9

CREATION AND ANALYSIS OF PROTEIN MULTIPLE SEQUENCE ALIGNMENTS

Geoffrey J. Barton

European Molecular Biology Laboratory
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge
UK

INTRODUCTION

When a protein sequence is newly-determined, an important goal is to assign possible functions to the protein. The first computational step is to search for similarities with sequences that have previously been deposited in the DNA and protein sequence databases. If similar sequences are found, they may match the complete length of the new sequence or only to subregions of the sequence. If more than one similar sequence is found, then the next important step in the analysis is to multiply align all of the sequences. Multiple alignments are a key starting point for the prediction of protein secondary structure, residue accessibility, function, and the identification of residues important for specificity. Multiple alignments also provide the basis for the most sensitive sequence searching algorithms (cf. Gribskov et al., 1987; Barton and Sternberg, 1990; Attwood et al., 2000). Effective analysis of a well-constructed multiple alignment can provide important clues about which residues in the protein are important for function and which are important for stabilizing the secondary and tertiary structures of the protein. In addition, it is often also possible to make predictions about which residues confer specificity of function to subsets of the sequences. In this chapter, some guidelines are provided toward the generation and analysis of protein multiple sequence alignments. This is not a comprehensive review of techniques; rather, it is a guide based on the software that have proven to be most useful in building alignments and using them to predict protein structure and function. A full summary of the software is available at the end of the chapter.

WHAT IS A MULTIPLE ALIGNMENT, AND WHY DO IT?

A protein sequence is represented by a string a of letters coding for the 20 different types of amino acid residues. A protein sequence alignment is created when the residues in one sequence are lined up with those in at least one other sequence. Optimal alignment of the two sequences will usually require the insertion of gaps in one or both sequences in order to find the best alignment. Alignment of two residues implies that those residues are performing similar roles in the two different proteins. This allows for information known about specific residues in one sequence to be potentially transferred to the residues aligned in the other. For example, if the active site residues of an enzyme have been characterized, alignment of these residues with similar residues in another sequence may suggest that the second sequence possesses similar catalytic activity to the first. The validity of such hypotheses depends on the overall similarity of the sequences, which in turn dictate the confidence with which an alignment can be generated. There are typically many millions of different possible alignments for any two sequences. The task is to find an alignment that is most likely to represent the chemical and biological similarities between the two proteins.

A *multiple sequence alignment* is simply an alignment that contains more than two sequences! Even if one is interested in the similarities between only two of the sequences in a set, it is always worth multiply-aligning all available sequences. The inclusion of these additional sequences in the multiple alignment will normally improve the accuracy of the alignment between the sequence pairs, as illustrated in Figure 9.1, as well as revealing patterns of conserved residues that would not have been obvious when only two sequences are directly studied. Although many programs exist that can generate a multiple alignment from unaligned sequences, extreme care must be taken when interpreting the results. An alignment may show perfect matching of a known active-site residue with an identical residue in a well-characterized protein family, but, if the alignment is incorrect, any inference about function will also be incorrect.

STRUCTURAL ALIGNMENT OR EVOLUTIONARY ALIGNMENT?

It is the precise arrangement of the amino acid side chains in the three-dimensional structure of the protein that dictates its function. Comparison of two or more protein three-dimensional structures will highlight which residues are in similar positions in space and hence likely to be performing similar functional roles. Such comparisons can be used to generate a sequence alignment from structure (e.g., see Russell and Barton, 1992). The *structural alignment* of two or more proteins is the gold standard against which sequence alignment algorithms are normally judged. This is because it is the structural alignment that most reliably aligns residues that are of functional importance. Unfortunately, structural alignments are only possible when the three-

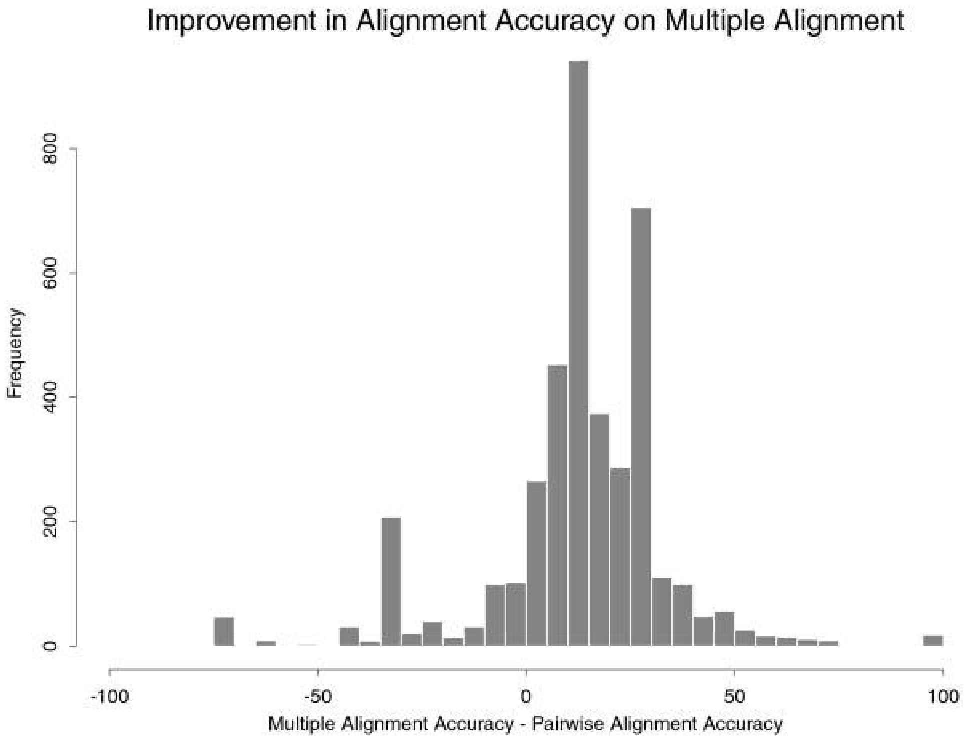


Figure 9.1. Histogram showing difference in accuracy between the same pairs of sequences aligned as a pair and as part of a larger multiple sequence alignment. On average, multiple alignments improve the overall alignment accuracy, which, in this example, is judged as the alignment obtained by comparison of the three-dimensional structures of the individual proteins rather than just their sequences (Russell and Barton, 1992).

dimensional structures of *all* the proteins to be aligned are known. This is not usually the case; therefore, the challenge for sequence alignment methods is to get as close as possible to the structural alignment without knowledge of structure. Although the structural alignment is the most important alignment for the prediction of function, it does not necessarily correspond to the *evolutionary alignment* implied by divergence from a common ancestor protein. Unfortunately, it is rarely possible to determine the evolutionary alignment of two divergent proteins with confidence because this would require knowledge of the precise history of substitutions, insertions, and deletions that have led to the creation of present-day proteins from their common ancestor.

HOW TO MULTIPLY ALIGN SEQUENCES

Automatic alignment programs such as CLUSTAL W (Thompson et al., 1994) will give good quality alignments for sequences that are more than 6σ similar (Barton and Sternberg, 1987). However, building good multiple alignments for sequences that are not trivially similar is a precise task even with the best available alignment tools. This section gives an overview of some of the steps to go through to make alignments that are good for structure/function predictions. This is *not* a universal recipe; in fact, there are very few universal recipes in bioinformatics in general. Each set of sequences presents its own biologically based problems, and only experience can guide the creation of high-quality alignments. Some collections of expertly created multiple alignments exist (described later), and these should always be consulted when studying sequences that are present there. The key steps in building a multiple alignment are as follows.

- Find the sequences to align by database searching or by other means.
- Locate the region(s) of each sequence to include in the alignment. *Do not* try to multiply align sequences that are substantially different in length. Most multiple alignment programs are designed to align sequences that are similar over their entire length; therefore, a necessary first step is to edit the sequences down to those regions that sequence database searches suggest are similar.
- Ideally, assess the similarities within the set of sequences by comparing them pairwise with randomizations. Select a subset of the sequences to align first that cluster above 6σ. Automatic alignment of such sequences are likely to be accurate (Barton and Sternberg, 1987). An alternative to doing randomization is to align only sequences that are similar to the query in a database search, say with an E-value of <1.
- Run the multiple alignment program.
- Manually inspect the alignment for problems. Pay particular attention to regions that appear to be speckled with gaps. Use an alignment visualization tool (e.g., ALSCRIPT/JalView, see below) to identify positions in the alignment that show conserved physicochemical properties across the complete alignment. If there are no such regions, then look at subsets of the sequences.
- Remove sequences that appear to disrupt the alignment seriously and then realign the remaining subset.
- After identifying key residues in the set of sequences that are straightforward to align, attempt to add the remaining sequences to the alignment so as to preserve the key features of the family.

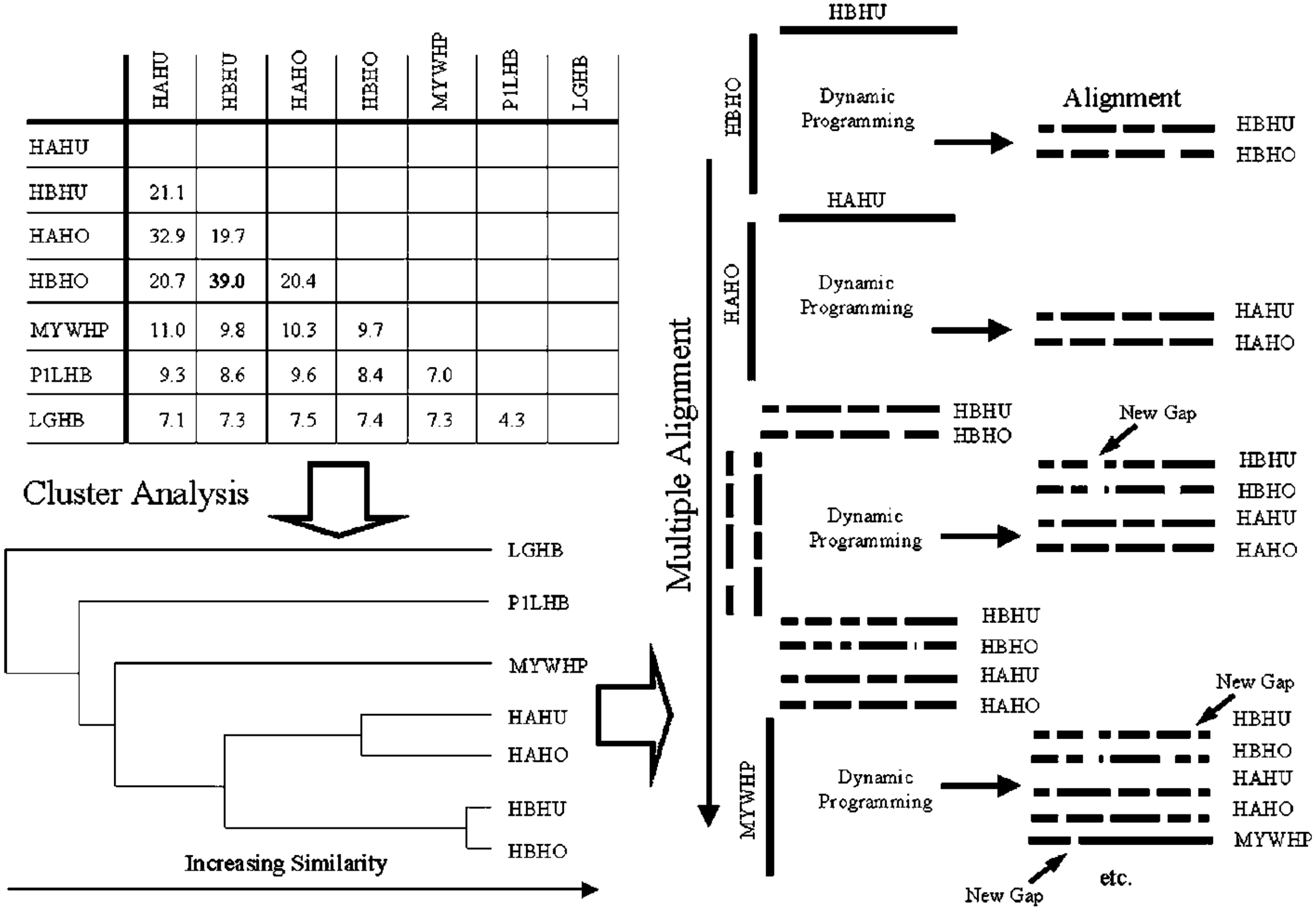
Assessing Quality of Alignment

Multiple alignment programs will align *any* set of sequences. However, the fact that the program produces an alignment does not mean that the alignment has any biological meaning. Most programs will take unrelated protein sequences and align them just as easily as two genuinely related sequences. Even for related sequences, there is no guarantee that the resulting alignment is in any way meaningful. One way of assessing whether an alignment is meaningful is to perform a randomization or ‘‘Monte Carlo’’ test of significance. To do this, the two sequences are first aligned and the score (S) for the alignment is recorded. The sequences are then shuffled so that they maintain their length and amino acid composition but have a randomized order. The shuffled sequences are then compared again, and the score is recorded. The shuffling and realigning process is repeated a number of times (typically 100), and the mean and standard deviation (σ) for the scores are calculated. The Z-score provides an indication of the significance of the alignment. If $Z > 6$, then it is highly likely that the two sequences are alignable, and the alignment correctly relates the key functional and structural residues in the individual proteins to one another (Barton and Sternberg, 1987). Unfortunately, this can only be a rough guide. An alignment that gives a $Z < 6$ may be poor, and some alignments with low Z-scores are actually correct. This is simply a reflection of the fact that, during evolution, sequence similarity has diverged faster than structural or functional similarity. Z-scores are preferable to simple percent identities as a measure of similarity because it corrects for both compositional bias in the sequences as well as accounting for the varying lengths of sequences. The Z-score, therefore, gives an indication of the *overall* similarity between two sequences. Although it is a powerful measure, it does *not* help to locate parts of the sequence alignment that are incorrect. As a general rule, if the alignment is between two or more sequences that do indeed share a similar three-dimensional structure, then the majority of errors will be concentrated around regions where there are gaps (insertions/deletions).

Hierarchical Methods

The most accurate, practical methods for automatic multiple alignment are hierarchical methods. These work by first finding a guide tree and then following the guide tree to build the alignment. The process is summarized in Figure 9.2. First, all pairs of sequences in the set to be aligned are compared by a pairwise method of sequence comparison. This provides a set of pairwise similarity scores for the sequences that can be fed into a cluster analysis or tree calculating program. The tree is calculated to place more similar pairs of sequences closer together on the tree than sequences that are less similar. The multiple alignment is then built by starting with the pair of sequences that is most similar and aligning them and then aligning the next most similar pair, and so on. Pairs to be aligned need not be single sequences but can be alignments that have been generated earlier in the tree. If an alignment is compared with a sequence or another alignment, then gaps that exist in the alignment are preserved. There are many different variations of this basic multiple alignment technique. Because errors in alignment that occur early in the process can get locked in and propagated, some methods allow for realignment of the sequences after the initial alignment (e.g., Barton and Sternberg, 1987; Gotoh, 1996). Other refinements include using different similarity scoring matrices at different stages in building up the alignment (e.g., Thompson et al., 1994). Gaps (insertions/deletions) do not occur randomly in protein sequences.

Since a stable, properly-folded protein must be maintained, proteins with an insertion or deletion in the middle of a secondary structure (α-helix or β-strand) are usually selected against during the course of evolution. As a consequence, present-day proteins show a strong bias toward localizing insertions and deletions to loop regions that link the core secondary structures. This observation can be used to improve the accuracy of multiple sequence alignments when the secondary structure is known for one or more of the proteins in practice by making the penalty for inserting a gap higher when in secondary structure regions than when in loops (Barton and Sternberg, 1987; Jones, 1999. A further refinement is to bias where gaps are most likely to be inserted in the alignment by examining the growing alignment for regions that are most likely to accommodate gaps (Pascarella and Argos, 1992).



