Lane tracking software for four-color fluorescence-based electrophoretic gel images.

M L Cooper, D R Maffitt, J D Parsons, et al.

Genome Res. 1996 6: 1110-1117
Access the most recent version at doi:10.1101/gr.6.11.1110

Lane Tracking Software for Four-color Fluorescence-based Electrophoretic Gel Images

Matthew L. Cooper,[1,3] David R. Maffitt,[2] Jeremy D. Parsons,[1] LaDeana Hillier,[1] and David J. States[2]

[1]Genome Sequencing Center, Department of Genetics, and [2]Institute for Biomedical Computing, Washington University School of Medicine, St. Louis, Missouri 63110

Software to track sample lanes automatically in four-color, fluorescence-based, electrophoretic gel images has been developed for application in large-scale DNA sequencing projects. Lanes and lane boundaries are tracked by analyzing a first difference approximation to the gradient of a vertically integrated and processed "brightness" profile. Initially lanes are located in a region of the gel image selected for good horizontal lane spacing and signal strength. The software uses models of expected lane and interlane spacing and lateral lane behavior to maintain accurate tracking on imperfect gels. In areas where intensity-based tracking is difficult, interpixel column correlation is also used to locate and define lane features. Summary statistics and compressed-in-time images are generated for user evaluation of tracking performance. The software developed has been tested successfully on gel images with degradations including significant horizontal lane motion (curving) and image artifacts, and is now in full-scale use in our sequencing projects.

Increasing the level of automation of electrophoretic DNA sequencing is an important and well-documented challenge to the Human Genome Project (HGP) (Watson 1985). Several advances toward this end have been made, including, most notably, the development of fluorescence-based sequencing instruments (Smith et al. 1986; Prober et al. 1987; Ansorge et al. 1988; Brumbaugh et al. 1988). Fluorescence-based sequencing systems allow the coupling of electrophoretic separation, fragment detection, image analysis, trace processing, and base-calling, without manual intervention. This level of automation is predicated on robust, high-performance software that, in the case of sample lane tracking, is only now becoming available (Berno 1996).

The genomes of several model organisms are being completely sequenced as part of the HGP, including that of *Caenorhabditis elegans*. In the context of sequencing the *C. elegans* genome, manual tracking of sample lanes in the gel images has proven to be a time-consuming obstacle to achieving increased throughput and further automation of accurate, reproducible sequence determination (Wilson et al. 1994). Recently, doubling the channel sampling across the gel has allowed us to increase the number of samples per gel from thirty-six to sixty-four, but has resulted in a concomitant increase in the difficulty and time required to manually reposition the tracking lanes on the appropriate samples. In the most problematic gel images, manual correction of the vendor-supplied tracking software on difficult gels can require up to 3 hr. This effort is necessitated by the failure, almost total in some cases, of the supplied data collection software to accurately track samples in gels exhibiting anomalous sample migration patterns.

To further automate the pathway from DNA sequence of individual shotgun clones to contiguous cosmid sequences, algorithms to accurately locate and track lanes of DNA sequencing fragments in fluorescence-based four-color electrophoretic gel images have been developed and tested successfully. The new software for lane tracking significantly reduces the manual processing time invested in gel image analysis, and represents the first step in the development of a complete UNIX-based gel analysis package.

Our goal is to locate and accurately track the lanes in a gel image that result from excitation and detection of electrophoretically separated,

[3]**Corresponding author.**
E-MAIL mlc@cis.wustl.edu; FAX (314) 286-1810.

6:1110–1117 ©1996 by Cold Spring Harbor Laboratory Press ISSN 1054-9803/96 $5.00

fluorescently labeled fragments of DNA. Many available gel analysis packages rely on vendor-supplied lane tracking software (Golden et al. 1993), manual tracking of the lanes (Giddings et al. 1993), or, most often, a combination of the two. The automation of lane tracking is difficult for a number of reasons. In principle, the negatively charged phosphate backbones of the DNA fragments should cause them to electrophorese in a straight path through the pores of the polymerized acrylamide gel in the presence of a linear potential gradient. In practice, gel or plate contaminants, excess salt in loaded samples, electric field or polymerization variability, or machine misuse or failures can cause the lanes to move laterally, sometimes significantly. Furthermore, image artifacts of undetermined origin are often observed. These can be described as "stripes" or "patches" in the gel image of approximately constant intensity that obscure the underlying data. Postulated causes of artifacts include gel impurities, plate scratches, dust on the detector, laser cutout, buffer leaks, and software failures. Observed artifacts, even within a single gel image, fail to exhibit a consistent set of characteristics, and can be mistaken for lanes by tracking software. Also, gels may contain empty lanes as a result of failed reactions or poorly formed wells, and lateral spacing, lane width, and signal strength can vary considerably with the location in the gel image and the comb used in loading.

