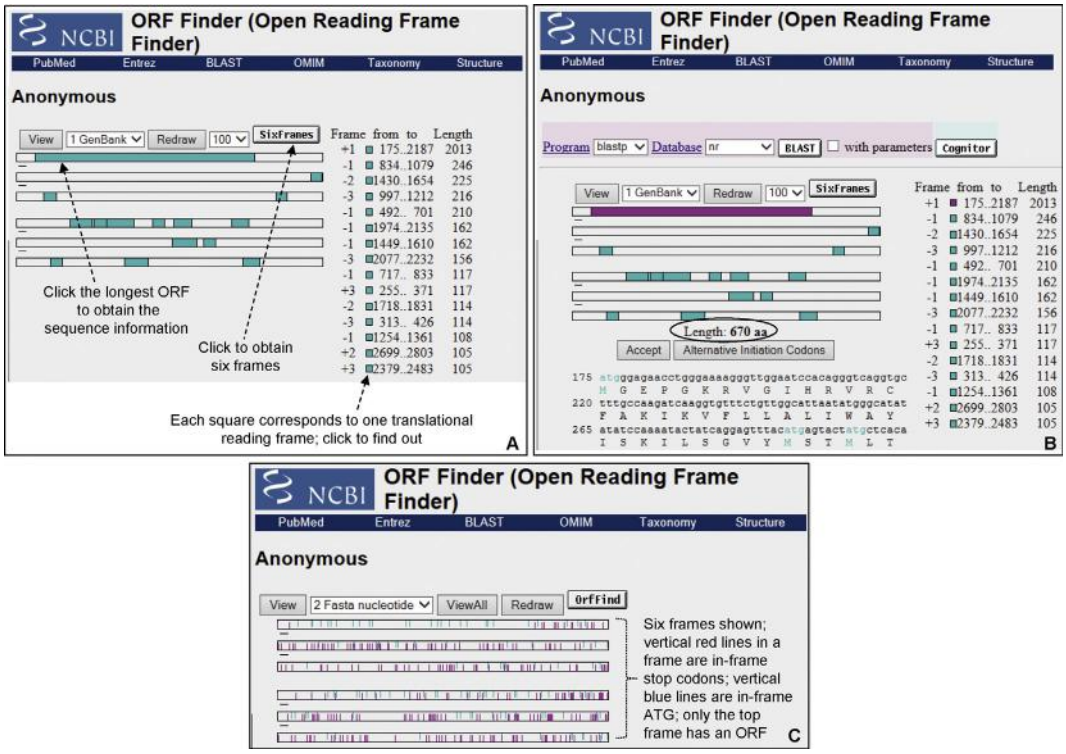


FIGURE 7.6 **NCBI ORF Finder.** (A) Computational translation of mouse *Slco1a6* mRNA in six frames, three sense and three antisense. (B) When the longest predicted ORF (top frame) is clicked, the sequence and other details of the sequence are displayed. Only the upper portion of the entire sequence is displayed. (C) Clicking the "SixFrames" link shows the six frames.

the ORF-prediction algorithm used, and partly due to the taxon examined. For example, genomes having high G + C content are particularly susceptible to ORF-prediction errors because of the existence of the alternative start codon GTG.[27]

Some of the publicly available online tools for the prediction of promoters, *cis*-regulatory elements, transcription start sites, translation initiation sites, and the ORF are listed in Table 7.1. There are many more prediction tools available. The reader can use these tools to obtain a rapid prediction about an input sequence, and compare the predictions of different tools.

7.5 RESTRICTION-SITE MAPPING OF THE INPUT SEQUENCE

Experiments involving DNA often require the experimenter to use various restriction enzymes. Restriction enzymes may be used to simply cut the DNA for gel electrophoresis or for advanced manipulation of DNA, such as making a vector, or a transgenic or knockout construct. Two online resources that can be used to analyze various restriction-enzyme cutting sites and generate a restriction map of an input DNA sequence are Webcutter 2.0[1] (http://rna.lundberg.gu.se/cutter2/) and NEBCutter 2.0[38] (http://tools.neb.com/NEBcutter2/).

7.6 RNA SECONDARY-STRUCTURE PREDICTION

RNA is single stranded but it can form significant secondary structure because of intrastrand base pairing. The three-dimensional shape of an RNA is its secondary structure. Some secondary structures observed in RNA are **short duplexes**, **stem−loops** (**hairpin stem−loops**), **bulges**, **internal loops**, **pseudoknots**, etc. (Figure 7.7A). The secondary structure of an RNA plays an important role in its maturation, regulation, and function. In fact, the formation of RNA secondary structure is the key to some of its functions regulating gene expression. For example, during **translational reprogramming**, or **recoding**, the gene-encoded reading frame is altered during translation, which allows for the generation of multiple ORFs from the same basic ORF encoded by the gene. This is achieved by

[1]©1997 Max Heiman.

TABLE 7.1 Some Online Tools for Prediction of Promoters, *Cis*-Regulatory Elements, Transcription Start and Initiation Sites, and the ORF

Online Analysis Tool	Comments and URL
BPROM	**Bacterial promoter prediction**. A SoftBerry utility that predicts putative transcription start positions of bacterial genes regulated by sigma70 promoters. The prediction accuracy is about 80%; the specificity is also about 80% when tested on equal numbers of promoter and non-promoter sequences. It uses the signal and content information of the sequence (e.g. consensus sequence). BPROM should be run on a region between two neighboring ORFs located on the same strand, or on a sequence upstream from an ORF (most promoters are located within 150 bp upstream of the ORF). BPROM should not be used for whole genomes, to avoid the many false positives (http://linux1.softberry.com/berry.phtml?topic=bprom&group=programs&subgroup=gfindb)
Virtual Footprint	**Prokaryotic promoter prediction**. Virtual Footprint is a software suite for analyzing transcription-factor-binding sites in whole bacterial genomes and their underlying regulatory networks. The result is a list of potential binding sites and corresponding genes defining the whole regulon. There are two types of analysis: analysis of a whole prokaryotic genome with one regulator pattern, and analysis of a promoter region with several regulator patterns[28] (http://www.prodoric.de/vfp/vfp_promoter.php)
BDGP (Berkeley *Drosophila* Genome Project)	**Prokaryotic and eukaryotic promoter prediction.** Neural network promoter prediction (NNPP)-based. NNPP is method that consists mainly of two recognition features for predicting eukaryotic promoters; one for recognizing the TATA-box and one for recognizing the initiator element. Both features are combined into one output unit, which gives output scores between 0 and 1. The default score is set at 0.8. The prediction accuracy for prokaryotic promoters is greater than that for eukaryotic promoters[29] (http://www.fruitfly.org/seq_tools/promoter.html)
FindTerm	**Rho-independent-terminator prediction in the bacterial genome**. A SoftBerry utility that predicts terminators in the bacterial genome. The search utilizes certain known features of bacterial terminators, such as T-rich regions, possible combinations of spacer lengths, all hairpins etc., and the result output shows all putative terminators (http://linux1.softberry.com/berry.phtml?topic=findterm&group=programs&subgroup=gfindb)
Promoter 2.0	**Vertebrate pol II transcription start site (TSS) prediction**. The program builds on principles that are common to neural networks and genetic algorithms[30] (http://www.cbs.dtu.dk/services/Promoter/)
Tfsitescan	**Eukaryotic promoter sequence and putative transcription-factor-binding site prediction**. Works best with sequences of ~500 nt. The output is in graphic display and shows expectation scores for the putative binding sites[a] (http://www.ifti.org/cgi-bin/ifti/Tfsitescan.pl)
SoftBerry Search for promoters/ functional motifs	**SoftBerry utility providing a suite of prediction tools for promoter/functional motif prediction**. For example: **1.** Plant promoter prediction (**TSSP**) **2.** Human pol II promoter prediction (**TSSG** and **TSSW**) **3.** Human promoter prediction (**FPROM**) **4.** Promoter prediction using orthologous sequences in eukaryotic genome (**PromH(G)** and **PromH(W)**) **5.** Regulatory motif prediction (**Nsite**) (http://linux1.softberry.com/berry.phtml?topic=index&group=programs&subgroup=promoter)
WWW Signal Scan	**Eukaryotic transcriptional elements prediction** based on scoring homologies of published *cis*-regulatory transcriptional signal sequences (e.g. in TFD, TRANSFAC databases) in the input sequence[b,31] (http://www-bimas.cit.nih.gov/molbio/signal/)
WWW Promoter Scan	**Eukaryotic promoter prediction** based on scoring homologies with eukaryotic pol II promoter sequences. If the program finds a putative promoter sequence, it reports the sequence range of the putative promoter, including the TATA box (if present) and the estimated transcription start site[32] (http://www-bimas.cit.nih.gov/molbio/proscan/)
Human Core-Promoter Finder	**Transcription start site (TSS) prediction in human core-promoters**. The input genomic DNA sequence should be longer than 240 bp and less than 2001 bp. The functional core-promoter is assumed to span between −60 and +40 nt with respect to the TSS (+1). The program is able to localize a TSS to a 100-bp interval ~60% of the time[c]. (http://rulai.cshl.org/tools/genefinder/CPROMOTER/human.htm)

