# Automated detection of point mutations using fluorescent sequence trace subtraction

James K. Bonfield, Cristina Rada and Rodger Staden\*

MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

Received April 10, 1998; Revised and Accepted May 26, 1998

# ABSTRACT

The final step in the detection of mutations is to determine the sequence of the suspected mutant and to compare it with that of the wild-type, and for this fluorescence-based sequencing instruments are widely used. We describe some simple algorithms for comparing sequence traces which, as part of our sequence assembly and analysis package, are proving useful for the discovery of mutations and which may also help to identify misplaced readings in sequence assembly projects. The mutations can be detected automatically by a new program called TRACE\_DIFF and new types of trace display in our program GAP4 greatly simplify visual checking of the assigned changes. To assess the accuracy of the automatic mutation detection algorithm we analysed 214 sequence readings from hypermutating DNA comprising a total of 108 497 bases. After the readings were assembled there were 1232 base differences, including 392 Ns and 166 alignment characters. Visual inspection of the traces established that of the 1232 differences, 353 were real mutations while the rest were due to base calling errors. The TRACE\_DIFF algorithm automatically identified all but 36, with 28 false positives. Further information about the software can be obtained from http://www.mrc-Imb.cam.ac.uk/pubseq/

# INTRODUCTION

Mutation detection is becoming increasingly important as a clinical diagnostic tool and as part of genome research and several techniques are currently in use. These include single-stranded conformational polymorphism analysis (1), denaturing gradient gel electrophoresis (2), heteroduplex analysis (3,4), chemical mismatch cleavage (5) and direct sequencing. A review of these techniques has been given by Grompe (6). Ultimately, direct sequencing is required to define the precise location and nature of any change. For this final step fluorescence-based sequencing instruments are now extensively used.

Because no single sequence reading obtained from a fluorescencebased sequencing instrument is 100% reliable, the search for mutations is not simply a matter of comparing a suspected divergent sequence against that of the wild-type: differences between two sequences can be due to mutations or to base calling errors and therefore the problem is to recognize real mutations within a background of base calling errors.

To eliminate base calling errors by visual comparison of wild-type and mutant traces, Tamary (7) used the ABI SeqEd program and Jonsson (8) used an early version of our sequence assembly software. Automated detection of heterozygotes using improvements to version 1.2 of the PE/ABI Factura program has been described by Phelps (9). In this case traces were analysed to identify base positions that exhibited a secondary signal strength greater than a selected percentage threshold of the primary peak. A further method of this type was described by Kwok (10) and a more recent method from Nickerson (11) has the advantage of being used in conjunction with a sequence alignment editor.

Our main interest is in automating the accurate location of base changes. The method we have devised employs sequences determined by a fluorescence-based sequencing instrument. The new software then normalizes and subtracts the pairs of wild-type and mutant traces to produce a new set of traces which represent their differences. The value of these calculations is two-fold: first, they enable reliable automatic detection of mutations; second, they greatly simplify visual inspection of traces because the software can display the traces of the two original sequences plus the trace of their differences. To illustrate the basic idea, in Figure 1 we show an example of the trace difference display from our sequence assembly program GAP4 (12). The two top panels contain the traces from a pair of readings and the bottom panel contains a plot of their trace differences. As can be seen, the difference traces are effectively flat except where a mutation occurs, resulting in a positive difference in one trace coincident with a negative difference in the other. It is obviously much easier to use this plot to direct visual inspection of the results than it is to look for differences throughout the whole of the two original traces. For automatic detection of mutations our new software analyses the trace differences to look for significant and coincident peaks; one positive and the other negative, and marks those above a given threshold.