	HAHU	HBHU	HAHO	HBHO	MYWHP	P1LHB	LGHB
HAHU							
HBHU	21.1						
HAHO	32.9	19.7					
HBHO	20.7	**39.0**	20.4				
MYWHP	11.0	9.8	10.3	9.7			
P1LHB	9.3	8.6	9.6	8.4	7.0		
LGHB	7.1	7.3	7.5	7.4	7.3	4.3	

Figure 9.2. Illustration of the stages in hierarchical multiple alignment of seven sequences. The codes for these sequences are HAHU, HBHU, HAHO, HBHO, MYWHP, P1LHB, and LGHB. The table at the top left shows the pairwise *Z*-scores for comparison of each sequence pair. Higher numbers mean greater similarity (see text). Hierarchical cluster analysis of the *Z*-score table generates the dendrogram or tree shown at the bottom left. Items joined toward the right of the tree are more similar than those linked toward the left. Based on the tree, LGHB is least similar to the other sequences in the set, whereas HBHU and HBHO are the most similar pair (most similar to each other). The first four steps in building the multiple alignment are shown on the right. The first two steps are pairwise alignments. The third step is a comparison of profiles from the two alignments generated in *steps 1* and *2*. The fourth step adds a single sequence (MYWHP) to the alignment generated at *step 3*. Further sequences are added in a similar manner.

CLUSTAL W and Other Hierarchical Alignment Software

CLUSTAL W combines a good hierarchical method for multiple sequence alignment with an easy-to-use interface. The software is free, although a contribution to development costs is required when purchasing the program. CLUSTAL W runs on most computer platforms and incorporates many of the techniques described in the previous section. The program uses a series of different pair-score matrices, biases the location of gaps, and allows you to realign a set of aligned sequences to refine the alignment. CLUSTAL W can read a secondary structure ‘‘mask’’ and bias the positioning of gaps according to it; the program can also read two preexisting alignments and align them to each other or align a set of sequences to an existing alignment. CLUSTAL W also includes options to calculate neighbor-joining trees for use in inferring phylogeny. Although CLUSTAL W does not provide general tools for viewing these trees, the output is compatible with the PHYLIP package (Felsenstein, 1989) and the resultant trees can be viewed with that program. CLUSTAL W can read a variety of different common sequence formats and produce a range of different output formats. The manual for CLUSTAL W is clearly written and explains possible limitations of the alignment process. Although CLUSTAL W can be installed and run locally, users can also access it through a faster Web service via the EBI server by clicking the ‘‘Tools page’’. With the exception of manual editing and visualization, CLUSTAL W contains most of the tools that are needed to build and refine a multiple sequence alignment. When combined with JalView, as described below, the process of building and refining a multiple alignment is greatly simplified. Although CLUSTAL W is probably the most widely used multiple alignment program and for most purposes is adequate, other software exists having functionality not found in CLUSTAL W. For example, AMPS (Barton, 1990) provides a pairwise sequence comparison option with randomization, allowing *Z*-scores to be calculated. The program can also generate alignments without the need to calculate trees first. For large numbers of sequences, this can save a lot of time because it eliminates the need to perform all pairwise comparisons of the sequences. AMPS also has software to visualize trees, thus helping in the selection of sequences for alignment. However, the program has no simple menu interface; therefore, it is more difficult for the novice or occasional user to use.

More Rigorous Nonhierarchical Methods

Hierarchical methods do not guarantee finding the one mathematically optimal multiple alignment for an entire set of sequences. However, in practice, the mathematical optimum rarely makes any more biological sense than the alignment that is found by hierarchical methods. This is probably because a great deal of effort has gone into tuning the parameters used by CLUSTAL W and other hierarchical methods to produce alignments that are consistent with those that a human expert or three-dimensional structure comparison might produce. The widespread use of these techniques has also ensured that the parameters are appropriate for a wide range of alignment problems. More rigorous alignment methods that attempt to find the mathematically optimal alignment over a set of sequences (cf. Lipman et al., 1989) may be capable of giving better alignments, but, as shown in recent benchmark studies, they are, on average, no better than the hierarchical methods.

Multiple Alignment by PSI-BLAST

Multiple sequence alignments have long been used for more sensitive searches of protein sequence databases than is possible with a single sequence. The program PSI-BLAST (Altschul et al., 1997) has recently made these profile methods more easily available. As part of its search, PSI-BLAST generates a multiple alignment. However, this alignment is not like the alignments made by CLUSTAL W, AMPS, or other traditional multiple alignment tools. In a conventional multiple alignment, all sequences in the set have equal weight. As a consequence, a multiple alignment will normally be longer than any one of the individual sequences, since gaps will be inserted to optimize the alignment. In contrast, a PSI-BLAST multiple alignment is *always* exactly the length of the query sequence used in the search. If alignment of the query (or query profile) to a database sequence requires an insertion in the query, then the inserted region from the database sequence is simply discarded. The resulting alignment thus highlights the amino acids that may be aligned to each position in the query. Perhaps for this reason, PSI-BLAST multiple alignments and their associated frequency tables and profiles have proved very effective as input for programs that predict protein secondary structure (Jones, 1999; Cuff and Barton, 2000).

Multiple Protein Alignment From DNA Sequences

Although most DNA sequences will have translations represented in the EMBL-TrEMBL or NCBI-GenPept databases, this is not true of single-pass EST sequences. Because EST data are accumulating at an exponential pace, an automatic method of extracting useful protein information from ESTs has been developed. In brief, the ProtEST server (Cuff et al., 1999) searches EST collections and protein sequence databases with a protein query sequence. EST hits are assembled into species-specific contigs, and an error-tolerant alignment method is used to correct probable sequencing errors. Finally, any protein sequences found in the search are multiply aligned with the translations of the EST assemblies to produce a multiple protein sequence alignment. The JPred server (version 7.3) will generate a multiple protein sequence alignment when presented with a single protein sequence by searching the SWALL protein sequence database and building a multiple alignment. The JPred alignments are a good starting point for further analysis with more sensitive methods.

TOOLS TO ASSIST THE ANALYSIS OF MULTIPLE ALIGNMENTS

A multiple sequence alignment can potentially consist of several hundred sequences that are 500 or more amino acids long. With such a volume of data, it can be difficult to find key features and present the alignments in a form that can be analyzed by eye. In the past, the only option was to print out the alignment on many sheets of paper, stick these together, and then pore over the massive poster with colored highlighter pens. This sort of approach can still be useful, but it is rather inconvenient! Visualization of the alignment is an important scientific tool, either for analysis or for publication. Appropriate use of color can highlight positions that are either identical in all the aligned sequences or share common physicochemical properties. ALSCRIPT (Barton, 1993) is a program to assist in this process. ALSCRIPT takes a multiple sequence alignment and a file of commands and produces a file in

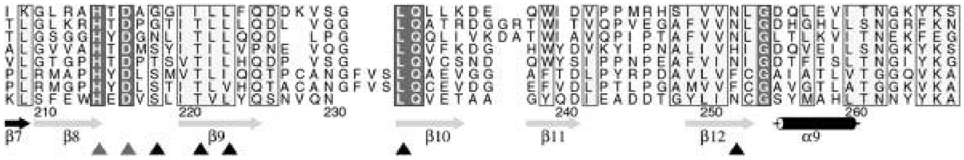


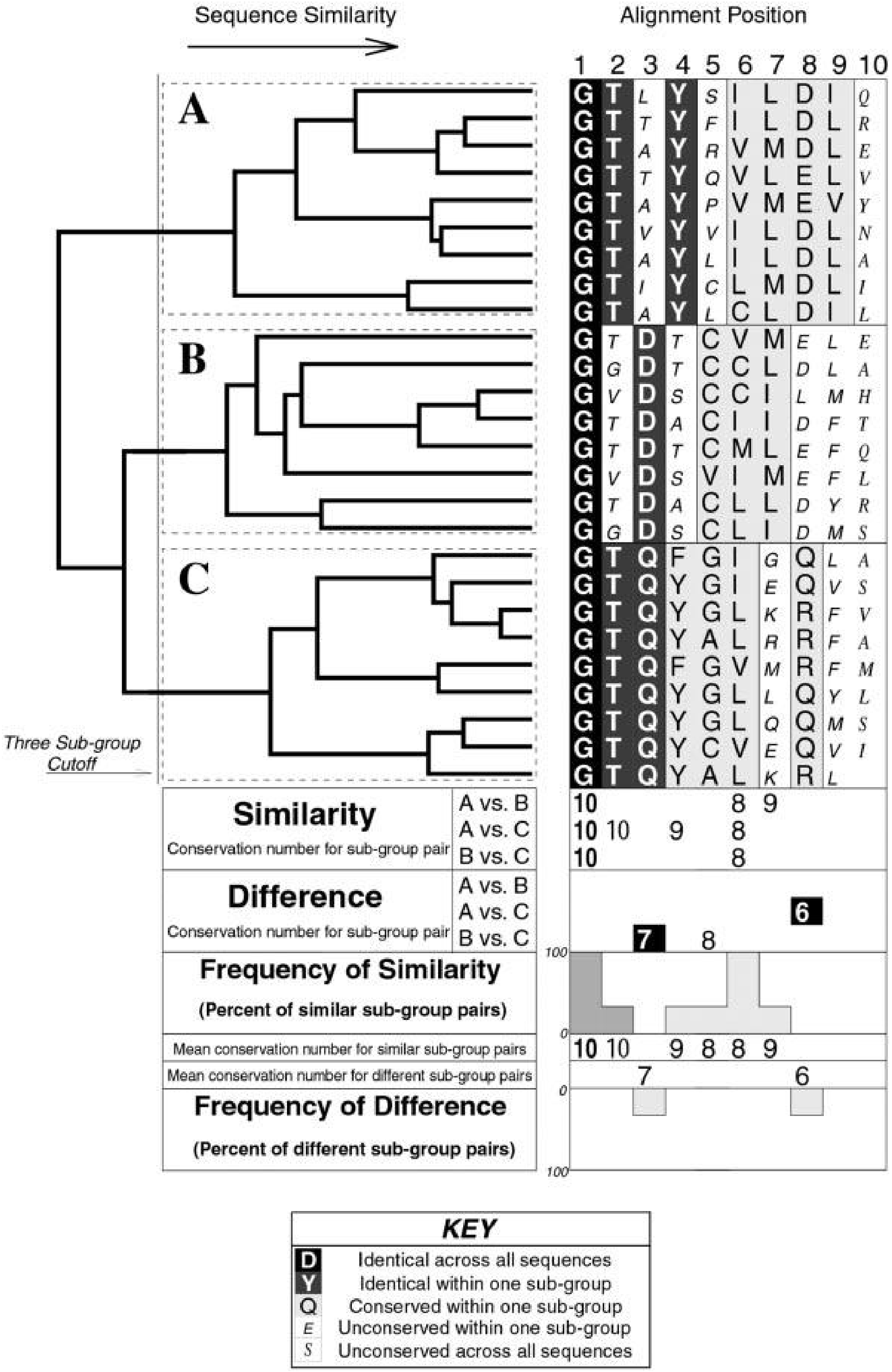
Figure 9.3. Example output from the program ALSCRIPT (Barton, 1993). Details can be found within the main text.

PostScript format suitable for printing out or viewing with a utility such as ghostview. Figure 9.3 illustrates a fragment of ALSCRIPT output (the full figure can be seen in color in Roach et al., 1995). In this example, identities across all sequences are shown in white on red and boxed, whereas positions with similar physicochemical properties are shown black on yellow and boxed. Residue numbering according to the bottom sequence is shown underneath the alignment. Green arrows illustrate the location of known β-strands, whereas α-helices are shown as black cylinders. Further symbols highlight specific positions in the alignment for easy cross-referencing to the text. ALSCRIPT is extremely flexible and has commands that permit control of font size and type, background coloring, and boxing down to the individual residue. The program will automatically split a large alignment over multiple pages, thus permitting alignments of any size to be visualized. However, this flexibility comes at a price. There is no point-and-click interface, and the program requires the user to be familiar with editing files and running programs from the command line. The ALSCRIPT distribution includes a comprehensive manual and example files that make the process of making a useful figure for your own data a little easier.

Subalignments—AMAS

ALSCRIPT provides a few commands for calculating residue conservation across a family of sequences and coloring the alignment accordingly. However, it is really intended as a display tool for multiple alignments rather than an analysis tool. In contrast, AMAS (Analysis of Multiply Aligned Sequences; Livingstone and Barton, 1993) is a program for studying the relationships between sequences in a multiple alignment to identify possible functional residues. AMAS automatically runs ALSCRIPT to provide one output that is a boxed, colored, and annotated multiple alignment.

Why might you want to run AMAS? A common question one faces is, ‘‘Which residues in a protein are important for its specificity?’’ AMAS can help identify these residues by highlighting similarities and differences between subgroups of sequences in a multiple alignment. For example, given a family of sequences that shows some variation, positions in a multiple alignment that are conserved across the entire family of sequences are likely to be important to stabilize the common fold of the protein or common functions. Positions that are conserved within a subset of the sequences, but different in the rest of the family, are likely to be those important to the specific function or specificity of that subset, and these positions can be easily identified using AMAS. There are a number of subtle types of differences that AMAS will search for, and these are summarized in Figure 9.4. To use AMAS, one must first have an idea of what subgroups of sequences exist in a multiple alignment of interest. One way to do this is to take a tree generated from the multiple alignment and



	1	2	3	4	5	6	7	8	9	10
A	G	T	L	Y	S	I	L	D	I	Q
	G	T	T	Y	F	I	L	D	L	R
	G	T	A	Y	R	V	M	D	L	E
	G	T	T	Y	Q	V	L	E	L	V
	G	T	A	Y	P	V	M	E	V	Y
	G	T	V	Y	V	I	L	D	L	N
	G	T	A	Y	L	I	L	D	L	A
	G	T	I	Y	C	L	M	D	L	I
	G	T	A	Y	L	C	L	D	I	L
B	G	T	D	T	C	V	M	E	L	E
	G	G	D	T	C	C	L	D	L	A
	G	V	D	S	C	C	I	L	M	H
	G	T	D	A	C	I	I	D	F	T
	G	T	D	T	C	M	L	E	F	Q
	G	V	D	S	V	I	M	E	F	L
	G	T	D	A	C	L	L	D	Y	R
	G	G	D	S	C	L	I	D	M	S
C	G	T	Q	F	G	I	G	Q	L	A
	G	T	Q	Y	G	I	E	Q	V	S
	G	T	Q	Y	G	L	K	R	F	V
	G	T	Q	Y	A	L	R	R	F	A
	G	T	Q	F	G	V	M	R	F	M
	G	T	Q	Y	G	L	L	Q	Y	L
	G	T	Q	Y	G	L	Q	Q	M	S
	G	T	Q	Y	C	V	E	Q	V	I
	G	T	Q	Y	A	L	K	R	L	
Similarity Conservation number for sub-group pair: A vs. B	10					8	9			
A vs. C	10	10		9		8				
B vs. C	10					8				
Difference Conservation number for sub-group pair: A vs. B										
A vs. C								6		
B vs. C			7		8					
Frequency of Similarity (Percent of similar sub-group pairs)										
Mean conservation number for similar sub-group pairs	10	10		9	8	8	9			
Mean conservation number for different sub-group pairs			7					6		
Frequency of Difference (Percent of different sub-group pairs)										

KEY

D	Identical across all sequences
Y	Identical within one sub-group
Q	Conserved within one sub-group
E	Unconserved within one sub-group
S	Unconserved across all sequences

