GENOMICS 46, 37–45 (1997)
ARTICLE NO. GE974984

A Tool for Analyzing and Annotating Genomic Sequences

Xiaoqiu Huang,[1] Mark D. Adams,* Hao Zhou, and Anthony R. Kerlavage*

Department of Computer Science, Michigan Technological University, Houghton, Michigan 49931; and
**The Institute for Genomic Research, 9712 Medical Center Drive, Rockville, Maryland 20850*

Received December 16, 1996; accepted August 15, 1997

We describe a tool for analyzing and annotating large genomic sequences containing introns. The analysis and annotation tool (AAT) includes two sets of programs, one for comparing the query sequence with a protein database and the other for comparing the query with a cDNA database. Each set contains a fast database search program and a rigorous alignment program. The database search program quickly identifies regions of the query sequence that are similar to a database sequence. Then the alignment program constructs an optimal alignment for each region and the database sequence. The alignment program also reports the coordinates of exons in the query sequence. Pairwise alignments of the query sequence with protein and cDNA database sequences are combined into multiple sequence alignments, which provide a view of all protein and cDNA sequences matching a query region. On a data set of 570 DNA sequences, AAT identified 94% of coding nucleotides correctly and 74% of exons exactly. Results of analyzing a human BAC sequence with the AAT tool are also presented. The AAT tool reduces the labor-intensive work of locating the exons of the query sequence and improves the process of defining intron–exon boundaries by using the wealth of available protein and cDNA data.

INTRODUCTION

Analysis and annotation of a newly determined DNA sequence involve identifying the protein-coding regions of the sequence. The problem of identifying a coding region with multiple exons amounts to determining the exact boundaries of each exon of the coding region. There are two complementary approaches to gene identification. One approach is based on use of sequence statistics to predict exons in the DNA sequence (see Fickett, 1996 for reviews). Burset and Guigo (1996) recently evaluated seven gene prediction programs on a data set of 570 sequences and found that the average percentage of exons exactly identified was less than 50% for most of the programs. The other approach is based on use of similarities between the DNA sequence and known protein sequences to identify exons in the DNA sequence (Gish and States, 1993; Gelfand *et al.*, 1996; Guan and Uberbacher, 1996; Huang and Zhang, 1996; Zhang, 1996; Zhang *et al.*, 1997). The Procrustes program (Gelfand *et al.*, 1996) was specially designed for recognizing genes using DNA–protein matches. All the other programs are general-purpose DNA–protein comparison programs. Alignments produced by the programs show exon–intron boundaries of the DNA sequence at various levels of resolution, respectively. The DNA–protein comparison programs differ significantly in capability and execution time. The BLASTX program (Gish and States, 1993) is perhaps the fastest program of all. The program compares a DNA sequence with a protein sequence database. It is suitable for quickly identifying protein database sequences that are similar to regions of the DNA sequence. In contrast, the NAP program (Huang and Zhang, 1996) is perhaps the slowest program of all. The program produces a high-resolution alignment between a DNA sequence and a protein sequence. It is suitable for displaying the similarity correlation between the DNA and the protein sequences and in particular for showing exon–intron boundaries of the DNA sequence at the highest level of resolution. Because of its high execution time requirement, it is not possible to use NAP on an ordinary computer to compare the DNA sequence with each protein sequence in the database. The remaining programs fall between the two programs in capability and execution time.

In this paper, we expand our work on the NAP program by developing an analysis and annotation tool (AAT) for identifying the coding regions of the DNA sequence that are similar to protein or cDNA sequences in the databases. The AAT tool makes it possible to produce accurate results at an affordable speed on an ordinary computer. The tool includes two sets of programs, one for comparing the query sequence with a protein database and the other for comparing the query with a cDNA database. Each set contains a fast database search program and a rigorous alignment program. The database search program quickly identifies

[1] To whom correspondence should be addressed at Department of Computer Science, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. Telephone: (906) 487-2123. Fax: (906) 487-2283. E-mail: huang@cs.mtu.edu.

0888-7543/97 $25.00

regions of the query sequence that are similar to a database sequence. Then the alignment program constructs a rigorous alignment for each region and the database sequence. The alignment program also reports the coordinates of exons in the query sequence. Pairwise alignments of the query sequence with protein and cDNA database sequences are combined into multiple sequence alignments, which provide a view of all protein and cDNA sequences matching a query region. On a data set of 570 DNA sequences, AAT identified 94% of coding nucleotides correctly and 74% of exons exactly. Results of analyzing a human BAC sequence with the AAT tool are also presented.

The AAT tool has several unique features. The DNA–protein alignment program in the tool allows frameshifts and introns within codons. The DNA–protein and DNA–cDNA alignment programs make use of the splice site consensus in alignment computation. The Show program allows the user to view all protein and cDNA sequences matching a query region. The AAT tool reduces the labor-intensive work of locating the exons of the query sequence and aids in defining intron–exon boundaries by using both protein and cDNA database information.

MATERIALS AND METHODS

The AAT package. A primary feature of the AAT package is the combination of fast database searching with rigorous sequence alignment, which achieves high accuracy at affordable speed. The database search program quickly identifies regions of the query sequence that are similar to a database sequence. Then the alignment program constructs a rigorous alignment for each region and the database sequence. Another feature of AAT is its modular design. Database search and sequence alignment programs are separate modules, which are connected by an adaptor program. Figure 1 shows the components of AAT and the data flow through them. We describe each program in the AAT package below.