# METHODS

## Methods background

The new routines we describe are additions to the extensive set of methods that we have developed for use in large scale sequencing projects. Of these existing methods, the following components are used in conjunction with the new routines described in this

\*To whom correspondence should be addressed. Tel: +44 1223 402389; Fax: +44 1223 213556; Email: rs@mrc-lmb.cam.ac.uk



**Figure 1.** This shows the traces as displayed by the GAP4 contig editor and illustrates the features of the trace differences as analysed by TRACE\_DIFF. The top trace is a consensus trace, the middle one is from an individual reading and the bottom one the difference between the other two. In general the difference trace contains very few features, however, three base changes are shown with their associated peaks in the differences plot. The left hand mutation  $(C \rightarrow T)$  causes a pair of strong peaks in opposite directions with a small context effect. The next mutation  $(T \rightarrow G)$  has strong opposite peaks and a strong context effect peak. The third mutation has quite strong opposite peaks and a single context effect peak.

paper. SCF files (13) are used for storing trace and sequence data generated by sequencing instruments; Experiment files (14) are used for transferring data about sequences between programs; the program GAP4 (12) is used for aligning and editing sequences. GAP4 obtains its data from Experiment files and stores the assembled sequences in its own database. The GAP4 contig editor can display and scroll (in register) sets of traces from aligned readings. Segments of sequences can be annotated and the annotated segments are marked with coloured boxes, called tags. Each tag type has its own identifying colour and the contig editor's cursor can jump from one tag to the next.

During conventional sequencing projects each base is generally determined from several independent readings and so the accuracy of each reading is less important than it is for mutation detection studies. Here, we cannot rely on base calling alone, but instead use it in conjunction with numerical analysis of the trace data.

#### Mutation detection criteria

Our assumption is that the traces are a more reliable picture of the data than the base calls, and that, aside from sequence context effects (15,16), the shapes of the traces are reproducible. By context effects we mean the differences in the heights of the peaks are a function of more than just a single base. Hence, a single base difference between two segments of DNA can result in trace differences, not only for the divergent base, but also for its immediate neighbours (see the bottom trace of Fig. 1). The nature of the context effects also depends on the specific sequencing protocol employed.

For the purpose of mutation detection a false positive will be a difference between two sequences which is the result of a base calling error. A false negative will be caused by the rare event of a base calling error disguising a real base difference. For automatic detection of mutations, we can search for trace differences above a specified minimum value that are accompanied by a change to the called base, although this will necessarily miss the rare cases where a base called in error disguises a real mutation. Context effect changes to the peak heights of bases adjacent to a mutated base will not be accompanied by a base change, and therefore will not be reported as mutations. So, our first criterion (**I**) for detection of a mutation is that the called base should be different from that of the wild-type.

In order to compare the traces from a pair of sequences we need to normalize the data. We have tried several methods of scaling the amplitudes of the traces (the *y* direction), including expressing the values as proportions of the total signal at each point, but at present we find that we obtain the best results by leaving the data unscaled in *y*. For scaling along the traces (the *x* direction), we need to get the peaks for the equivalent bases from the two sets of traces in register. If there are no insertions or deletions this is straightforward, otherwise there are a number of possible cases. The first case is that the base calling software has called an extra base but there is no extra peak and the base calling is incorrect. The second is that the base calling software has missed a base for which there is a peak. The other cases are not base calling errors but genuine insertion and deletion differences between wild-type and mutant. The software uses a dynamic programming algorithm to align the sequences and then the traces are re-sampled to produce two new sets of traces.

Let  $t_i^{w}(x)$  and  $t_i^{m}(x)$  be the heights of the wild-type and mutant traces at trace sample position *x* for each base type *i* = A, G, C or T. (In our data, for each peak/base there are typically around 12 sample points for each base type.)

The differences between the wild-type and mutant traces are

$$d_{\mathbf{i}}(x) = t_{\mathbf{i}}^{\mathbf{w}}(x) - t_{\mathbf{i}}^{\mathbf{m}}(x)$$

It is these values that we assess and, for example, where a single base is changed, say A to T, we would expect to observe abnormally large magnitudes for  $d_A(x)$  and  $d_T(x)$ , one positive and the other negative. The significance of the values  $d_i(x)$  is assessed by first calculating their mean m and standard deviation s over the length of the reading and then locating all positions for which the observed value exceeds a threshold defined by the user. The values of *m* and *s* are the combined values for all  $d_i(x)$ . We chose to use the mean and standard deviation because we needed a value that depended on the sequence being analysed (rather than use a fixed value for all sequences), but, as described above, this calculation is dependent on the number of mutations. Therefore, the calculation of the mean and standard deviation is restricted to the segments of the trace differences that correspond to the troughs between the peaks of the original data. We also restrict the search of  $d_i(x)$  to the segments that correspond to the positions of the peaks in the original data. Context effects may result in changes to the heights of peaks, but are not expected to produce a positive change for one base type and a negative change for another. Referring again to Figure 1, we see that at base position 179 in the trace difference plot there is a C $\rightarrow$ T mutation resulting in strong peaks (in opposite directions) for each of these base types, but this is accompanied by a small context effect peak at base position 180. A more striking example can be seen at base position 184 in the differences plot: here a  $T \rightarrow G$  mutation (with peaks in opposite directions) is followed by a large change in the A trace. Both of these examples show that real mutations exhibit strong peaks in opposite directions and context effects tend to produce peaks in only one direction.