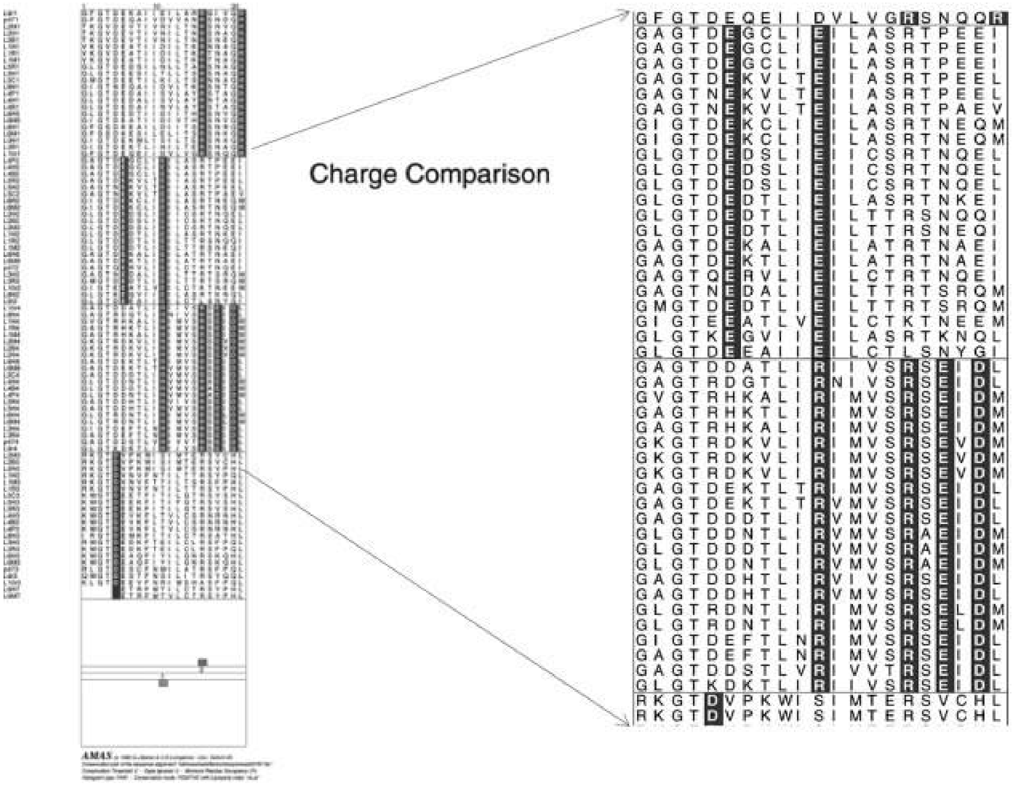
identify clusters of sequences at some similarity threshold. This is also illustrated in Figure 9.4, in which three groups have been selected on the basis of the tree shown at the top left. Alternatively, if one knows in advance that finding common features and differences between, for example, sequences 1–20 and 21–50 in a multiple alignment is important, one can specify these ranges explicitly. The output of AMAS is a detailed text summary of the analysis as well as a colored and shaded multiple sequence alignment. By default, AMAS searches for general features of amino acid physicochemical properties. However, this can be narrowed down just to a single feature of amino acids such as charge. An example of a charge analysis is shown in Figure 9.5 for repeats within the annexin supergene family of proteins (Barton et al., 1991). The analysis highlights a charge swap within two subgroups of the sequences, correctly predicting the presence of a salt bridge in the native folded protein (Huber et al., 1990). The AMAS program may either be downloaded and run locally, or a subset of its options can be accessed over the Web at a server hosted by EBI.

Secondary Structure Prediction and the Prediction of Buried Residues From Multiple Sequence Alignment

When aligning sequences, it is important to remember that the protein is a three-dimensional molecule and not just a string of letters. Predicting secondary structure either for the whole collection of sequences or subsets of the sequences can be used to help discover how the protein might fold locally and guide the alignment of more distantly related sequences. For example, it is common for proteins with similar topologies to have quite different sequences and be unalignable by an automatic alignment method (e.g., see Russell and Barton, 1994; *cf.* the SCOP database, see Murzin et al., 1995, Chapter 5). In these circumstances, the secondary structure may suggest which blocks of sequences should be equivalent. The prediction of secondary structure (α-helix and β-strand) is enhanced by around 6% when performed from a multiple alignment, compared with prediction from a single sequence (Cuff and

<

Figure 9.4. Stylized output from the program AMAS. The sequence alignment has been shaded to illustrate similarities within each subgroup of sequences. *Conservation numbers* (Livingstone and Barton, 1993; Zvelebil et al., 1987) run from 0 to 10 and provide a numerical measure of the similarity in physicochemical properties of each column in the alignment. Below the alignment, the lines "Similar Pairs" show the conservation values obtained when each pair of subgroups is combined and the combined conservation number is not less than a threshold. For example, at position 7, subgroups *A* and *B* combine with a conservation number of 9. The lines "Different Pairs" illustrate positions at which a combination of subgroups lowers the conservation number below the threshold. For example, at position 3, there is an identity in subgroup *B* and one in *C*, but, when the groups are combined, the identity is lost and the conservation drops below the threshold of 8 to 7. A summary of the similarities and differences is given as a frequency histogram. Each upward bar represents the proportion of subgroup pairs that preserve conservation, whereas each downward bar shows the percentage of differences. For example, at position 6, 3/3 pairs are conserved (100%), whereas at positions 3 and 8, 1/3 pairs show (33%) differences With a large alignment, the histogram can quickly draw the eye to regions that are highly conserved or to regions where there are differences in conserved physicochemical properties.



```
G F G T D E Q E I I D V L V G H S N Q Q H
G A G T D E G C L I E I L A S R T P E E I
G A G T D E G C L I E I L A S R T P E E I
G A G T D E G C L I E I L A S R T P E E I
G A G T D E K V L T E I I A S R T P E E L
G A G T N E K V L T E I I A S R T P E E L
G A G T N E K V L T E I L A S R T P A E V
G I G T D E K C L I E I L A S R T N E Q M
G I G T D E K C L I E I L A S R T N E Q M
G L G T D E D S L I E I I C S R T N Q E L
G L G T D E D S L I E I I C S R T N Q E L
G L G T D E D S L I E I I C S R T N Q E L
G L G T D E D T L I E I L A S R T N K E I
G L G T D E D T L I E I L T T R S N Q Q I
G L G T D E D T L I E I L T T R S N E Q I
G A G T D E K A L I E I L A T R T N A E I
G A G T D E K T L I E I L A T R T N A E I
G A G T Q E R V L I E I L C T R T N Q E I
G A G T N E D A L I E I L T T R T S R Q M
G M G T D E D T L I E I L T T R T S R Q M
G I G T E E A T L V E I L C T K T N E E M
G L G T K E G V I I E I L A S R T K N Q L
G L G T D E E A I I E I L C T L S N Y G I
G A G T D D A T L I R I I V S R S E I D L
G A G T R D G T L I R N I V S R S E I D L
G V G T R H K A L I R I M V S R S E I D M
G A G T R H K T L I R I M V S R S E I D M
G A G T R H K A L I R I M V S R S E I D M
G K G T R D K V L I R I M V S R S E V D M
G K G T R D K V L I R I M V S R S E V D M
G K G T R D K V L I R I M V S R S E V D M
G A G T D E K T L T R I M V S R S E I D L
G A G T D E K T L T R V M V S R S E I D L
G A G T D D D T L I R V M V S R S E I D L
G L G T D D N T L I R V M V S R A E I D M
G L G T D D D T L I R V M V S R A E I D M
G L G T D D N T L I R V M V S R A E I D M
G A G T D D H T L I R V I V S R S E I D L
G A G T D D H T L I R V M V S R S E I D L
G L G T R D N T L I R I M V S R S E L D M
G L G T R D N T L I R I M V S R S E L D M
G I G T D E F T L N R I M V S R S E I D L
G A G T D E F T L N R I M V S R S E I D L
G A G T D D S T L V R I V V T R S E I D L
G L G T K D K T L I R I I V S R S E I D L
R K G T D V P K W I S I M T E R S V C H L
R K G T D V P K W I S I M T E R S V C H L
```

Figure 9.5. Illustration of an AMAS output used to find a charge pair in the annexins. There are four groups of sequences in the alignment. The highlighted positions highlight locations where the charge is conserved in each group of sequences yet different between groups. A change from glutamine to arginine is shown at position 1.

Barton 1999). The best current methods [PSIPRED (Jones, 1999) and JNET (Cuff and Barton, 2000)] give over 76% accuracy for the prediction of three states (α-helix, β-strand, and random coil) in rigorous testing. This high accuracy is possible because the prediction algorithms are able to locate regions in the sequences that show patterns of conserved physicochemical properties across the aligned family. These patterns are characteristic of particular secondary structure types and can often be seen by eye in a multiple sequence alignment, as summarized below:

- Short runs of conserved hydrophobic residues suggest a buried β-strand.
- i, $i + 2$, and $i + 4$ patterns of conserved hydrophobic amino acids suggest a surface β-strand, since the alternate residues in a strand point in the same direction. If the alternate residues all conserve similar physicochemical properties, then they are likely to form one face of a β-strand.
- i, $i + 3$, $i + 4$, and $i + 7$, and variations of that pattern, (e.g., i, $i + 4$, $i + 7$) of conserved residues suggest an α-helix with one surface facing the solvent.
- Insertions and deletions are normally only tolerated in regions not associated with the buried core of the protein. Thus, in a good multiple alignment, the location of indels suggests surface loops rather than α-helices or β-strands.

- Although glycine and proline may be found in all secondary structure types, a glycine or proline residue that is conserved across a family of sequences is a strong indicator of a loop.

Secondary structure prediction programs such as JNET (Cuff and Barton, 2000) and PHD (Rost and Sander, 1993) also exploit multiply aligned sequences to predict the likely exposure of each residue to solvent. Knowledge of solvent accessibility can help in the identification of residues key to stabilizing the fold of the protein as well as those that may be involved in binding. Both the JNET and PHD programs may be run from the JPred prediction server, whereas JNET may also be run from within JalView. [For further discussion of methods used to predict secondary structure, the reader is referred to Chapter 11.]

JalView

AMAS and ALSCRIPT are not interactive: they run a script or set of commands and produce a PostScript file, which can be viewed on-screen using a Postscript viewer or just printed out. Although this provides the maximum number of options and flexibility in its display, it is comparatively slow and sometimes difficult to learn. In addition, the programs require a separate program to be run to generate the multiple alignment for analysis. If the alignment requires modification or subsets of the alignment are needed, a difficult cycle of editing and realigning is often required. The program JalView overcomes most of these problems. JalView encapsulates many of the most useful features of AMAS and ALSCRIPT in an interactive, mouse-driven program that will run on most computers with a Java interpreter. The core of JalView is an interactive alignment editor. This allows an existing alignment to be read into the program and individual residues or blocks of residues to be moved around. A few mouse clicks permit the sequences to be subset into a separate copy of JalView. JalView can call CLUSTAL W (Thompson et al., 1994) either as a local copy on the same computer that is running JalView or the CLUSTAL W server at EBI. Thus, one can also read in a set of unaligned sequences, align them with CLUSTAL W, edit the alignment, and take subsets with great ease. Further functions of JalView will calculate a simple, neighbor-joining tree from a multiple alignment and allow an AMAS-style analysis to be performed on the subgroups of sequences. If the tertiary structure of one of the proteins in the set is available, then the three-dimensional structure may be viewed alongside the alignment in JalView. In addition, the JNET secondary structure prediction algorithm (Cuff and Barton, 2000) may be run on any subset of sequences in the alignment and the resulting prediction displayed along with the alignment. The JalView application is available for free download and, because it is written in Java, can also be run as an applet in a Web browser such as Netscape or Internet Explorer. Many alignment services such as the CLUSTAL W server at EBI and the Pfam server include JalView as an option to view the resulting multiple alignments. Figure 9.6 illustrates a typical JalView session with the alignment editing and tree windows open.

COLLECTIONS OF MULTIPLE ALIGNMENTS

This chapter has focused on methods and servers for building multiple protein sequence alignments. Although proteins that are clearly similar by the *Z*-score measure

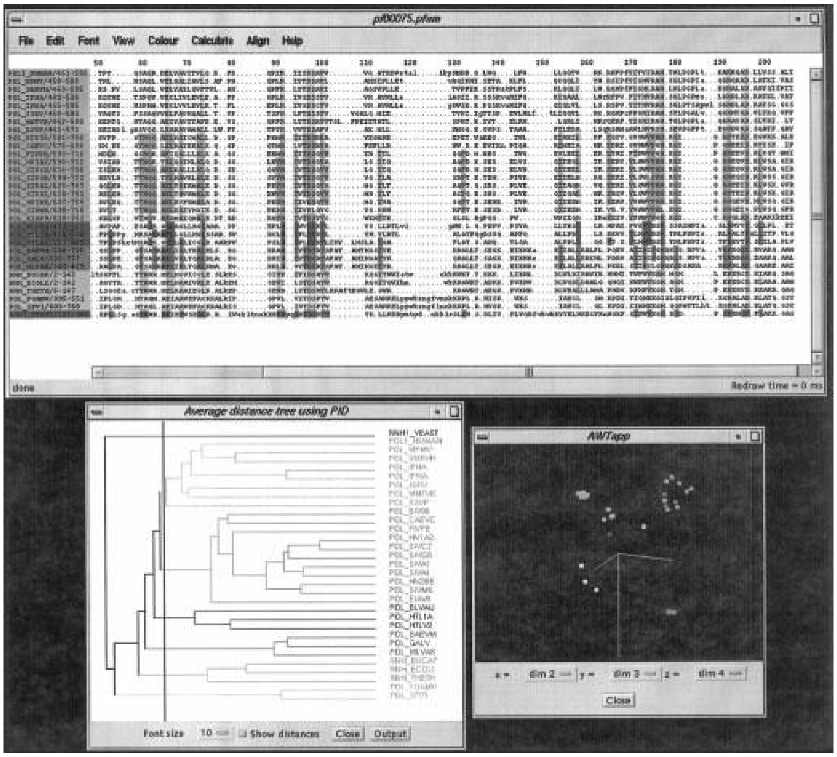

pf00075.pfam
File Edit Font View Colour Calculate Align Help
done
Redraw time = 0 ms
Average distance tree using PID
RNH1_YEAST
POL_BLVAU
POL_HTL1A
POL_HTLV2
POL_BAEVM
POL_GALV
POL_MLVAK
Font size 10 Show distances Close Output
AWTapp
x = dim 2 y = dim 3 z = dim 4
Close

Figure 9.6. An example JalView alignment editing and analysis session. The top panel contains a multiple alignment, and the bottom left is the similarity tree resulting from that alignment. A vertical line on the tree has separated the sequences into subgroups, which have been colored to highlight conservation within each subgroup. The panel at the bottom right illustrates an alternative clustering method.

should be straightforward to align by the automatic methods discussed here, getting good alignments for proteins with more remote similarities can be a very time-consuming process. A number of groups have built collections of alignments using a combination of automation and expert curation [e.g., SMART (Schultz et al., 1998), Pfam (Bateman et al., 1999), and PRINTS (Attwood et al., 2000)], and these, together with the tools available at their Web sites, can provide an excellent starting point for further analyses.

INTERNET RESOURCES FOR TOPICS PRESENTED IN CHAPTER 9

CLUSTAL W	*ftp://ftp.ebi.ac.uk/pub/software*
AMAS	*http://barton.ebi.ac.uk/servers/amas.html*
JPred	*http://barton.ebi.ac.uk/servers/jpred.html*
ProtEST	*http://barton.ebi.ac.uk/servers/protest.html*
JalView	*http://barton.ebi.ac.uk/new/software.html*
AMPS	*http://barton.ebi.ac.uk/new/software.html*
European Bioinformatics Institute	*http://www.ebi.ac.uk*

PROBLEM SET

The following problems are based on the annexin supergene family, the same family used throughout the discussion in this chapter. This family contains a 100 amino acid residue unit that repeats either four, eight, or 16 times within each protein. The analysis required below will focus on the individual repeat units, rather than the organization of the repeat units within the full-length protein sequences.

The problems will require the use of CLUSTAL W and Jalview, which you may have to install (or have installed) on a UNIX- or Linux-based system to which you have access. The files referred to below are available on the book's Web site.

The file `ann_rep1.fa` contains the sequence of a single annexin domain. This sequence has been used as the query against the SWALL protein sequence database, using the program `scanps` to make the pairwise sequence comparisons. A partial listing of the results can be found in the file named `ann_rep1_frags.fa`.

