Proc. Natl. Acad. Sci. USA
Vol. 89, pp. 4698–4702, May 1992
Genetics

Finding errors in DNA sequences

(reading frames/frameshifts)

JANOS POSFAI* AND RICHARD J. ROBERTS†‡

†Cold Spring Harbor Laboratory, P.O. Box 100, Cold Spring Harbor, NY 11724; and *Institute of Biophysics, Biological Research Centre of the Hungarian Academy of Science, P.O. Box 521, Szeged, H-6701 Hungary

Communicated by Barbara McClintock, August 29, 1991 (received for review August 29, 1991)

ABSTRACT **An algorithm is described that can detect certain errors within coding regions of DNA sequences. The algorithm is based on the idea that an insertion or deletion error within a coding sequence would interrupt the reading frame and cause the correct translation of a DNA sequence to require one or more frameshifts. If the coding sequence shows similarity to a known protein sequence then such errors can be detected by comparing the conceptual translations of DNA sequences in all six reading frames with every sequence in a protein sequence data base. We have incorporated these ideas into a computer program, called DETECT, that can serve as an aid to the experimentalist who is determining new DNA sequences so that obvious errors may be located and corrected. The program has been tested using raw experimental data and against sequences from the European Molecular Biology Laboratory data base, annotated as containing frameshifts. We have also tested it using unidentified open reading frames that flank known, annotated genes in the GenBank data base. Many potential errors are apparent and in some cases functions can be suggested for the "corrected" versions of these reading frames leading to the identification of new genes. As more sequences are determined the power of this method will increase substantially.**

During the determination of a DNA sequence there are many opportunities for errors. These include simple human errors, such as incorrectly recording a sequence from one medium to another or misinterpreting experimental data, as might occur at a compression in a sequencing gel. Such errors can lead to changes in protein coding sequences and obscure the interpretation of the final sequence. Often they are detected serendipitously because a particular sequence is known to encode a product and the absence of a continuous open reading frame leads to a suspicion of error. Finding the location of such errors can be time-consuming, although programs such as BLAST (1) can greatly assist in the endeavor. Nevertheless, the use of manual methods to locate potential frameshifts has several disadvantages. They are rarely systematic and often require considerable interpretation of the results before their significance can be assessed.

Already computational aids are available to avoid many of the pitfalls of error accumulation during DNA sequence determination. Semi-automated gel readers can help to ensure the correct recording of a sequence from a radioautogram to a computer (2–6), while many programs can manipulate raw sequence data and assemble it into a final sequence (7–9). More recently, new DNA sequencing machines have been developed that reduce the labor required for the accumulation of sequence data and further reduce the possibility for manual introduction of errors. However data interpretation remains a problem and the initial sequence is still likely to contain errors. In this paper we introduce a program, called DETECT, that can scan newly determined DNA sequences for the presence of errors in regions that code for proteins.

METHODS

The experimental DNA sequence is translated in all six reading frames and look-up tables are used to facilitate rapid searching. Individual entries from a protein sequence data base are then compared with the experimental sequence to identify matching segments. A match is scored when two segments contain, within a 10-amino acid stretch, at least one identical tripeptide and an overall similarity that corresponds to a preset value, typically 85%. The degree of similarity required for a match can be set by the user and is measured as an accepted point mutation (PAM) score (10). A single pass over the library sequence finds all significant local matches with the six reading frames of the experimental sequence.

To be accepted as an indicator of a potential error the two local matches must involve different reading frames of the experimental DNA sequence. The matches must lie in the same order on their respective sequences and must not show extensive overlap. Also the relative distances between the matches must be comparable. The allowable variation in distance is controlled by a user-defined parameter that has a default of ±20% of the distance observed in the data base sequence. Pairs of significant local matches, between the data base sequence and the experimental protein sequences, that satisfy the above conditions are reported as sites of possible sequence errors. Output from the program consists of a schematic diagram (e.g., Fig. 1*A*) showing the regions of similarity and also including the locations of termination codons. The likely site of the error is indicated as lying between the end of the similarity block and the closest relevant terminator. However, note that other errors may precede the closest terminator since there can be no guarantee that only a single error is present. In addition to the schematic output, a more detailed report defines each similarity block and shows the amino acid sequences at the ends of each block that contributed to the schematic (Fig. 1*B*).