We developed a program named DPS for computing high-scoring chains of segment pairs between a query DNA sequence and a protein database (Huang, 1996). A segment pair between a DNA sequence and a protein sequence is an ungapped alignment of two segments of the sequences, where each codon of the DNA segment is aligned with a residue of the protein segment. A chain of segment pairs is an abstract representation of an alignment, with each segment pair being an ungapped portion of the alignment. The DPS program is an improvement over the BLASTX program of Gish and States (1993) by combining close segment pairs in proper order. Computing a chain of close segment pairs enables DPS to determine the full extent of the coding region present in the query sequence. The DPS program is able to compare the 1.8-Mb sequence of the *Haemophilus influenzae* Rd genome with the Swiss-Prot database in a few hours on a typical Unix workstation.

Using a similar approach, we wrote a program named DDS for searching a query DNA sequence against a DNA or cDNA database. The DDS program is an improvement over the BLASTN program of Altschul *et al.* (1990) by combining close segment pairs in proper order. Special care was taken to address the problem of introns in the query sequence and in the database sequence. There are two gap length parameters q and l, q for the query and l for the database sequence. Any query region of length $\leq q$ between two adjacent segment pairs in a chain is given a linear gap penalty, and any query region of length $>q$ is penalized as a gap of length q. The same is used for penalizing a region of the database sequence between two adjacent segment pairs in a chain with l being the gap length parameter. If the query contains introns, then a small value is used for q. If the database contains only cDNA sequences, a large value is used for l since cDNA sequences typically do not contain introns. This scheme does not penalize heavily an intron region between two adjacent segment pairs in a chain.

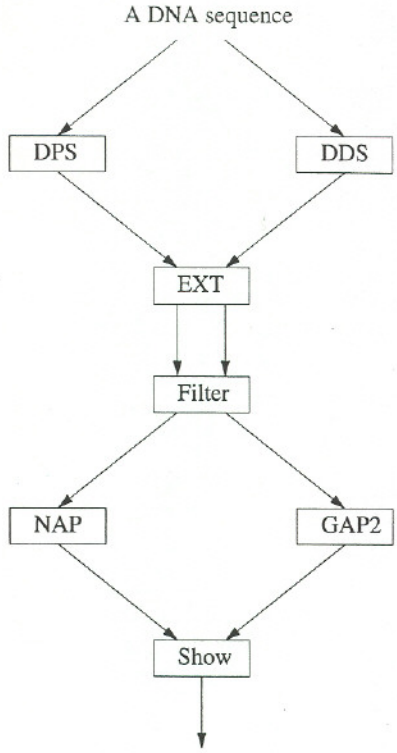

FIG. 1. The components of AAT and the data flow through them. Each box represents a program. The AAT package includes two pairs of database search and sequence alignment programs, DPS/NAP for a protein database and DDS/GAP2 for a cDNA database. The two pairs share an adaptor program named EXT. Large volumes of output produced by the database search programs are fed into a filtration program named Filter. DNA–cDNA and DNA–protein alignments are integrated by another program named Show.

A program named EXT serves as an adaptor between a database search program and a sequence alignment program. The EXT program extracts the coordinates of each database match from the output of DPS or DDS and passes them on to an alignment program. Specifically, for each chain, EXT obtains the coordinates of the query region and the database sequence region and the identification information of the database sequence. The EXT program produces a list of coordinate records, one per chain, sorted by starting coordinates of query regions. The EXT program has an option to obtain information only from chains whose database entry descriptions do not contain a word specified by the user. For example, if the word "ALU" is specified, then EXT does not pass on to the alignment program any database match involving *Alu* entries (e.g., the "*Alu* warning" sequences in Swiss-Prot). This option is useful for removing any match involving an *Alu* protein sequence entry.

It is possible that some query regions have a large number of database matches. For instance, the DNA region coding for the human β globin protein matches several hundred globin protein sequences in the Swiss-Prot database. However, for the purpose of annotation, it usually suffices to examine several of the best database hits for each query region. A program named Filter was designed to remove a large number of weaker matches. The Filter program takes the output of EXT and lets at most n highest scoring chains for each query region covered by overlapping chains pass through, where the

value for the n parameter is specified by the user. Two chains overlap if the query regions of the chains overlap.

We recently developed a program named NAP for constructing a global alignment of a DNA sequence and a protein sequence (Huang and Zhang, 1996). The alignment model of NAP accommodates introns and frameshifts within codons. The scheme for scoring an alignment has several features that allow NAP to identify the exact locations of introns. A nucleotide insertion gap of length $\leq k$ is given a linear penalty, and a nucleotide insertion gap of length $>k$ is penalized as a gap of length k, where the value for k is specified by the user. The splice site consensus information (GT and AG) is used to identify RNA splice sites. The NAP program can handle long sequences because it requires only computer memory proportional to the sum of the sequence lengths.