Generation of a Multiple Sequence Alignment

1. Copy the file `ann_rep1_frags.fa` to a new directory.
2. Run CLUSTAL W on `ann_rep1_frags.fa`. Accept all defaults, and create an output file called `ann_rep1_frags.aln`.
3. Pass this output file to Jalview by typing `Jalview ann_rep1_frags.aln CLUSTAL`.
4. Select the fragment sequences by clicking on the ID code. Select Delete Selected Sequences from the Edit menu.
5. Save the modified alignment to a CLUSTAL-formatted file called `ann_rep1_frags_del1.aln`.
6. Select Average Distance Tree from the Calculate menu. A new window will now appear, and after a few moments, a tree (dendrogram) will be rendered within that window. There should be outliers at the very top of that tree, and these outliers will need to be eliminated.
7. Click on the tree to the left of where the outliers join the tree. A vertical line should now appear, and the outliers will be highlighted in a different color.
8. Return to the Alignment window and delete the outliers from the alignment, in the same way as was done in Step 4. Save the resulting alignment to a file named `ann_rep1_frags_del2.aln`.

This series of steps produces a "clean alignment" for inspection. Positions within the alignment can be colored in different ways to highlight certain features of the amino acids within the alignment. For example, selecting Conservation from the Calculate menu will shade each column on the basis of the relative amino acid conservation seen at that particular position in the alignment. By doing so, it immediately becomes apparent which parts of the protein may lie within regions of secondary structure. Examine the area around positions 60 to 70 of the alignment;

the pattern observed should be two conserved, two unconserved, and two conserved residues, a parttern that is characteristic of an alpha-helix.

Select Jnet from the Align menu. This will return a secondary structure prediction based on the alignment. Alternatively, the alignment file can be submitted to the JPRED2 server at EBI. In order to submit the alignment to the JPRED2 server, the alignment must first be saved in MSF format (`ann_rep1_frags_del2.msf`). Either of these methods should corroborate that there is an alpha-helical region in the area around residues 60–70.

By "cleaning" the alignment in this way, information about sequences (and sequences themselves) has been discarded. It is advisable to always save files at intermediate steps: the clean alignment will be relatively easy to interpret, but the results of the intermediate steps will have information about the parts of the alignment requiring more thought.

Subfamily Analysis

The following steps will allow a subfamily analysis to be performed on the annexin family. The input file is `ideal_annexins.als`.

1. Start Jalview and read in the alignment file by typing `ideal_annexins.blc BLC`.
2. Select Average Distance Tree from the Calculate menu. The resultant tree will have four clear clusters with one outlier. Click on the tree at an appropriate position to draw a vertical line and highlight the four clusters.
3. Return to the Alignment window. Select Conservation from the Calculate menu. The most highly-conserved positions within each subgroup of sequences will be colored the brightest. Examine the alignment, and identify the charge-pair shown as an example in this Chapter. Selecting either the Taylor or Zappo color schemes may help in identifying the desired region.
4. Submit the file `ideal_annexins.blc` to the AMAS Web server. On the Web page, paste the contents of `ideal_annexins.blc` into the Alignment window, then paste the contents of the file `ideal_annexins.grp` into the Sensible Groups window. The server should return results quickly, providing links to a number of output files. The Pretty Output file contains the PostScript alignment, which should be identical to `ideal_annexins_amas.ps` provided here.

REFERENCES

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

Attwood, T. K., Croning, M. D. R., Flower, D. R., Lewis, A. P., Mabey, J. E., Scordis, P., Selley, J., and Wright. W. (2000). Prints-s: the database formerly known as prints. *Nucl. Acids Res.* 28, 225–227.

Barton, G. J. (1990). Protein multiple sequence alignment and flexible pattern matching. *Methods Enz.* 183, 403–428.

Barton, G. J. (1993). ALSCRIPT: A tool to format multiple sequence alignments. *Prot. Eng.* 6, 37–40.

Barton, G. J., Newman, R. H., Freemont, P. F., and Crumpton, M. J. (1991). Amino acid sequence analysis of the annexin super-gene family of proteins. *European J. Biochem.* 198, 749–760.

Barton, G. J., and Sternberg, M. J. E. Evaluation and improvements in the automatic alignment of protein sequences. (1987). *Prot. Eng.* 1, 89–94.

Barton, G. J., and Sternberg, M. J. E. (1987). A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons. *J. Mol. Biol.* 198, 327–337.

Barton, G. J., and Sternberg, M. J. E. (1990). Flexible protein sequence patternsa sensitive method to detect weak structural similarities. *J. Mol. Biol.* 212, 389–402.

Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Finn, R. D., and Sonnhammer, E. L. L. Pfam 3.1: 1313 multiple alignments match the majority of proteins. *Nucl. Acids Res.* 27, 260–262.

Cuff, J. A., and Barton, G. J. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins* 34, 508–519.

Cuff, J. A., and Barton, G. J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* 40, 502–511.

Cuff, J. A., Birney, E., Clamp, M. E., and Barton, G. J. (2000). ProtEST: Protein multiple sequence alignments from expressed sequence tags. *Bioinformatics* 6: 111–116.

Felsenstein, J. (1989). Phylip—phylogeny inference package (version 3.2). *Cladistics* 5, 164–166.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536–540.

Gotoh, O. (1996). Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.* 264, 823–838.

Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Nat. Acad. Sci.* USA 84, 4355–4358.

Huber, R., Romsich, J., and Paques, E.-P. (1990). The crystal and molecular structure of human annexin v, an anticoagulant protein that binds to calcium and membranes. *EMBO J.* 9, 3867–3874.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* 17, 195–202.

Lesk, A. M., Levitt, M., and Chothia, C. (1986). Alignment of the amino acid sequences of distantly related proteins using variable gap penalties. *Prot. Eng.* 1, 77–78.

Lipman, D. J., Altschul, S. F., and Kececioglu, J. D. (1989). A tool for multiple sequence alignment. *Proc. Nat. Acad. Sci.* USA 86, 4412–4415.

Livingstone, C. D., and Barton, G. J. (1993). Protein sequence alignments: A strategy for the hierarchical analysis of residue conservation. *Comp. App. Biosci.* 9, 745–756.

Pascarella, S., and Argos, P. (1992). Analysis of insertions/deletions in protein structures. *J. Mol. Biol.* 224, 461–471.

Roach, P. L., Clifton, I. J., Fulop, V., Harlos, K., Barton, G. J., Hajdu, J., Andersson, I., Schofield, C. J., and Baldwin, J. E. (1995). Crystal structure of isopenicillin n synthase is the first from a new structural family of enzymes. *Nature* 375, 700–704.

Rost, B., and Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* 232, 584–599.

Russell, R. B., and Barton, G. J. (1992). Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins* 14, 309–323.

Russell, R. B., and Barton, G. J. (1994). Structural features can be unconserved in proteins with similar folds. *J. Mol. Biol.* 244, 332–350.

Schultz, J., Milpetz, F., Bork, P., and Ponting, C. P. (1998). Smart, a simple modular architecture research tool: Identification of signalling domains. *Proc. Nat. Acad. Sci.* USA 95, 5857–5864.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.* 22, 4673–4680.

Zvelebil, M. J. J. M., Barton, G. J., Taylor, W. R., and Sternberg, M. J. E. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J. Mol. Biol.* 195, 957–961.

10

PREDICTIVE METHODS USING DNA SEQUENCES

Andreas D. Baxevanis

Genome Technology Branch
National Human Genome Research Institute
National Institutes of Health
Bethesda, Maryland

With the announcement of the completion of a ''working draft'' of the sequence of the human genome in June 2000 and the Human Genome Project targeting the completion of sequencing in 2002, investigators will be faced with the challenge of developing a strategy by which they can deal with the oncoming flood of both unfinished and finished data, whether the data are generated in their own laboratories or at one of the major sequencing centers. These data undergo what can best be described as a maturation process, starting as single reads off of a sequencing machine, passing through a phase where the data become part of an assembled (yet incomplete) sequence contig, and finally ending up as part of a finished, completely assembled sequence with an error rate of less than one in 10,000 bases. Even before such sequencing data reach this highly polished state, investigators can begin to ask whether or not given stretches of sequence represent coding or noncoding regions. The ability to make such determinations is of great relevance in the context of systematic sequencing efforts, since all of the data being generated by these projects are, in essence, ''anonymous'' in nature—nothing is known about the coding potential of these stretches of DNA as they are being sequenced. As such, automated methods will become increasingly important in annotating the human and other genomes to increase the intrinsic value of these data as they are being deposited into the public databases.

In considering the problem of gene identification, it is important to briefly go over the basic biology underlying what will become, in essence, a mathematical problem (Fig. 10.1). At the DNA level, upstream of a given gene, there are promoters and other regulatory elements that control the transcription of that gene. The gene itself is discontinuous, comprising both introns and exons. Once this stretch of DNA is transcribed into an RNA molecule, both ends of the RNA are modified, capping the 5′ end and placing a polyA signal at the 3′ end. The RNA molecule reaches maturity when the introns are spliced out, based on short consensus sequences found both at the intron-exon boundaries and within the introns themselves. Once splicing has occurred and the start and stop codons have been established, the mature mRNA is transported through a nuclear pore into the cytoplasm, at which point translation can take place.

Although the process of moving from DNA to protein is obviously more complex in eukaryotes than it is in prokaryotes, the mere fact that it can be described in its entirety in eukaryotes would lead one to believe that predictions can confidently be made as to the exact positions of introns and exons. Unfortunately, the signals that control the process of moving from the DNA level to the protein level are not very well defined, precluding their use as foolproof indicators of gene structure. For example, upward of 70% of the promoter regions contain a TATA box, but, because the remainder do not, the presence (or absence) of the TATA box in and of itself cannot be used to assess whether a region is a promoter. Similarly, during end modification, the polyA tail may be present or absent or may not contain the canonical

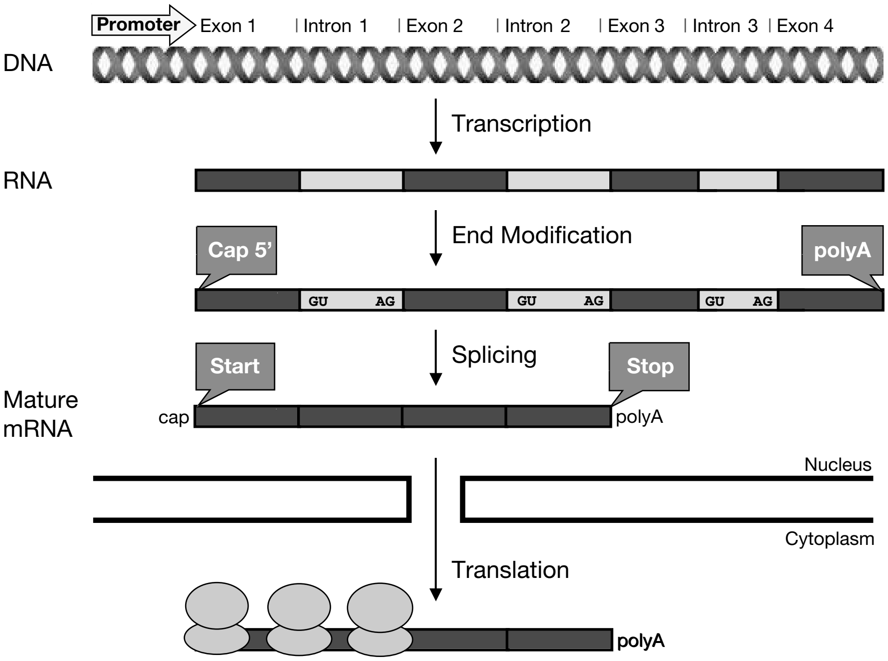


Figure 10.1. The central dogma. Proceeding from the DNA through the RNA to the protein level, various sequence features and modifications can be identified that can be used in the computational deduction of gene structure. These include the presence of promoter and regulatory regions, intron-exon boundaries, and both start and stop signals. Unfortunately, these signals are not always present, and, when they are present, they may not always be in the same form or context. The reader is referred to the text for greater detail.

AATAAA. Adding to these complications is the fact that an open reading frame is required *but is not sufficient* for judging a region as being an exon. Given these and other considerations, there is at present no straightforward method that will allow for 100% confidence in the prediction of an intron or an exon. Despite this, a combinatorial approach can be used, relying on a number of methods, to increase the confidence with which gene structure is predicted.

Briefly, gene-finding strategies can be grouped into three major categories. ***Content-based methods*** rely on the overall, bulk properties of a sequence in making a determination. Characteristics considered here include how often particular codons are used, the periodicity of repeats, and the compositional complexity of the sequence. Because different organisms use synonymous codons with different frequency, such clues can provide insight into determining regions that are more likely to be exons. In ***site-based methods***, the focus turns to the presence or absence of a specific sequence, pattern, or consensus. These methods are used to detect features such as donor and acceptor splice sites, binding sites for transcription factors, polyA tracts, and start and stop codons. Finally, ***comparative methods*** make determinations based on sequence homology. Here, translated sequences are subjected to database searches against protein sequences (cf. Chapter 8) to determine whether a previously characterized coding region corresponds to a region in the query sequence. Although this is conceptually the most straightforward of the methods, it is restrictive because most newly discovered genes do not have gene products that match anything in the protein databases. Also, the modular nature of proteins and the fact that there are only a limited number of protein motifs (Chothia and Lesk, 1986) make predicting anything more than just exonic regions in this way difficult. The reader is referred to a number of excellent reviews detailing the theoretical underpinnings of these various classes of methods (Claverie, 1997a; Claverie, 1997b; Guigó, 1997; Snyder and Stormo, 1997; Claverie, 1998; Rogic et al., 2001). Although many of the gene prediction methods belong strictly to one of these three classes of methods, most of the methods that will be discussed here use the strength of combining different classes of methods to optimize predictions.

With the complexity of the problem at hand and the various approaches described above for tackling the problem, it becomes important for investigators to gain an appreciation for when and how each particular method should be applied. A recurring theme in this chapter will be the fact that, *depending on the nature of the data, each method will perform differently*. Put another way, although one method may be best for human finished sequences, another may be better for unfinished sequences or for sequences from another organism. In this chapter, we will examine a number of the commonly used methods that are freely available in the public domain, focusing on their application to human sequence data; this will be followed by a general discussion of gene-finding strategy.

GRAIL

GRAIL, which stands for Gene Recognition and Analysis Internet Link (Uberbacher and Mural, 1991; Mural et al., 1992), is the elder statesman of the gene prediction techniques because it is among the first of the techniques developed in this area and enjoys widespread usage. As more and more has become known about gene structure in general and better Internet tools have become more widespread, GRAIL has continuously evolved to keep in step with the current state of the field.