Our software, Getlanes, has been designed for use with four-color fluorescence-based electrophoretic gel images. All code is written in ANSI C. Thus far, the gel images have been generated by the Applied Biosystems (ABI) 373A Sequencing Instrument, and accompanying 388 Channel Data Collection and Analysis Software (Version 1.2.388) or the ABI 377 Sequencing Machine and 194 Channel Data Collection and Analysis Software (Version 2.1). The 194 Channel Software has also been used with the ABI 373A. Once data collection (on the Macintosh connected directly to the sequencing machine) is complete, the gel file is transferred to a UNIX file system using one of many available UNIX/Macintosh file-sharing packages, Columbia Appleshare Protocol (CAP, publicly available from Columbia University). This transfer separates the image data, contained in the ABI gel file data fork, and the resource data, contained in the ABI gel file resource fork, into two UNIX files. These two files are then processed by Getlanes in the UNIX environment.

After lane tracking, one-dimensional traces are extracted down the lanes from each of the four ABI filter images. These traces are then processed and used for base calling. Presently, ABI gel files are transferred to a UNIX machine for automatic tracking by a daemon that runs Getlanes using default options, and then back to a Macintosh for manual inspection. The daemon process runs continuously, and regularly checks specified directories spawning processes to execute the lane tracking software on any new gel files it discovers. Statistics are compiled in log files. Once the gel is back on a Macintosh, the ABI-supplied software is used for both trace extraction and preprocessing.

In the future, additional UNIX tools will be available for gel inspection, manual tracking, and trace preprocessing. Locally optimized UNIX-based base-calling software is already in use in our project (P. Green, unpubl.). Thus our overall aim is to use only the ABI Data Collection software and to complete the gel analysis using improved UNIX-based software, including the software documented here. The design of these tools is modular to permit comparative testing of each individual component prior to the completion of our entire gel analysis package.

RESULTS

We have tested Getlanes in three ways. First, we have tested the software using gel images exhibiting extreme (and unusual) imperfections. Figures 1–3 show three such gel images with artifacts and extreme lateral sample. The gel of Figure 3 possesses an artifact in the lower left that crosses

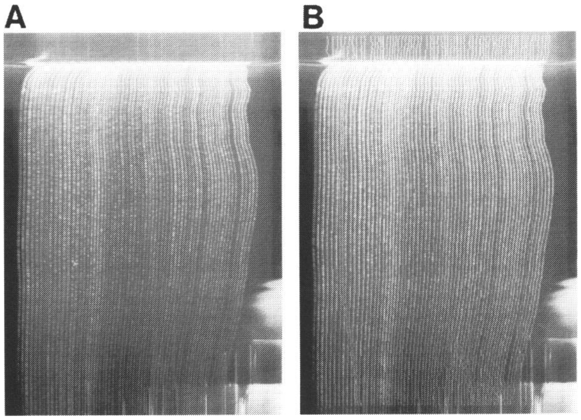

Figure 1 Example gel imperfections. (*A*) A prominent artifact is crossing the curving lanes. (*B*) The gel is shown tracked using lane-behavior models.

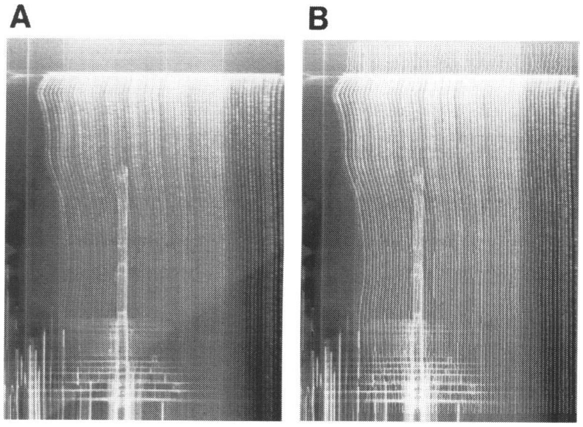

Figure 2 The most curved section of the gel in *A* is shown retracked in *B,* demonstrating the success of our definition of a lane using peaks and boundaries even in an area of large lateral movement. This gel also has "missing" lanes in which the signal is weak. The gel is retracked with missing lanes added, automatically enabling the application of our spacing models.

