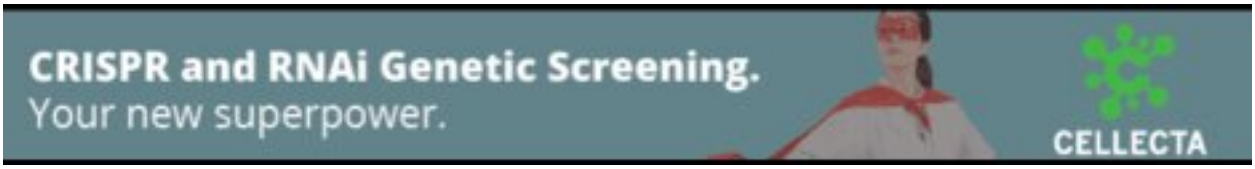
A graph theoretic approach to the analysis of DNA sequencing data.

A J Berno

Genome Res. 1996 6: 80-91
Access the most recent version at doi:10.1101/gr.6.2.80



RESEARCH

A Graph Theoretic Approach to the Analysis of DNA Sequencing Data

Anthony J. Berno[1]

Stanford DNA Sequence and Technology Center, Department of Biochemistry B403, Stanford University School of Medicine, Stanford, California 94305-5307

The analysis of data from automated DNA sequencing instruments has been a limiting factor in the development of new sequencing technology. A new base-calling algorithm that is intended to be independent of any particular sequencing technology has been developed and shown to be effective with data from the Applied Biosystems 373 sequencing system. This algorithm makes use of a nonlinear deconvolution filter to detect likely oligomer events and a graph theoretic editing strategy to find the subset of those events that is most likely to correspond to the correct sequence. Metrics evaluating the quality and accuracy of the resulting sequence are also generated and have been shown to be predictive of measured error rates. Compared to the Applied Biosystems *Analysis* software, this algorithm generates 18% fewer insertion errors, 80% more deletion errors, and 4% fewer mismatches. The tradeoff between different types of errors can be controlled through a secondary editing step that inserts or deletes base calls depending on their associated confidence values.

Large-scale DNA sequencing efforts typically sequence DNA samples using automated fluorescence-based electrophoresis instruments (Hunkapiller et al. 1991). Samples of a particular DNA fragment are prepared by creating a population of duplicate fragments that are truncated at random locations along their length. These subfragments are then tagged with one of four different fluorescent dyes, depending on their terminal nucleotide. When combined and electrophoresed through a suitable medium, each subfragment can be detected by a fixed, four-channel laser scanner as a peak in the fluorescence signal, and the identity of its terminator determined by the relative response in each channel. The order in which these fragments pass the detector then corresponds to the sequence of the original DNA sample. This general strategy has been employed widely in large-scale DNA sequencing and has been the subject of extensive research aimed at improving the accuracy and efficiency of this process.

However, the development of new DNA sequencing technology is often limited by the difficulty of analyzing the data from automated gel or capillary electrophoresis systems. A central problem in this analysis is the diversity of data that is obtained even when using a single chemistry and electrophoresis apparatus. This diversity limits the usefulness of many conventional signal processing techniques; for example, pattern recognition based on peak shapes and widths is not reliable when the typical peak shape varies widely even within individual gel runs. Furthermore, the irregularity of the data in terms of variable peak heights and spacing between peaks makes it very difficult to find metrics that can discriminate between true oligomer events and those that result from contamination or DNA secondary structure.

In addition, because the development of new sequencing technology involves a great deal of experimentation, it is desirable to be able to process the data in a way that is as general and flexible as possible. A base-calling algorithm used in this situation should be able to adapt itself to the data without requiring tedious, manual analysis of its characteristics while offering accuracy comparable to that of algorithms tuned for particular sequencing instruments. Values related to the quality of the data and the correctness of the called sequence are useful in the systematic improvement of sequencing technology and in the large-scale assembly of genomic DNA.

Although several approaches to the interpretation of DNA sequencing data have been published (Giddings et al. 1993; Tibbetts et al. 1993),

[1]E-MAIL aberno@genome.stanford.edu; FAX (408) 481-0422.

6:80–91 ©1996 by Cold Spring Harbor Laboratory Press ISSN 1054-9803/96 $5.00

each with their own strengths and limitations, most of the software in use today is proprietary and thus not available to new sequencing technology. A new base-calling algorithm, designed to be readily applicable to data from novel instruments, has been developed and demonstrated to be successful on data from the Applied Biosystems (ABI) 373 sequencing system. This algorithm is unique in that it requires minimal configuration for different sequencing chemistries and electrophoresis conditions, makes no assumptions about the data that are unique to any particular sequencing system, and requires no human intervention. This flexibility is made possible by a graph-theoretic editing step that relies on relatively invariant characteristics of the data to optimize the set of called bases. The software generates a number of metrics associated with the called sequence that can be used for quantitative evaluation of data quality and to aid in the assembly process.

This algorithm has been incorporated into *Sax*, a Macintosh application that performs lane tracking and base calling on data from ABI 373 sequencers. *Sax* allows for visualization of individual files and the automated processing of large batches of data, and can be extended for new types of analysis tasks. It is currently available via the World-Wide Web at http://genome-www.stanford.edu/sax/.

Overview of the Algorithm

The input to the algorithm is the one-dimensional data trace extracted from a two-dimensional gel image. This consists of four channels of fluorescence data over several thousand sample points. The process of converting these data into usable sequence can be broken down into the following series of steps: (1) Lowpass filtering; (2) channel separation; (3) dye mobility correction; (4) baseline removal; (5) deconvolution and event detection; (6) spacing estimation and event editing; (7) event confidence assessment; (8) identity confidence assessment; and (9) estimation of regional data quality.

To improve computational efficiency, most filtering operations are performed in the Fourier domain, with the resulting restriction that the data array be a power of 2 in length. All computations are performed using single precision (4 byte) floating point values, which offer sufficient accuracy while using reatively little memory.

Low-pass Filtering

The first step is to apply a low-pass filter to the signal to remove noise. A Gaussian filter of half-height width equal to half the estimated peak spacing is used. This step is performed primarily to improve data visualization, as excessive noise can obscure peak shapes. It does not have a significant effect on the results, as the later application of a deconvolution filter also incorporates a noise reduction component.