So our second criterion (II) for a mutation at position x is that there exist two base types i and j such that

$$d_{\mathbf{i}}(x) \ge m + n.s \qquad \qquad \mathbf{1}$$

$$d_{i}(x) \leq m - n.s \qquad 2$$

and where *n* is a value set by the user.

We assume that if criterion  $\mathbf{II}$  alone is satisfied then it corresponds to a context effect, and if criterion  $\mathbf{I}$  alone is satisfied then it corresponds to a base calling error. If they are both satisfied at the same position then a mutation is reported.

## Automatic mutation detection

For any individual prospective mutant sequence the processing is generally performed in two steps. First, a new program, TRACE\_DIFF, is used to align and compare the mutant and wild-type sequences and traces and to locate possible mutations. Second, the sequence is assembled into a GAP4 database from where users can visually check the differences between the wild-type and mutant traces. As explained below, using one of the search modes in GAP4, the visual inspection can be performed very rapidly. To facilitate batch processing of sequence readings for mutation detection we have added a new type of Experiment file record (WT) to contain the name of each reading's wild-type trace file. Then TRACE\_DIFF can get all the information it needs from the readings' Experiment file. Also, all mutations detected by TRACE\_DIFF are written back to the reading's Experiment file as mutation tags and the peak height from the difference trace is recorded in the tags' comment field. As all the information needed for processing (apart from user-supplied parameters that are common to all readings) are in each reading's Experiment file, any number of readings can be processed in a single batch.

The mutation detection program TRACE\_DIFF performs the following steps.

Get the options and parameters selected by the user such as the threshold n used in equations 1 and 2 and the name(s) of input file(s) and possibly an output file. Because, in general, the traces are poor at both ends of readings, the user selects a start and end point in base positions to define the range over which the peak detection routine should be applied.

Read-the wild type and mutant traces and sequences.

Use dynamic programming to align the two sequences.

Use the sequence alignment to align the two traces.

Compare the two sets of traces to create a new set that represents the differences between them.

Calculate the mean and standard deviation of the difference traces.

Scan the difference traces and locate all peaks where criterion **II** is satisfied. If such a peak corresponds to a base difference (i.e. criterion **I** is also satisfied) report it. Note that TRACE\_DIFF can also be used in a mode which reports all positions with significant trace differences, even though the called base is identical to that of the wild-type.

If requested, write the trace differences as a new SCF format file. The sequence of this file denotes all the suspected mutations discovered by TRACE\_DIFF using the IUBC (17) redundant base symbols in upper and lower case, e.g. Y denotes a C $\rightarrow$ T change and y a T $\rightarrow$ C change. This SCF file can be viewed just like a normal trace file.

If the mutant sequence's trace file was accessed indirectly by using the data on the WT line of its Experiment file, then TRACE\_DIFF can write annotation data into this Experiment file to denote the positions of all the detected mutations. If this file is assembled into a GAP4 database these annotated bases will appear as 'mutation tags'.

#### Checking for mutations with the GAP4 contig editor

Since the original version of the GAP4 contig editor was described many new features have been added. A view of the editor exhibiting some of the modes relevant to mutation detection is shown in Figure 2. Datasets for mutation studies are often very large and so we have added a vertical scroll bar to enable all readings to be viewed whatever the depth of the alignment. Reading names can now also be scrolled horizontally. Base accuracy estimates for machine-read sequence data were introduced in Bonfield and Staden (18) by writing bases below a given quality threshold in red. We have extended this to use grey scales to depict base quality or confidence. In Figure 2 we have switched on a GAP4 display mode which, by changes to the background colour, highlights all edits that have been made: deletions are shown in red, changes in pink, padding characters in light green and modified confidence values in blue. Note that