The reference protein sequence data base that we have used routinely is Swiss-Prot (release 14). A user option within the program allows the use of other protein sequence data bases. The European Molecular Biology Laboratory (EMBL; release 24) and GenBank (release 56) data bases have been used as sources of DNA sequences. The program, DETECT, is written in C and has been implemented on SUN workstations. We currently use a SUN SparcStation running SUN UNIX version 4.1. The program has been copyrighted and will be made available upon request.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

Abbreviations: EMBL, European Molecular Biology Laboratory; m^5C methylase, 5-methylcytosine DNA methyltransferase.

‡To whom reprint requests should be addressed.

Table 1. Annotated errors and frameshifts

	DNA sequence	EMBL identification	Hits	Comment
1	κ Immunoglobulin (human)	HSIGK13	49	
2	κ Immunoglobulin (human)	HSIGK6	342	
3	κ Immunoglobulin (human)	HSIGKP2	28	
4	κ Immunoglobulin (human)	HSIGKVII	226	
5	T-cell receptor (human)	HSTCRAVJ	1	
6	κ Immunoglobulin (human)	HSVKII	56	
7	T-cell receptor α chain (mouse)	MMTCRA27	4	
8	T-cell receptor α chain (mouse)	MMTCRA31	1	
9	T-cell receptor α chain (mouse)	MMTCRA80	2	
10	Steroid 18-hydroxylase A (human)	HSMHCP51	4 (2)	+ one new error
11	Peptide chain release factor (*E. coli*)	ECRF2X	1 (1)	+ one genuine frameshift
12	κ Immunoglobulin (human)	HSIGKP1	36 (2)	Reduced stringency, 80%
13	T-cell receptor α chain (mouse)	MMTCRAC	0 (1)	No homolog
14	T-cell receptor α chain (mouse)	MMTCRAR4	23	Reduced stringency, 75%
15	T-cell receptor α chain (mouse)	MMTCRA37	0	Similarity at 3′ end
16	Polymerase (mouse hepatitis virus)	COMHVPOL	0 (1)	No homolog
17	Putative polyprotein (mosaic virus)	RNMRS1	0 (1)	No homolog

The examples are from the prokaryotic section of the EMBL data base. Listed under "Hits" is the number of pairs of similarity blocks predicting an error, summed over all data base matches. The number in parentheses signifies false hits due to a chance match. These false hits are readily distinguished because the similarity rarely extends beyond the minimal block required for the hit. In two cases (lines 12 and 14) a reduced stringency search was necessary to detect matching segments. *E. coli, Escherichia coli.*

RESULTS

Initial Tests to Locate Known Errors. Using the key word "frameshift," present in the Feature Table of the EMBL data base, a few examples of single base insertions and deletions were chosen to test the ability of the program to find the errors. From Table 1 it can be seen that 11 entries are identified at high stringency and 2 more are identified at reduced stringency. In human steroid 21-hydroxylase A pseudogene (line 10, Table 1) a second frameshift was found that was annotated as a miscellaneous feature. For *E. coli* peptide chain release factor (line 11, Table 1) the program reported a second potential error, which proved to be a situation of known biological interest, ribosomal frameshifting (11). Several genes are known to contain extra bases within their coding sequences that are skipped over during translation (12). This illustrates a useful feature of the program. When a genuine hit occurs and the sequence is correct, then a feature of biological interest is usually present. In one case, a mouse T-cell receptor α chain (line 15, Table 1), the error was missed because it lay at the extreme 3′ end of the sequence and lacks the second extended similarity demanded by our algorithm. Three other frameshift errors were missed because the sequences had no clear homologs in the data base, as judged by the use of FASTA (13).