several of the underlying lanes, but its effects are minimized by the application of the horizontal spacing model (see Methods). Such artifacts confuse algorithms lacking lane behavior models, as artifacts often have greater intensity than the lanes nearby, as seen in the intensity plot of Figure 4 of a vertically integrated horizontal slice of this gel image. The gel in Figure 2A exhibits severe curving of the lanes at the top of the image (physically, the bottom of the gel). This gel tests the robustness of our definitions for lanes as image features. The gel of Figure 3A contains an artifact that makes manual tracking nearly impossible. This necessitates a combination of horizontal and vertical lane behavior models, described in Methods.

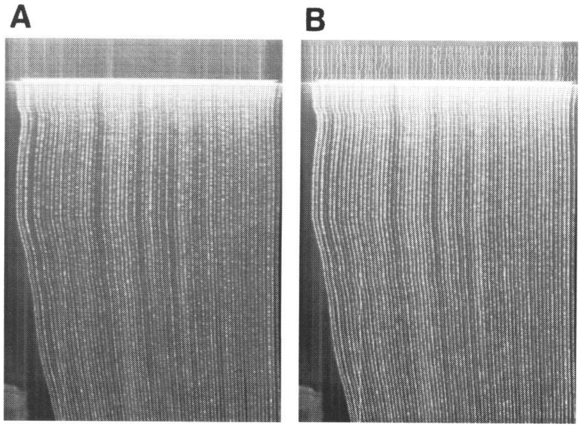

Figure 3 (*A*) Artifacts that make tracking in the *lower left* corner extremely difficult and erase most hopes of determining the underlying sequence. (*B*) The gel is shown tracked by Getlanes salvaging much of the data.

Second, we have tested the software's performance statistically against ABI software as well as human retracking. A comparison of Getlanes, the software supplied by ABI and human tracking is shown in the Tables 1 and 2, prepared using production data from our lab. Table 2 presents the results of Table 1 normalized by the corresponding manual results. In preparing these statistics, ABI lane-tracking software placed more than one lane on a single sample in some instances. This has skewed our statistics in favor of ABI in some cases. To partially correct for this, ABI results were not allowed to exceed the corresponding human result used for normalization. Obviously, double tracking a single sample can create major problems in any subsequent data processing.

To obtain summary statistics, we subjected the data to our normal laboratory protocols for data processing. Data (tracked with either ABI or Getlanes software or manually) for each lane was extracted and processed using the ABI analysis software for both. Base calls were performed using the program PHRED (P. Green, unpubl.). Each trace was evaluated for quality and removed entirely or trimmed based on measures of peak to highest noncalled peak values and peak to shoulder ratios. Regardless of trace quality, trimming was not allowed to extend past 400 bases. Sequences were screened for sequencing and cloning vector (10–12% of the sequences are entirely vector) and removed or clipped accordingly. The percentage success shown in Table 1, then, is the number of vector-free, high-quality reads divided by the total number of reads processed (number of lanes loaded) for ABI-tracked data versus Getlanes-tracked data, both without manual intervention. The mean number of bases per successful trace is the mean clipped read length (clipped for vector and for low-quality data) for the successful reads. This method provides a measure of the ability of the tracking to successfully follow the lane throughout the length of the gel. The mean bases per lane loaded is the sum of the quality, vector-clipped read lengths divided by the total number of reads processed (total number of lanes loaded). Finally, the number of entered bases is the sum of the quality, vector-clipped read lengths of the successful reads.