Several modifications were made to NAP to make NAP become an annotation tool. The NAP program was modified to report the starting and ending coordinates of each exon. Here an exon is defined to be a longest region of the query sequence aligned with a region of the protein sequence such that the alignment contains no nucleotide insertion gap of length at least k. A minimum length requirement is also imposed on the exon, say, 15 bp. The modified NAP program performs the translation of each exon, which is guided by the alignment of the exon and a protein region. The input to the NAP program includes the query sequence, the protein database, and a coordinate file produced by EXT from the output of DPS. The NAP program scans the protein database and finds the protein sequence for each coordinate record. Then for each coordinate record, NAP locates the query region, extends the region in both directions by a certain number of bases, and computes an alignment of the extended region and the protein sequence. Note that NAP corrects frameshifts in the query sequence.

For alignment of two DNA sequences, we use a program named GAP (Huang, 1994). Since a gap of length $>k$ in one sequence is given a constant penalty, the GAP program is specially suitable for aligning a genomic sequence containing introns with a cDNA sequence. Note that a gap of length $\leq k$ is given a linear penalty. The GAP program is able to handle large sequences because its computer memory requirement is proportional to the sum of the sequence lengths.

We made several improvements to GAP. The first improvement is use of the GT and AG dinucleotides to recognize splice sites. An insertion gap is one where a region of the query is aligned with gap symbols. An insertion gap of length greater than k in the query sequence is given a 5′ bonus if it starts with GT and a 3′ bonus if it ends with AG. This feature encourages a long insertion gap to occur between donor and receptor sites. The use of the GT and AG dinucleotides also leads to identification of the coding strand of a query region. Note that some partial cDNA sequences such as ESTs in the database are anticoding strands. For a region R of the query sequence and a cDNA sequence C in the database, an optimal alignment of R and C, and an optimal alignment of the reverse complements of R and C are computed. If the two alignments have different scores, then the one with a larger score is likely to be the coding strand and hence reported. The difference in score between the two alignments is due to the GT or AG bonus given to long insertions in the alignment with a larger score. The GAP program was modified to report the starting and ending coordinates of each exon. The resulting program is named GAP2. The input to the GAP2 program includes the query sequence, the cDNA database, and a coordinate file produced by EXT from the output of DDS. The GAP2 program locates and extends the query region in a way similar to NAP.

We also developed a program named Show for merging both DNA–protein alignments by NAP and DNA–cDNA alignments by GAP2 into multiple alignments. The Show program constructs a multiple alignment for each query region with database hits. The multiple alignment is consistent with all the individual pairwise alignments. The database sequence residues that are aligned to a query residue on pairwise alignments are also aligned to the same query residue on the multiple alignment. Extra gaps are included in the multiple alignment to accommodate database sequence residues that do not correspond to any query residue on pairwise alignments. The Show program allows the user to visualize all protein and cDNA database matches to a query region at the same time.

Comparison with existing programs. The idea of combining fast database searching with rigorous alignment was employed in a tool named PowerBLAST (Zhang, 1996). The PowerBLAST tool processes the BLAST output and computes gapped alignments using the SIM2 program of Chao *et al.* (1995). The SIM2 program is an improved version of the SIM program of Huang and Miller (1991), which was designed to compare two large genomic sequences. A similar paradigm of combining rapid heuristic sequence comparison with rigorous sequence alignment was used earlier in a DNA sequence assembly program named CAP (Huang, 1992). The PowerBLAST and AAT tools have different strengths. The PowerBLAST tool has several useful user interface features, while work is in progress to implement a user interface for the AAT tool. On the other hand, both DNA–protein and DNA–cDNA alignment programs in the AAT tool use the consensus information for splice sites to determine the exact locations of introns. The DNA–protein alignment program in the AAT tool can accommodate a large number of frameshifts.

The Procrustes program (Gelfand *et al.*, 1996) was specially designed to recognize genes in a DNA sequence using DNA–protein matches. The program is based on a two-stage approach. It first finds a set of regions called blocks of the DNA sequence. Then it computes an assembly of blocks with the maximum similarity score to a known protein sequence. The algorithm used in Procrustes addresses the problem of introns within codons, but not the problem of frameshifts. On the other hand, the NAP program (Huang and Zhang, 1996) uses a one-stage approach. It directly compares the DNA sequence with the protein sequence using dynamic programming. The regions of the DNA sequence that are aligned to regions of the protein sequence by NAP are taken as exons. In the one-stage approach, any region of the DNA sequence can be a potential exon, and a subset of DNA regions with the maximum similarity to the protein sequence is efficiently computed. In addition, the NAP program allows frameshifts both between and within codons. It also allows introns within codons. However, since NAP is a general DNA–protein alignment program, it does not require that an initial exon identified by NAP begin with a potential start site and that a terminal exon identified by NAP end with a potential stop site. Thus, initial and terminal exon coordinates identified by NAP are less likely to be exactly correct.

Guan and Uberbacher (1996) developed an alignment algorithm that compares a protein sequence with the 3-frame translations of a DNA sequence. The algorithm allows the alignment to shift from one frame translation to another frame translation. Since the algorithm uses the 3-frame translations of the DNA sequence, it handles only situations where frameshifts occur between codons. For the same reason, the algorithm handles only situations where introns occur between codons. Because the algorithm of Guan and Uberbacher (1996) handles fewer cases, it is faster than the algorithm of Huang and Zhang (1996).

Zhang *et al.* (1997) developed a program named FASTX for comparing a DNA sequence with a protein database. The approach used in the FASTX program is similar to that of Guan and Uberbacher (1996). Zhang *et al.* (1997) noted that the FASTX program is faster than the program of Guan and Uberbacher (1996).