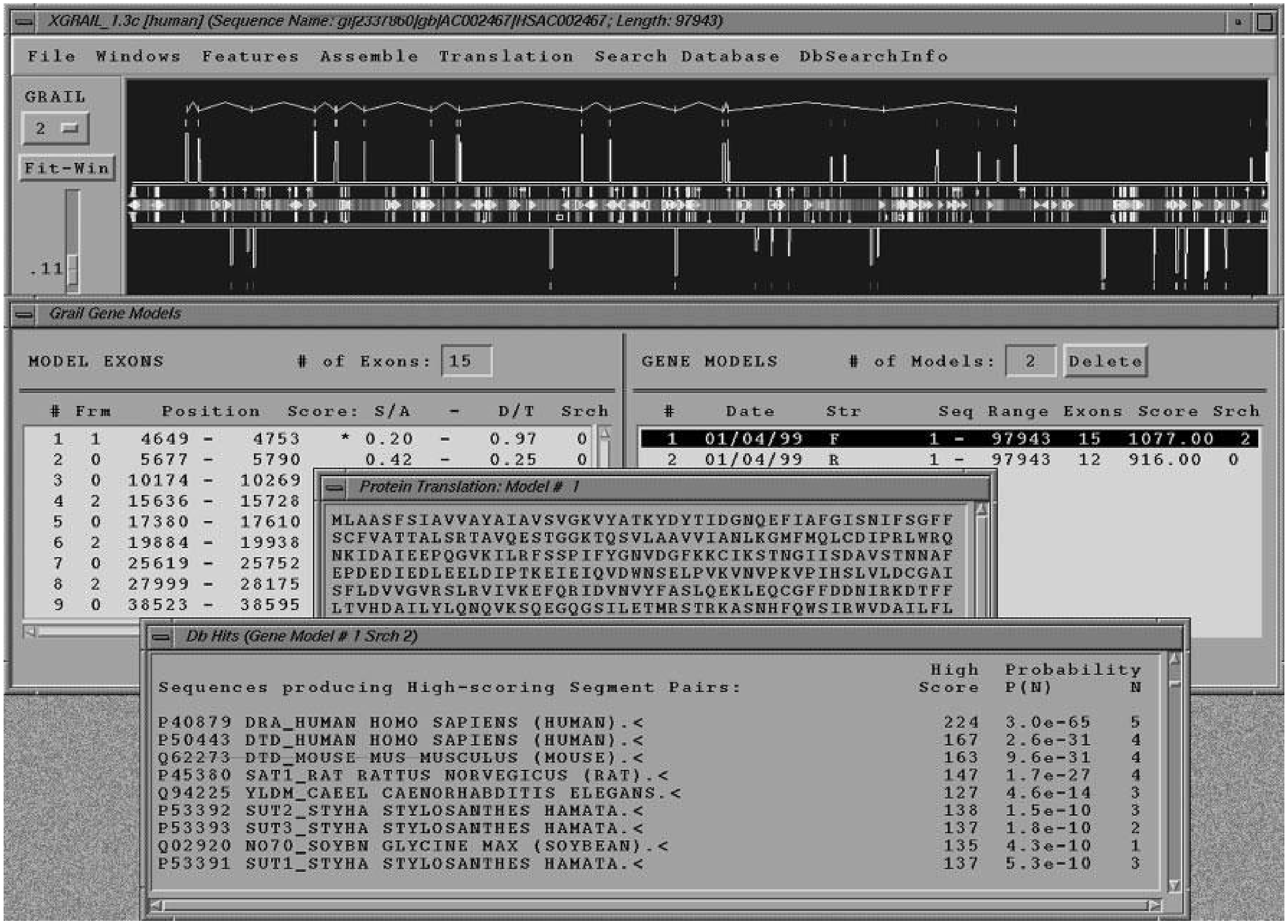
There are two basic GRAIL versions that will be discussed in the context of this discussion. GRAIL 1 makes use of a neural network method to recognize coding potential in fixed-length (100 base) windows considering the sequence itself, without looking for additional features such as splice junctions or start and stop codons. An improved version of GRAIL 1 (called GRAIL 1a) expands on this method by considering regions immediately adjacent to regions deemed to have coding potential, resulting in better performance in both finding true exons and eliminating false positives. Either GRAIL 1 or GRAIL 1a would be appropriate in the context of searching for single exons. A further refinement led to a second version, called GRAIL 2, in which variable-length windows are used and contextual information (e.g., splice junctions, start and stop codons, polyA signals) is considered. Because GRAIL 2 makes its prediction by taking genomic context into account, it is appropriate for determining model gene structures.

In this chapter, the output of each of the methods discussed will be shown using the same set of input data as the query. The sequence that will be considered is that of a human BAC clone RG364P16 from 7q31, a clone established as part of the systematic sequencing of chromosome 7 (GenBank AC002467). By using the same example throughout, the strengths and weaknesses of each of the discussed methods can be highlighted. For purposes of this example, a client-server application called XGRAIL will be used. This software, which runs on the UNIX platform, allows for graphical output of GRAIL 1/1a/2 results, as shown in Figure 10.2. Because the DNA sequence in question is rather large and is apt to contain at least one gene, GRAIL 2 was selected as the method. The large, upper window presents an overview of the ~98 kb making up this clone, and the user can selectively turn on or off particular markings that identify features within the sequence (described in the figure legend). Of most importance in this view is the prediction of exons at the very top of the window, with the histogram representing the probability that a given region represents an exon. Information on each one of the predicted exons is shown in the Model Exons window, and the model exons can be assembled and shown as both Model Genes and as a Protein Translation. Only putative exons with acceptable probability values (as defined in the GRAIL algorithm) are included in the gene models. The protein translation can, in turn, be searched against the public databases to find sequence homologs using a program called genQuest (integrated into XGRAIL), and these are shown in the Db Hits window. In this case, the 15 exons in the first gene model (from the forward strand) are translated into a protein that shows significant sequence homology to a group of proteins putatively involved in anion transport (Everett et al., 1997).

Most recently, the authors of GRAIL have released GRAIL-EXP, which is based on GRAIL but uses additional information in making the predictions, including a database search of known complete and partial gene messages. The inclusion of this database search in deducing gene models has greatly improved the performance of the original GRAIL algorithm.

FGENEH/FGENES

FGENEH, developed by Victor Solovyev and colleagues, is a method that predicts internal exons by looking for structural features such as donor and acceptor splice


XGRAIL_1.3c [human] (Sequence Name: gi|2337860|gb|AC002467|HSAC002467; Length: 97943)
File Windows Features Assemble Translation Search Database DbSearchInfo
GRAIL
2
Fit-Win
.11
Grail Gene Models
Protein Translation: Model # 1
Db Hits (Gene Model # 1 Srch 2)
Delete

```
MODEL EXONS          # of Exons: 15

 # Frm   Position    Score: S/A   -   D/T  Srch
 1  1   4649 -  4753    * 0.20   -   0.97   0
 2  0   5677 -  5790      0.42   -   0.25   0
 3  0  10174 - 10269
 4  2  15636 - 15728
 5  0  17380 - 17610
 6  2  19884 - 19938
 7  0  25619 - 25752
 8  2  27999 - 28175
 9  0  38523 - 38595
```

```
GENE MODELS     # of Models: 2

 #    Date     Str    Seq Range   Exons   Score   Srch
 1  01/04/99   F     1 - 97943    15    1077.00   2
 2  01/04/99   R     1 - 97943    12     916.00   0
```

```
MLAASFSIAVVAYAIAVSVGKVYATKYDYTIDGNQEFIAFGISNIFSGFF
SCFVATTALSRTAVQESTGGKTQSVLAAVVIANLKGMFMQLCDIPRLWRQ
NKIDAIEEPQGVKILRFSSPIFYGNVDGFKKCIKSTNGIISDAVSTNNAF
EPDEDIEDLEELDIPTKEIEIQVDWNSELPVKVNVPKVPIHSLVLDCGAI
SFLDVVGVRSLRVIVKEFQRIDVNVYFASLQEKLEQCGFFDDNIRKDTFF
LTVHDAILYLQNQVKSQEGQGSILETMRSTRKASNHFQWSIRWVDAILFL
```

```
                                                    High  Probability
Sequences producing High-scoring Segment Pairs:    Score  P(N)       N

P40879 DRA_HUMAN HOMO SAPIENS (HUMAN).<             224   3.0e-65    5
P50443 DTD_HUMAN HOMO SAPIENS (HUMAN).<             167   2.6e-31    4
Q62273 DTD_MOUSE MUS MUSCULUS (MOUSE).<             163   9.6e-31    4
P45380 SAT1_RAT RATTUS NORVEGICUS (RAT).<           147   1.7e-27    4
Q94225 YLDM_CAEEL CAENORHABDITIS ELEGANS.<          127   4.6e-14    3
P53392 SUT2_STYHA STYLOSANTHES HAMATA.<             138   1.5e-10    3
P53393 SUT3_STYHA STYLOSANTHES HAMATA.<             137   1.8e-10    2
Q02920 NO70_SOYBN GLYCINE MAX (SOYBEAN).<           135   4.3e-10    1
P53391 SUT1_STYHA STYLOSANTHES HAMATA.<             137   5.3e-10    3
```

Figure 10.2. XGRAIL output using the human BAC clone RG364P16 from 7q31 as the query. The upper window shows the results of the prediction, with the histogram representing the probability that a given stretch of DNA is an exon. The various bars in the center represent features of the DNA (e.g., arrows represent repetitive DNA, and vertical bars represent repeat sequences). Exon and gene models, protein translations, and the results of a genQuest search using the protein translation are shown. (See color plate.)

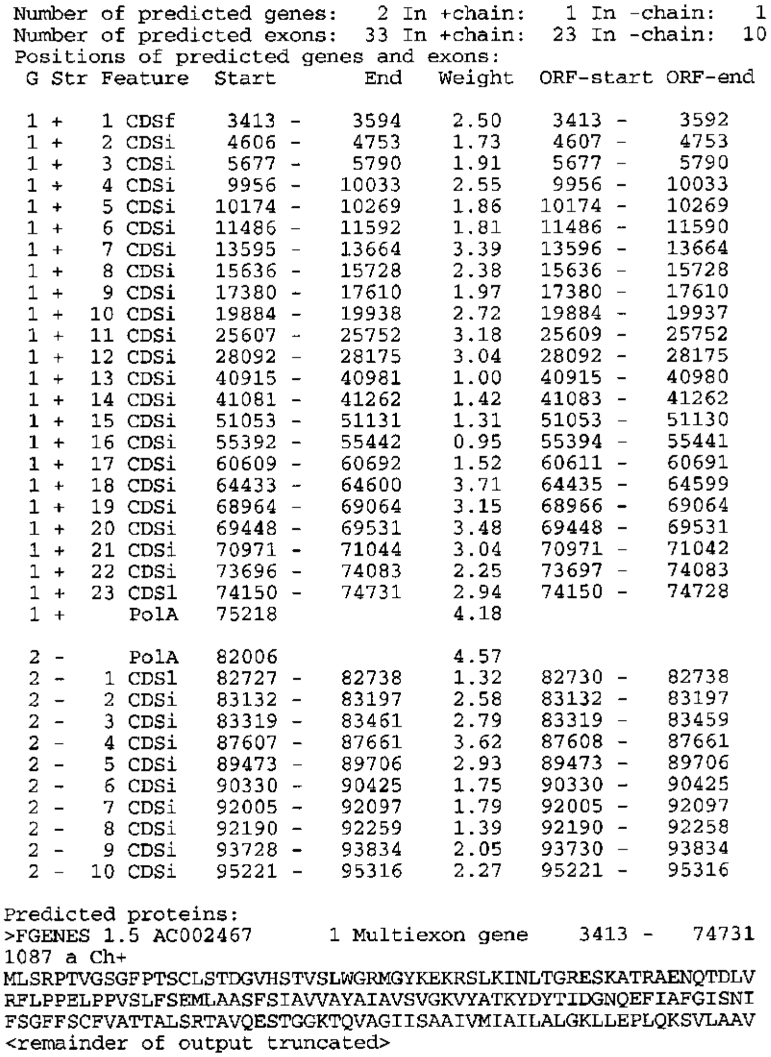
sites, putative coding regions, and intronic regions both 5′ and 3′ to the putative exon (Solovyev et al., 1994a; Solovyev et al., 1994b; Solovyev et al., 1995). The method makes use of *linear discriminant analysis*, a mathematical technique that allows data from multiple experiments to be combined. Once the data are combined, a linear function is used to discriminate between two classes of events—here, whether a given stretch of DNA is or is not an exon. In FGENEH, results of the linear discriminant approach are then passed to a dynamic programming algorithm that determines how to best combine these predicted exons into a coherent gene model. An extension of FGENEH, called FGENES, can be used in cases when multiple genes are expected in a given stretch of DNA.

The Sanger Centre Web server provides a very simple front-end for performing FGENES. The query sequence (again, the BAC clone from 7q31) is pasted into the query box, an identifier is entered, and the search can then be performed. The results are returned in a tabular format, as shown in Figure 10.3. The total number of predicted genes and exons (2 and 33, respectively) is shown at the top of the output. The information for each gene (`G`) then follows. For each predicted exon, the strand (`Str`) is given, with + indicating the forward strand and – indicating the reverse. The Feature list in this particular case includes initial exons (`CDSf`), internal exons (`CDSi`), terminal exons (`CDSl`), and polyA regions (`PolA`). The nucleotide region for the predicted feature is then given as a range. In the current example, the features of the second predicted gene are shown in reverse order, since the prediction is based on the reverse strand. On the basis of the information in the table, predicted proteins are given at the bottom of the output in FASTA format. The definition line for each of the predicted proteins gives the range of nucleotide residues involved, as well as the total length of the protein and the direction (`+/-`) of the predicted gene.

MZEF

MZEF stands for ‘‘Michael Zhang’s Exon Finder,’’ after its author at the Cold Spring Harbor Laboratory. The predictions rely on a technique called *quadratic discriminant analysis* (Zhang, 1997). Imagine a case in which the results of two types of predictions are plotted against each other on a simple *XY* graph (for instance, splice site scores vs. exon length). If the relationship between these two sets of data is nonlinear or multivariate, the resulting graph will look like a swarm of points. Points lying in only a small part of this swarm will represent a ‘‘correct’’ prediction; to separate the correctly predicted points from the incorrectly predicted points in the swarm, a quadratic function is used, hence the name of the technique. In the case of MZEF, the measured variables include exon length, intron-exon and exon-intron transitions, branch sites, 3′ and 5′ splice sites, and exon, strand, and frame scores. MZEF is intended to predict internal coding exons and does not give any other information with respect to gene structure.

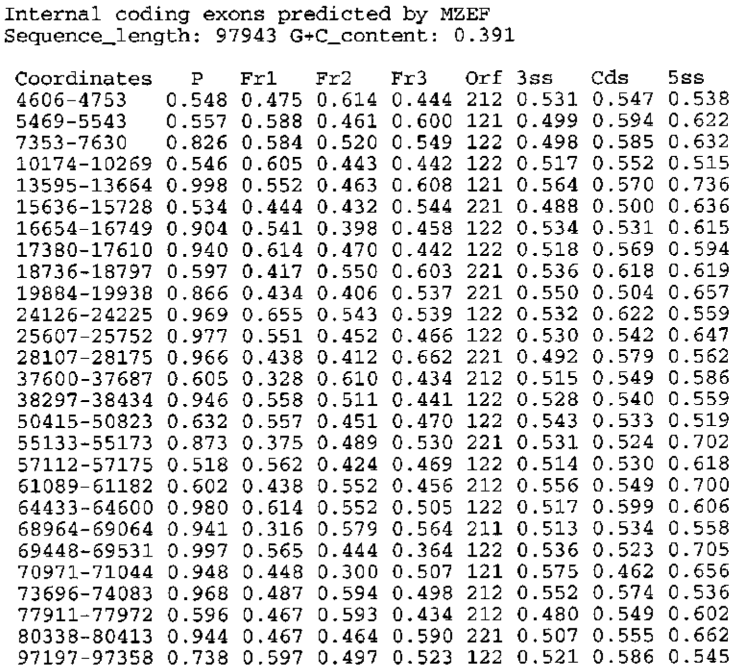
There are two implementations of MZEF currently available. The program can be downloaded from the CSHL FTP site for UNIX command-line use, or the program can be accessed through a Web front-end. The input is a single sequence, read in only one direction (either the forward *or* the reverse strand); to perform MZEF on both strands, the program must be run twice. Returning to the BAC clone from chromosome 7, MZEF predicts a total of 27 exons in the forward strand (Fig. 10.4). Focusing in on the first two columns of the table, the region of the prediction is



```
Number of predicted genes:   2 In +chain:   1 In -chain:   1
Number of predicted exons:  33 In +chain:  23 In -chain:  10
Positions of predicted genes and exons:
 G Str Feature  Start       End   Weight  ORF-start ORF-end

 1 +   1 CDSf    3413 -    3594    2.50    3413 -    3592
 1 +   2 CDSi    4606 -    4753    1.73    4607 -    4753
 1 +   3 CDSi    5677 -    5790    1.91    5677 -    5790
 1 +   4 CDSi    9956 -   10033    2.55    9956 -   10033
 1 +   5 CDSi   10174 -   10269    1.86   10174 -   10269
 1 +   6 CDSi   11486 -   11592    1.81   11486 -   11590
 1 +   7 CDSi   13595 -   13664    3.39   13596 -   13664
 1 +   8 CDSi   15636 -   15728    2.38   15636 -   15728
 1 +   9 CDSi   17380 -   17610    1.97   17380 -   17610
 1 +  10 CDSi   19884 -   19938    2.72   19884 -   19937
 1 +  11 CDSi   25607 -   25752    3.18   25609 -   25752
 1 +  12 CDSi   28092 -   28175    3.04   28092 -   28175
 1 +  13 CDSi   40915 -   40981    1.00   40915 -   40980
 1 +  14 CDSi   41081 -   41262    1.42   41083 -   41262
 1 +  15 CDSi   51053 -   51131    1.31   51053 -   51130
 1 +  16 CDSi   55392 -   55442    0.95   55394 -   55441
 1 +  17 CDSi   60609 -   60692    1.52   60611 -   60691
 1 +  18 CDSi   64433 -   64600    3.71   64435 -   64599
 1 +  19 CDSi   68964 -   69064    3.15   68966 -   69064
 1 +  20 CDSi   69448 -   69531    3.48   69448 -   69531
 1 +  21 CDSi   70971 -   71044    3.04   70971 -   71042
 1 +  22 CDSi   73696 -   74083    2.25   73697 -   74083
 1 +  23 CDSl   74150 -   74731    2.94   74150 -   74728
 1 +     PolA   75218              4.18

 2 -     PolA   82006              4.57
 2 -   1 CDSl   82727 -   82738    1.32   82730 -   82738
 2 -   2 CDSi   83132 -   83197    2.58   83132 -   83197
 2 -   3 CDSi   83319 -   83461    2.79   83319 -   83459
 2 -   4 CDSi   87607 -   87661    3.62   87608 -   87661
 2 -   5 CDSi   89473 -   89706    2.93   89473 -   89706
 2 -   6 CDSi   90330 -   90425    1.75   90330 -   90425
 2 -   7 CDSi   92005 -   92097    1.79   92005 -   92097
 2 -   8 CDSi   92190 -   92259    1.39   92190 -   92258
 2 -   9 CDSi   93728 -   93834    2.05   93730 -   93834
 2 -  10 CDSi   95221 -   95316    2.27   95221 -   95316

Predicted proteins:
>FGENES 1.5 AC002467       1 Multiexon gene    3413 -   74731
1087 a Ch+
MLSRPTVGSGFPTSCLSTDGVHSTVSLWGRMGYKEKRSLKINLTGRESKATRAENQTDLV
RFLPPELPPVSLFSEMLAASFSIAVVAYAIAVSVGKVYATKYDYTIDGNQEFIAFGISNI
FSGFFSCFVATTALSRTAVQESTGGKTQVAGIISAAIVMIAILALGKLLEPLQKSVLAAV
<remainder of output truncated>
```