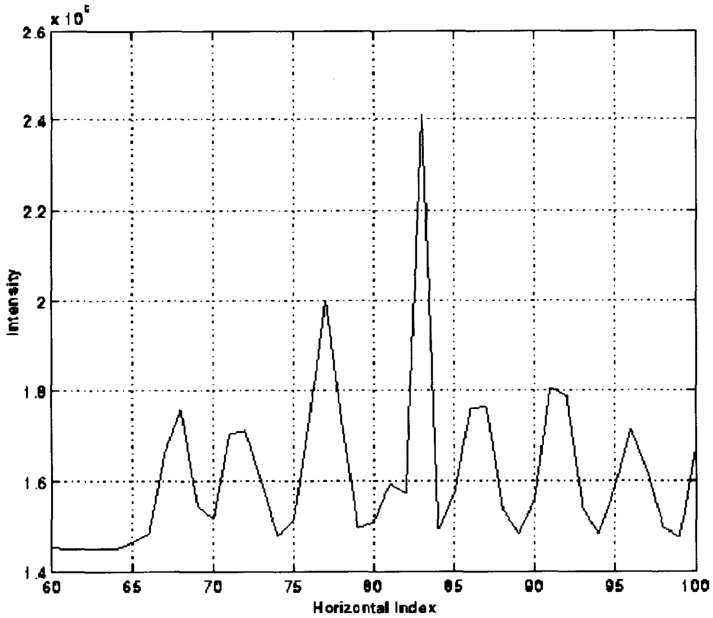

Figure 4 This plot shows the effect of the artifact in the *lower left* of the gel from Fig. 1. The artifact obscures the peak resulting from the true lane at index 81. The peak resulting from the true lane becomes the shoulder on the *left* of the peak resulting from the artifact at index 83. This effect is overcome by the use of horizontal spacing models.

Judging from the successful trace percentage in Table 1, the sample of gels used here is below average in quality, and Getlanes is successful in salvaging data. The statistics demonstrate that the software performs approximately as well as a human, and significantly better than ABI. In six of the 20 cases summarized above, Getlanes tracking yielded >2000 more entered bases than ABI, and in one instance >10,000 more. In four cases where ABI tracking yielded more bases, only once did ABI successfully track an additional sample, indicating that it tracks the same sample with more than one lane when it fails to locate an expected number of lanes. In these cases, using the human result as a correction, the number of bases entered using ABI tracking never exceeded the Getlanes result by 1000 bases (on a 64-lane gel). In five cases, the number of bases entered using Getlanes tracking exceeded the human result.

M13 subclones for the data summarized above came from both human expressed sequence tags, plasmid artificial chromosomes, and *C. elegans* cosmids. In most cases, sample lanes are tracked very similarly by humans and Getlanes. Human tracking does not yield major gains in the number of bases called and entered into our cosmid sequence data bases. Thus, data tracked by humans and missed by Getlanes are often of poor quality and cannot be included in assemblies of larger contiguous sequences.

Finally, we have tested Getlanes by full-scale introduction into our production groups. Although transferring the gel files from the ABI Macintosh to the UNIX system and back again is cumbersome, it enables us to evaluate the performance of Getlanes prior to the completion of the downstream processing. Nonetheless, the performance of Getlanes' has dramatically reduced time spent manually retracking sample lanes as documented below, despite these unnecessary data transfers. It is estimated that the use of this software in our group has reduced the number of people performing this task by up to 60% and time spent per gel by up to 80%.

DISCUSSION

The lane tracking approach described here was designed as a useful compromise between speed and performance, and run times on a Sun SPARC5 workstation for 388 Channel ABI 373A gels containing 6000 rows or more are ~1–2 min. The memory required is ~14 megabytes, but up to

Table 1. Raw Performance Statistics

	Success (%)	Mean bases per successful trace	Mean bases per lane loaded	Entered bases
ABI	57	287.5	203.4	10743
Getlanes	69	312.55	241.1	12923

Statistics shown compiled over 19 standard 64-lane production gels. Entries are averaged over the 19 gels.

Table 2. Normalized Performance Statistics

	Success (%)	Mean bases per successful trace	Mean bases per lane loaded	Entered bases
ABI	81	95	84	81
Getlanes	100	100	99	98

Statistics normalized by corresponding results of manual tracking. Compiled over 19 standard 64-lane production gels. Entries are averaged over the 19 gels, then normalized by the corresponding human average.

two-thirds of this may be the gel image data, and could be compressed by an order of magnitude if needed (the compression would merely store intensity information as it is laterused: multiple rows summed together so no useful information would be lost). The brightness technique, though simplistic, provides great flexibility in the adaptation of our models to the data and was found to be significantly better on gels with poor signal strength, though poor, at times, in the presence of artifacts. The correlation approach was more effective when lanes merged and in the presence of certain artifacts. Its pitfall is the use of the four-column neighborhood that implicitly assumes that lane and interlane features can be distinguished using correlation over any four columns. This may not be true for unusually large gaps between lanes or wide lanes. The choice of a four-column neighborhood yielded the best overall performance for our data, and the two approaches to tracking complement each other very well. Other more sophisticated approaches were considered and sometimes tested, but deemed unnecessary given the success of the current version.