RESULTS

The programs in the AAT package were written in the C programming language. The programs read a DNA sequence and a sequence database in the FASTA format. We think that the programs are easy to use. We performed two experiments with the AAT tool. In the first experiment, we evaluated the performance of the tool on a large data set of 570 DNA sequences. This data set was created to evaluate the accuracy of seven gene prediction programs (Burset and Guigo, 1996). Each DNA sequence in the data set contains a gene

with multiple exons. In the second experiment, we used the tool to annotate a human BAC sequence containing several genes with multiple exons.

In the first experiment, we tested the AAT tool on the benchmark data set of Burset and Guigo (1996). This data set contains 570 vertebrate multiexon gene sequences. The tool was used to compare each sequence in the data set with a nonredundant protein database (NR) of 73.0 million amino acids from National Center for Biotechnology Information (NCBI). To provide a general situation where protein sequences coded by DNA sequences are not in the database, we removed from the NR database any sequence that was exactly identical to a protein sequence coded by a DNA sequence in the data set. To reduce the number of undesirable matches due to interspersed repeats, each DNA sequence was screened for interspersed repeats using the RepeatMasker program (Smit and Green, unpublished results). The masked DNA sequence was used for database searching, and the unmasked DNA sequence was used for sequence alignment, which allowed the alignment program to identify the exact coordinates of exons even if parts of the exons were masked. Human involvement is required to synthesize a number of matches produced by AAT for a DNA sequence. To avoid human intervention in the test, we decided to take only a best match for each DNA sequence. Default values were used for the parameters of each program. The test took 22.5 h on a Sun enterprise 2 server.

For each DNA sequence in the data set, we repeated the following steps. The interspersed repeats of the DNA sequence were masked by RepeatMasker. The masked version of the DNA sequence was compared by DPS with the modified NR protein database. Then the output of DPS was parsed by EXT to select a chain with the maximum score, which was a best match between a region of the DNA sequence and a protein database sequence. Next an alignment of the region from the unmasked DNA sequence and the protein sequence was computed by NAP. Finally, the accuracy measures per nucleotide and per exon used by Burset and Guigo (1996) were calculated by comparing the exon coordinates produced by NAP with the true exon coordinates.

Table 1 shows the average accuracy measures of AAT on the data set. The AAT tool identified 94% of coding nucleotides correctly with a specificity of 97% and 74% of exons exactly with a specificity of 78%. For comparison, the accuracy data of the GENSCAN program (Burge and Karlin, 1997) on the data set are also included in Table 1. The GENSCAN program produced significantly more accurate results than other gene prediction programs on the Burset–Guigo data set (Burge and Karlin, 1997). At the nucleotide level, AAT performed slightly better than GENSCAN in sensitivity (Sn), specificity (Sp), approximate correlation (AC), and correlation coefficient (CC). At the exon level, GENSCAN did slightly better than AAT in sensitivity, specificity, average (Avg.), and missed exons (ME), but slightly worse in wrong exons (WE). Note that AAT is based on database similarity, while GENSCAN on sequence statistics. Thus AAT depends on existence of a similar gene sequence in the database.

If there is a similar gene sequence in the database, then an alignment produced by the AAT can help the user determine exact exon coordinates even though exon coordinates reported by AAT are not exactly correct. The alignment can also help the user determine the likelihood that a region is coding. On the other hand, if there is no match between the DNA sequence and the database, AAT cannot identify any coding region in the DNA sequence. For example, the tool did not find any database match and hence did not identify any exon for each of nine DNA sequences in the data set. Note that AAT was used only for gene identification here. Obviously, alignments produced by AAT are useful for other purposes.

In the second experiment, the AAT tool was used to analyze and annotate a human chromosome 22 BAC sequence (BAC CIT987SK_384D8, GenBank Accession No. U62317) of 139,887 bp produced at The Institute for Genomic Research (TIGR). The query DNA sequence was compared by DDS with a cDNA sequence database at TIGR of 87.9 million nucleotides. Nondefault values were used for the following parameters: 300 for the chain score cutoff, 90 for the percentage identity cutoff, and 93 for the percentage similarity cutoff. This means that each reported chain must have a percentage identity of 90% or higher and a percentage similarity of 93% or higher. Here, in addition to identities, insertions and deletions (indels) are counted as similarities because indel errors occur more frequently than mismatch errors in EST sequences. The use of the high values for the percentage identity and similarity cutoffs significantly reduced the number of *Alu* matches. The DDS program produced 44 chains in 12.4 min. All comparisons were performed on a Sun Sparcstation 4 with 64 Mb of main memory. The GAP2 program was used to construct an optimal alignment for each query region and the corresponding cDNA sequence. The GAP2 program computed 44 alignments in 4.3 min. A portion of an alignment produced by GAP2 is shown in Fig. 2.