(*Continued*)

TABLE 7.1 (Continued)

Online Analysis Tool	Comments and URL
EP3 (Easy Promoter Prediction Program)	**Eukaryotic core promoter prediction**. Performs very well in identifying regions in human genes that are associated with transcription initiation. EP3 uses universal properties of the promoter to detect those regions in a whole-genome context[33] (downloadable) (http://bioinformatics.psb.ugent.be/webtools/ep3/)
Eponine	**Transcription start site prediction in mammalian genomic sequence**. A probabilistic method with good specificity and excellent positional accuracy. Eponine is estimated to detect > 50% of transcription start sites, with ~70% specificity[34] (downloadable from Sanger Center) (http://www.sanger.ac.uk/resources/software/eponine/)
Footprinter	**Prediction of regulatory elements in DNA sequences based on phylogenetic footprinting**. Phylogenetic footprinting method identifies regions of DNA that are highly conserved across a set of orthologous sequences[35] (downloadable from the University of Washington (Motif Discovery link) (http://bio.cs.washington.edu/software)
ORF Finder	**Open reading frame (ORF) prediction**. A very user-friendly ORF finder on the web. It is a graphical analysis tool that finds all ORFs in the input sequence, using the standard or alternative genetic codes. The putative ORFs are displayed in six frames, three sense and three antisense[d] (http://www.ncbi.nlm.nih.gov/gorf/gorf.html)
NetStart 1.0	**Translation initiation site prediction**. NetStart produces neural network predictions of translation start sites in vertebrate and *Arabidopsis thaliana* nucleotide sequences. The program has been trained on cDNA-like sequences; therefore, it shows better performance for cDNAs and ESTs. It has not been tested on genomic data[36] (http://www.cbs.dtu.dk/services/NetStart/)
ATGPr	**Translation initiation site prediction**. ATGpr can be used to predict whether an initiation codon is present or absent in a piece of cDNA, and predict which ATG is the initiation codon for cases where there are multiple ATG codons. The method uses linear discriminant analysis, and has been tested on a non-redundant data set of 660 sequences[37] (http://atgpr.dbcls.jp/)

[a] *Made available by the Institute for Transcriptional Informatics (IFTI) at the IFTI-MIRAGE website.*
[b] *WWW implementation by Robin Hart and Rao Parasa.*
[c] *The web version is offered by Michael Zhang.*
[d] *Tatiana Tatusov and Roman Tatusov are credited on the ORF Finder home page.*

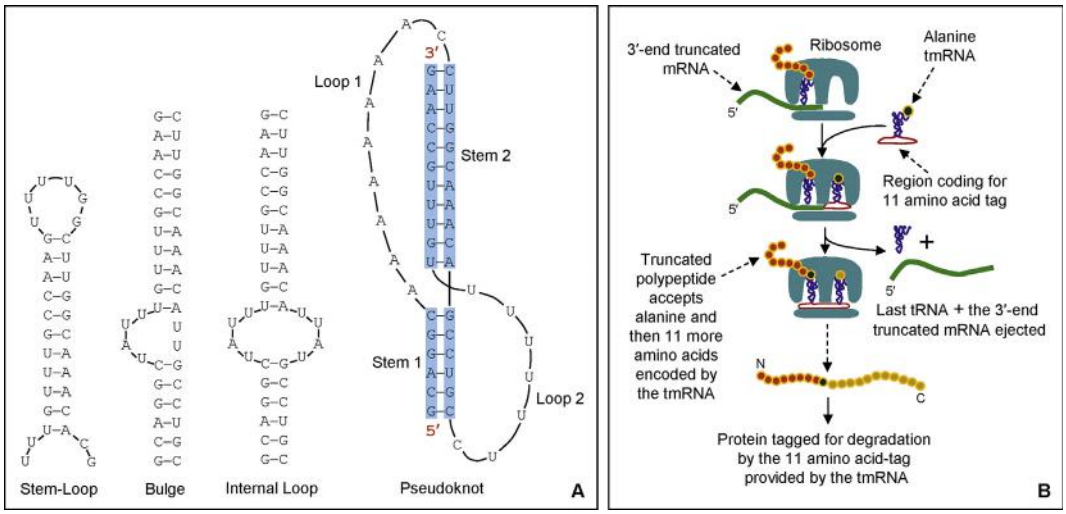


FIGURE 7.7 RNA secondary structure. (A) Some secondary structures of RNA. RNA pseudoknots can be more complex than the one shown here. (B) The transfer-messenger RNA (tmRNA; 10Sa RNA) and *trans*-translation. Alanine-charged tmRNA helps resume translation of a 3′-end-truncated mRNA by first providing alanine and then providing its own coding sequence, which adds the 11-amino-acid sequence to the C-terminal of the previously translated truncated polypeptide. The 11-amino-acid sequence tags the protein for degradation.

switching the reading frame during translation by one base, the so-called −1 or +1 frameshift mechanism. The efficiency of frame shifting is directly correlated with the extent of ribosomal pause. The *cis*-acting structural motifs of the mRNA that apparently facilitate ribosomal pause and consequent frame shifting include a heptanucleotide **slippery sequence** at the shift site, and a **pseudoknot** secondary structure that begins five or six nucleotides downstream from the shift site.

It is well recognized that the secondary structures of tRNA and ribozyme are necessary for their function. The telomerase RNAs in different species of ciliates and vertebrates have very different sequences but they all fold into similar secondary structures, strongly suggesting that the conserved secondary structure is important for the specific function of telomerase RNA.[39]

The **transfer-messenger RNA** (**tmRNA**) in bacteria that mediates ***trans*-translation** also has a unique secondary structure that is needed for its function. The phenomenon of *trans*-translation involves **ribosomal hopping**, involving two distinct RNA templates in succession. In various bacteria, this 10Sa RNA species acts as an alanyl tRNA because it is charged with alanine by alanyl-tRNA synthetase. The 10Sa RNA also has mRNA features because it encodes an 11-amino-acid oligopeptide that tags proteins for degradation. Because 10Sa RNA possesses such dual features of tRNA and mRNA, it is called transfer-messenger RNA (tmRNA). When ribosomes carrying a peptidyl-tRNA pause at the end of a 3′-end-truncated mRNA and accept the alanyl-10Sa RNA molecule as the alanyl-tRNA surrogate, the alanyl-10Sa RNA first provides the alanine and then provides its internal reading frame for the translation of the 11-amino-acid oligopeptide tag. This results in the incorporation of the oligopeptide tag to the already synthesized truncated polypeptide, which is thus flagged for degradation (Figure 7.7B).