The most common error made by the software, although relatively infrequent, is to miss one of the outermost lanes of the gel. It is a straightforward error to correct in general, because additional lanes are placed to the right side of the tracked lanes, so they do not interfere with the tracking of correctly identified lanes. This error is due to stringent requirements of peaks that are labeled as lanes in the starting window. These requirements are used to avoid mistaking intensity maxima resulting from artifacts or background fluctuations for lanes. The performance of the software on gels with low signal, missing lanes, or curving lanes is satisfactory in large part because of the definitions of a lane as a global image feature, and the behavior models that are used to track each lane, as described earlier.

Tolerances of the spacing models vary with the tracking techniques to best exploit the different profiles, though the overall structure is the same. By tracking boundaries closely, it is hoped that in extracted one-dimensional traces, signal strength will be maximized, as the horizontal integrations between the boundaries will span the full lane width. Two hundred locations of lane boundaries for each lane are recorded, and summary statistics, as well as compressed-in-time tagged image file format (TIFF) images are produced. Code is also available to extract traces down the tracked lanes using a horizontal integration over a user-specified or automatically computed lane width moving down the lanes. The program's use of command line flags to control its options is very flexible for interactive use and allows its easy integration into scripts.

Future versions of the software could, if necessary, use larger scale modeling of gel features, one- and two-dimensional image enhancement, reduced memory requirements, and dynamic programming. We also plan to make it compatible with other sequencing technologies as well as the aforementioned complete UNIX-based gel tracking and analysis interface and package currently in development.

Conclusion

The development of this software has addressed a significant obstacle to automatic high-throughput, reproducible sequence determination. The software is routinely used by our production sequencing groups. The results of its use on 40–50 gel images per day for several months indicate that it improves upon vendor-supplied software markedly, and approximates human tracking performance. The further automation of lane tracking and gel file management has significantly reduced the time invested by our production groups in this tedious task. It is hoped that our completion of a robust UNIX-based software package to track lanes, extract and preprocess traces, and make base calls will both increase the throughput and improve the sequence quality of large-scale DNA sequencing projects.

METHODS

The algorithm's structure is depicted in Figure 4, and detailed below. For additional details, the source code is available as noted at the conclusion of this paper. First the image data is loaded and preprocessed to form a composite image. From the composite image, a starting point is selected, and horizontal tracking profiles are generated by filtering the vertical sum of 40–50 rows of the image (described in more detail below). Using the starting tracking profile, global characteristics of the gel are identified. With these global characteristics and lane behavior models, tracking of lanes and lane boundaries proceeds iteratively over the whole image.

Image Preprocessing

The ABI gel file contains four filter images, produced by filtering the scan data with four bandpass filters tuned to the emission spectra of the four fluorescent dye primers used. The four filter images are loaded and summed to form a brightness image. Several image-processing steps are performed to facilitate subsequent tracking. The first is a crude background subtraction. After determining the maximum and minimum intensities of the image, the minimum value is subtracted from the entire image, and the image is normalized to a maximum intensity of 512. Then, a histogram equalization is performed to the normalized image, to increase its contrast and dynamic range (Gonzales and Woods 1992).

A starting point for tracking is then selected from a sample of gel regions by comparison of signal strength and horizontal lane spacing. Greater signal strength improves the detection of lanes and lane boundaries. Likewise, regular spacing increases the effectiveness of the lane behavior models applied later and facilitates the determination of characteristics of the gel as a whole, including missing lanes, horizontal boundaries of the lane data, and the number of lanes in the image. Algorithms that attempt to process disjoint portions of the gel without these global parameters are unable to track lanes in more difficult regions of the gel (Blanchard 1993).