Similarly, the query sequence was searched by DPS against TIGR's nonredundant amino acid sequence database (NRAA) of 61.8 million amino acids. The DPS program produced 6255 chains of score > 150 in 42.5 min, most of which were *Alu* matches. The EXT program was used to select those chains of score >300 such that the corresponding protein database sequence does not contain the word ALU in its database description. Ninety-three chains were selected. The Filter program was applied to the 93 chains by choosing at most 5 chains for each query region covered by overlapping chains. A total of 39 chains passed through. The NAP program constructed 39 alignments in 28.6 min. The longest alignment is 8.5 kb and contains 18 exons. Figure 3 shows a portion of an alignment produced by NAP. The Show program combined the DNA–cDNA

TABLE 1

Performance on Burset–Guigo Set of 570 Sequences

		Accuracy per nucleotide				Accuracy per exon				
Program	Sequences	Sn	Sp	AC	CC	Sn	Sp	Avg.	ME	WE
AAT	570 (9)	0.94	0.97	0.95	0.95	0.74	0.78	0.76	0.11	0.01
GENSCAN	570 (8)	0.93	0.93	0.91	0.92	0.78	0.81	0.80	0.09	0.05

Note. The accuracy measures per nucleotide and per exon were calculated for each sequence and averaged over all sequences for which they were defined using the formulas described in Burset and Guigo (1996). Results for GENSCAN are from Burge and Karlin (1997). In the Sequences column are the number of sequences out of 570 processed by each program and the number of sequences (given in parentheses) for which no database match was found (AAT) or no gene was predicted (GENSCAN). Four statistics are shown at the nucleotide level. Sensitivity (Sn) is the proportion of coding nucleotides that have been correctly identified as coding. Specificity (Sp) is the proportion of identified coding nucleotides that are actually coding. Approximate Correlation (AC) and Correlation Coefficient (CC), which range from −1.0 to 1.0, with 1.0 corresponding to a perfect identification, are overall measures of accuracy. Five statistics are shown at the exon level. Sensitivity (Sn) and Specificity (Sp) are defined similarly, where an exon is correctly identified if both coordinates are exactly correct. Average (Avg.) of Sn and Sp is an overall measure of accuracy. Missed Exons (ME) is the proportion of true exons not overlapped by any identified exon, and Wrong Exons (WE) is the proportion of identified exons not overlapped by any true exon.

and DNA–protein alignments into multiple alignments, one for each query region. A portion of an alignment produced by Show is given in Fig. 4.

We have defined seven coding regions in this BAC sequence. All seven regions have partial cDNA matches that contain long splicing gaps. A splicing gap of an alignment between the query sequence and a database sequence is a gap where nucleotides of the query DNA sequence correspond to no residues of the database sequence. The seven regions (1, NCBI ID g1399961; 2,

```
   2040     .    :    .    :    .    :    .    :    .    :    .    :
  33547 CTGCCCACCTCTGCTCTAGTTCCAGCGCTGCAGCCATTTCTTCCAGATGAAGAACATCTC
        -------------------|||||||||||||||||||||||||||||||||||||||||
    576                    TTCCAGCGCTGCAGCCATTTCTTCCAGATGAAGAACATCTC

Exon  6    33528    33458  Confidence: 100 100

   2100     .    :    .    :    .    :    .    :    .    :    .    :
  33487 CTTCTGCGGCTGCCATCCCCGCAACAGCTGGTGAGAGTCTGACCGTCCCCGAGTCCCCCA
        ||||||||||||||||||||||||||||||-----------------------------
    617 CTTCTGCGGCTGCCATCCCCGCAACAGCTG

   2160     .    :    .    :    .    :    .    :    .    :    .    :
  33427 ACCGCCCCCACAAACCCTGAGAGTCCAACCGCCCCCTGAGTCCCCACAGACCCTGAGGGC
        ------------------------------------------------------------
    647

              ~~~ 3060 bp removed ~~~

   5280     .    :    .    :    .    :    .    :    .    :    .    :
  30307 TACATCATTTGTGGGGTGGCCAAGGCTGGGCCAGGGCTCTGACGTGCCCTCCACACTGAC
        ------------------------------------------------------------
    647

   5340     .    :    .    :    .    :    .    :    .    :    .    :
  30247 TTGCCCGCAGCTATTTCGGCTTCATCACCAAACACCCCCTGCTGAGCCGCTTCGCCTGCC
        ----------|||||||||||||||||||||||||||||||||||||||||||||||||||
    647           CTATTTCGGCTTCATCACCAAACACCCCCTGCTGAGCCGCTTCGCCTGCC

Exon  7    30237    30139  Confidence: 100  96

   5400     .    :    .    :    .    :    .    :    .    :    .    :
  30187 ACGTCTTTGTCTCCCAGGAGTCCATGAGGCCGGTGGCGCAGAGTGTGGGGTGAGTGGGGC
        |||||||||||||||||||||||||||| ||||||||||||||||-----------
    697 ACGTCTTTGTCTCCCAGGAGTCCATGAGGCCrGTGGCGCAGAGTGTGGG
```

FIG. 2. Part of an alignment of a query region (NCBI ID g1399960) and a partial cDNA sequence (TIGR Human cDNA Database (HcD) Accession No. THC126740). Upper lines show the reverse strand of the query sequence. Two exons (30,735–30,627 and 30,531–30,434) predicted by GRAIL 2 are completely contained in the region (33,457–30,238) of the query in a splicing gap. The alignment was produced by the GAP2 program.