Tests to Find New Errors. Initial tests of our program used fresh sequence data from some of our colleagues. The results were encouraging and several potential errors were found, but only a limited data set was available. To obtain an expanded data set, which would provide a wider range of experimental errors, we examined some of the unidentified reading frames present in GenBank, arguing that these were more likely to be error-prone than the well-characterized reading frames that had prompted the original sequence determination. We chose the prokaryotic section of that data base and selected only entries, or portions of entries, not annotated as encoding a gene product. We examined 1.3 million bases, including almost 6000 reading frames, each of length 21 amino acid residues or longer; 156 showed clear

Table 2. Potential errors in EMBL data base sequences

Region of error	Extended similarity	Test sequence	Data base identification	Sequence descriptor
1347–1438	←36 27→	BCIAMY	CDGT$BACLI	Cyclomaltodextrin glucanotransferase precursor
322–382	←11 1→	BPECYA	HLY3$ECOLI	Hemolysin C, *E. coli*
1227–1290	← 6 9→	BSTPFK	KPY1$ECOLI	Pyruvate kinase I
4129–4128	←66 66→	BTHKURHD	TRAC$BACTB	IS231C probable transposase
3621–3642	←42 9→	BTHMSQB	CRYS$BACTI	130-kDa crystal protein (δ endotoxin)
130–130	← 2 0→	ECOELTA2	CHTA$VIBCH	Cholera enterotoxin, A chain precursor
453–593	← 9 3→	ECOGLNACR	EF2$CRIGR	Elongation factor 2 (EF-2)
7007–7008	←22 94→	ECOHSDMSR	YID2$SHIDY	ISO-IS1D hypothetical 15.9-kDa protein
1898–1913	← 9 21→	ECOTAU	FUMH$BACSU	Fumarate hydratase (fumarase)
785–809	← 8 19→	LEPTRPEG	SODF$PHOLE	Superoxide dismutase (Fe)
164–170	←56 10→	PSMNOD2	NOD1$BRAJA	Nodulation protein D1
57–137	← 4 5→	RCAPETG	OMPR$ECOLI	Regulatory protein OmpR
203–220	← 0 7→	RSSFBC	PHOB$ECOLI	Phosphate regulon regulatory protein PhoB
198–194	←24 15→	STYCRPA	TCR3$ECOLI	Tetracycline-resistance protein (transposon Tn*1721*)
3548–3548	←26 68→	STYCYSJIH	YI41$ECOLI	IS421 41-kDa hypothetical protein
4374–4395	← 3 21→	TRN917	TNPA$BACTH	TNP A transposase

Examples of sequences from the prokaryotic section of the EMBL data base showing potential errors against at least three other sequences. The stringency was 85% over 10 residues. The extent of the similarity in each direction beyond the initial 10-residue match is shown in column 2. The last columns show one of the data base sequences that led to the hit.