An example of the importance of RNA secondary structure in its maturation is the biogenesis of micro RNA (miRNA). Transcription of a miRNA gene produces primary miRNA (pri-miRNA), which has a stem−loop structure with additional internal loops. Processing of pri-miRNA in the nucleus by Drosha produces precursor miRNA (pre-miRNA) which has a shortened stem−loop structure compared to pri-miRNA. Processing of pre-miRNA in the cytoplasm produces miRNA. The secondary structure of these precursors is necessary for the biogenesis of miRNA. An RNA hairpin is an essential secondary structure of RNA that can guide RNA folding, determine interactions in a ribozyme, protect mRNA from degradation, serve as a recognition motif for RNA-binding proteins, and also regulate gene expression.[40] A recent study using a high-throughput sequencing-based structure-mapping approach in *Drosophila melanogaster* and *Caenorhabditis elegans* transcriptomes identified both paired (double-stranded) and unpaired (single-stranded) RNA components. The authors observed that these RNAs are significantly correlated with specific epigenetic modifications. They also uncovered highly base-paired RNAs, many of which likely encode lncRNAs (long noncoding RNAs). Additionally, they identified conserved features of mRNA secondary structure that indicate that RNA folding demarcates regions of protein translation. Finally, they identified and characterized 546 mRNAs whose folding pattern is significantly correlated between these two species even though they are so far apart in evolution, thereby suggesting that the observed mRNA secondary structure has some function.[41]

The formation and stability of RNA secondary structure are dependent on a number of factors. For example, more GC base pairs and longer stem regions result in greater stability of the secondary structure, whereas unpaired bases, such as bulges and internal loops, tend to decrease the stability of the secondary structure. Similarly, the formation of hairpin loops with more than 10 or less than 5 bases requires more energy; hence, it reduces the stability of the secondary structure. In general, a secondary structure is thermodynamically favored (hence more stable) if its formation releases energy (ΔG is negative, i.e. negative free energy). Conversely, a secondary structure becomes thermodynamically unfavorable (hence less stable) if its formation requires energy (ΔG is positive, i.e. positive free energy). This fact is used to predict the secondary structure of a particular sequence. Free energies are additive, so one can determine the total free energy of a secondary structure by adding all the component free energies (as kcal/mole).

Given the importance of RNA secondary structure, a number of prediction algorithms have been developed and are available online to analyze an RNA sequence to predict its putative secondary structure. Some of the publicly available online tools for RNA secondary-structure prediction are listed in Table 7.2.

Secondary-structure-predicting algorithms often generate an output made up of brackets and dots (sometimes brackets and hyphens). The character string denoted by brackets and dots represents the number of residues of the input sequence and their base-pairing status. In the bracket notation, the base pairs are indicated by opening and closing parentheses. Some program outputs have these brackets and dots above the bases. Some program outputs may contain the base-pairing probability as well (Figure 7.8).

RNA secondary-structure prediction based on thermodynamic parameters has been in practice since the 1980s. Such predictions owe their success to the application of various experimentally verified thermodynamic parameters. However, like every other method, thermodynamic predictions have their limitations. In

TABLE 7.2 Some Online Tools for RNA Secondary-Structure Prediction

Online Analysis Tool	Comments and URL
RNAfold	RNAfold predicts secondary structures of single-stranded RNA or DNA sequences based on the classic minimum-free-energy algorithm of Zuker and Stiegler[42] as well as the partition-function algorithm of McCaskill.[43] Current limits are 10,000 nt for minimum-free-energy-only predictions and 7500 nt for partition-function calculations. The server function can be tested using the sample sequence provided[44] (http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi)
RNAsoft	RNAsoft is a collection of online services for the computational prediction and design of RNA/DNA structures based on a standard free-energy model.[45] The underlying algorithms have been designed and implemented by members of the Bioinformatics, Empirical and Theoretical Algorithmics (BETA) Lab at the Department of Computer Science of the University of British Columbia (http://www.rnasoft.ca/)
CONTRAfold	CONTRAfold is a novel secondary-structure prediction method based on conditional log-linear models (CLLMs), a flexible class of probabilistic models with high prediction accuracy[46] (http://contra.stanford.edu/contrafold/server.html)
RNAstructure	RNAstructure uses several secondary-structure prediction algorithms, including thermodynamic and partition-function algorithms. It is a complete package for RNA and DNA secondary-structure prediction and analysis. It can take different types of experiment mapping data to constrain or restrain structure prediction[47] (http://rna.urmc.rochester.edu/RNAstructureWeb/)
IPKnot	IPKnot performs integer-programming (IP)-based prediction of RNA pseudoknots. IPknot can also predict the consensus secondary structure when a multiple alignment of RNA sequences is given[48] (http://rna.naist.jp/ipknot/)
CYLOFOLD	RNA secondary-structure (including pseudoknot) prediction tool. Some examples of RNA sequences are provided that can be used to perform a test run. The bracket notation output is in brackets and hyphens instead of brackets and dots* (http://cylofold.abcc.ncifcrf.gov/)
CentroidHomfold and CentroidFold	CentroidHomfold predicts the secondary structure of an input RNA sequence by employing automatically collected homologous sequences of the target[49,50] CentroidFold uses the CONTRAfold model as the default setting to calculate base-pairing probabilities, and predicts RNA secondary structure using a γ-centroid estimator. Currently, the input sequence should be less than or equal to 2000 bases[51] (http://www.ncrna.org/)
pknotsRG	pknotsRG is a tool for predicting RNA secondary structures, including the class of simple recursive pseudoknots. It uses the thermodynamic energy model extended by some pseudoknot-specific values.[52] The program on the BiBiserv is limited to sequences of length up to 800 bases (http://bibiserv.techfak.uni-bielefeld.de/pknotsrg/submission.html) pknotsRG will be discontinued and replaced by pKiss in the near future (http://bibiserv2.cebitec.uni-bielefeld.de/pkiss)

*Made available by Dr Bruce A. Shapiro and his research group at the National Cancer Institute, Frederick, MD.

order to circumvent this problem, various probabilistic and statistical models have been developed that seemingly outperform thermodynamic-parameter-based predictions.[54] Figure 7.8A shows secondary-structure prediction of the input RNA sequence based on minimal-free-energy (MFE) calculation by pknotsRG-MFE. Figure 7.8B shows secondary-structure prediction of the input RNA sequence based on the partition functions and base-pair probabilities model[m] by IPKnot; the output is the McCaskill model. In contrast, Figure 7.8C shows an alternative output by IPKnot, based on a conditional log-linear probabilistic model known as CONTRAfold.[46] The figure also shows the respective bracket notations of each model. The free energy of a secondary structure is calculated by summing energy parameters of respective loop substructures, which can be experimentally determined and computationally estimated.[55]

7.7 MICROARRAY ANALYSIS

Most researchers doing microarray experiments use the analysis software provided by the manufacturer of

[m]Partition functions estimate statistical properties of a system with respect to thermodynamic probabilities, such as melting behavior and base-pair probabilities; properties and probabilities of a myriad of alternative structures in thermodynamic equilibrium.