In each of three regions, a horizontal gel profile is formed by vertically integrating over a fixed window with a size of between 30 and 50 rows. The window size used by these algorithms in the iterations described later is computed as a function of the total number of rows in the gel image. In each of these profiles, maxima and minima are located, and signal strength is compared by computing the mean peak intensity. The standard deviation of the intermaxima widths is used to evaluate the regularity of the horizontal spacing of the lanes. The ratio of the mean peak height to the standard deviation of the intermaxima distance is maximized to select the starting region.

Next, the starting window intensity histogram is computed and used to identify the horizontal boundaries of the portion of the image containing the lanes. The modes of this histogram tend to be background intensities. Thus, the maximum background intensity is estimated by comparison of the incremental integral of the histogram and the total integral, as in Equation 1.

$$X_b = \max_{0 \leqslant I < 512} \left\{ I \colon \sum_{i=0}^{I} x_i \leqslant \left(0.3 \sum_{0 \leqslant i < 512} x_i \right) \right\} \tag{1}$$

The subscript i represents intensity, between 0 and 511, and x_i represents the number of pixels in the starting window with intensity i. Thus the x_i comprise the histogram of the starting window.

This equation is a thresholding of the gel image, using a threshold computed by approximating the unnormalized cumulative distribution function of the intensity with the integral of the histogram. In each row of the starting window, the outermost columns in which the intensity equals or exceeds X_B as computed in Equation 1 are found. Of these, the outermost columns are marked as horizontal boundaries for the area containing the lane data in the starting window. These boundaries are adjusted according to the lateral movements of the outermost lanes throughout the gel image.

The number of lanes in the image, as well as missing lanes, are identified and compared with the expected number, determined according to ABI machine type. If the expected number of lanes is not found, a compensatory number of "artificial" lanes are added on the right side of the image to be available for subsequent manual tracking. Alternatively, the positions of the leftmost and rightmost lanes (horizontal boundaries for lane data), the number of lanes, and starting region can be specified explicitly in the command line.

The software then iterates in both directions vertically to cover the entire image. At each iteration, the software locates and tracks lanes over a 50% redundant window of the gel consisting of typically 30–50 rows. After one or more horizontal gel profiles are used to locate and define the lanes, lane behavior models are used to fine-tune the tracking, and to work through difficult portions of the images.

Tracking Techniques

The software uses multiple tracking techniques. The first approach, which has proven the most robust in practice, is the brightness-based approach. An intensity-based profile is formed by vertical integration over the current window of the brightness image. This profile is filtered for peak enhancement, and a first-difference approximation to the gradient is computed to identify maxima, which are marked as lane locations. The output of the peak-enhancement filter, at horizontal index i, is given in Equation 2 for input profile $\bar{f}$ a row vector with elements indexed from 1 to 388, as the ABI channels, and μ_f the mean of the elements of that row vector:

$$h_i = \frac{f_{i-1} \cdot f_i \cdot f_{i+1}}{\mu f^2} \tag{2}$$

Location of gradient extrema allows us to determine lane boundaries. Gradient maxima are labeled as left lane boundaries, and gradient minima are labeled as right lane

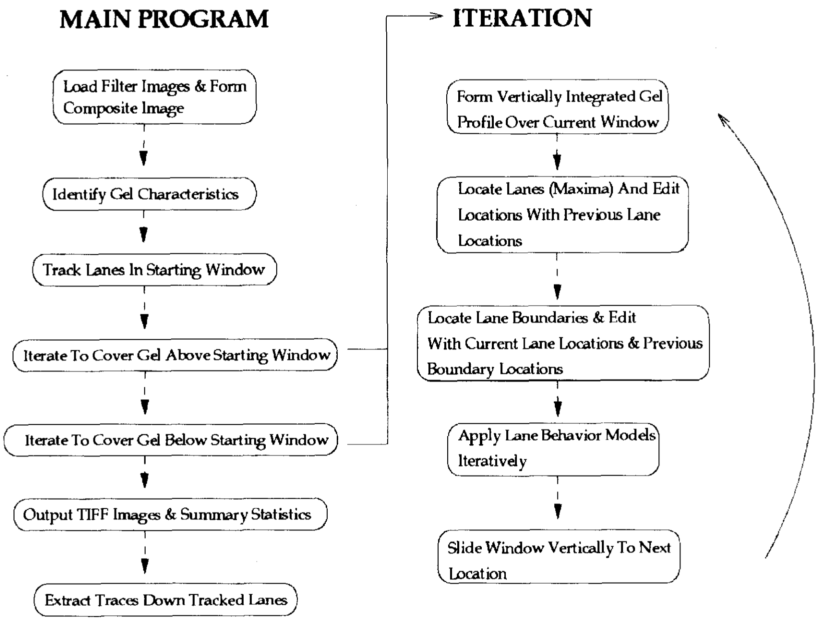
MAIN PROGRAM
Load Filter Images & Form Composite Image
Identify Gel Characteristics
Track Lanes In Starting Window
Iterate To Cover Gel Above Starting Window
Iterate To Cover Gel Below Starting Window
Output TIFF Images & Summary Statistics
Extract Traces Down Tracked Lanes
ITERATION
Form Vertically Integrated Gel Profile Over Current Window
Locate Lanes (Maxima) And Edit Locations With Previous Lane Locations
Locate Lane Boundaries & Edit With Current Lane Locations & Previous Boundary Locations
Apply Lane Behavior Models Iteratively
Slide Window Vertically To Next Location