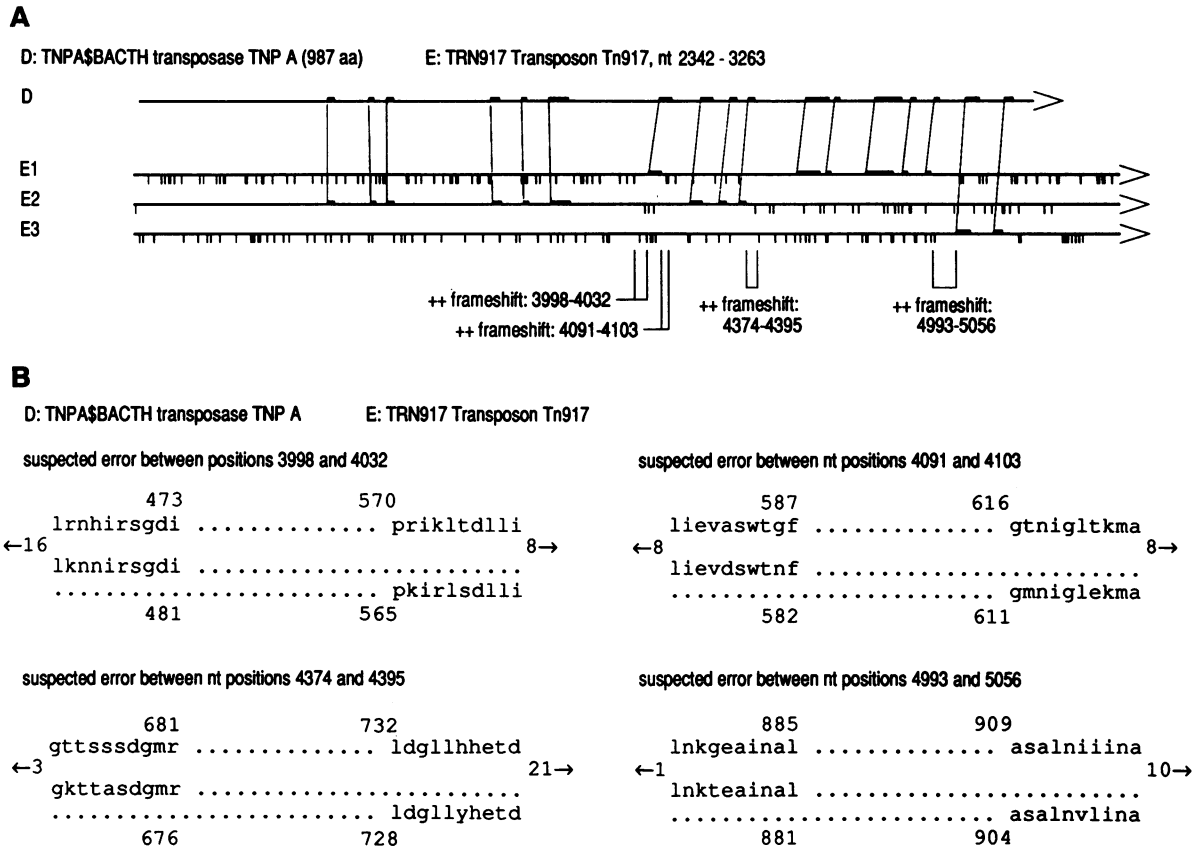

FIG. 1. Output from the program DETECT. (*A*) Schematic showing matches at the protein level between a translated segment of the Tn*917* transposon of *S. faecalis* and the transposase gene of *B. thuringiensis*. D indicates the data base sequence; E1, E2, and E3 indicate the three reading frames of the experimental sequence. Regions of strong similarity are indicated with bold boxes; termination codons are indicated with short vertical lines; ++ indicates a strong probability of a frameshift error (based either on one unusually long similarity block or on several similarity blocks flanking the frameshift). The coordinates within which the error is likely to occur are indicated. nt, Nucleotides. (*B*) Amino acid sequences at the junctions of the similarity blocks that led to error prediction. The arrows indicate the number of additional amino acids extending the similarity block.

examples of potential errors. A sample is shown in Table 2, and three are discussed in detail below.

The Tn*917* Transposase. Fig. 1 shows one particularly clear example of error detection, where a long sequence from a transposon responsible for erythromycin resistance in *Streptococcus faecalis* (14) showed substantial interrupted similarity to a transposase gene from *Bacillus thuringiensis* (15, 16). Four other transposase sequences—TRA$PSEAE, TRA3$ECOLI, TRA4$ECOLI, TRA7$ECOLI—also showed strong similarities to this sequence and provided additional support for the assignments of errors. Some of these regions of similarity had been noted previously (15), although their full extent had not been appreciated because of the frameshifts necessary to line up the complete sequences. Recently, this region has been resequenced and the regions identified by our program have been shown to contain errors. The revised sequence predicts a single open reading frame containing all of the similarity blocks identified by our program (17).