Channel Separation

To eliminate the cross talk between the four channels, it is necessary to apply a linear transformation to each sample point. It is convenient to represent this transformation as the 5×4 matrix $\mathbf{M}$, with the fifth row being used to subtract the constant background component from the signal. The n raw data points may then be represented as the $n \times 5$ matrix $\mathbf{R}$, where each row contains the 4 data values at each sample point and the fifth column is filled with the value 1. The equation for this transformation is then $\mathbf{S} = \mathbf{RM}$, where $\mathbf{S}$ is the $n \times 4$ matrix containing transformed data values.

In most base-calling algorithms, the matrix $\mathbf{M}$ is computed in advance for each sequencing instrument. However, this matrix can be computed automatically from the data, given data of sufficiently high quality. If we assume that once the appropriate transformation is applied, oligomer events from different channels will not overlap, one can compute the matrix that will best satisfy this assumption. Whereas this assumption is not strictly true, the overlaps that do occur between adjacent events are small compared to typical peak heights and tend to cancel out so as to not affect the results significantly.

Computing $\mathbf{M}$ is an iterative process. $\mathbf{M}$ is initialized to the identity matrix, and the following steps are performed to successively improve it:

1. Compute $\mathbf{S} = \mathbf{RM}$.
2. Subtract the background from $\mathbf{S}$ and normalize each channel so that the median peak height is 1.
3. Estimate a target $\mathbf{S}'$ for the desired transform by setting all values at each point in $\mathbf{S}$, other than the largest value, to zero.
4. Use least-squares approximation to find the matrix that best transforms $\mathbf{R}$ into $\mathbf{S}'$, that is, the solution to $\mathbf{R}^T\mathbf{S}' = \mathbf{R}^T\mathbf{RM}$.

This is accomplished using the gaussj routine from Press et al. (1992).

This procedure is repeated until the values in the matrix converge; three iterations are generally sufficient. This operation works best when it is applied only to the highest quality data. The inclusion of a large primer peak in the data used to compute the matrix can bias the results, whereas very low resolution data does not satisfy the assumption of nonoverlapping events. Once an appropriate matrix is computed on a high quality data trace, it may be applied to lower quality traces. Although it is certainly possible to compute a new matrix for each data set, this results in a lower average accuracy.

Mobility Correction

The next step is to correct for differences in the mobility of different oligomers attributable to the varying molecular weight of their fluorescent tags. (This step is not necessary when using dye terminator chemistry.) For most of the data, a simple translation of each channel is sufficient to cause the oligomer peaks to be spaced evenly. However, the peaks near the primer peak are not aligned correctly by this technique; whereas mathematical models of the electrophoresis process suggest a fairly complicated strategy for this correction, it is sufficient in practice simply to apply a translation that decreases exponentially from some initial value to a constant value as one moves past the primer peak. The equation for this translation is: $\Delta t = c + \Delta c e^{-(t-t_0/k)}$, where Δt is the required translation at each sample point t, c is the mobility correction component required far from the primer peak, Δc is the adjustment to c required near the primer peak, t_0 is the location of the primer peak, and k is the decay constant. The values of c and Δc differ for each channel, whereas t_0 and k are constant. Although no fully automatic means of finding these coefficients has been determined, and it is necessary for the user to supply them, their values can be found to sufficient accuracy through estimation and manual adjustment. Base-calling accuracy is not affected by variations in their values that are smaller than the natural (10%–20%) variation in peak spacing, and a single set of values works well with a particular dye set under a variety of electrophoresis conditions. Nevertheless, a more rigorous and automated strategy for performing this correction would be a desirable improvement.

Baseline Removal

After the mobility correction, the slowly varying component of the baseline is subtracted from each channel. This is accomplished by first constructing a piecewise baseline determined by evaluating the second percentile of data values in overlapping segments of the data. The length of these segments is 20 times the estimated peak spacing, and they are spaced at half their length. The piecewise baseline is then smoothed by a Gaussian filter whose characteristic width is the same as the segment length and subtracted from the data.

Deconvolution and Event Detection

Because the electrophoretic process often fails to separate peaks adequately, some form of deconvolution filter must be applied to the data to resolve overlapping events. This process is complicated by the variability of peak shapes, meaning that conventional deconvolution often fails.

Neural network processing (Tibbetts et al. 1993) has been shown to be highly effective at this step but requires tedious retraining of the neural network whenever novel data are encountered. Experiments with neural network processing using code borrowed from Masters (1993) indicated strong correlations between the neural network output and the higher derivatives of the data; in effect, the network was acting as a differentiating filter.

On the basis of this result, a nonlinear filter that operates on the second and fourth derivatives of the data was developed and has been found to be useful for separating poorly resolved peaks while being robust enough to handle data with widely varying peak widths and separations. The differentiating components of the filter are implemented in the Fourier domain, with a high-cutoff component to dampen any resulting noise. It corresponds to the time domain equation

$$x' = \ln\left[1 + x\sqrt{\frac{d^2x}{dt^2} \times \frac{d^4x}{dt^4}}\right]$$

When one or both derivatives are positive, the filter output is set to zero so as to avoid negative outputs and to exclude regions of upward curvature. The logarithm is then applied to reduce the large variance in data values that results from the differentiation components. The motivation for

taking the product of the second and fourth derivatives arises from the observation that both derivatives are negative only near the center of typical oligomer peaks, and that the fourth derivative's negative magnitude near the peak center is generally greater than its positive magnitudes elsewhere. This distinctive fourth-derivative "signature" is what allows the filter to separate strongly overlapping peaks.

The output of this filter is normalized to the local average intensity by filtering the output signal with a Gaussian filter whose width is equal to 10 times the estimated base spacing and then dividing the output signal by the locally averaged version. The resulting data shows clearly the number and location of peaks, even when the original trace is difficult to interpret (Fig. 1). It is then possible to find the locations of likely oligomer events by simply finding maxima in this signal that are above a small (0.1) threshold. The intensity of these events is taken to be the value of the deconvoluted signal at these maxima.