Figure 5 Flowcharts depicting the stucture of Getlanes1.0. The flowchart on the *right* details the two iterations in the flow chart on the *left*.

boundaries. The algorithm then applies models of expected lane behavior, as detailed below, and iterates to cover the entire image.

The second approach calculates intercolumn correlation coefficients in the four filter images independently over four column neighborhoods, in which the current column is the third column from the left. The columns used in these computations span the current window (30–50 rows). Portions of the four images being used are stored in cache memory to reduce the memory needed for access to the large color images. The correlation coefficients are combined to create an alternative horizontal gel profile. Let the quantity $\bar{x}_i \diamond \bar{x}_j$ equal the correlation coefficient of the vectors $\bar{x}_i$ and $\bar{x}_j$. In Equation 3 below, $\bar{x}_i$ denotes the column of the current window at horizontal index *i*. The value of the profile at horizontal index *i* is computed as follows:

$$X_i = \sum_{\textit{Filter Images}} [(\bar{x}_{i-2} \diamond \bar{x}_i) + (\bar{x}_{i-1} \diamond \bar{x}_i) + (\bar{x}_{i-2} \diamond \bar{x}_{i+1}) + (\bar{x}_{i-1} \diamond \bar{x}_{i+1})] \qquad (3)$$

This profile is made nonnegative by subtraction of its minimum value, then normalized and scaled to a maximum value equal to the mean of the maxima of the brightness data. This profile is processed similarly to the previously mentioned brightness approach, and either method can be used exclusively throughout the gel via a command line option.

The default approach is a combination of the two. The brightness profile is computed initially, and in cases where it fails to reveal expected lanes or lane boundaries as required by the models below, the correlation profile is checked for improved feature definition. Brightness is the initial technique applied for its speed and generally reliable performance.

Lane Behavior Models

The lane behavior models are applied in two steps. In the first step, each lane location is analyzed individually for consistency with lane locations in the previous iteration. The lane locations are constrained to lie between the corresponding lane boundaries tracked in the previous iteration. This restriction is based on empirical observations that the maximum lateral velocity of a lane is approximately one pixel per 20 rows and that typical lane widths are three to four pixels. This underlies the concept of lane momentum, that the lateral movements of lanes occur in consistent directions in localized areas of the gel image. When no peak is found within a lane's boundaries from the previous iteration, the software estimates its location based on momentum observed in the two previous iterations. Lane boundaries must be within an absolute distance of the lane and their previous locations, or they are estimated similarly. This approach precludes mislabeling maxima or gradient extrema due to background fluctuations, gel artifacts, or even neighboring lanes as a particular lane or lane boundary.

In the second step, the models applied reflect collective characteristics of the lane locations in any horizontal window of the gel. After the current lane and lane boundary locations are established as above, first-order statistics are computed by which outliers in terms of lane width or interlane width are identified. These outliers are processed in an iterative routine to correct tracking errors similar to those above. The vertical models used are the same, but here are applied to the lane and interlane widths rather than to the lane and lane boundary locations. The horizontal models used impose regularity on the lane and interlane widths across the gel. An estimate for lane location or boundaries for statistical outliers is computed by averaging estimates based on the horizontal and vertical behavior models. First-order statistics are recomputed and the iteration continues until the spacing satisfies the constraints of the model, or until a maximum number of iterations is reached, when horizontal lane spacing is unusually irregular. Generally this second step efficiently fine-tunes the tracking locations established in the first step, and is no more than a quality check.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health grant HG00956. The authors would also like to acknowledge the review of the code by Darren Platt and Asif Chinwalla, the readings of this manuscript by Elaine Mardis, and the direction of Richard Wilson and Robert Waterston. The informatics and, above all, production groups at the Genome Sequencing Center at the Washington University School of Medicine are especially thanked.