Figure 10.3. FGENES output using the human BAC clone RG364P16 from 7q31 as the query. The columns, going from left to right, represent the gene number (G), strand (Str), feature (described in the main text), start and end points for the predicted exon, a scoring weight, and start and end points for corresponding open reading frames (ORF-start and ORF-end). Each predicted gene is shown as a separate block. The tables are followed by protein translations of any predicted gene products.

given as a range, followed by the probability that the prediction is correct (P). Predictions with $P > 0.5$ are considered correct and are included in the table. Immediately, one begins to see the difference in the predictions between methods. MZEF is again geared toward finding single exons; therefore, the exons are not shown in the context of a putative gene, as they are in GRAIL 2 or FGENES. However, the exons predicted by these methods are not the same, a point that we will return to later in this discussion.



```
Internal coding exons predicted by MZEF
Sequence_length: 97943 G+C_content: 0.391

 Coordinates     P    Fr1   Fr2   Fr3   Orf 3ss   Cds   5ss
 4606-4753    0.548 0.475 0.614 0.444 212 0.531 0.547 0.538
 5469-5543    0.557 0.588 0.461 0.600 121 0.499 0.594 0.622
 7353-7630    0.826 0.584 0.520 0.549 122 0.498 0.585 0.632
 10174-10269 0.546 0.605 0.443 0.442 122 0.517 0.552 0.515
 13595-13664 0.998 0.552 0.463 0.608 121 0.564 0.570 0.736
 15636-15728 0.534 0.444 0.432 0.544 221 0.488 0.500 0.636
 16654-16749 0.904 0.541 0.398 0.458 122 0.534 0.531 0.615
 17380-17610 0.940 0.614 0.470 0.442 122 0.518 0.569 0.594
 18736-18797 0.597 0.417 0.550 0.603 221 0.536 0.618 0.619
 19884-19938 0.866 0.434 0.406 0.537 221 0.550 0.504 0.657
 24126-24225 0.969 0.655 0.543 0.539 122 0.532 0.622 0.559
 25607-25752 0.977 0.551 0.452 0.466 122 0.530 0.542 0.647
 28107-28175 0.966 0.438 0.412 0.662 221 0.492 0.579 0.562
 37600-37687 0.605 0.328 0.610 0.434 212 0.515 0.549 0.586
 38297-38434 0.946 0.558 0.511 0.441 122 0.528 0.540 0.559
 50415-50823 0.632 0.557 0.451 0.470 122 0.543 0.533 0.519
 55133-55173 0.873 0.375 0.489 0.530 221 0.531 0.524 0.702
 57112-57175 0.518 0.562 0.424 0.469 122 0.514 0.530 0.618
 61089-61182 0.602 0.438 0.552 0.456 212 0.556 0.549 0.700
 64433-64600 0.980 0.614 0.552 0.505 122 0.517 0.599 0.606
 68964-69064 0.941 0.316 0.579 0.564 211 0.513 0.534 0.558
 69448-69531 0.997 0.565 0.444 0.364 122 0.536 0.523 0.705
 70971-71044 0.948 0.448 0.300 0.507 121 0.575 0.462 0.656
 73696-74083 0.968 0.487 0.594 0.498 212 0.552 0.574 0.536
 77911-77972 0.596 0.467 0.593 0.434 212 0.480 0.549 0.602
 80338-80413 0.944 0.467 0.464 0.590 221 0.507 0.555 0.662
 97197-97358 0.738 0.597 0.497 0.523 122 0.521 0.586 0.545
```

Figure 10.4. MZEF output using the human BAC clone RG364P16 from 7q31 as the query. The columns, going from left to right, give the location of the prediction as a range of included bases (Coordinates), the probability value (P), frame preference scores, an ORF indicator showing which reading frames are open, and scores for the 3′ splice site, coding regions, and 5′ splice site.

GENSCAN

GENSCAN, developed by Chris Burge and Sam Karlin (Burge and Karlin, 1997; Burge and Karlin, 1998), is designed to predict complete gene structures. As such, GENSCAN can identify introns, exons, promoter sites, and polyA signals, as do a number of the other gene identification algorithms. Like FGENES, GENSCAN does not expect the input sequence to represent one and only one gene or one and only one exon: it can accurately make predictions for sequences representing either partial genes or multiple genes separated by intergenic DNA. The ability to make these predictions accurately when a sequence is in a variety of contexts makes GENSCAN a particularly useful method for gene identification.

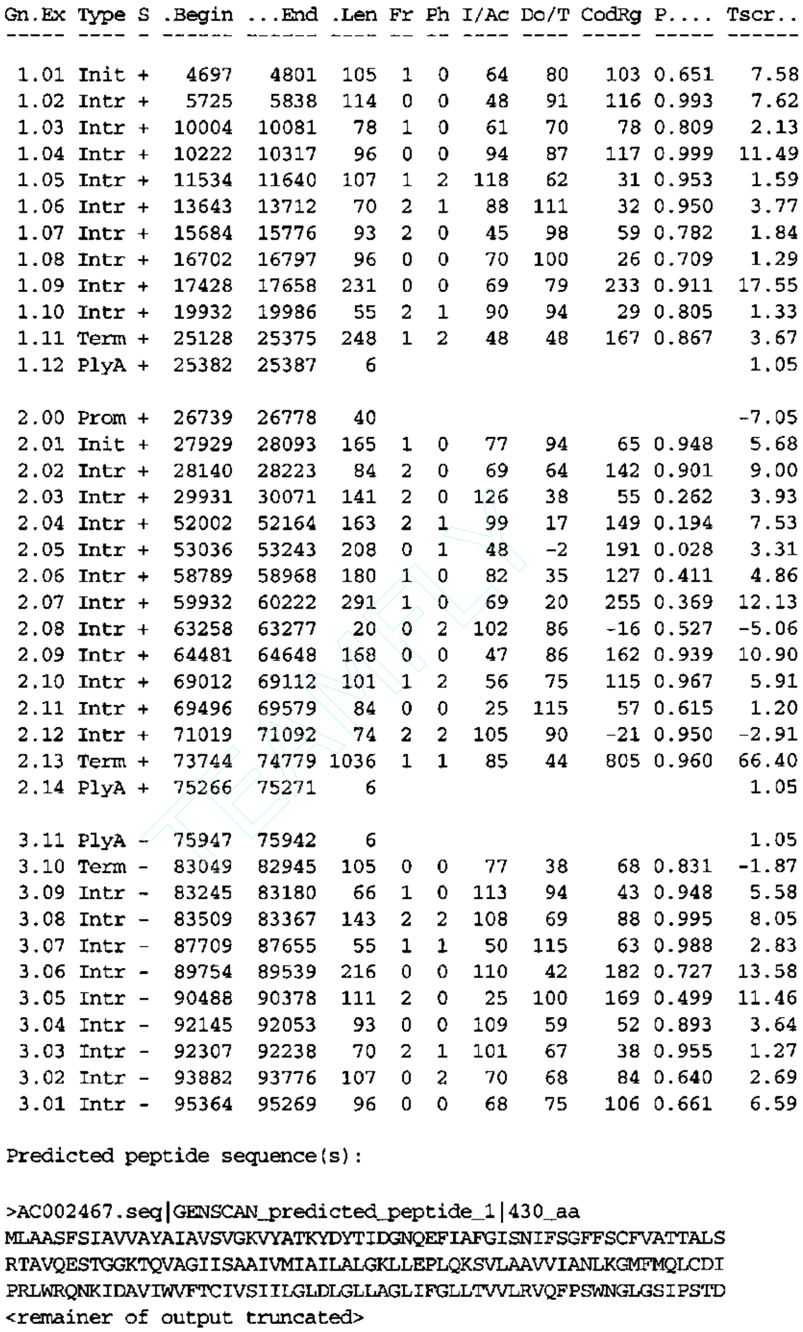
GENSCAN relies on what the author terms a "probabilistic model" of genomic sequence composition and gene structure. By looking for gene structure descriptions that match or are consistent with the query sequence, the algorithm can assign a probability as to the chance that a given stretch of sequence represents an exon, promoter, and so forth. The "optimal exons" are the ones with the highest probability and represent the part of the query sequence having the best chance of actually being an exon. The method will also predict "suboptimal exons," stretches of sequence having an acceptable probability value but one not as good as the optimal one. The authors of the method encourage users to examine both sets of predictions so that alternatively spliced regions of genes or other nonstandard gene structures are not missed.

With the use of the human BAC clone from 7q31 again, the query can be issued directly from the GENSCAN Web site, using Vertebrate as the organism, the default suboptimal cutoff, and Predicted Peptides Only as the print option. The results for this query are shown in Figure 10.5. The output indicates that there are three genes in this region, with the first gene having 11 exons, the second gene having 13 exons, and the third gene having 10 exons. The most important columns in the table are those labeled `Type` and `P`. The `Type` column indicates whether the prediction is for an initial exon (`Init`), an internal exon (`Intr`), a terminal exon (`Term`), a single-exon gene (`Sngl`), a promoter region (`Prom`), or a polyA signal (`PlyA`). The `P` column gives the probability that this prediction is actually correct. GENSCAN exons having a very high probability value ($P > 0.99$) are 97.7% accurate where the prediction matches a true, annotated exon. These high-probability predictions can be used in the rational design of PCR primers for cDNA amplification or for other purposes where extremely high confidence is necessary. GENSCAN exons that have probabilities in the range from 0.50 to 0.99 are deemed to be correct most of the time; the best-case accuracies for *P*-values over 0.90 is on the order of 88%. Any predictions below 0.50 should be discarded as unreliable, and those data are not given in the table. An alternative view of the data is shown in Figure 10.6. Here, both the optimal and suboptimal exons are shown, with the initial and terminal exons showing the direction in which the prediction is being made ($5' \rightarrow 3'$ or $3' \rightarrow 5'$). This view is particularly useful for large stretches of DNA, as the tables become harder to interpret when more and more exons are predicted.

By the time of this printing, a new program named GenomeScan will be available from the Burge laboratory at MIT. GenomeScan assigns a higher score to putative exons that overlap BLASTX hits than to comparable exons for which similarity evidence is lacking. Regions of higher similarity (according to BLASTX E-value, for example) are accorded more confidence than regions of lower similarity, since weak similarities sometimes do not represent homology. Thus, the predictions of GenomeScan tend to be consistent with all or almost all of the regions of high detected similarity but may sometimes ignore a region of weak similarity that either has weak intrinsic properties (e.g., poor splice signals) or is inconsistent with other extrinsic information. The accuracy of GenomeScan tends to be significantly higher than that of GENSCAN when a moderate or closely related protein sequence is available. An example of the improved accuracy of GenomeScan over GENSCAN, using the human BRCA1 gene as the query, is shown in Figure 10.7.

PROCRUSTES

Greek mythology heralds the story of Theseus, the king of Athens who underwent many trials and tribulations on his way to becoming a hero, along with Hercules. As if Amazons and the Minotaur were not enough, in the course of his travels, Theseus happened upon Procrustes, a bandit with a warped idea of hospitality. Procrustes, which means “he who stretches,” would invite passersby into his home for a meal and a night’s stay in his guest bed. The problem lay, quite literally, in the bed, in that Procrustes would make sure that his guests fit in the bed by stretching them out on a rack if they were too short or by chopping off their legs if they were too long.



```
Gn.Ex Type S .Begin ...End .Len Fr Ph I/Ac Do/T CodRg P.... Tscr..
----- ---- - ------ ------ ---- -- -- ---- ---- ----- ----- ------

 1.01 Init +   4697   4801  105  1  0   64   80   103 0.651   7.58
 1.02 Intr +   5725   5838  114  0  0   48   91   116 0.993   7.62
 1.03 Intr +  10004  10081   78  1  0   61   70    78 0.809   2.13
 1.04 Intr +  10222  10317   96  0  0   94   87   117 0.999  11.49
 1.05 Intr +  11534  11640  107  1  2  118   62    31 0.953   1.59
 1.06 Intr +  13643  13712   70  2  1   88  111    32 0.950   3.77
 1.07 Intr +  15684  15776   93  2  0   45   98    59 0.782   1.84
 1.08 Intr +  16702  16797   96  0  0   70  100    26 0.709   1.29
 1.09 Intr +  17428  17658  231  0  0   69   79   233 0.911  17.55
 1.10 Intr +  19932  19986   55  2  1   90   94    29 0.805   1.33
 1.11 Term +  25128  25375  248  1  2   48   48   167 0.867   3.67
 1.12 PlyA +  25382  25387    6                               1.05

 2.00 Prom +  26739  26778   40                              -7.05
 2.01 Init +  27929  28093  165  1  0   77   94    65 0.948   5.68
 2.02 Intr +  28140  28223   84  2  0   69   64   142 0.901   9.00
 2.03 Intr +  29931  30071  141  2  0  126   38    55 0.262   3.93
 2.04 Intr +  52002  52164  163  2  1   99   17   149 0.194   7.53
 2.05 Intr +  53036  53243  208  0  1   48   -2   191 0.028   3.31
 2.06 Intr +  58789  58968  180  1  0   82   35   127 0.411   4.86
 2.07 Intr +  59932  60222  291  1  0   69   20   255 0.369  12.13
 2.08 Intr +  63258  63277   20  0  2  102   86   -16 0.527  -5.06
 2.09 Intr +  64481  64648  168  0  0   47   86   162 0.939  10.90
 2.10 Intr +  69012  69112  101  1  2   56   75   115 0.967   5.91
 2.11 Intr +  69496  69579   84  0  0   25  115    57 0.615   1.20
 2.12 Intr +  71019  71092   74  2  2  105   90   -21 0.950  -2.91
 2.13 Term +  73744  74779 1036  1  1   85   44   805 0.960  66.40
 2.14 PlyA +  75266  75271    6                               1.05

 3.11 PlyA -  75947  75942    6                               1.05
 3.10 Term -  83049  82945  105  0  0   77   38    68 0.831  -1.87
 3.09 Intr -  83245  83180   66  1  0  113   94    43 0.948   5.58
 3.08 Intr -  83509  83367  143  2  2  108   69    88 0.995   8.05
 3.07 Intr -  87709  87655   55  1  1   50  115    63 0.988   2.83
 3.06 Intr -  89754  89539  216  0  0  110   42   182 0.727  13.58
 3.05 Intr -  90488  90378  111  2  0   25  100   169 0.499  11.46
 3.04 Intr -  92145  92053   93  0  0  109   59    52 0.893   3.64
 3.03 Intr -  92307  92238   70  2  1  101   67    38 0.955   1.27
 3.02 Intr -  93882  93776  107  0  2   70   68    84 0.640   2.69
 3.01 Intr -  95364  95269   96  0  0   68   75   106 0.661   6.59

Predicted peptide sequence(s):

>AC002467.seq|GENSCAN_predicted_peptide_1|430_aa
MLAASFSIAVVAYAIAVSVGKVYATKYDYTIDGNQEFIAFGISNIFSGFFSCFVATTALS
RTAVQESTGGKTQVAGIISAAIVMIAILALGKLLEPLQKSVLAAVVIANLKGMFMQLCDI
PRLWRQNKIDAVIWVFTCIVSIILGLDLGLLAGLIFGLLTVVLRVQFPSWNGLGSIPSTD
<remainer of output truncated>
```

Figure 10.5. GENSCAN output using the human BAC clone RG364P16 from 7q31 as the query. The columns, going from left to right, represent the gene and exon number (`Gn.Ex`), the type of prediction (`Type`), the strand on which the prediction was made (`S`, with + as the forward strand and − as the reverse), the beginning and endpoints for the prediction (`Begin` and `End`), the length of the prediction (`Len`), the reading frame of the prediction (`Fr`), several scoring columns, and the probability value (`P`). Each predicted gene is shown as a separate block; notice that the third gene has its exons listed in reverse order, reflecting that the prediction is on the reverse strand. The tables are followed by the protein translations for each of the three predicted genes.