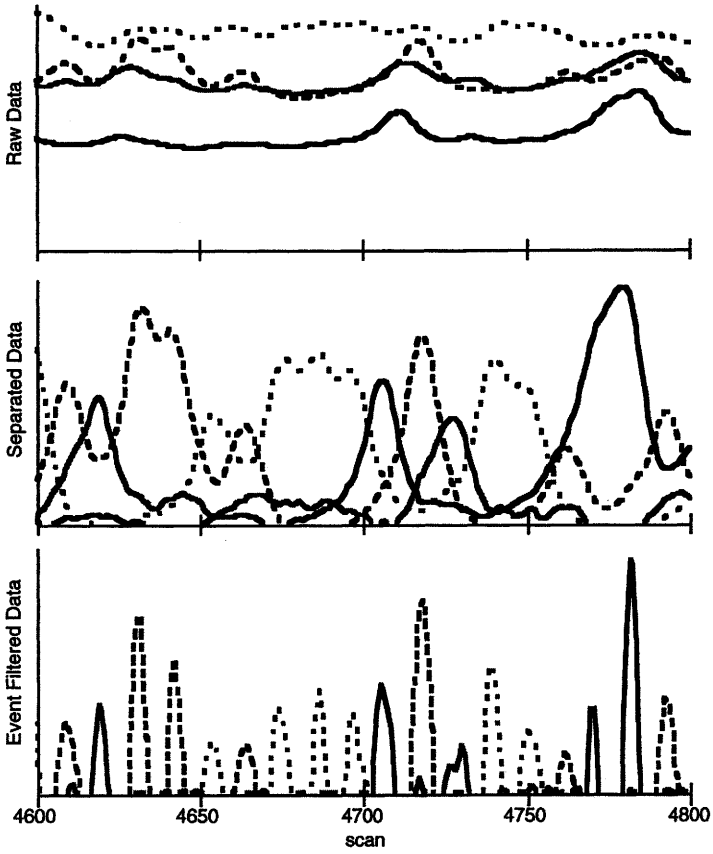


Figure 1 Steps in data processing. An example of raw data produced by an ABI 373 sequencing system, compared with the same data after normalization, separation, and processing by a nonlinear deconvolution filter.

Spacing Estimation and Event Editing

These events correspond loosely to the actual sequence, but some bases will be missed, and there will be a great number of events that do not correspond to correct base calls. To achieve the high accuracy required in most sequencing applications, there must be some way to accurately discriminate between oligomer events and those that are produced as a result of contamination, noise, low resolution, or the formation of DNA secondary structure. Unfortunately, there does not appear to be any single metric or combination of metrics that can discriminate accurately between real and spurious events when they are considered individually. In particular, the spacing between an event and its nearest neighbors is not a suitable criterion, because its neighbors may not represent oligomer events themselves.

It is necessary to employ an approach that considers the entire set of candidate events simultaneously and selects the subset that optimizes a particular scoring function. In choosing a scoring function, it is important to find one that can be maximized using an algorithm that is both reasonably efficient and provably optimal. An exhaustive search through all possible subsets is computationally intractable, whereas greedy or simulated annealing algorithms tend to become trapped in locally optimal solutions. One suitable class of functions considers pairs of adjacent base calls together with their intensities. Several functions were tried, and the one that yielded the best results was

$$s = (I_1 + I_2) - a\left[\frac{d - \bar{d}}{d}\right]^4$$

where s is the score for one pair of candidate base calls, d is their spacing, $\bar{d}$ is the estimated average spacing, I_1 and I_2 are the intensities of the base calls normalized to the local average event intensity, and a is an arbitrary parameter that controls the tradeoff between peak intensity and conformity to the expected spacing. This function allows a small amount of variability in peak spacing while strongly penalizing pairs of

peaks whose spacing is significantly greater or lesser than the estimated average.

Because the function considers pairs of events, the optimal set of called bases can be represented as a path through a directed acyclic graph (Fig. 2), in which events are represented by nodes, and the arcs drawn between nodes represent pairs of adjacent base calls. The weight of each arc is simply the value of the scoring function, and the called sequence is the set of nodes that is on the maximum-weight path through the graph. There is a simple, efficient algorithm, a special case of Dijkstra's algorithm (Cormen et al. 1992), that can find this path correctly and, hence, the optimal sequence. Because the graph is already topologically sorted, and the number of plausible arcs is linearly related to the number of nodes, the complexity of this step is linear in the number of candidate base calls, and its execution time is insignificant.

Editing is first performed using the initial spacing estimate provided by the user, which need only be within about 30% of the actual spacing. This estimate is refined after one application of the editing algorithm, and if the new estimate differs significantly from the original one, the editing process is repeated using the new one. At present, a single spacing value is used, representing the average spacing over the entire run. Although it is certainly possible to use a more sophisticated model to account for the changes in spacing over the length of the run, experiments using a quadratic model resulted in lower overall performance attributable to an increased tendency for the model coefficients to diverge.

The use of information related to base identity in event editing, discussed by Golden et al. (1993) is implicit in this algorithm. As suggested in this work, most variation in oligomer event separations is attributable to the effect of different dyes on the mobility of each oligomer. It has been found that once the mobility correction step described above is applied to the data, no simple residual correlations between base identities and peak separations can be found. Whereas the formation of sequence-dependent secondary structure in the oligomers certainly has an influence on their mobility, any function that could account for such structures would have to consider more than just pairs of base calls and would not be compatible with this editing strategy.

Graph-theoretic editing can be also be compared to Giddings' algorithm (Giddings et al. 1993) in which an object-oriented architecture allows an arbitrary collection of informative metrics to be combined to generate a score for each base call. The set of called bases is improved iteratively, inserting and deleting calls on the basis of their individual scores, until the resulting sequence converges. Whereas this method allows the consideration of more parameters, such as the peak spacing over a larger neighborhood, the resulting sequence may not be optimal. In choosing a base-calling algorithm, the benefits of guaranteed optimality must be weighed against those arising from the use of additional parameters, and the correct choice would likely depend on the particular data being considered.