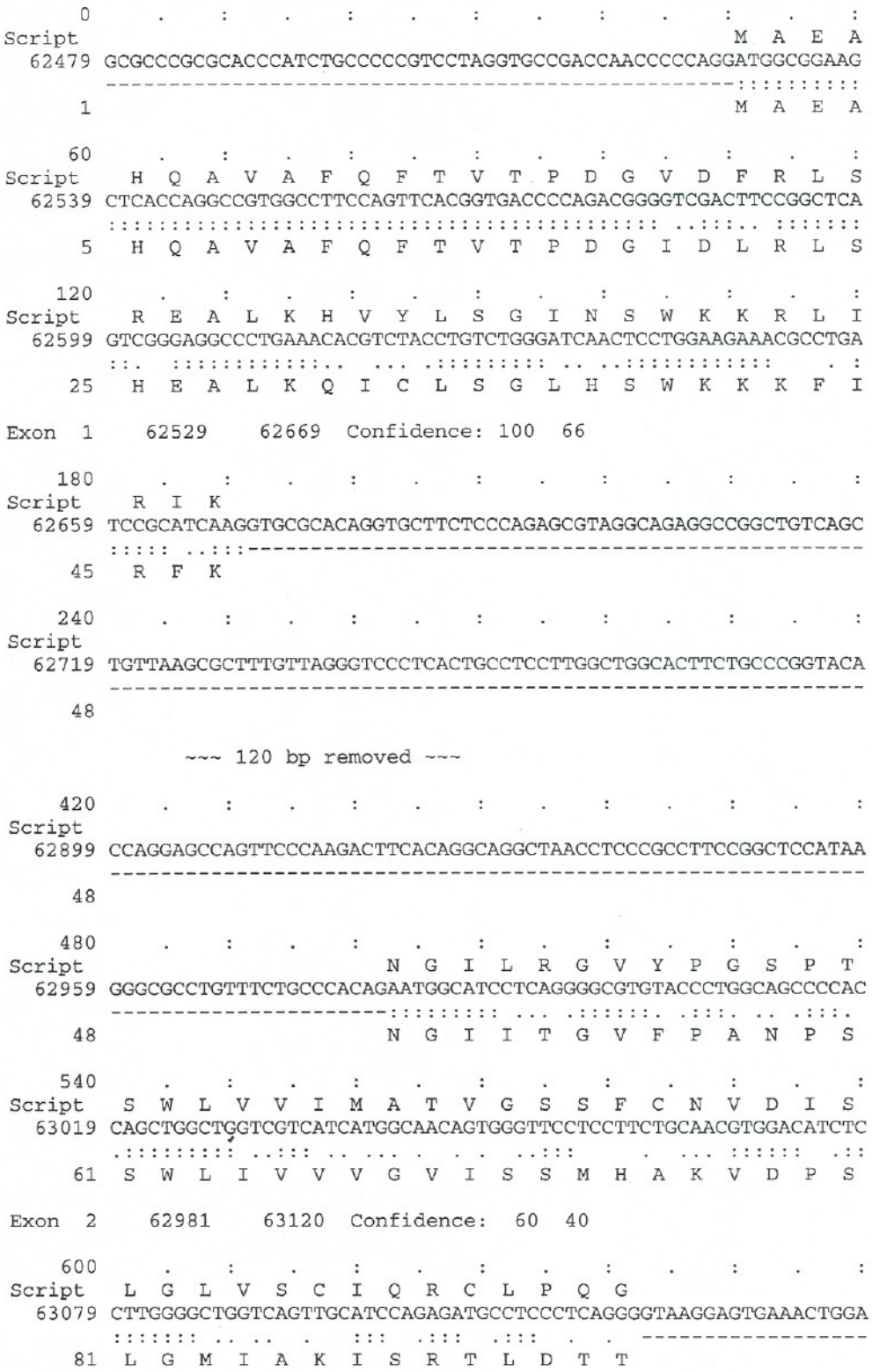

FIG. 3. Initial part of an alignment of a query region (NCBI ID g1399963) and a protein sequence (Swiss-Prot Accession No. P32198). A match is indicated by three colons and a partial match by periods. Each amino acid symbol is under base 1 of a codon. A line beginning with "Script" shows the translation of an exon. The alignment was produced by the NAP program. The starting and ending coordinates of each exon are computed by NAP. Note that exon 2 ends at base 2 of a codon.