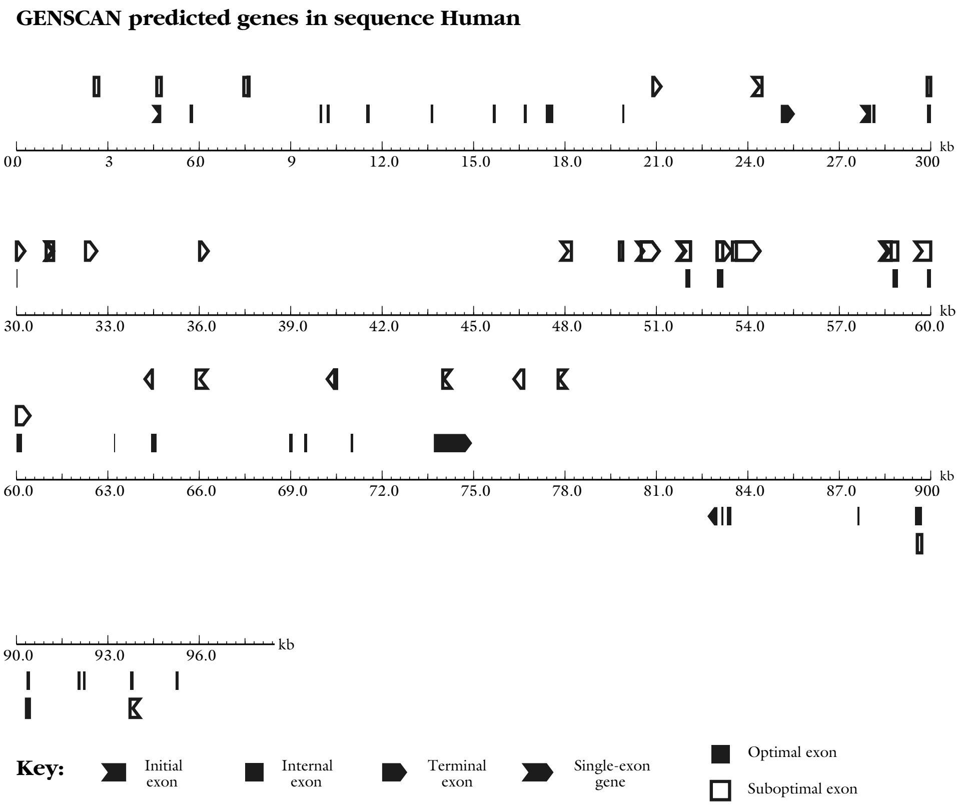


Figure 10.6. GENSCAN output in graphical form, using the human BAC clone RG364P16 from 7q31 as the query. Optimal and suboptimal exons are indicated, and the initial and terminal exons show the direction in which the prediction is being made (5′ → 3′ or 3′ → 5′).

Theseus made short order of Procrustes by fitting him to his own bed, thereby sparing any other traveler the same fate. On the basis of this story, the phrase ‘‘bed of Procrustes’’ has come to convey the idea of forcing something to fit where it normally would not.

Living up to its namesake, PROCRUSTES takes genomic DNA sequences and ‘‘forces’’ them to fit into a pattern as defined by a related target protein (Gelfand et al., 1996). Unlike the other gene prediction methods that have been discussed, the algorithm does not use a DNA sequence *on its own* to look for content- or site-based signals. Instead, the algorithm requires that the user identify putative gene products *before* the prediction is made, so that the prediction represents the best fit of the given DNA sequence to its putative transcription product. The method uses a spliced alignment algorithm to sequentially explore all possible exon assemblies, looking for the best fit of predicted gene structure to candidate protein. If the candidate protein is known to arise from the query DNA sequence, correct gene structures can be predicted with an accuracy of 99% or better. By making use of candidate proteins in the course of the prediction, PROCRUSTES can take advantage of information known about this protein or related proteins in the public databases to better deter-

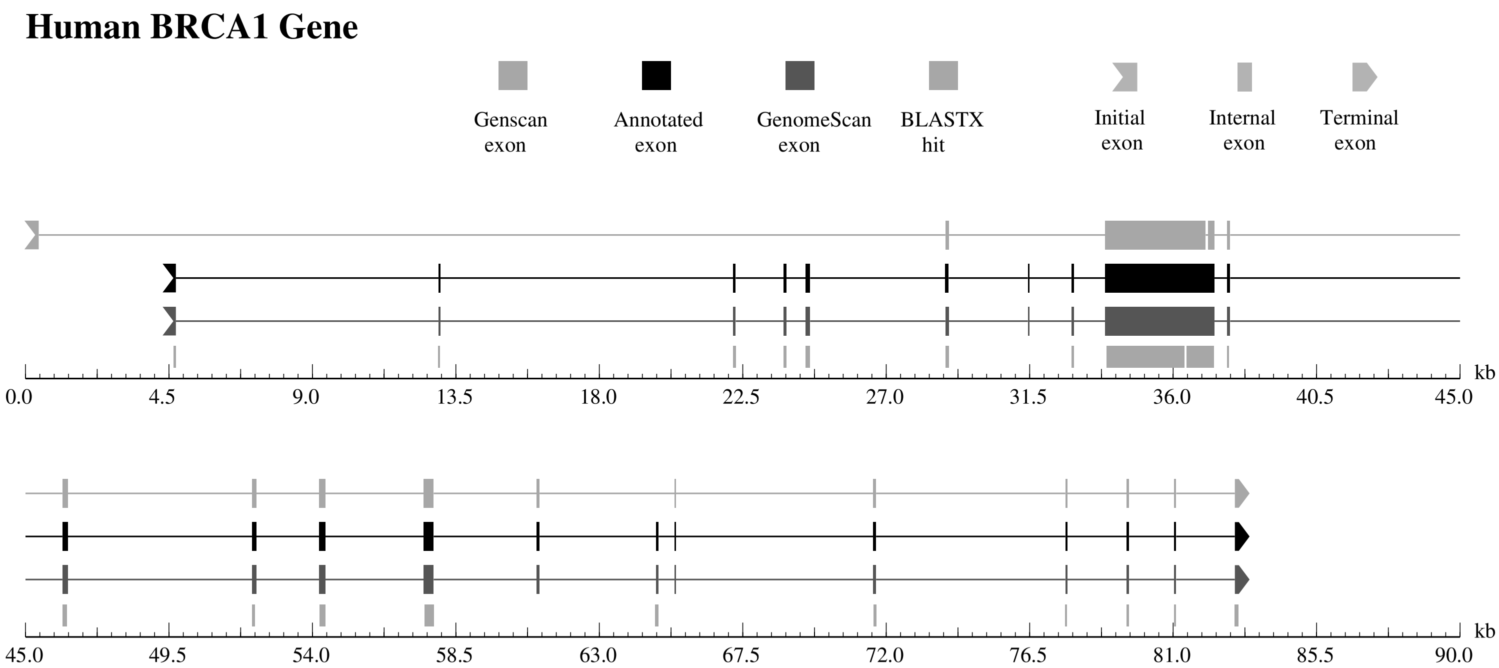


Figure 10.7. Comparison of GENSCAN with GenomeScan, using the human BRCA1 gene sequence as the query. The GENSCAN prediction (top line) is missing a number of the exons that appear in the annotation for the BRCA1 gene (second line; GenBank L78833), and the GENSCAN prediction is slightly longer than the actual gene at the 5′ end. The inclusion of BLASTX hit information (vertical bars closest to the scale) in GenomeScan produces a more complete and accurate prediction (third line).

mine the location of the introns and the exons in this gene. PROCRUSTES can handle cases where there are either partial or multiple genes in the query DNA sequence.

The input to PROCRUSTES is through a Web interface and is quite simple. The user needs to supply the nucleotide sequence *and* as many protein sequences as are relevant to this region. The supplied protein sequences will be treated as being similar, though not necessarily identical, to that encoded by the DNA sequence. Typical output from PROCRUSTES (not shown here) includes an aligned map of the predicted intron-exon structure for all target proteins, probability values, a list of exons with their starting and ending nucleotide positions, translations of the gene model (which may not be the same as the sequence of the initially supplied protein), and a ‘‘spliced alignment’’ showing any differences between the predicted protein and the target protein. The nature of the results makes PROCRUSTES a valuable method for refining results obtained by other methods, particularly in the context of positional candidate efforts.

GeneID

The current version of GeneID finds exons based on measures of coding potential (Guigó et al., 1992). The original version of this program was among the fastest in that it used a rule-based system to examine the putative exons and assemble them into the ‘‘most likely gene’’ for that sequence. GeneID uses position-weight matrices to assess whether or not a given stretch of sequence represents a splice site or a start or stop codon. Once this assessment is made, models of putative exons are built. On the basis of the sets of predicted exons that GeneID develops, a final refinement round is performed, yielding the most probable gene structure based on the input sequence.

The interface to GeneID is through a simple Web front-end, in which the user pastes in the DNA sequence and specifies whether the organism is either human or *Drosophila*. The user can specify whether predictions should be made only on the forward or reverse strand, and available output options include lists of putative acceptor sites, donor sites, and start and stop codons. Users can also limit output to only first exons, internal exons, terminal exons, or single genes, for specialized analyses. It is recommended that the user simply select All Exons to assure that all relevant information is returned.

GeneParser

GeneParser (Snyder and Stormo, 1993; Snyder and Stormo, 1997) uses a slightly different approach in identifying putative introns and exons. Instead of predetermining candidate regions of interest, GeneParser computes scores on all ‘‘subintervals’’ in a submitted sequence. Once each subinterval is scored, a neural network approach is used to determine whether each subinterval contains a first exon, internal exon, final exon, or intron. The individual predictions are then analyzed for the combination that represents the most likely gene. There is no Web front-end for this program, but the program itself is freely available for use on Sun, DEC, and SGI-based systems.

HMMgene

HMMgene predicts whole genes in any given DNA sequence using a hidden Markov model (HMM) method geared toward maximizing the probability of an accurate prediction (Krogh, 1997). The use of HMMs in this method helps to assess the confidence in any one prediction, enabling HMMgene to not only report the ‘‘best’’ prediction for the input sequence but alternative predictions on the same sequence as well. One of the strengths of this method is that, by returning multiple predictions on the same region, the user may be able to gain insight onto possible alternative splicings that may occur in a region containing a single gene.

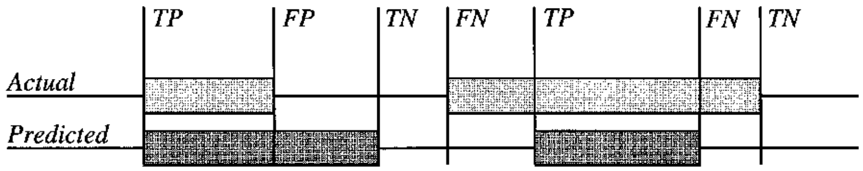
The front-end for HMMgene requires an input sequence, with the organismal options being either human or *C. elegans*. An interesting addition is that the user can include known annotations, which could be from one of the public databases or based on experimental data that the investigator is privy to. Multiple sequences in FASTA format can be submitted as a single job to the server. Examples of sequence input format and resulting output are given in the documentation file at the HMMgene Web site.

HOW WELL DO THE METHODS WORK?

As we have already seen, different methods produce different types of results—in some cases, lists of putative exons are returned but these exons are not in a genomic context; in other cases, complete gene structures are predicted but possibly at a cost of less-reliable individual exon predictions. Looking at the absolute results for the 7q31 BAC clone, anywhere between one and three genes are predicted for the region, and those one to three genes have anywhere between 27 and 34 exons. In cases of similar exons, the boundaries of the exons are not always consistent. Which method is the ‘‘winner’’ in this particular case is not important; what is important is the variance in the results.

Returning to the cautionary note that different methods will perform better or worse, depending on the system being examined, it becomes important to be able to quantify the performance of each of these algorithms. Several studies have systematically examined the rigor of these methods using a variety of test data sets (Burset and Guigó, 1996; Claverie, 1997a; Snyder and Stormo, 1997, Rogic et al., 2001). Before discussing the results of these studies, it is necessary to define some terms.

For any given prediction, there are four possible outcomes: the detection of a true positive, true negative, false positive, or false negative (Fig. 10.8). Two measures of accuracy can be calculated based on the ratios of these occurrences: a *sensitivity* value, reflecting the fraction of actual coding regions that are correctly predicted as truly being coding regions, and a *specificity* value, reflecting the overall fraction of the prediction that is correct. In the best-case scenario, the methods will try to optimize the balance between sensitivity and specificity, to be able to find all of the true exons without becoming so sensitive as to start picking up an inordinate amount of false positives. An easier-to-understand measure that combines the sensitivity and specificity values is called the *correlation coefficient*. Like all correlation coefficients, its value can range from − 1, meaning that the prediction is always wrong, through zero, to + 1, meaning that the prediction is always right.



		Predicted	
		Positives (*PP*)	Negatives (*PN*)
Actual	Positives (*AP*)	*TP*	*FN*
	Negatives (*AN*)	*FP*	*TN*

$$\textit{Sensitivity}\ (Sn) = TP\,/\,(TP+FN)$$
$$\textit{Specificity}\ (Sp) = TP\,/\,(TP+FP)$$

$$\textit{Correlation Coefficient}\ (CC) = \frac{TP \bullet TN + FP \bullet FN}{\sqrt{PP \bullet PN \bullet AP \bullet AN}}$$

Figure 10.8. Sensitivity vs. specificity. In the upper portion, the four possible outcomes of a prediction are shown: a true positive (*TP*), a true negative (*TN*), a false positive (*FP*), and a false negative (*FN*). The matrix at the bottom shows how both sensitivity and specificity are determined from these four possible outcomes, giving a tangible measure of the effectiveness of any gene prediction method. (Figure adapted from Burset and Guigó, 1996; Snyder and Stormo, 1997.)