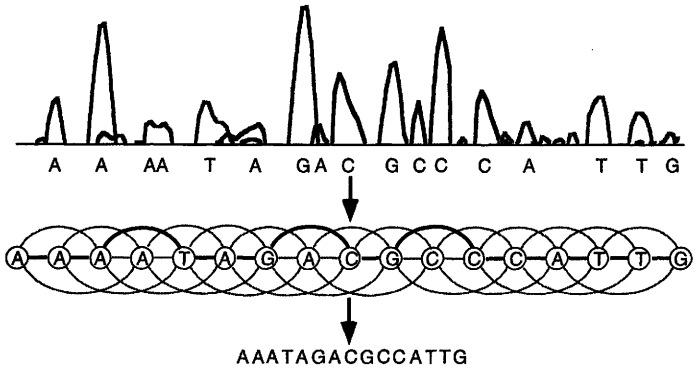


Figure 2 Event editing. Maxima in the processed data that lie above a small threshold value represent candidate base calls. A subset of these events, represented by a maximal weight path through a directed acyclic graph, constitutes an optimal set of base calls with respect to a scoring function that selects for peak intensity and conformance to predicted spacing among pairs of candidate events.

Event Confidence Estimation

A value related to the probability that any particular base call is present in the sequence can be obtained through the statistical analysis of the base calls and errors therein. These values can then be used by automated assembly software to assist the reconstruction of large sequences.

Two types of event confidence metrics can be computed: one associated with the probability that a base call is not part of the correct sequence (an insertion error), and another related to that probability that a base call was missed (a deletion error). Because this base-calling strategy is somewhat con-

servative, insertion errors were much less frequent than deletions. In each case, the most informative metric was found to be the spacing between a base call and its neighbors, normalized to the local average spacing as measured over a 10-base window. In the case of deletion errors, bases that were not called were postulated to exist in the gaps between base calls and the metric was computed as if they were present in the called sequence.

Surprisingly, event intensity alone did not correlate strongly with the probability of either insertion or deletion errors. Although a more sophisticated multivariate analysis may well show event intensity to be significant, it was not used in this algorithm. Efforts at using neural networks to evaluate base-call confidences on the basis of multiple parameters have shown them to be useful as well (C. Tibbetts, pers. comm.); however, it was felt that the additional complexity that they introduce outweighed their potential advantages over a more straightforward statistical analysis.

Once event confidence metrics are computed, they can be used to insert or delete base calls according to their probability of correctness. Because the desired balance of insertion versus deletion errors and the interpretation of the confidence metric is dependent on the particular sequencing application, this algorithm does not specify how the second-pass editing process is to be carried out.

Identity Confidence Assessment

Assessment of confidence in base identity is an issue independent of event confidence. Frequently, an event that is present with high confidence consists of two or more superimposed peaks of nearly equal intensity in separate channels, making its identity ambiguous. It has been found experimentally that the probability of a base being identified correlates well with the ratio of the intensity of a called peak to the combined intensity of all peaks between the previous and next called bases. A ratio of <0.4 for any base call is taken to mean that the overall confidence in its correctness is too low to assign a particular identity to the base, so it is called as an "N."

Estimation of Regional Data Quality

The final step is to determine the quality of the sequence on a regional basis. This is needed to estimate read length and for determining regions of sequence that would be useful for constructing primers in directed sequencing. By examining regions of sequence rather than individual base calls, it becomes possible to place a likely upper bound on the number of errors in that region.

One useful metric is simply the average identity confidence, which is a good predictor of mismatch errors. However, neither this metric nor the average event confidence correlates particularly well with frameshift errors. It is possible for a sequence to have a high confidence on each individual event yet still contain insertion or deletion errors. A more informative metric is the standard deviation of base spacing normalized to the average spacing. In regions of otherwise high data quality, high values are indicative of G-C compressions, which in turn tend to generate base-calling errors. In areas of low resolution, the deconvolution filter cannot accurately specify the locations of events; this uncertainty is reflected in erratic base spacing and more frequent frameshift errors.

The usable portion of the sequence can be taken to be the smallest region that contains all of the sequence with a spacing deviation of <0.1 and an average identity confidence of >0.9, as measured over 20-base segments. Data past the boundaries of this region tend to degrade quickly so that little high-quality sequence is missed, although it is possible for poor-quality data to be present in the midst of a high-quality region.

Performance Evaluation

This algorithm was implemented in C++ on a Macintosh computer using the Metrowerks CodeWarrior development environment and tested with data obtained from regular production sequencing at the Stanford DNA Sequence and Technology Center. Test data were selected from a large set of files representing the 6-kb cosmid cloning vector pHC79 (GenBank accession no. L08873). The frequent sequencing of this vector is a by-product of the Stanford sequencing operation, so by assembling a large number of the resulting sequence fragments, its entire sequence could be found to high accuracy. Furthermore, because these data had been collected over a period of several months, they were representative of what could be expected from a typical large-scale sequencing effort, rather

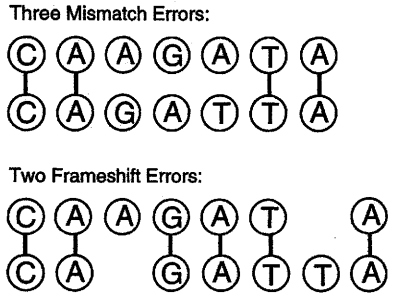


Figure 3 Ambiguities in error classification. This illustration shows two possible alignments between the sequences CAAGATA and CAGATTA, which can be interpreted as either three mismatch errors or two frameshift errors. The most meaningful choice between these two interpretations is not well defined and cannot be determined easily in software.

than the higher quality data that might have been obtained from a few, carefully controlled gel runs.

Sequencing was performed using the ABI 373 system, with 34-cm gels containing 32 lanes each. Samples were prepared using *Taq* cycle sequencing with dye primer chemistry and the M-13 universal primer. Typical gels contained 300–400 readable bases in each lane before the peak resolution or fluorescence intensity became insufficient.

Two hundred fifty trace files, representing over 100,000 base calls, were randomly selected from the set of cosmid cloning vector data. Each file had been extracted previously from the gel image and base-called using the ABI *Analysis* software, and had been analyzed by FASTA to establish that it represented sequence from the cosmid vector. These files were then reanalyzed using this algorithm.