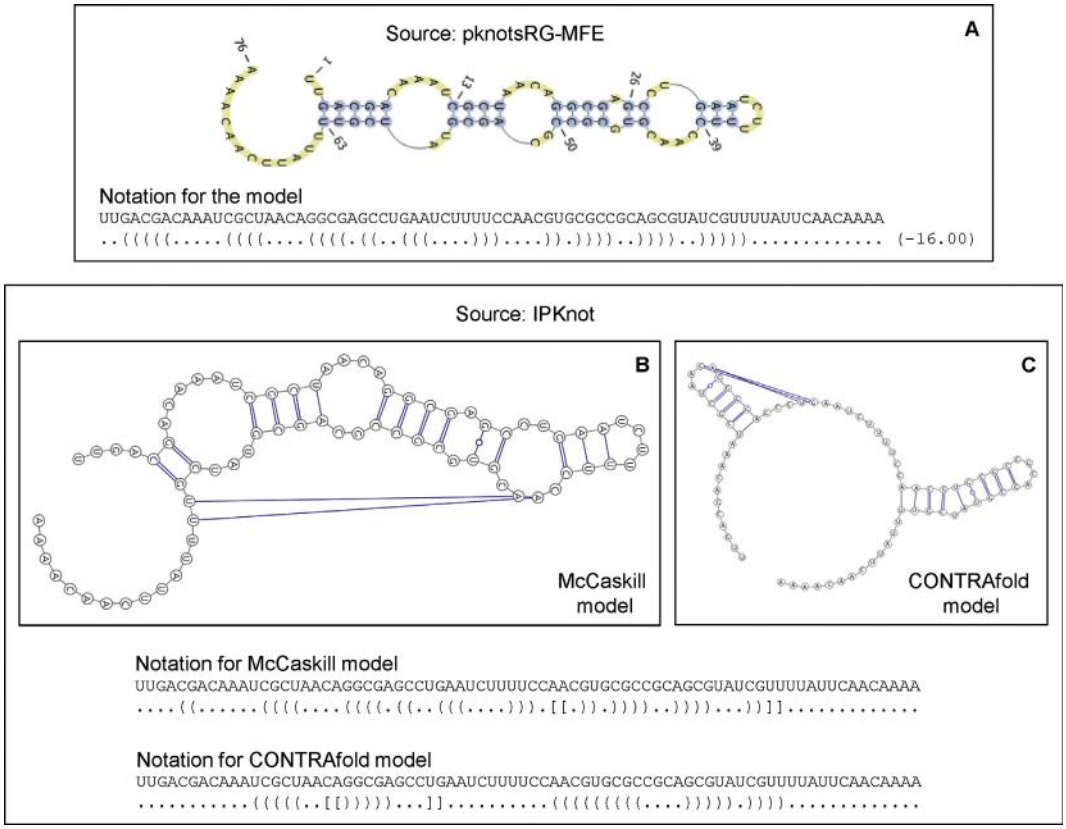


FIGURE 7.8 **RNA secondary-structure prediction by two web-based programs using default parameters.** (A) Prediction using pknotsRG-MFE of the Bielefeld University Bioinformatics Server (BiBiServ).[53] (B and C) Integer-programming (IP)-based prediction using IPKnot of the Nara Institute of Science and Technology, Japan. The default is the McCaskill model shown in (B); an alternative is the CONTRAfold model shown in (C). The respective bracket notations are also shown. In the bracket notation, the base pairs are indicated by opening and closing parentheses. Residues not involved in base pairing are denoted by dots. Every base with a "(" notation below is base-paired with a downstream base with a ")" notation below it. Some program outputs may also contain the base-pairing probability.

the microarray platforms. Therefore, some basic concepts of microarray data analysis are discussed here.

An outline of the microarray technique has been discussed in Chapter 3. The system described is also called **two-color** or **two-channel microarrays** because it involves the use of two different fluorescently labeled probes; one labeled with the fluorescent dye Cy3[n] (fluorescein, with fluorescence emission at ~565 nm; hence green), and the other labeled with the fluorescent dye Cy5 (biotin, with fluorescence emission at ~665 nm; hence red). The goal of DNA microarray is to screen the expression profile of genes, and the technique is useful because of its high-throughput nature.

Scanning of the microarray slide is the first step following post-hybridization processing and drying. The slide is scanned by a laser scanner hooked to a confocal laser microscope. The laser excites each spot in the microarray and the fluorescence emission is captured through a photomultiplier connected to the confocal laser microscope. The scanning is done in both green and red channels (at both wavelengths), each producing an individual image. The individual images are merged to obtain a composite image, in which the spot images can be green, red, or yellow; yellow means there are equal amounts of green and red fluorescence. However, the color of all the spots may not be perfectly green, red, or yellow, and may show a range, such as black/dark blue, blue, green, yellow, orange, and red. The image is usually reported as the ratio of Cy5 and Cy3 fluorescence intensity.

The next step is **image processing**. The features on the array—that is, what is contained in each grid/spot—are already defined. The image captured is a

[n]Cy3 (cyanine 3) dye is red (dark pink) in color and Cy5 (cyanine 5) dye is blue in color. However, the absorption and florescence emission maxima for Cy3 are ~547 and ~565 nm, respectively, whereas those of Cy5 are ~647 and ~665 nm, respectively. Hence, Cy3 is detected as green florescence in the green channel, and Cy5 is detected as red florescence in the red channel. Therefore, the physical colors of these dyes are not to be confused with their fluorescence emission colors.

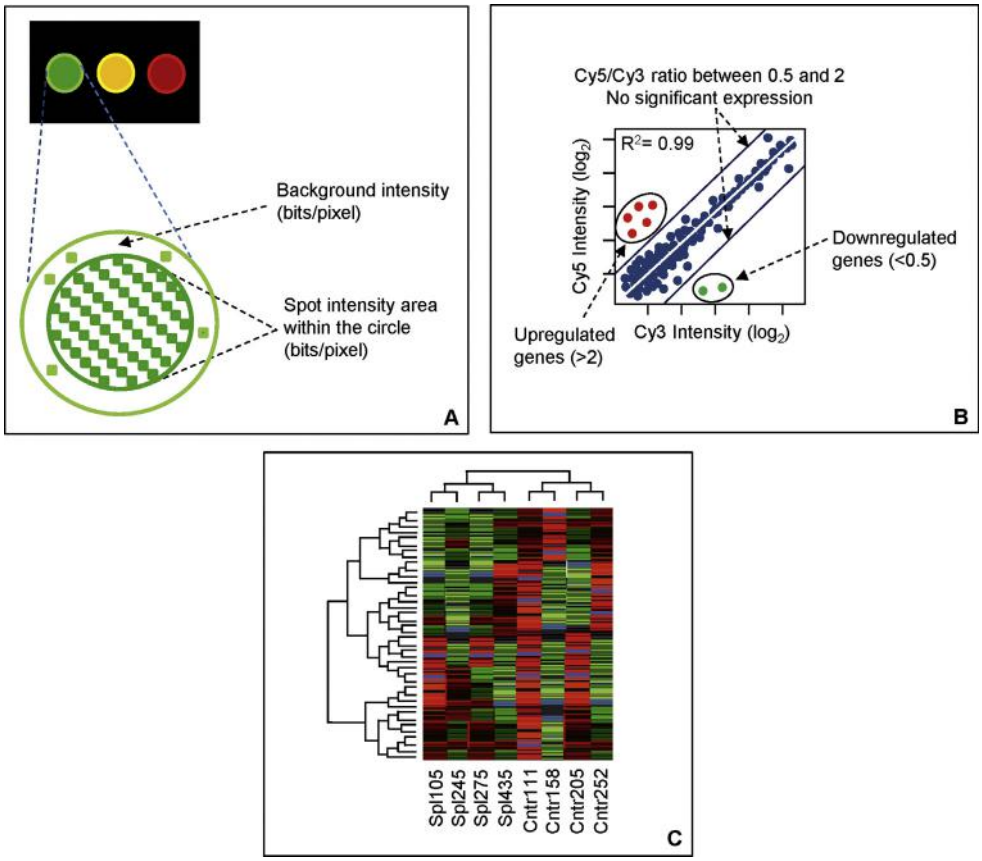


FIGURE 7.9 **Microarray image normalization and clustering.** (A) The captured microarray image is digital in nature. A digital image is composed of pixels, its smallest individual elements; each pixel has a value that represents the brightness of a given color at a point. Microarray scanners typically capture the color images as 16 bits/pixel. Therefore, the higher the bits/pixel, the greater is the color depth. For each spot, the true signal intensity is determined by subtracting the median background value. (B) Following image processing, the data are normalized in order to adjust for differences in labeling and detection efficiencies for Cy5 and Cy3. In the Lowess (locally weighted scatterplot smoothing; regression) method of normalization, it is assumed that mRNAs from closely related samples should cluster, producing a straight line in a scatter plot of Cy5 versus Cy3 intensities (or their $\log_2$ values), with a slope value close to 1. If such linearity is missing, the data are normalized to create the desired slope. If the cutoff for significant changes in expression is set at 2, the values ranging between 0.5 and 2 are not considered to be significant. (C) Hierarchical clustering dendrogram and heat map commonly used to display microarray data. The dendrogram represents relationships amongst genes and the branch lengths represent the degree of similarity in terms of their expression. In this method, using a distant matrix method, the algorithm first joins the two closest genes into a cluster; then the next most similar genes are joined together, and so on. This repetitive agglomeration first creates smaller clusters, which are similarly joined to form larger clusters. This process continues until all of the genes are joined into one giant cluster.

digital image, which is a rectangular array of intensity values in the spot; each intensity value is a pixel. The color depth is expressed as bits/pixel; hence the higher the bits/pixel, the greater is the color depth. During image processing, the spot boundaries are defined so that the true signal and the background values can be assigned. The median background value is then subtracted to obtain the true signal value (Figure 7.9A). The true signal is the fluorescence intensity due to specific hybridization, whereas the background signal is the fluorescence intensity due to non-specific hybridization that has survived post-hybridization washing, as well as non-specific binding of the fluorescently labeled nucleic acid fragments to a "sticky" surface, or even any dirt on the slide.