The *Bordetella pertussis* Adenylate Cyclase Gene. *B. pertussis* makes a calmodulin-sensitive adenylate cyclase that is secreted from the bacterium into the host cell where it contributes to pathogenicity. The gene for the cyclase, *cyaA*, has been sequenced, together with flanking genes, *cyaB*, *cyaD*, and *cyaE*, involved in secretion of the cyclase (18) (see Fig. 2*A*). Upstream of *cyaA* and in the opposite orientation is an open reading frame that, with one frameshift, closely matches a gene, *hlyC*, from *E. coli* (Fig. 2*B*). *hlyC* is an activator of the hemolysin in *E. coli* and its gene forms part of an operon, including *hlyA*, *hlyB*, and *hlyD*, involved in hemolysin synthesis and transport (19). These *E. coli* genes show strong similarities to the *Bordetella* genes, but no counterpart for the regulatory gene, *hlyC*, could be found in *B. pertussis* (18). Fig. 2 shows why. The *Bordetella* gene is encoded on the opposite strand from its *E. coli* counterpart and the sequence contains at least one error that separates two regions of clear similarity. This interpretation is now known to be correct, the error has been found, and the gene product has been characterized (20).

Fig. 2*B* illustrates an important aspect of the program. When the search is run at relatively high stringency (85% similarity in a stretch of 10 amino acid residues) two short regions of similarity are found. However, rerunning the two sequences at lower stringency (55% similarity in a 15 amino acid stretch) reveals more extensive similarity and helps to pinpoint the location of the error more precisely. In general, this strategy of a high stringency run against the complete data base, followed by a low-stringency run against the subset of sequences identified in the first run, can help to distinguish false positives from true hits. The true hits show an increase in the length of the matching regions, whereas false hits caused by chance similarities rarely improve at lower stringency.

New 5-Methylcytosine DNA Methyltransferase (m^5C Methylase). The large-scale testing phase used 85% similarity as the initial stringency of matching. However, occasionally lower stringencies were used to detect weaker homologies. An error located at 65% stringency in a window of 12 residues is shown in Fig. 3. A previously unidentified reading frame from the *Bacillus subtilis* bacteriophage Φ3T showed a partial

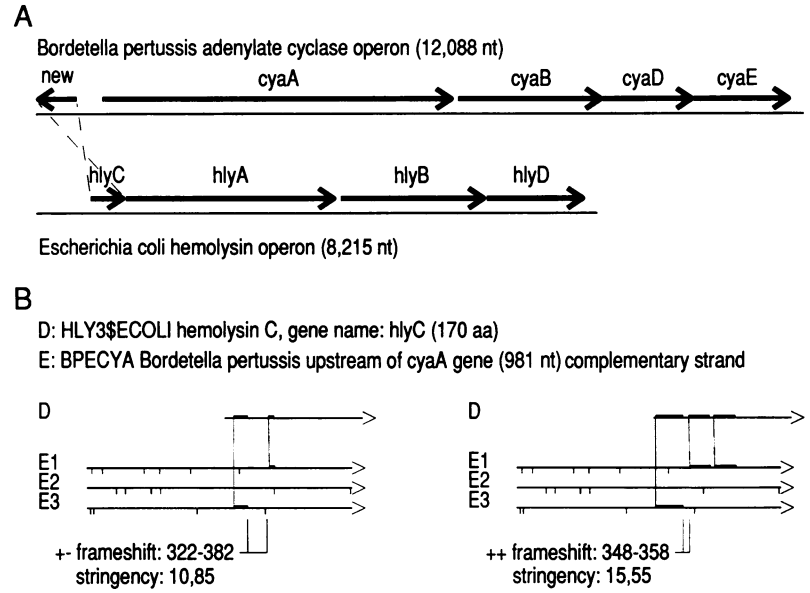

FIG. 2. Detection of a new gene regulating the *B. pertussis* calmodulin-dependent adenylate cyclase. (*A*) Organization of the two operons in *B. pertussis* and *E. coli*. "New" indicates the predicted homolog of *hlyC*. nt, Nucleotides. (*B*) Schematic output from DETECT that led to the error prediction. On the left, the search was carried out at high stringency (85% over 10 residues); on the right, the same search was carried out at low stringency (55% over 15 residues). Symbols are described in the legend to Fig. 1.

match to five m^5C methylases and a few unrelated sequences. The data base sequence was from the original paper (21) and lay upstream of the known coding region for another m^5C methylase. The matches correspond to motifs that are well-conserved in m^5C methylase sequences (22, 23). With an appropriate shift, one reading frame encoding two m^5C methylase similarities can be joined to a third similarity in the C-terminal portion of the gene. This led to the strong prediction that this unidentified reading frame encoded a new m^5C methylase. This has subsequently been confirmed. First, the reading frame change predicted by DETECT was corrected when two sequencing errors were found (24), although these errors were not annotated in the data base!