An initial spacing estimate of 10 sample points was used, with the parameter a in the event editing step set to 30, the mobility correction decay constant k set to 350, and the following channel-specific mobility correction coefficients:

Channel	ΔC	C
C	0	5
A	0	6
G	14	0
T	14	3

Surprisingly, the value of a had little effect on the accuracy of the results when using dye primer data; any value between 20 and 50 worked equally well for the test data that were used to optimize it. Dye terminator data, however, generally give the best results when a is near the high end of this range.

Errors were located using a modification of the Needleman-Wunsch algorithm (Needleman et al. 1970) to align the called sequence to the known sequence. Because the conventional Needleman-Wunsch algorithm can result in nonunique optimal alignments, it was necessary to modify its scoring mechanism so as to take into account some estimate of the relative probability of the correctness of each base call. Correct matches were rewarded with small additional scores that reflected the intensity of the corresponding oligomer event, whereas the scores for gaps in the sequence were adjusted in proportion to the spacing of these events. The scores for matches, mismatches, and gaps

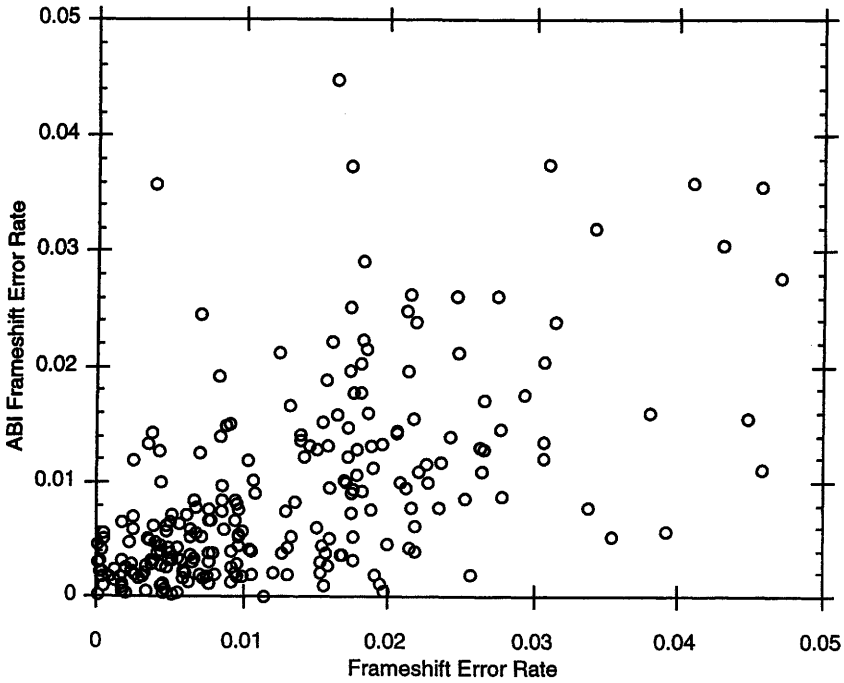


Figure 4 The illustration shows the distribution of frameshift error rates for both ABI's *Analysis* software and this algorithm for 254 samples as measured over the first 300 bases.

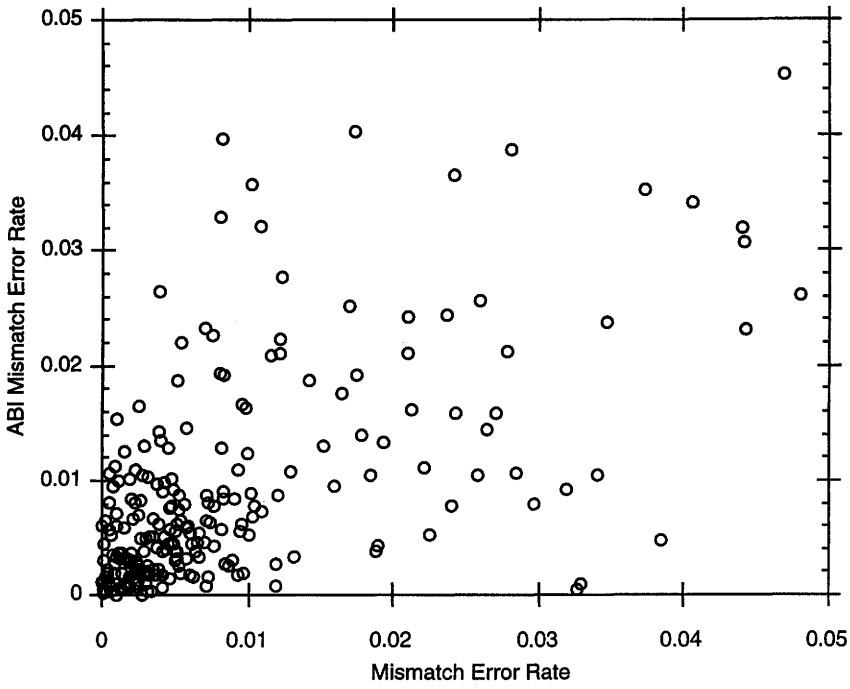


Figure 5 The illustration shows the distribution of mismatch error rates or both ABI's *Analysis* software and this algorithm for 254 samples as measured over the first 300 bases.

in the alignment were determined as follows:

Match score:	3 + 0.001 *i*
Mismatch score:	–1 + 0.001 *i*
Gap opening score:	–3 + 0.001 *d*
Gap extension score:	–2

where *i* is the value of the event filter output at the location of the base call and *d* is the width of a gap as measured in sample points. In cases where a gap is encountered in the reference sequence rather than the called sequence, this adjustment component is not used. The specific value for the coefficients for *i* and *d* are unimportant, as long as they are small compared to the fixed component of the scores. The inclusion of these adjustments ensures that in cases that would otherwise be ambiguous, base calls associated with larger peaks are those that are matched to the reference sequence, whereas missed bases are associated with areas between the most widely spaced calls.

It should be noted that although the modified Needleman–Wunsch algorithm results in unique alignments, there is still some ambiguity inherent in determining the types of base-calling errors. An example of such an ambiguity is illustrated in the two possible alignments between the sequences CAAGATA and CAGATTA, shown in Figure 3.