The next step is **data normalization**. Following image processing and analysis, the data are normalized. The purpose of normalization is to adjust for differences in labeling and detection efficiencies for Cy5 and Cy3, as well as to adjust for any differences in the RNA samples. Without normalization, the Cy5/Cy3 ratio could be artificially skewed. Normalized samples are ready for further analysis. Normalization can be done by (1) the total intensity normalization method, (2) the regression method, or (3) the ratio statistics method. The regression method is called the "**Lowess**" (locally weighted scatterplot smoothing) method, which is a locally weighted linear regression used to estimate systemic biases in the data. In the regression method, which is often used, it is assumed that mRNAs from closely related samples should be

expressed at similar levels. Under this assumption these mRNAs should cluster, producing a straight line in a scatter plot of Cy5 versus Cy3 intensities (or their $\log_2$ values). The scatter plot is thus a **ratio-intensity (R-I) plot**. If the labeling and detection efficiencies were the same for both samples, the slope of the scatter plot should be 1 or close to 1. If such linearity is missing, Lowess normalizes the data to create the desired slope. Normalized data are then used to report the expression ratios of genes between the samples, such as between the control and the experimental sample, or between normal and disease tissue samples. The cutoff for significant changes in expression can be set at 2—that is, values ranging between 0.5 and 2 are not considered to be significant. In this scenario, >2-fold difference means significant upregulation of expression, and <0.5-fold difference means significant downregulation of expression. However, these can be adjusted depending on the experiment, as well as the variability of the data (Figure 7.9B).

Cluster analysis of microarray data is a very widely used way to demonstrate gene-expression differences between the objects being studied, such as normal versus diseased tissue, control versus treatment group. Because genes involved in a common pathway, genes that are coordinately regulated, and genes involved in similar physiological response may be expressed similarly, the expressions of these genes are related. Microarray expression data can be used to find the relationships between genes in terms of their expression and consequently categorize such genes. This method is called cluster analysis. Therefore, in cluster analysis, the genes that are upregulated or downregulated in response to a specific condition (exposure, disease), can be identified and the biological relevance of such gene expression can be further investigated. Additionally, such gene expression can also be used as a biological marker of specific physiological response. Clustering can be supervised or unsupervised. In **supervised clustering**, the expression pattern of the gene(s) is known and this knowledge is used to group genes into clusters. In **unsupervised clustering**, there is no prior knowledge regarding the expression pattern of the gene(s) in a specific condition. Similar expression profiles are then connected to form the groups until all expression data have been included.

The most widely used method of unsupervised clustering is known as **hierarchical clustering**. Hierarchical clustering is commonly used in microarray as well as in phylogenetic analysis because it computes a tree (dendrogram). In DNA microarray analysis, the tree represents relationships amongst genes and the branch lengths represent the degree of similarity in terms of their expression. *Hierarchical clustering is a bottom-up agglomerative approach*. In this method, the algorithm starts by calculating the pairwise distance matrix for all of the genes in the so-called "gene space." Next, the algorithm joins the two genes that are the closest into a cluster. If there are multiple gene pairs that share the same degree of similarity, then the first cluster is formed based on some predetermined rule. Then, the next most similar genes are joined together, and so on. Once the small clusters are formed, the algorithm computes the pairwise distance matrix for all of the clusters in the so-called "cluster space." Next, the algorithm joins the two small clusters that are the closest into a larger cluster. This repetitive agglomeration process continues until all of the genes are joined into one giant cluster (Figure 7.9C). The other means of unsupervised clustering is known as ***k*-means clustering**. Contrary to the hierarchical clustering, *k-means clustering is a top-down divisive approach*. Obviously it does not produce dendrograms; instead, in this method data are partitioned into a prespecified set of *k*-clusters. Another divisive clustering method based on neural networks is **self-organizing maps** (SOM). The *k*-means clustering and SOM methods will not be further discussed here.

The **TM4 suite** of tools (http://www.tm4.org/)[56] consists of four major applications, Microarray Data Manager (MADAM), The Institute for Genomic Research (TIGR) Spotfinder, Microarray Data Analysis System (MIDAS), and Multiexperiment Viewer (MeV). **TIGR Spotfinder** is a microarray image-processing and quantification tool, whereas **TIGR's MIDAS** is a normalization and filtering tool. Another microarray image-analysis tool, **ScanAlyze**, is provided by the Eisen Lab at http://rana.lbl.gov/EisenSoftware.htm. The same link at Eisen Lab also provides **Cluster** and **TreeView**, which are cluster-analysis and graphical visualization software tools. They can perform hierarchical clustering, self-organizing maps (SOMs), *k*-means clustering, and principal component analysis.[57] Another web server for the normalization and standardization of DNA microarray data is **SNOMAD**[o] (http://pevsnerlab.kennedykrieger.org/snomadinput.html), made available by the Pevsner Lab at Johns Hopkins University School of Medicine.

7.8 DETECTION OF SEQUENCE POLYMORPHISM AND THE SNP DATABASE

Mutations can be point mutations, small deletions and insertions, or large-scale changes in the chromosome. Point mutations can be common or rare types of mutations. By definition, a point mutation that occurs

[o]© 2000 by Carlo Colantuoni, George Henry, and Jonathan Pevsner.

in at least 1% of the population is called a **single nucleotide polymorphism** (**SNP**; pronounced "**snip**").

SNPs constitute a very important class of mutations; they generally occur at a frequency of at least 0.1% (1/1000 bases) in the genome but may occur more frequently in certain regions. In the human genome, >65% of all SNPs involve C→T transition mutations. A set of linked SNPs that tend to inherit together as a unit is referred to as **SNP haplotype**. SNPs can occur in both coding and noncoding regions of genes. SNPs in the coding region may alter the characteristics of the protein while SNPs in the regulatory regions may alter the expression profile of genes.

Some SNPs can predispose people to disease or influence their response to a drug. For example, two SNPs in the *ApoE* gene result in three possible alleles of the gene: *E2*, *E3* (wild type), and *E4*. The corresponding protein product of each gene differs by one amino acid ($ApoE2^{C112,C158}$, $ApoE3^{C112,R158}$, $ApoE4^{R112,R158}$). Individuals inheriting two *E4* alleles have the highest chance of getting Alzheimer's disease, while those inheriting two *E2* alleles are the least likely to get the disease; so the order of risk associated with various *ApoE* alleles is *E4* > *E3* > *E2*. Apparently, one amino acid change in the ApoE protein alters its structure and function enough to influence the risk of disease development associated with each allele.[58]

The **International HapMap Project** is a multi-country (USA, UK, Canada, Japan, China, and Nigeria) effort to identify and catalog genetic similarities and differences in human beings. In doing so, the project expects to identify and catalog SNPs and SNP haplotypes that confer susceptibility/resistance to disease or therapy.

Sequence polymorphisms can be detected through pairwise alignment of two DNA sequences from two individuals. Deep resequencing of specific regions of the genome can also identify sequence polymorphisms.

The NCBI SNP database (**dbSNP**; http://www.ncbi.nlm.nih.gov/projects/SNP/ or http://www.ncbi.nlm.nih.gov/snp/) is the largest public database of short genetic variations (SNVs). The dbSNP is a broad collection of simple genetic polymorphisms, which includes single-base nucleotide substitution (**SNPs**), small-scale multi-base deletions or insertions (deletion−insertion polymorphisms or **DIPs**[P]), and retroposable element insertions and microsatellite repeat variations (also called short tandem repeats or **STRs**). Each dbSNP entry includes the sequence context of the actual polymorphism, such as the surrounding sequence; the occurrence frequency of the polymorphism (by population or individual); and the experimental method(s), protocols, and conditions used to assay the variation.[60]

A new submission to dbSNP is assigned a unique **ss#** (**submitted SNP ID number**). The submission is verified by alignment to the appropriate genomic contig. If several ss# entries map to the same position, the records are merged into a cluster that is given a unique **rs#** (**reference SNP cluster ID**).