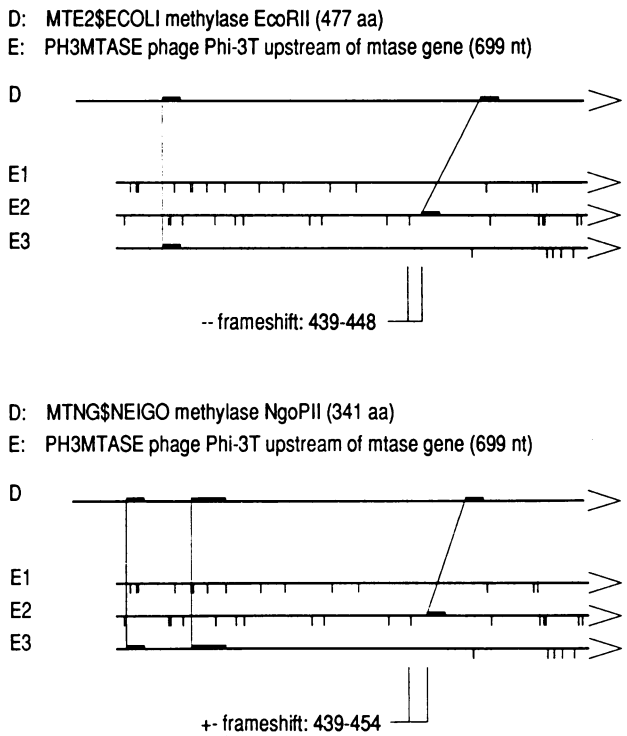

FIG. 3. Detection of a new methylase gene encoded by bacteriophage Φ3T. Shown is the interrupted similarity between the data base sequence and two known m^5C methylases, *Eco*RII and *Ngo*PII. In both cases the search was carried out at a stringency of 65% over 12 residues. nt, Nucleotides. Symbols are described in the legend to Fig. 1.

Second, the complete gene corresponding to this open reading frame has now been cloned, sequenced, and demonstrated to be a novel m^5C methylase (T. A. Trautner, personal communication). This example shows that potential errors can be spotted even when the regions of similarity are far apart. In this case 96 amino acids, which are quite variable in the m^5C methylases, separate the conserved regions in the *Ngo*PII methylase and 116 residues separate the regions in the case of the *Eco*RII methylase.

Sensitivity of the Method. We have attempted to quantitate the sensitivity of the method by collecting several sets of aligned sequences with different levels of similarity, from 15% identity to >50% identity, and analyzing the likelihood of our program to find an error at various stringency levels. Thus, within a sample of 210 aligned pairs that contain 20–25% identity, 172 contained two blocks of similarity that in principle could be used for error detection. Of these 172, the similarity blocks occurred such that on average 69% of randomly introduced errors would lie between them and so could be detected by the program. These numbers were generated using 55% stringency in matching, which we have found to be a useful level. The numbers would be reduced at higher stringency. As the level of similarity between aligned pairs of sequences increases above 25%, the ability of the program to detect an error approaches 80% or better, even using a very high stringency of matching.

It is important to note that the ability to detect errors depends completely upon the presence of a homolog in the data base and the location of the region where the error occurs. Neither of these parameters can be rigorously analyzed at the present time. For instance, in two proteins that might have low overall similarity, but which have two highly conserved motifs at their ends, a frameshift error would be detected almost every time. In contrast, for proteins in which similarities are localized in the middle of the sequence, although the overall identity may be very high, the ability of the program to detect errors at the ends would be very low.