As can be seen in the illustration, these two sequences can be said to differ either by two frameshift errors or three mismatches. Such alignments are characteristic of GC compressions, which typically involve a number of closely spaced peaks followed by a region of unusually wide spacing. Although any alignment algorithm can be adjusted to favor one interpretation over the other, it is not clear which one offers a more useful measure of base-calling performance. Using the scoring scheme described above, the alignment algorithm favors mismatches over frameshifts, with the possible consequence that some frameshift errors might be missed.

Error rates were calculated for each file as analyzed by this algorithm and Applied Biosystems' *Analysis* software. Care was taken to ensure that exactly the same region of sequence was evaluated in both cases and that sequence from the M13 cloning vector used in the template preparation was skipped. Eighteen samples were excluded from this analysis because of extremely poor data quality (>10% overall error rate in the first 300 bases with both base-calling methods) or because they were chimeric.

RESULTS

The following table summarizes the average error rates over the first 300 bases past the end of the cloning vector for each file in the test data set:

	This algorithm (%)	ABI software (%)
Insertion errors	0.23	0.28
Deletion errors	0.89	0.49
Mismatch errors	0.70	0.73
Ambiguities	2.0	1.1

Figures 4 and 5 plot the relative frameshift and mismatch error rates for each file over the same region. It is interesting to note that the rel-

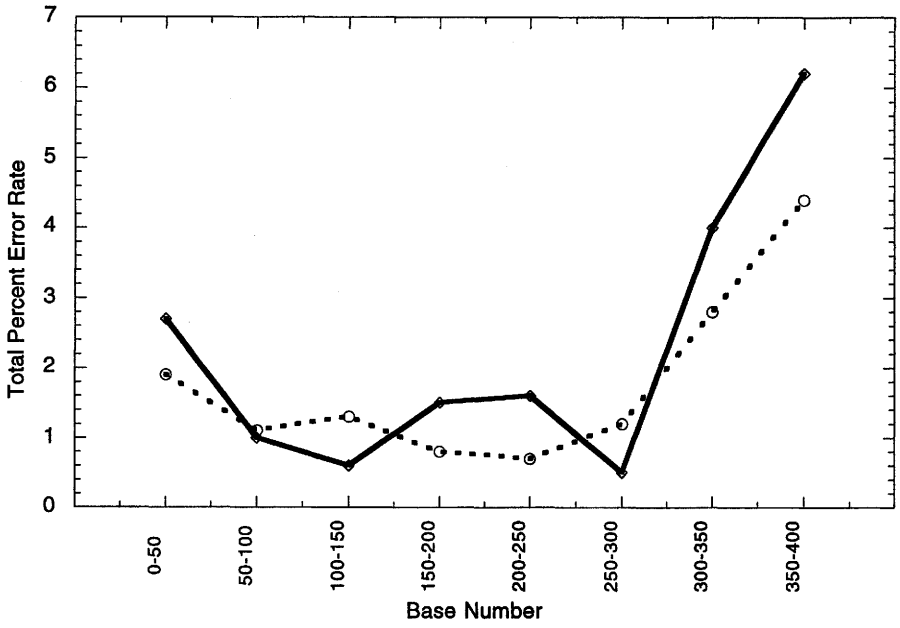


Figure 6 Average error rates by region. The average percent error rate for all types of errors in 50-base regions of the called sequence is compared for this algorithm (solid line) and for ABI *Analysis* software (broken line).

ative performance of the ABI software and this algorithm varies widely on a file-by-file basis with a relatively low correlation between the error rates they produce.

If errors are broken down by region (Fig. 6), the greatest source of errors for both this and the ABI software lies toward the end of the sequence, where peak widths are too great to be resolved individually. This algorithm generates mostly deletion errors in this region, whereas ABI's *Analysis* tends to produce relatively more insertion and mismatch errors.

Figures 7 and 8 show the distribution of the event confidence metric and its relation to the probability of an insertion or deletion error, as measured over bases 0–450 of each file. In cases where the probability of such an error is >50%, inserting or deleting a base would improve the overall frameshift error rate. In the case of insertion errors, such cases are rare, but a significant fraction of deletion errors fall into this category. It is therefore possible to improve the overall error rate by adding bases wherever their event confidence would be greater than ~0.8.

Similarly, the balance between incorrect calls and ambiguity calls can be adjusted using the identity confidence metric. Figure 9 shows the distribution of this metric, together with its relationship to the probability of a mismatch error, also over the first 450 bases. Whereas an identity confidence cutoff of 0.4 was used for the purposes of this evaluation, another value might be optimal depending on the particular sequencing application.

Figure 10 illustrates the relationship between the average identity confidence and the rate of mismatch errors in regions 10 bp in length. Note

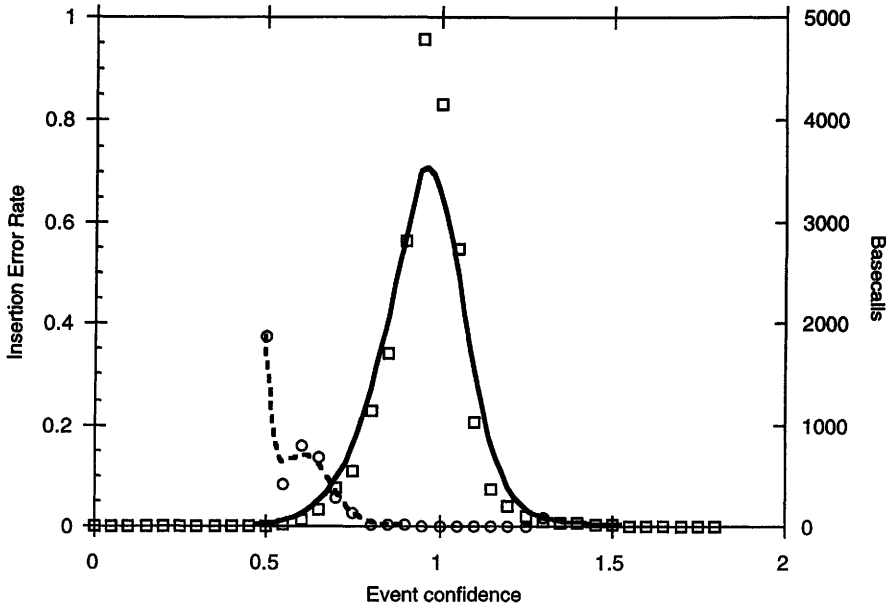


Figure 7 Event confidence distribution and its relation to the probability of an insertion error. The solid line indicates the distribution of the event confidence metric; the broken line gives the probability that a base call with a given event confidence represents an insertion error.