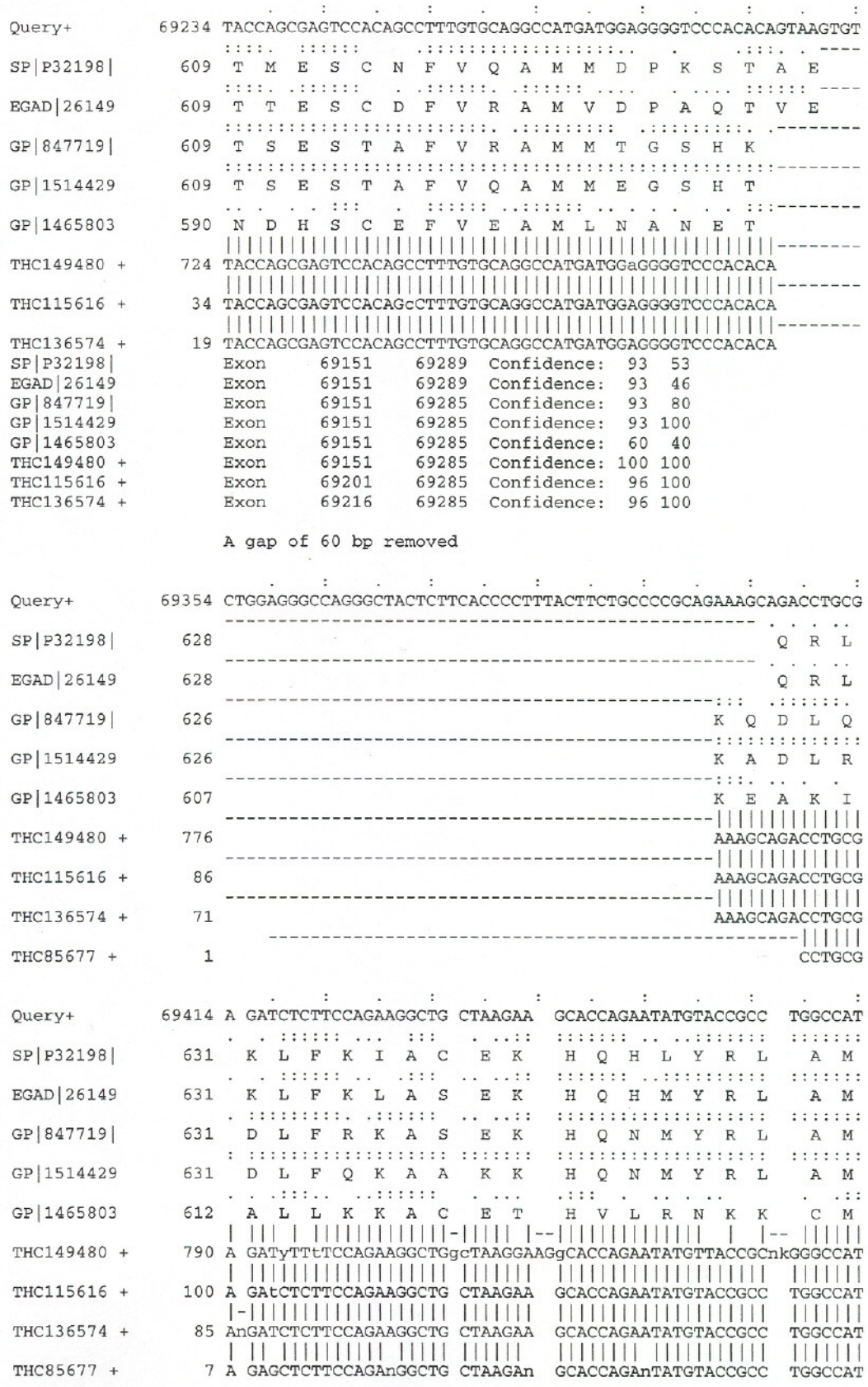

FIG. 4. Part of a multiple sequence alignment of a query region (NCBI ID g1399963) with protein and cDNA sequences. The alignment was produced by the Show program. A plus sign indicates the forward strand, and a minus sign indicates the reverse strand. A line indicating matches, mismatches, and gaps between a query region and a database sequence region is shown above the database sequence line. The alignment shows two potential pairs of splice sites, one of which was confirmed by partial cDNA sequences.

g1399960; 3, g1399962; 4, g1399963; 5, g1399964; 6, g1399965; 7, g1399966) are likely to be coding regions. There are nine additional regions that have only weak protein matches that are not strong enough for us to conclude that any are coding. Regions 1, 4, and 5 are at least 99% similar to protein sequences and code for arylsulfatase A, carnitine palmitoyltransferase, and endothelial cell growth factor, respectively. The NAP program identified the exact location of each exon for each of the three regions. Region 3 is 52 to 54% similar to a few protein sequences and two-thirds of the region has partial cDNA matches. Region 3 codes for choline kinase. Using the multiple alignment of the region with protein and cDNA sequences, we were able to locate the start site, the splice sites, and the stop site of the region.

For each of the three remaining regions with only partial cDNA matches, the GRAIL 2 program (Xu *et al.,* 1994) was used to construct a gene model of the region. The three regions code for hypothetical proteins. For region 2, 12 exons were predicted by GRAIL 2, and 14 of the 24 exon coordinates were confirmed by cDNA alignments produced by GAP2. One difference was observed between the gene model produced by GRAIL 2 and a DNA–cDNA alignment produced by GAP2. Two of the predicted exons are completely contained in a 3220-bp splicing gap of the DNA–cDNA alignment (Fig. 2). For region 6, 19 exons were predicted and 27 of the 38 exon coordinates were confirmed by DNA–cDNA alignments. A DNA–cDNA alignment shows a splicing gap of about 8 kb 5′ to the predicted 5′ exon. In other words, the alignment shows that the predicted 5′ exon is part of an internal exon and that there is another exon 8 kb 5′ to the internal exon. For region 7, 11 exons were predicted by GRAIL 2, and 8 of the 22 exon coordinates were confirmed by cDNA alignments produced by GAP2. No difference was found between the GRAIL 2 prediction and the GAP2 alignments.

DISCUSSION

Annotation of genomic sequences is a bottleneck in large-scale sequencing projects. Currently, annotation is typically performed by working manually with results of BLAST or other search programs and by working with results of gene prediction programs. Use of the AAT tool will significantly reduce the amount of manual work in annotation. The development of the AAT tool is a first step toward completely automated annotation of genomic sequences. The AAT tool is being used to analyze and annotate sequences from the human chromosome 16 and *Arabidopsis thaliana* sequencing projects at TIGR.