As a result of a Cold Spring Harbor Laboratory meeting on gene prediction,[1] a Web site called the "Banbury Cross" was created. The intent behind creating such a Web site was twofold: for groups actively involved in program development to post their methods for public use and for researchers actively deriving fully characterized, finished genomic sequence to submit such data for use as "benchmark" sequences. In this way, the meeting participants created an active forum for the dissemination of the most recent findings in the field of gene identification. Using these and other published studies, Jean-Michel Claverie at CNRS in Marseille compared the sensitivity and specificity of 14 different gene identification programs (Claverie, 1997, and references therein); PROCRUSTES was not one of the 14 considered, since the method varies substantially from that employed by other gene prediction programs. In examining data from these disparate sources, either the best performance found in an independent study *or* the worst performance reported by the authors of the method themselves was used in making the comparisons. On the basis of these comparisons, the best overall individual exon finder was deemed to be MZEF and the best gene structure prediction program was deemed to be GENSCAN. (By back-calculating as best as possible from the numbers reported in the Claverie paper, these two methods gave the highest correlation coefficients within their class, with $CC_{\mathrm{MZEF}} \sim 0.79$ and $CC_{\mathrm{GENSCAN}} \sim 0.86$.)

[1] Finding Genes: Computational Analysis of DNA Sequences. Cold Spring Harbor Laboratory, March 1997.

Because these gene-finding programs are undergoing a constant evolution, adding new features and incorporating new biological information, the idea of a comparative analysis of a number of representative algorithms was recently revisited (Rogic et al., 2001). One of the encouraging outcomes of this study was that these newer methods, as a whole, did a substantially better job in accurately predicting gene structures than their predecessors did. By using an independent data set containing 195 sequences from GenBank in which intron-exon boundaries have been annotated, GENSCAN and HMMgene appeared to perform the best, both having a correlation coefficient of 0.91. (Note the improvement of CC_{GENSCAN} from the time of the Burset and Guigó study to the time of the Rogic et al. study.)

STRATEGIES AND CONSIDERATIONS

Given these statistics, it can be concluded that both MZEF and GENSCAN are particularly suited for differentiating introns from exons at different stages in the maturation of sequence data. However, this should *not* be interpreted as a blanket recommendation to *only* use these two programs in gene identification. Remember that these results represent a compilation of findings from different sources, so keep in mind that the reported results may *not* have been derived from the same data set. It has already been stated numerous times that any given program can behave better or worse depending on the input sequences. It has also been demonstrated that the actual performance of these methods can be highly sensitive to G + C content. For example, Snyder and Stormo (1997) reported that GeneParser (Snyder and Stormo, 1993) and GRAIL2 (with assembly) performed best on test sets having high G + C content (as assessed by their respective *CC* values), whereas GeneID (Guigó et al., 1992) performed best on test sets having low G + C content. Interestingly, both GENSCAN and HMMgene were seen to perform "steadily," regardless of G + C content, in the Rogic study (Rogic et al., 2001).

There are several major drawbacks that most gene identification programs share that users need to be keenly aware of. Because most of these methods are "trained" on test data, they will work best in finding genes most similar to those in the training sets (that is, they will work best on things similar to what they have seen before). Often methods have an absolute requirement to predict both a discrete beginning and an end to a gene, meaning that these methods may miscall a region that consists of either a partial gene or multiple genes. The importance given to each individual factor in deciding whether a stretch of sequence is an intron or an exon can also influence outcomes, as the weighing of each criterion may be either biased or incorrect. Finally, there is the unusual case of genes that are transcribed but not translated (so-called "noncoding RNA genes"). One such gene, NTT (noncoding transcript in T cells), shows no exons or significant open reading frames, even though RT-PCR shows that NTT is transcribed as a polyadenlyated 17-kb mRNA (Liu et al., 1997). A similar protein, IPW, is involved in imprinting, and its expression is correlated to the incidence of Prader-Willi syndrome (Wevrick et al., 1996). Because hallmark features of gene structure are presumably absent from these genes, they cannot be reliably detected by any known method to date.

It begins to become evident that no one program provides the foolproof key to computational gene identification. The correct choice will depend on the nature of the data and where in the pathway of data maturation the data lie. On the basis of the studies described above, some starting points can be recommended. In the case of incompletely assembled sequence contigs (prefinished genome survey sequence), MZEF provides the best jumping-off point, since, for sequences of this length, one would expect no more than one exon. In the case of nearly finished or finished data, where much larger contigs provide a good deal of contextual information, GENSCAN or HMMgene would be an appropriate choice. In either case, users should supplement these predictions with results from *at least* one other predictive method, as consistency among methods can be used as a qualitative measure of the robustness of the results. Furthermore, utilization of comparative search methods, such as BLAST (Altschul et al., 1997) or FASTA (Pearson et al., 1997), should be considered an absolute requirement, with users targeting both dbEST and the protein databases for homology-based clues. PROCRUSTES again should be used when some information regarding the putative gene product is known, particularly when the cloning efforts are part of a positional candidate strategy.

A good example of the combinatorial approach is illustrated in the case of the gene for cerebral cavernous malformation (CCM1) located at 7q21–7q22; here, a combination of MZEF, GENSCAN, XGRAIL, and PowerBLAST (Zhang and Madden, 1997) was used in an integrated fashion in the prediction of gene structure (Kuehl et al., 1999). Another integrated approach to this approach lies in ‘‘workbenches’’ such as Genotator, which allow users to simultaneously run a number of prediction methods and homology searches, as well as providing the ability to annotate sequence features through a graphical user interface (Harris, 1997).

A combinatorial method developed at the National Human Genome Research Institute combines most of the methods described in this chapter into a single tool. This tool, named GeneMachine, allows users to query multiple exon and gene prediction programs in an automated fashion (Makalowska et al., 1999). A suite of Perl modules are used to run MZEF, GENSCAN, GRAIL2, FGENES, and BLAST. RepeatMasker and Sputnik are used to find repeats within the query sequence. Once GeneMachine is run, a file is written that can subsequently be opened using NCBI Sequin, in essence using Sequin as a workbench and graphical viewer. Using Sequin also has the advantage of presenting the results to the user in a familiar format—basically the same format that is used in Entrez for graphical views. The main feature of GeneMachine is that the process is fully automated; the user is only required to launch GeneMachine and then open the resulting file with NCBI Sequin. GeneMachine also does not require users to install local copies of the prediction programs, enabling users to pass-off to Web interfaces instead; although this reduces some of the overhead of maintaining the program, it does result in slower performance. Annotations can then be made to these results before submission to GenBank, thereby increasing the intrinsic value of these data. A sample of the output obtained using GeneMachine is shown in Figure 10.9, and more details on GeneMachine can be found on the NHGRI Web site.

The ultimate solution to the gene identification problem lies in the advancement of the Human Genome Project and other sequencing projects. As more and more gene structures are elucidated, this biological information can in turn be used to develop better methods, yielding more accurate predictions. Although the promise of such computational methods may not be completely fulfilled before the Human Genome Project reaches completion, the information learned from this effort will play a major role in facilitating similar efforts targeting other model genomes.

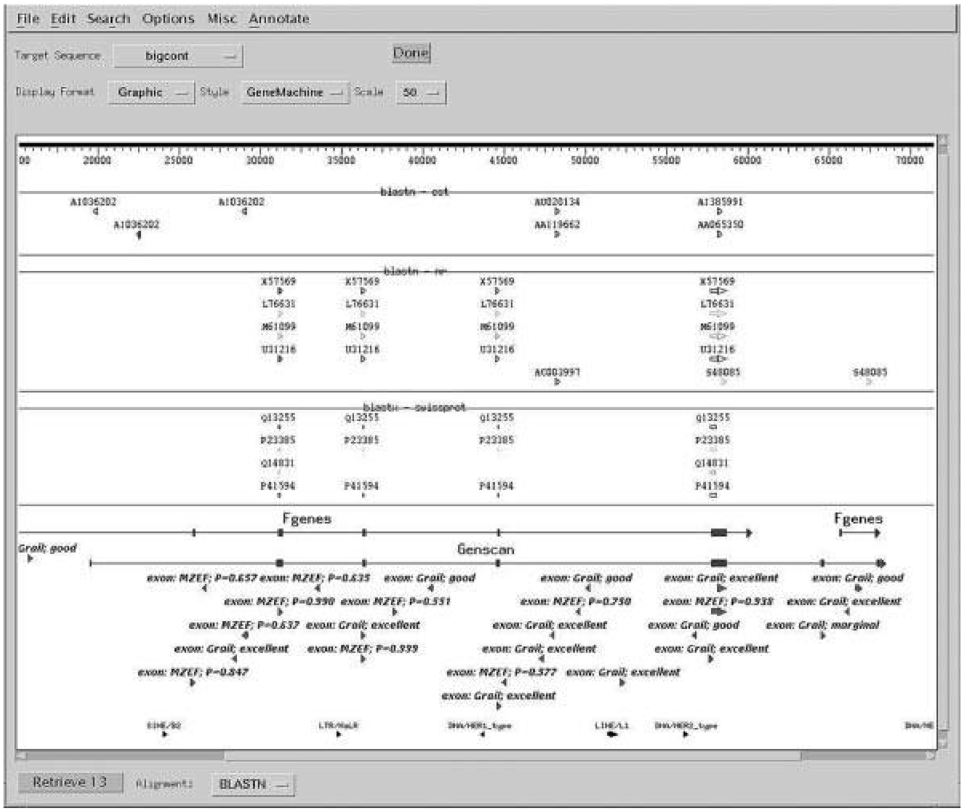


Figure 10.9. Annotated output from GeneMachine showing the results of multiple gene prediction program runs. NCBI Sequin is used at the viewer. The top of the output shows the results from various BLAST runs (BLASTN *vs*. dbEST, BLASTN vs. nr, and BLASTX vs. SWISS-PROT). Toward the bottom of the window are shown the results from the predictive methods (FGENES, GENSCAN, MZEF, and GRAIL 2). Annotations indicating the strength of the prediction are preserved and shown wherever possible within the viewer. Putative regions of high interest would be areas where hits from the BLAST runs line up with exon predictions from the gene prediction programs. (See color plate.)

INTERNET RESOURCES FOR TOPICS PRESENTED IN CHAPTER 10

Banbury Cross	*http://igs-server.cnrs-mrs.fr/igs/banbury*
FGENEH	*http://genomic.sanger.ac.uk/gf/gf.shtml*
GeneID	*http://www1.imim.es/geneid.html*
GeneMachine	*http://genome.nhgri.nih.gov/genemachine*
GeneParser	*http://beagle.colorado.edu/~eesnyder/GeneParser.htl*
GENSCAN	*http://genes.mit.edu/GENSCAN.html*
Genotator	*http://www.fruitfly.org/~nomi/genotator/*
GRAIL	*http://compbio.ornl.gov/tools/index.shtml*
GRAIL-EXP	*http://compbio.ornl.gov/grailexp/*
HMMgene	*http://www.cbs.dtu.dk/services/HMMgene/*
MZEF	*http://www.cshl.org/genefinder*

PROCRUSTES	*http://www-hto.usc.edu/software/procrustes*
RepeatMasker	*http://ftp.genome.washington.edu/RM/RepeatMasker.html*
Sputnik	*http://rast.abajian.com/sputnik/*

PROBLEM SET

An anonymous sequence from 18q requiring computational analysis is posted on the book's Web site (http://www.wiley.com/bioinformatics). To gain a better appreciation for the relative performance of the methods discussed in this chapter and how the results may vary between methods, use FGENES, GENSCAN, and HMMgene to answer each of the following questions.

1. How many exons are in the unknown sequence?
2. What are the start and stop points for each of these exons?
3. Which strand (forward or reverse) are the putative exons found on?
4. Are there any unique features present, like polyA tracts? Where are they located?
5. Can any protein translations be derived from the sequence? What is the length (in amino acids) of these translations?
6. For HMMgene only, can alternative translations be computed for this particular DNA sequence? If so, give the number of exons and the length of the coding region (CDS) for each possible alternative prediction. Note on which strand the alternative translations are found.

REFERENCES

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

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

Burge, C. B., and Karlin, S. (1998). Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.* 8, 346–354.

Burset, M., and Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics* 34, 353–367.

Chothia, C., and Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J* 5, 823–826.

Claverie, J. M. (1998). Computational methods for exon detection. *Mol. Biotechnol.* 10, 27–48.

Claverie, J. M. (1997a). Computational methods for the identification of genes in vertebrate genomic sequences. *Hum. Mol. Genet.* 6, 1735–1744.

Claverie, J. M. (1997b). Exon detection by similarity searches. *Methods Mol. Biol.* 68, 283–313.

Everett, L. A., Glaser, B., Beck, J. C., Idol, J. R., Buchs, A., Heyman, M., Adawi, F., Hazani, E., Nassir, E., Baxevanis, A. D., Sheffield, V. C., and Green, E. D. (1997). Pendred syndrome is caused by mutations in a putative sulphate transporter gene (PDS). *Nat. Genet.* 17, 411–422.

Gelfand, M. S., Mironov, A. A., and Pevzner, P. A. (1996). Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA* 93, 9061–9066.

Guigó, R. (1997). Computational gene identification. *J. Mol. Med.* 75, 389–393.

Guigó, R., Knudsen, S., Drake, N., and Smith, T. (1992). Prediction of gene structure. *J. Mol. Biol.* 226, 141–157.

Harris, N. L. (1997). Genotator: a workbench for sequence annotation. *Genome Res.* 7, 754–762.

Krogh, A. (1997). Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, Gaasterland, T., Karp, P., Karplus, K., Ouzounis, C., Sander, C., and Valencia, A., eds. (AAAI Press, Menlo Park, CA), p. 179–186.

Kuehl, P., Weisemann, J., Touchman, J., Green, E., and Boguski, M. (1999). An effective approach for analyzing “prefinished genomic sequence data. *Genome Res.* 9, 189–194.

Liu, A. Y., Torchia, B. S., Migeon, B. R., and Siliciano, R. F. (1997). The human NTT gene: identification of a novel 17-kb noncoding nuclear RNA expressed in activated CD4+ T cells. *Genomics* 39, 171–184.

Makalowska, I., Ryan, J. F., and Baxevanis, A. D. (1999) GeneMachine: A Unified Solution for Performing Content-Based, Site-Based, and Comparative Gene Prediction Methods. 12th Cold Spring Harbor Meeting on Genome Mapping, Sequencing, and Biology, Cold Spring Harbor, NY.

Mural, R. J., Einstein, J. R., Guan, X., Mann, R. C., and Uberbacher, E. C. (1992). An artificial intelligence approach to DNA sequence feature recognition. *Trends Biotech.* 10, 67–69.

Pearson, W. R., Wood, T., Zhang, Z., and Miller, W. (1997). Comparison of DNA sequences with protein sequences. *Genomics* 46, 24–36.

Rogic, S., Mackworth, A., and Ouellette, B. F. F. (2001). Evaluation of Gene-Finding Programs. In press.

Snyder, E. E., and Stormo, G. D. (1993). Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Res.* 21, 607–613.

Snyder, E. E., and Stormo, G. D. (1997). Identifying genes in genomic DNA sequences. In DNA and Protein Sequence Analysis, M. J. Bishop and C. J. Rawlings, eds. (New York: Oxford University Press), p. 209–224.

Solovyev, V. V., Salamov, A. A., and Lawrence, C. B. (1995). Identification of human gene structure using linear discriminant functions and dynamic programming. *Ismb* 3, 367–375.

Solovyev, V. V., Salamov, A. A., and Lawrence, C. B. (1994a). Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res.* 22, 5156–5163.

Solovyev, V. V., Salamov, A. A., and Lawrence, C. B. (1994b). The prediction of human exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Ismb* 2, 354–362.

Uberbacher, E. C., and Mural, R. J. (1991). Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci. USA* 88, 11261–11265.

Wevrick, R., Kerns, J. A., and Francke, U. (1996). The IPW gene is imprinted and is not expressed in the Prader-Willi syndrome. *Acta Genet. Med. Gemellol.* 45, 191–197.

Zhang, J., and Madden, T. L. (1997). PowerBLAST: a new network BLAST application for interactive or automated sequence analysis and annotation. *Genome Res.* 7, 649–656.