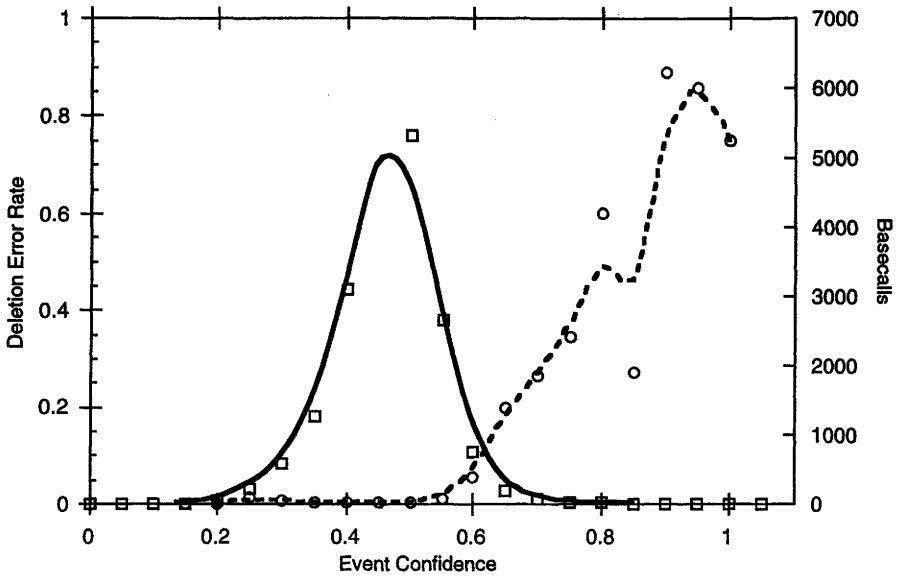


Figure 8 Event confidence distribution for uncalled bases and relation to rate of deletion errors. The solid line gives the distribution of the event confidence metric for bases that were not called but were postulated to be part of the sequence; the broken line indicates the probability that such a base call would be correct, thus causing a deletion error by its absence.

that it is possible to find an upper bound for the error rate such that 90% of regions with a given average identity confidence will have an error rate lower than this bound. Furthermore, for a large number of cases, one can state with high confidence that there are no mismatch errors whatsoever.

The distribution of the normalized spacing deviation, and the mean and 90th percentile frameshift error rates against its value, are shown in Figure 11. The utility of this metric is evident in the relatively large proportion of regions that can be nearly guaranteed to be error free, coupled with a stronger sensitivity to errors than is provided by the event confidence alone.

With appropriate adjustments to the mobility correction and event editing parameters, this algorithm has also been found to be effective with data obtained from dye-terminator sequencing, using both *Taq* and *Taq*–FS chemistries. Dye-terminator data differ from dye primer data in that they do not require any mobility correction, peak spacing is more regular, and peak intensity is more variable. Results are therefore improved by adjusting the event editing step to favor regular spacing over high peak intensity. Additionally, collaborations with other researchers have indicated that it is also applicable to data from new sequencing instruments, including the ABI 377 sequencing system (L. Stein, pers. comm.) and a capillary array sequencer which incorporates a full-spectrum fluorescence detector (A. Miller, pers. comm.).

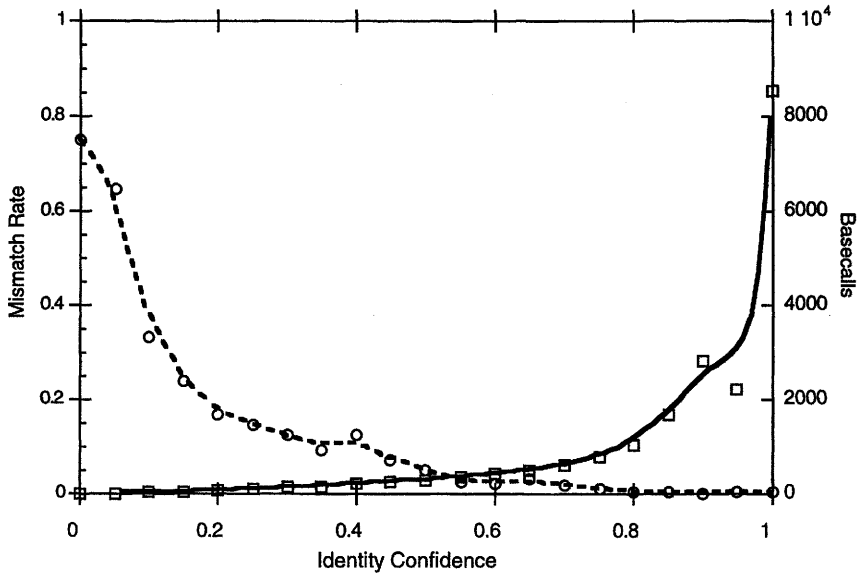


Figure 9 Identity confidence distribution and its relation to the probability of a mismatch error. The solid line gives the distribution of the identity confidence metric; the broken line indicates the probability that a base call with a given identity confidence is a mismatch error.

CONCLUSIONS

Although this algorithm exhibits a higher overall error rate than is produced by the ABI *Analysis* software when evaluated with data from ABI instruments, it has the advantage of

being relatively independent of any particular sequencing technology and more tolerant of data that do not conform to expected parameters. In addition, it generates information that can assist in sequence assembly and other sequencing applications, and affords control over the balance between different error types.

A qualitative assessment of the results indicates that the ABI software offers the best performance on data in which the peak spacing lies within a characteristic range, typically 9–12 sample points. In addition, the ABI software can interpret regions with extremely low resolution more accurately, whereas the event filtering mechanism used here cannot resolve individual peaks consistently. However, even high-quality data that do not satisfy this peak spacing expectation will fail using ABI software, whereas this algorithm will operate without any additional difficulty.