A search was made for the mouse *Slco1a6* gene in dbSNP. The search produced 2092 hits as of July 2013 (Figure 7.10).

Selecting "Summary" from the "Display Settings" drop-down menu returns the summary of information on that SNP (figure not shown). Selecting "Graphic Summary" from the drop-down menu returns the display shown in Figure 7.10. Clicking "rs266211819" returns its **cluster report**. The top portion of the cluster report is shown in Figure 7.11A. The "Variation Class" field shows that it is a single nucleotide variation (SNV), the "RefSNP Alleles" field shows that the SNV is either A or C (circled). In other words, one of the alleles would be termed the "A" allele and the other allele would be termed the "C" allele, and the SNP is located on the "forward strand" ("Fwd"; circled). The information is organized into a few sections, such as GeneView, Map, etc. Figure 7.11B shows that rs266211819 is an intronic SNP. Clicking "view" in the "Neighbor SNP" field (circled in Figure 7.11A) shows that there are two SNPs within 100 bases upstream and four SNPs within 100 bases downstream of rs266211819 (Figure 7.12).

Figure 7.13 shows the graphic view of SNP rs266211819.

The SNP cluster page also has a section on the submitted SNP ID number (ss#) (Figure 7.14A). The ss370364874 has the longest flanking sequence and is shown. Clicking on the ss# (Figure 7.14A; circled) returns the details of the submitted SNP (Figure 7.14B). In the left-hand top corner there is "Submitter" information. The "Handle" field provides the submitter information. Clicking "SC_MOUSE_GENOMES" reveals the submitter contact information. In this case, the submitter is from the Wellcome Trust Sanger Institute, Cambridge, UK. In the right-hand top corner is "Resource Links." The submission can be viewed by clicking the "view" field (circled). Figure 7.15A shows the details of the original submission, including the SNP (A/C) as well as the 5′- and 3′-flanking sequences. Note that the original submission shows the SNP as A/C, but in the NCBI cluster report (the FASTA sequence part from the cluster report is displayed in Figure 7.15B) this (A/C) is replaced by M. This substitution of the original SNP is done following the IUPAC (International Union of Pure and Applied Chemistry) nucleotide codes shown in Table 7.3.

[P]DIP (deletion−insertion) or indel (insertion−deletion) polymorphisms consist of the presence or absence of short sequences (typically 1−50 bp).[59]

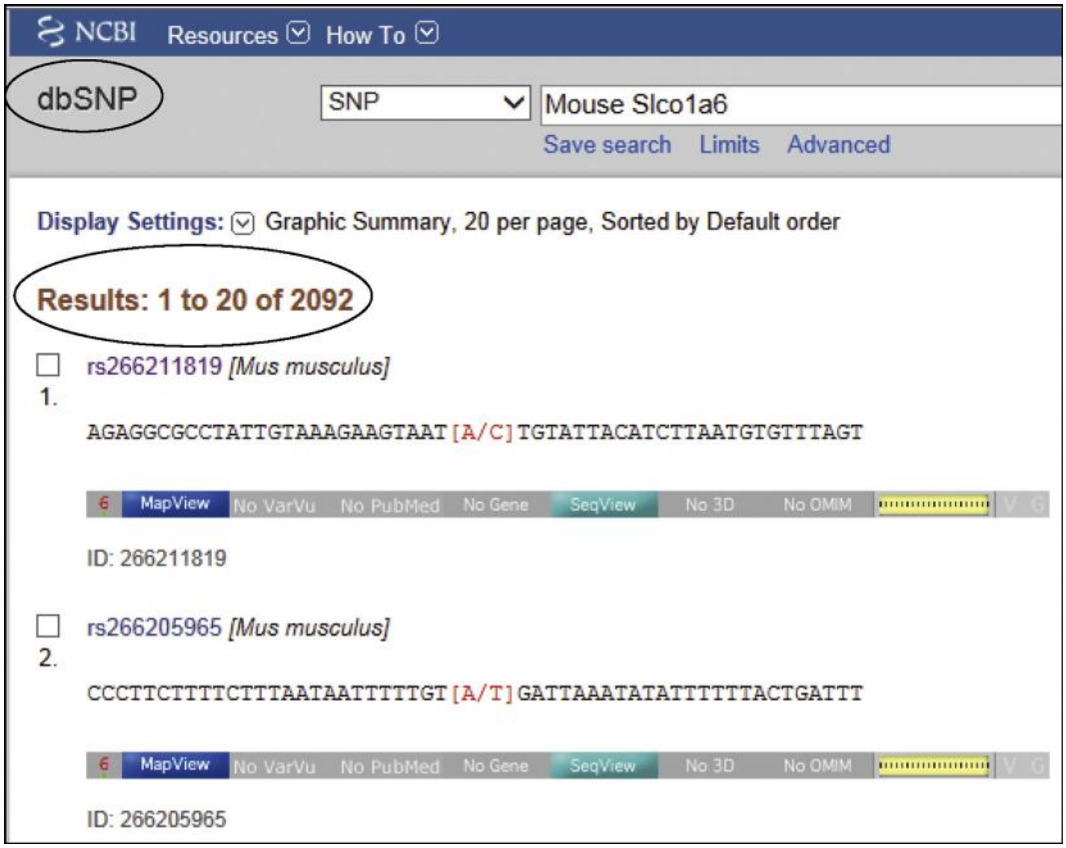


FIGURE 7.10 **A search for the mouse *Slco1a6* gene in the SNP database.**

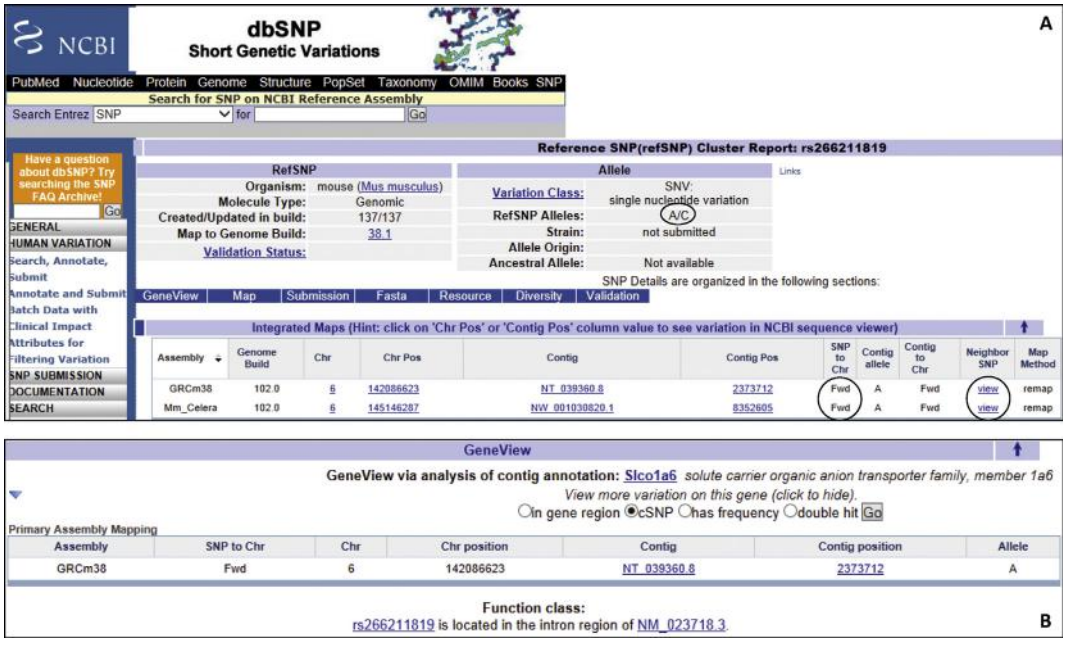


FIGURE 7.11 **Clicking the first entry rs266211819 returns its cluster report.** (A) The top portion of the cluster report is shown, see text for explanation; (B) GeneView shows that the rs266211819 is an intronic SNP.