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.

NOTE

The following sites on the World Wide Web are available: to obtain CAP and related documentation: http://www.cs.mu.oz.au/appletalk/cap.page; to obtain Getlanes and related documentation (note: distribution or use of Getlanes and all software listed here for commercial purposes without permission of the Genome Sequencing Center of the Washington University School of Medicine is strictly prohibited): http://genome.wustl.edu/gsc/gschmpg.html; to obtain the Getlanes Daemon (Perl Script for automatic running of Getlanes) and related documentation: http://genome.wustl.edu/gsc/gschmpg.html; and to obtain the TIFF software library and related documentation: http://www.sgi.com/Fun/tiff/tiff-v3.4beta018/html/index.html.

REFERENCES

Ansorge, W., B. Sproat, J. Stegemann, C. Schwager, and M. Zenke. 1987. Automated DNA sequencing: Ultrasensitive detection of fluorescent bands during electrophoresis. *Nucleic Acids Res.* **15:** 4593–4602.

Berno, A.J. 1987. A graph theoretic approach to the analysis of DNA sequencing data. *Genome Res.* **6:** 80–91.

Blanchard, A. 1993. "Sequence-specific effects on the incorporation of dideoxynucleotides by a modified T7 polymerase." Ph.D. thesis, California Institute of Technology, Pasadena, CA.

Brumbaugh, J.A., L.R. Middendorf, D.L. Grone, and J.L. Ruth. 1988. Continuous on-line DNA sequencing using oligodeoxynucleotide primers with multiple fluorophores. *Proc. Natl. Acad. Sci.* **85:** 5610–5614.

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

Golden, J.B., D. Torgersen, and C. Tibbetts. 1993. Pattern recognition for automated DNA sequencing: I. On-line signal conditioning and feature extraction for basecalling. Proceedings of the First International Conference on Intelligent Systems for Molecular Biology. AAAI Press.

Gonzales, R.C and R. E. Woods. 1992. *Digital image processing*. Addison-Wesley, Reading, MA.

Prober, J.M., G.L. Trainor, R.J. Dam, F.W. Hobbs, C.W. Robertson, R.J. Zagursky, A.J. Cocuzza, M.A. Jensen, and K. Baumeister. 1987. A system for rapid DNA sequencing with fluorescent chain-terminating dideoxynucleotides. *Science* **238:** 336–341.

Smith, L.M., J.Z. Sanders, R.J. Kaiser, P. Hughes, C. Dodd, C.R. Connell, C. Heiner, S.B.H. Kent, and L.E. Hood. 1986. Fluorescence detection in automated DNA sequence analysis. *Nature* **321:** 674–679.

Watson, J.D. 1985. The Human Genome Project: Past, present, and future. *Science* **248:** 44–48.

Wilson, R., R. Ainscough, K. Anderson, C. Baynes, M. Berks, J. Bonfield, J. Burton, M. Connell, T. Copsey, J. Cooper, A. Coulson, M. Craxton, S. Dear, Z. Du, R. Durbin, A. Favello, L. Fulton, A. Gardner, P. Green, T. Hawkins, L. Hillier, M. Jier, L. Johnston, M. Jones, J. Kershaw, J. Kirsten, N. Laister, P. Latreille, J. Lightning, C. Lloyd, A. McMurray, B. Mortimore, M. O'Callaghan, J. Parsons, C. Percy, L. Rifken, A. Roopra, D. Saunders, R. Shownkeen, N. Smaldon, A. Smith, E. Sonnhammer, R. Staden, J. Sulston, J. Thierry-Mieg, K. Thomas, M. Vaudin, K. Vaughan, R. Waterston, A. Watson, L. Weinstock, J. Wilkinson-Sproat, and P. Wohldman. 1994. The C. elegans genome project: Contiguous nucleotide sequence of over two megabases from chromosome III. *Nature* **368:** 32–38.

Received January 5, 1996; accepted in revised form August 21, 1996.