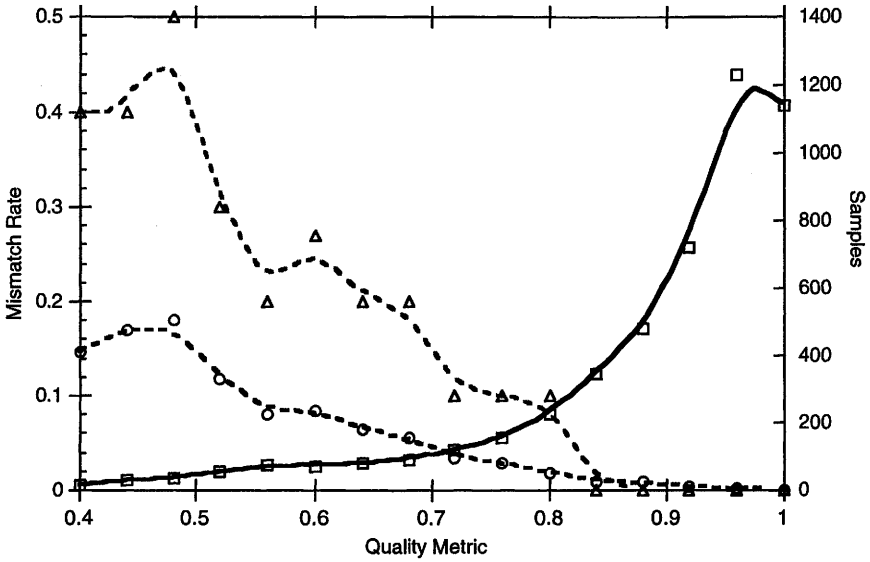


Figure 10 Distribution of average identity confidence and its relation to the mismatch error rate over small regions. The solid line gives the distribution of the identity confidence metric as averaged over regions 10 bp in length; the broken lines indicate the mean (○) and 90th percentile (△) error rates in those regions.

This algorithm's generic approach to base calling makes it attractive for the development of new sequencing technology as well as selected aspects of production sequencing. Desirable improvements include a mechanism for finding the mobility correction coefficients automatically, as well as an alternative base-calling mechanism that can be employed in regions of low resolution.

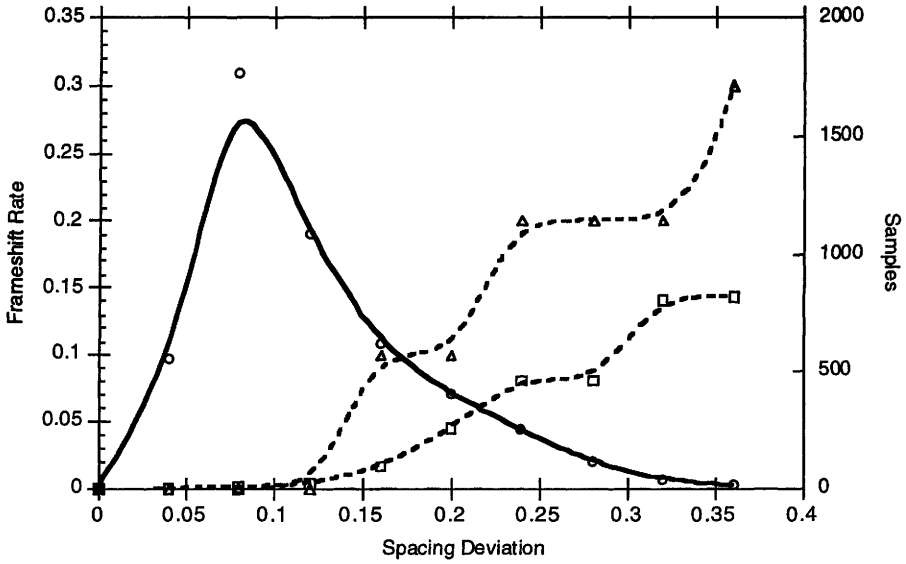


Figure 11 Distribution of the normalized spacing deviation and its relation to the frameshift error rate. The solid line gives the distribution of the standard deviation of base spacing normalized to the average spacing; the broken lines indicate the mean (□) and 90th percentile (△) error rates in those regions.

ACKNOWLEDGMENTS

I acknowledge Ron Davis, Michael Walker, Fred Dietrich, and Clark Tibbetts for their invaluable assistance with this work. Research was funded by National Institutes of Health grant 1P01HG00205.

The publication costs of this article were defrayed in part by payment of page charges. This article must therefore be hereby marked "advertisement" in accordance with 18 USC section 1734 solely to indicate this fact.

REFERENCES

Cormen, T.H., C.E. Leiserson

and R.L. Rivest. 1992. Single-source shortest paths. In *Introduction to algorithms*, pp. 514–550. The MIT Press, Cambridge, MA.

Giddings, M.C., R.L. Brumley Jr., M. Haker, and L. Smith. 1993. An adaptive, object oriented strategy for base calling in DNA sequence analysis. *Nucleic Acids Res.* **21:** 4530–4540.

Golden, J.B. III, D. Torgersen, and C. Tibbetts. 1993. Pattern recognition for automated DNA sequencing I: On-line signal conditioning and feature extraction for basecalling. In *First International Conference on Intelligent Systems for Molecular Biology* (ed. Hunter, Searls, and Shavlik). AAAI Press, Washington, D.C.

Hunkapiller, T., R.J. Kaiser, B.F. Koop, and L. Hood. 1991. Large-scale and automated DNA sequence determination. *Science* **254:** 59–67.

Masters, T. 1993. *Practical neural network recipes in C++*. Academic Press, San Diego, CA.

Needleman, S.B. and C.D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48:** 443–453.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 1992. Gauss-Jordan elimination. In *Numerical recipes in C*, pp. 36–41. Cambridge University Press, Cambridge, MA.

Tibbetts, C., J.M. Bowling, and J.B. Golden III. 1993. Neural networks for automated base calling of gel-based DNA sequencing ladders. In *Automated DNA sequencing and analysis techniques* (ed. J. Craig Venter). Academic Press, San Diego, CA.

Received May 10, 1995; accepted in revised form January 18, 1996.