Neighbor (within 100 bases) SNP for rs266211819:

distance (base)	rs	map weight	validation	assembly	Contig accession	Contig position
-76	rs240960243	1		Mm_Celera	NW_001030820.1	8352529
-62	rs244941523	1	N.D.	Mm_Celera	NW_001030820.1	8352543
0	rs266211819	1	N.D.	Mm_Celera	NW_001030820.1	8352605
34	rs228719522	1	N.D.	Mm_Celera	NW_001030820.1	8352639
54	rs215448038	1	N.D.	Mm_Celera	NW_001030820.1	8352659
72	rs236998816	1	N.D.	Mm_Celera	NW_001030820.1	8352677
74	rs254203577	1	N.D.	Mm_Celera	NW_001030820.1	8352679

Note:
- When distance is negative, it means the neighbor snp is upstream to the rs266211819.
- When distance is 0 and the snp is other than rs266211819, then it means these snps hit on the same contig position on the assembly, which means:
A: These rs will be merged in future builds.
B: Some snp that hit the same positions are not merged because they have different variation class.

FIGURE 7.12 **Neighboring SNPs of rs266211819.** Information retrieved by clicking "view" in the "Neighbor SNP" field circled in Figure 7.11A, showing six flanking SNPs.

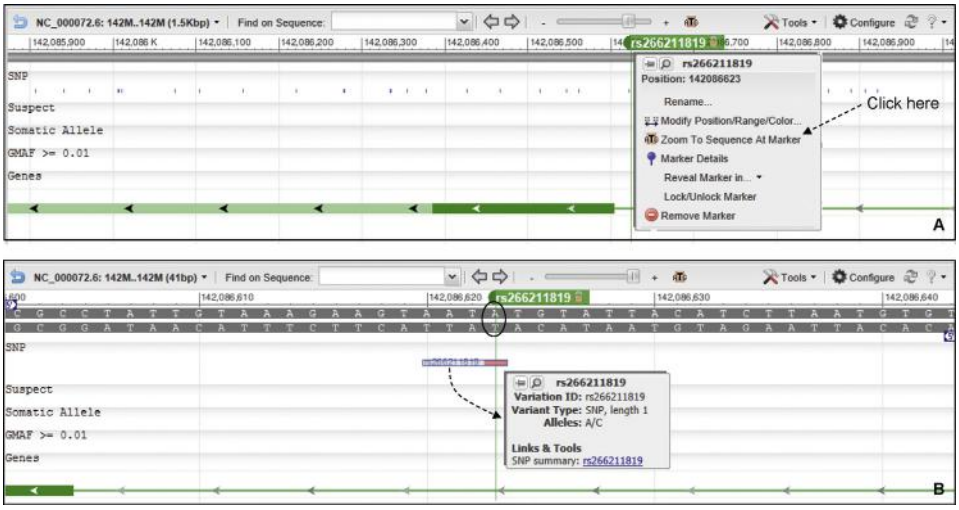


FIGURE 7.13 **The graphic view of rs266211819.** (A) Holding the cursor next to the green bar with the rsID (rs266211819) produces a drop-down menu. (B) Selecting "Zoom to Sequence At Marker" from this drop-down menu returns the sequence and the SNP. Selecting the bar with the rs# returns the drop-down menu shown. The drop-down menu contains information about the SNP (A/C).

A

Submitter records for this RefSNP Cluster

The submission **ss370364874** has the longest flanking sequence of all cluster members and was used to instantiate sequence for **rs266211819** during BLAST analysis for the cur

NCBI Assay ID	Handle|Submitter ID	Validation Status	ss to rs Orientation /Strand	Alleles	5' Near Seq 30 bp	3' Near Seq 30 bp	Entry Date	Update Date	Build Added	Molecule Type
ss370364874	SC_MOUSE_GENOMES|MGP_WTSI_6_142035144		fwd/	A/C	agatagaggcgcctattgtaaagaagtaat	tgtattacatcttaatgtgtttagtgaaag	04/20/11	04/20/11	137	Genomic

B

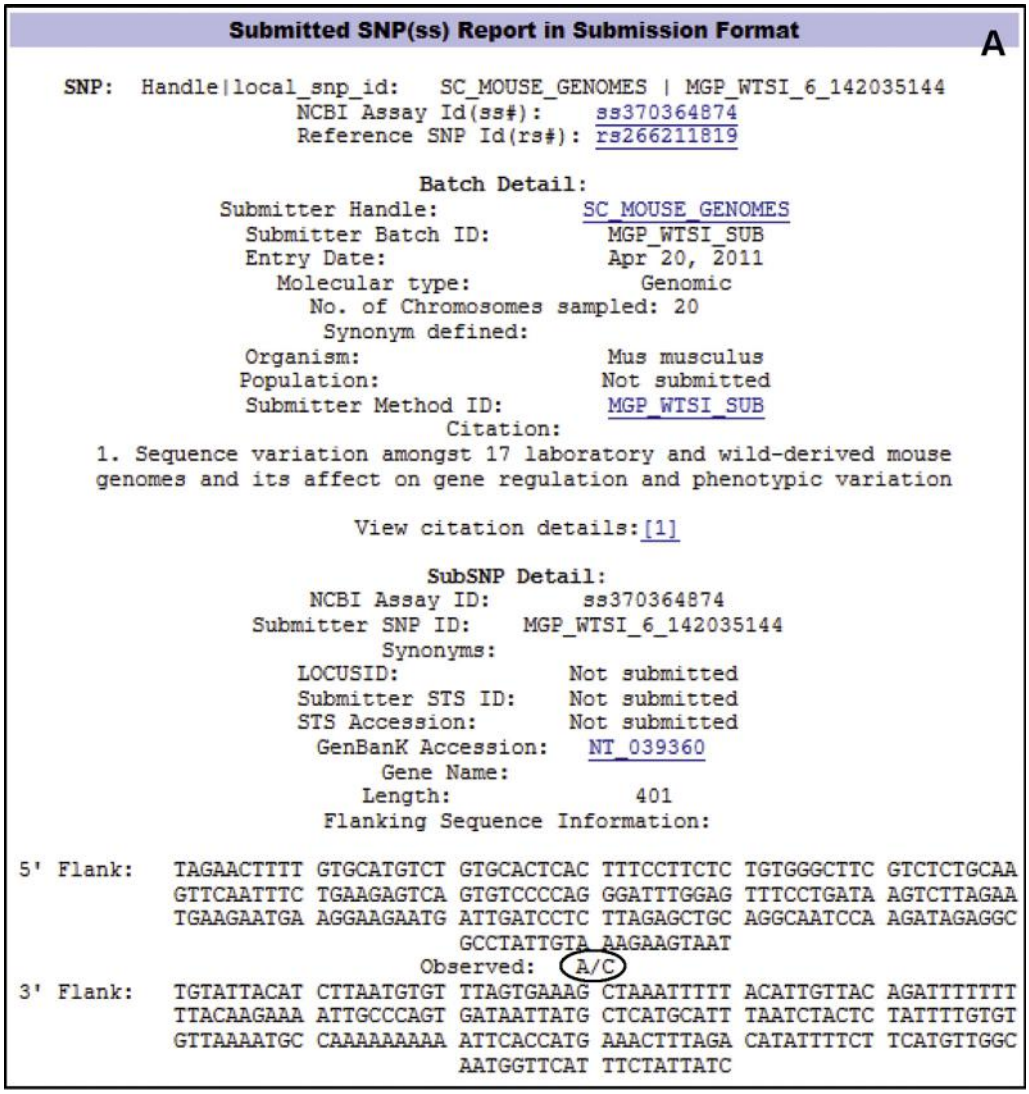
Submitted SNP(ss) Details: ss370364874

Submitter		Resource Links	
Handle	SC_MOUSE_GENOMES	GenBank Accession	NT_039360
Submitter SNP ID	MGP_WTSI_6_142035144	Submitted Gene Name	N.D
RefSNP(rs#)	rs266211819	Submitted Gene ID	N.D.
Submitted Batch ID	MGP_WTSI_SUB	Submitted SNP Synonyms	N.D
Submitted Date	Apr 20, 2011	Submitted linkout	N.D
Publication Cited	[1] Sequence variation amongst 17 laboratory and wild-derived mouse genomes and its affect on gene regulation and phenotypic variation	Submission report	view
First entry to dbSNP	Apr 20 2011 12:00:00:000AM		

Assay		Allele	
Species	Mus musculus	Observed Allele	A/C
Molecular Type	Genomic	Ancestral Allele	N.D.
Method	MGP_WTSI_SUB	Allele Origin	N/A
Ascertainment Samplesize	20	SNP Class	SNV
Population	N.D.	CpG Code	N.D.