DISCUSSION

The program described in this paper can detect errors in DNA sequences that result in reading frame changes provided that a protein of related sequence is already present in the sequence data bases. In principle, the same kind of insertion and deletion errors could be detected directly at the DNA sequence level, regardless of the presence of coding regions, provided the similarity between the experimental sequence

and the data base is sufficiently strong. However, comparison of amino acid sequences is more sensitive and increases the chance of finding a match within the data base (25). Although this approach only finds potential errors in sequences that have relatives in the data base, this limitation will diminish rapidly as more sequence is determined.

We have developed the program, DETECT, to serve as a preliminary check on a newly determined sequence to find potential regions of error. The idea is to focus attention on those portions of the original experimental data that should be checked carefully for possible interpretative errors. Note that the program will only identify regions in which errors are suspected and cannot predict how many errors might be present. If the sequence is found to be correct, then regions of the sequence highlighted by this program are likely to be biologically interesting. For instance, they may be indicative of a pseudogene, ribosomal frame-shifting, post-transcriptional editing (26), or perhaps some evolutionary property of the gene such as an earlier gene duplication followed by a frameshift and change in function. Such possibilities are not readily detected by existing programs designed to analyze DNA and protein sequences.

One might imagine additional uses for this program. For instance, each sequence in the existing DNA data bases could be analyzed against the rest of the data base in an attempt to locate potential errors. This might be quite profitable in identifying new functions for previously unidentified reading frames, as illustrated above. From the subset of the prokaryotic section of GenBank, which represented <4% of the whole data base and gave 156 potential errors, one might anticipate that several thousand potential errors could be uncovered by our program. This estimate considers only the unidentified reading frames and ignores potential errors in the well-characterized reading frames. Any estimate of errors in identified genes would necessarily be lower, since much of the checking described here will have already been carried out manually during the initial assembly of the sequence. Another use arises because our algorithm relies on the detection of short coding regions that are independent of reading frame. Thus long gaps of nonhomology can be present without affecting the performance of our algorithm. This situation arises naturally when a gene contains introns and so a simple extension, which included keeping track of in-frame similarities, could help to locate exons in genes.

Whenever our program detects a potential error in a DNA sequence it could be viewed as a special case of the broader concept of detecting an anomaly within a potential coding region. Intuitively it is unlikely that a significant stretch of protein sequence homology would occur in one reading frame and then continue in a second reading frame unless it resulted from an error superimposed on a true match. This suggests that it may be possible to use the more general statistical properties of DNA sequences to increase the range of errors that might be detected computationally. Within coding regions, one might compute codon usage in each reading frame and use the resulting statistics to assess regions of potential error. Such calculations are presently used to help identify coding regions (27–29), but they are based on the assumption that the sequence is accurate! Other statistical properties of DNA sequences, such as dinucleotide or oligonucleotide periodicities, could be exploited similarly. As we learn more about the organization and evolution of genomes it is likely that other properties could provide a basis for improved algorithms to detect errors.

Many of the experimental aspects of DNA sequence determination have benefited from computer assistance. Nevertheless, errors still creep in and their detection and elimination are often a major and time-consuming part of any sequence project. DETECT represents a first step in bringing computational tools to bear on this important problem. Although one can detect similarities between proteins, even when errors are present (30), this does not make it desirable, or even acceptable, to settle for less than an accurate sequence. As accuracy decreases, so our ability to provide a correct interpretation of that sequence diminishes. Using the polymerase chain reaction (PCR) (31), experimental verification of a newly determined sequence, which is suspected to contain an error, is easily achieved. The combined use of DETECT to find potential errors and PCR to check them could allow great improvements in sequence accuracy. One early windfall from the Human Genome Project will be the automatic determination of DNA sequences. Their interpretation is likely to become increasingly dependent on computer programs. To avoid "garbage out" we should take advantage of all possible tools to see that our sequence data bases contain accurate data.

We thank many of our colleagues at Cold Spring Harbor Laboratory for comments on the implementation of this program. Drs. D. Beach and M. Wigler provided much raw sequence data for early tests of the program. Thanks also go to Kevin Zachmann, who wrote an early version of this program. This work was supported by a grant from the National Library of Medicine (LM04971).

1. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) *J. Mol. Biol.* **215,** 403–410.
2. Gingeras, T. R., Rice, P. R. & Roberts, R. J. (1982) *Nucleic Acids Res.* **10,** 103–114.
3. Lautenberger, J. A. (1982) *Nucleic Acids Res.* **10,** 27–38.
4. Staden, R. (1984) *Nucleic Acids Res.* **12,** 499–503.
5. West, J. (1988) *Nucleic Acids Res.* **16,** 1847–1856.
6. Sjoberg, S., Carlsson, P., Enerback, S. & Bjurrel, G. (1989) *Comput. Appl. Biosci.* **5,** 41–46.
7. Gingeras, T. R., Milazzo, J. P., Sciaky, D. & Roberts, R. J. (1979) *Nucleic Acids Res.* **7,** 529–545.
8. Grymes, R. A., Travers, P. & Engelberg, A. (1986) *Nucleic Acids Res.* **14,** 87–98.
9. Staden, R. (1986) *Nucleic Acids Res.* **14,** 217–231.
10. Dayhoff, M. O., Barker, W. C. & Hunt, L. T. (1983) *Methods Enzymol.* **91,** 524–545.
11. Craigen, W. J., Cook, R. G., Tate, W. P. & Caskey, C. T. (1985) *Proc. Natl. Acad. Sci. USA* **82,** 3616–3620.
12. Atkins, J. F., Weiss, R. B. & Gesteland, R. F. (1990) *Cell* **62,** 413–423.
13. Pearson, W. R. & Lipman, D. J. (1988) *Proc. Natl. Acad. Sci. USA* **85,** 2444–2448.
14. Shaw, J. H. & Clewell, D. B. (1985) *J. Bacteriol.* **164,** 782–796.
15. Mahillon, J. & Lereclus, D. (1988) *EMBO J.* **7,** 1515–1526.
16. Mahillon, J. & Seurinck, J. (1988) *Nucleic Acids Res.* **16,** 11827–11828.
17. An, F. Y. & Clewell, D. B. (1991) *Plasmid* **25,** 121–124.
18. Glaser, P., Sakamoto, H., Bellalou, J., Ullmann, A. & Danchin, A. (1988) *EMBO J.* **7,** 3997–4004.
19. Welch, R. A. & Pellet, S. (1988) *J. Bacteriol.* **170,** 1622–1630.
20. Barry, E. M., Weiss, A. A., Ehrmann, I. E., Gray, M. C., Hewlett, E. L. & St. Mary Goodwin, M. (1991) *J. Bacteriol.* **173,** 720–726.
21. Tran-Betcke, A., Behrens, B., Noyer-Weidner, M. & Trautner, T. A. (1986) *Gene* **42,** 89–96.
22. Posfai, J., Bhagwat, A. S., Posfai, G. & Roberts, R. J. (1989) *Nucleic Acids Res.* **17,** 2421–2435.
23. Lauster, R., Trautner, T. A. & Noyer-Weidner, M. (1989) *J. Mol. Biol.* **206,** 305–312.
24. Behrens, B., Noyer-Weidner, M., Pawlek, B., Lauster, R., Balganesh, T. S. & Trautner, T. A. (1987) *EMBO J.* **6,** 1137–1142.
25. Henikoff, S. & Wallace, J. C. (1988) *Nucleic Acids Res.* **16,** 6191–6204.
26. Simpson, L. & Shaw, J. (1989) *Cell* **57,** 355–366.
27. Shepherd, J. C. W. (1981) *Proc. Natl. Acad. Sci. USA* **78,** 1596–1600.
28. Fickett, J. W. (1982) *Nucleic Acids Res.* **10,** 5303–5318.
29. Staden, R. (1984) *Nucleic Acids Res.* **12,** 505–519.
30. States, D. J. & Botstein, D. (1991) *Proc. Natl. Acad. Sci. USA* **88,** 5518–5522.
31. Saiki, R. K., Gelfand, D. H., Stoffel, S., Scharf, S. J., Higuchi, R., Horn, G. T., Mullis, K. B. & Erlich, H. A. (1988) *Science* **239,** 487–491.