A few gene prediction programs include comparison to a protein database, for example, GeneParser3 (Snyder and Stormo, 1995) and Genie (Kulp *et al.,* 1997). The GeneParser3 and Genie programs use the BLASTX program (Gish and States, 1993) to compare the query sequence with a protein database. A match produced by BLASTX is used as evidence that the query region of the match is a potential exon. Test data showed that the prediction programs produced more accurate results by using BLASTX database matches (Snyder and Stormo, 1995; Kulp *et al.,* 1997). We think that the accuracy of the prediction programs will further increase through use of the AAT tool. Since BLASTX was designed to search a huge protein database, the speed requirement did not allow BLASTX to handle gaps, frameshifts, and introns in the query sequence. Thus, the boundaries of the DNA region of a BLASTX segment pair may be off by a number of bases from the boundaries of an exon. On the other hand, in the AAT tool, a rigorous DNA–protein alignment program is used to refine results of a database search program. Since the NAP program handles gaps, frameshifts, and introns in the query sequence, the boundaries of a DNA region on a NAP alignment are closer to the boundaries of an exon. The DDS/GAP2 set in the AAT tool can also be employed by the gene prediction programs to make use of similarities between the query and an EST database.

The NAP and GAP2 alignment programs currently use the GT and AG dinucleotides to identify splice sites. In calculating the score of an alignment, a splicing gap is given a 5′ bonus if it begins with GT and a 3′ bonus if it ends with AG. We will improve the accuracy of the alignment programs in identifying splice signals by using a weight matrix method (Staden, 1984). The weight matrix method calculates the strengths of 5′ and 3′ splice signals at each position of the query sequence. In aligning a query region with a protein or cDNA database sequence, a splicing gap is given a larger 5′ bonus if it begins with a stronger 5′ splice signal and a larger 3′ bonus if it ends with a stronger 3′ splice signal.

Availability. The programs are freely available for academic use on the WWW from TIGR at http://www.tigr.org/software/software.html and from MTU at http://www.cs.mtu.edu/faculty/huang.html. For commercial use, contact X.H. at huang@cs.mtu.edu. The programs can also be used through an electronic mail server. To receive information on using the AAT server, send an electronic mail message containing the word "HELP" in the body of the message to aat@cs.mtu.edu.

ACKNOWLEDGMENTS

We thank the following people for discussions: Phil Green, LaDeana Hillier, Brendan Loftus, Steve Rounsley, Granger Sutton, Jinghui Zhang, and Lixin Zhou. We also thank Christian Burks and the reviewers for suggestions that significantly improved the paper. The work benefitted from discussions at the 1996 Aspen workshop on identifying features in biological sequences. Part of the work was done while X.H. was on sabbatical leave at TIGR. This project was supported in part by NIH Grant R01HG01502-01.

REFERENCES

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215:** 403–410.

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

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

Chao, K.-M., Zhang, J., Ostell, J., and Miller, W. (1995). A local alignment tool for very long DNA sequences. *Comput. Appl. Biosci.* **11:** 147–153.

Fickett, J. W. (1996). The gene identification problem: An overview for developers. *Comput. Chem.* **20:** 103–118.

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.

Gish, W., and States, D. J. (1993). Identification of protein coding regions by database similarity search. *Nature Genet.* **3:** 266–272.

Guan, X., and Uberbacher, E. C. (1996). Alignments of DNA and protein sequences containing frameshift errors. *Comput. Appl. Biosci.* **12:** 31–40.

Huang, X. (1992). A contig assembly program based on sensitive detection of fragment overlaps. *Genomics* **14:** 18–25.

Huang, X. (1994). On global sequence alignment. *Comput. Appl. Biosci.* **10:** 227–235.

Huang, X. (1996). Fast comparison of a DNA sequence with a protein sequence database. *Microb. Comp. Genomics* **1:** 281–291.

Huang, X., and Miller, W. (1991). A time-efficient, linear-space local similarity algorithm. *Adv. Appl. Math.* **12:** 337–357.

Huang, X., and Zhang, J. (1996). Methods for comparing a DNA sequence with a protein sequence. *Comput. Appl. Biosci.* **12:** 497–506.

Kulp, D., Reese, M. G., Eeckman, F. H., and Haussler, D. (1997). Integrating database homology in a probabilistic gene structure model. "Pacific Symposium on Biocomputing," World Scientific Press, New York.

Snyder, E. E., and Stormo, G. D. (1995). Identification of protein coding regions in genomic DNA. *J. Mol. Biol.* **248:** 1–18.

Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res.* **12:** 505–519.

Xu, Y., Mural, R. J., Shah, M., and Uberbacher, E. C. (1994). Recognizing exons in genomic sequence using GRAIL II. *In* "Genetic Engineering: Principles and Methods" (J. Setlow, Ed.), pp. 241–253, Plenum, New York.

Zhang, J. (1996). PowerBLAST: A new network BLAST application for genomic sequence analysis. "Genome Mapping and Sequencing," p. 19. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

Zhang, Z., Pearson, W., and Miller, W. (1997). Aligning a DNA sequence with a protein sequence. *In* "Proceedings of the First Annual International Conference on Computational Molecular Biology," pp. 337–343, Santa Fe, NM.