Validation		Variation	
Validation Status	Not Validated	Frequency Submission	N.D.
HWE Goodness of Fit	not applicable	Genotype Summary	N.D.
Homozygote Detected		Genotype Submission	N.D.
PCR Confirmed		Haplotype	N.D.
In Expressed Sequence			

FIGURE 7.14 **Submitter information for a SNP ID number.** See text for details.



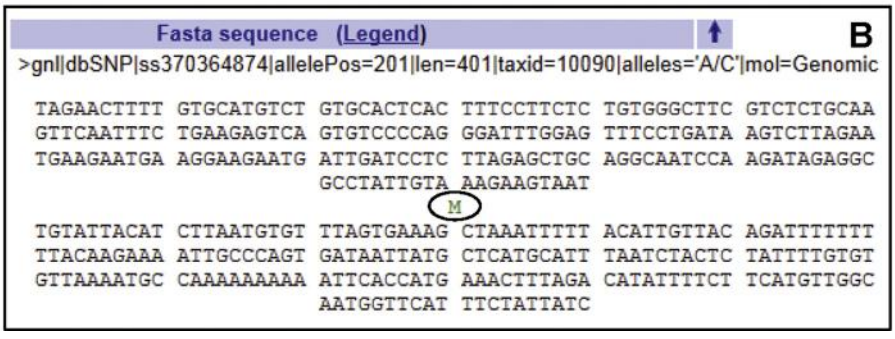


FIGURE 7.15 **IUPAC designation of the SNP in the database.** (A) The original submission showing the SNP (A/C) and the flanking sequence. (B) The substitution of A/C by M in the SNP database following the IUPAC nucleotide codes, as shown in Table 7.3.

TABLE 7.3 IUPAC Codes for Nucleotides

A = adenine	**T = thymine**	**G = guanine**	**C = cytosine**		
R = A/G	Y = C/T	S = G/C	W = A/T	K = G/T	M = A/C
B = C/G/T	D = A/G/T	H = A/C/T	V = A/C/G	N = any base	

/ means "or" (e.g. A/G means A or G)

References

1. Lander ES, Waterman MS. *Genomics* 1988;**2**:231−9.
2. Pevzner P, Shamir R, editors. *Bioinformatics for biologists*. Cambridge University Press; 2011.
3. Miller JR, et al. *Genomics* 2010;**95**:315−27.
4. Nagarajan N, Pop M. *Nat Rev Genet* 2013;**14**:157−67.
5. Compeau PEC, et al. *Nat Biotechnol* 2011;**29**:987−91.
6. Magoč T, Salzberg SL. *Bioinformatics* 2011;**27**:2957−63.
7. Baker M. *Nat Methods* 2012;**9**:333−7.
8. Yandell M, Ence D. *Nat Rev Genet* 2012;**13**:329−42.
9. Kapustin Y, et al. *Biol Direct* 2008;**3**:20.
10. Wheelan SJ, et al. *Genome Res* 2001;**11**:1952−7.
11. Taft RJ, et al. *Bioessays* 2007;**29**:288−99.
12. Tocchini-Valentini GD, et al. *Proc Natl Acad Sci USA* 2011;**108**:4782−7.
13. Yang X, et al. *Genome Res* 2011;**21**:634−41.
14. Yoon B-J. *Curr Genomics* 2009;**10**:402−15.
15. Salzberg SL, et al. *Nucl Acids Res* 1998;**26**:544−8.
16. Stanke M, et al. *Nucl Acids Res* 2006;**34**:W435−9 (Web Server issue).
17. Burge C, Karlin S. *J Mol Biol* 1997;**268**:78−94.
18. Burset M, Guigó R. *Genomics* 1996;**34**:353−67.
19. Lander ES, et al. *Nature* 2004;**431**:931−45.
20. Alexandersson M, et al. *Genome Res* 2003;**13**:496−502.
21. Dewey C, et al. *Genome Res* 2004;**14**:661−4.
22. Wang L, et al. *Nucl Acids Res* 2004;**32**:3689−702.
23. Butler JEF, Kadonaga JT. *Genes Dev* 2002;**16**:2583−92.
24. Choudhuri S, et al. *DNA Seq* 2002;**13**:103−7.
25. Hewitt SC, et al. *Mol Endocrinol* 2012;**26**:887−98.
26. Choudhuri S. In: Choudhuri S, Carlson DB, editors. *Genomics: fundamentals and applications*. New York: Informa; 2009. p. 3−48.
27. Klassen JL, Currie CR. *PLoS ONE* 2013;**8**:e58387.
28. Münch R, et al. *Bioinformatics* 2005;**21**:4187−9.
29. Reese MG. *Comput Chem* 2001;**26**:51−6.
30. Knudsen S. *Bioinformatics* 1999;**15**:356−61.
31. Prestridge DS. *CABIOS* 1991;**7**:203−6.
32. Prestridge DS. *J Mol Biol* 1995;**249**:923−32.
33. Abeel T, et al. *Genome Res* 2008;**18**:310−23.
34. Down TA, Hubbard TJ. *Genome Res* 2002;**12**:458−61.
35. Blanchette M, Tompa M. *Nucl Acids Res* 2003;**31**:3840−2.
36. Pedersen AG, Nielsen H. *ISMB* 1997;**5**:226−33.
37. Nishikawa T, et al. *Bioinformatics* 2000;**16**:960−7.
38. Vincze T, et al. *Nucl Acids Res* 2003;**31**:3688−91.
39. Chen JL, et al. *Cell* 2000;**100**:503−14.
40. Svoboda P, Di Cara A. *Cell Mol Life Sci* 2006;**63**(7-8):901−8.
41. Li F, et al. *Cell Rep* 2012;**1**:69−82.
42. Zuker M, Stiegler P. *Nucl Acid Res* 1981;**9**:133−48.
43. McCaskill JS. *Biopolymers* 1990;**29**:1105−19.
44. Hofacker IL. *Nucl Acids Res* 2003;**31**:3429−31.
45. Andronescu M, et al. *Nucl Acids Res* 2003;**31**:3416−22.
46. Do CB, et al. *Bioinformatics* 2006;**22**:e90−8.
47. Reuter JS, Mathews DH. *BMC Bioinformatics* 2010;**11**:129.
48. Sato K, et al. *Bioinformatics* 2011;**27**:i85−93.
49. Hamada M, et al. *Bioinformatics* 2009;**25**:i330−8.
50. Frith MC, et al. *BMC Bioinformatics* 2010;**11**:80.
51. Sato K, et al. *Nucl Acids Res* 2009;**37**:W277−80 (Web Server issue).
52. Reeder J, et al. *Nucl Acids Res* 2007;**35**:W320−4 (Web server issue).
53. Reeder J, Giegerich R. *BMC Bioinformatics* 2004;**5**:104.
54. Rivas E. *RNA Biol* 2013;**10**:1−12.
55. Mathews DH, et al. *J Mol Biol* 1999;**288**:911−40.
56. Saeed AI, et al. *Biotechniques* 2003;**34**:374−8.
57. Eisen MB, et al. *Proc Natl Acad Sci USA* 1998;**95**:14863−8.
58. Choudhuri S. In: Choudhuri S, Carlson DB, editors. *Genomics: fundamentals and applications*. New York: Informa; 2009. p. 49−99.
59. Pepinski W, et al. *Mol Biol Rep* 2013;**40**:4333−8.
60. Kitts A, Sherry S. The single nucleotide polymorphism database (dbSNP) of nucleotide sequence variation (chapter 5). In: McEntyre J, Ostell J, editors. *The NCBI handbook*. Bethesda (MD): National Center for Biotechnology Information; 2011.