-	Contig Editor: 34 LEA960															
	< C:	93 >	< 9: 51	>	📕 Inse	rt Edit	Modes		Unda	Next Problem	Comma	nds Setting:			(	auit Help
	**	<	>	>>	LI.			11	0.0		1					
2				~	170		18	0	190	2	00	210	220	230	240	
	40	LEA938			TTTTG	GGGGA	GGTG	CCTGAC	TTCTG	TEGGACC	GTGCAT	GTAATTT	ACACTTGAGCI	<b><i>IGGCACTGCAC</i></b>	GTCATGGT	GACC
	41	LEA936			TITIG	GGGGGA	GGTG	CCTGAC	TTCTG	CTGGTACC	GTGCAT	<b>IGTAACTT</b>	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC
	42	LEA943			TTTTG	GGGGGA	GGTG	CCTGAC	ITCTG	CTGCTACC	GTGCAT	<b>IGTAACTT</b>	ACACTTGAGCI	GGCACTGCAC	GTCATGGT	GACC
	43	LEA9472			TTTTG	GGGCA	GGTG	CCTGAC	TCTG	CTEGTACC	GTGCAT	GTAACTT	ACACTTGAGCI	CACTGCAG	GTC	GACC
L	52	LEA939			TTTTG	GGGGA	GGTG	CCTGAC	TCTG	CTEGTACC	GTGCAT	<b>IGTAACTT</b>	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC
	51	LEA945			TTTTG	GGGGGA	GGTG	CCTGAC	TCTG	CTEGTACC	GTGCAT	GTAACTT	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC
	53	LEA946			TITIG	GGGGA	GGTG	CCTGAC	TCTG	CTEGTACC	GTGCAT	GTAACTT	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC _
	57	LEA966			TTTTG	GGGGA	GGTG	CCTGAC	TCTG	CTGGTACC	GTGCAT	<b>IGTAACTT</b>	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC
	54	LEA957			TTTTG	GGG	GGTG	CCTGAC	TCTG	CTGGTACC	GTGCAT	GTAACTT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
	55	LEA959			TETTG	GGGGA	GGTG	CCTGAC	TCTG	CTGGTACCI	GTGCAT	GTAACTT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
	56	LEA968			TITIG	GGGGA	GGTG	CCTGAC	TCTG	CTGGTACC	GTGCAT	GTAACTT	ACACTTGAGCI	GGCACTGCAG	GTCATGGT	GACC
	20	LEA903			TITIG	GGGGA	GGTG	CCTGAC	TCTG	CTGGTACC	GTGCAT	GTAACTT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
	23	LEA906		-	TITIG	GGGGA	GGTG	CCTGAC	TCTG	CTGGTACC	GTGCAT	TAATT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
	18	LEA934			TITIG	GGGGA	GGTG	CCTGACT	TCTG	CTGGTACC	GTGCAT	GTAACTT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
<	>	CONSENS	US	-##-	TITTG	GGGGA	GGTG	CCTGACT	TCTG	CTGGTACC	GTGCAT	GTAACTT	ACACTTGAGCT	GGCACTGCAG	GTCATGGT	GACC
		Frame 2	+		F	GG	G	A 🗕 I	. L	LVP	V H	V T	Y T 🕷 A	GTA	GHG	DY

Figure 2. An example of the GAP4 contig editor showing modes of operation useful for mutation detection. A vertical scroll bar enables alignments of any depth to be viewed. Individual readings are numbered, named and written left to right. Accuracy estimates or confidence values for each base are shown using grey scales: the darker the background the poorer the data. Mutation tags are shown in dark green (for example there are three visible in the top sequence). Changes to the original base calls or accuracy estimates are also colour coded by changes to the background colour: deletions are shown in red, base changes in pink, padding characters in light green and modified confidence values in blue.

the edit positions are not stored as tags, but are retained automatically each time a change is made to the data. Mutation tags are dark green and so, for example, there are three on the top sequence (LEA938), whose trace was shown in Figure 1.

Within the GAP4 contig editor, to facilitate the use of trace comparisons for checking and detecting mutations the following new features have been added. Although, for the convenience of users, there are several methods of selection, there are, in essence, two new modes of operation. The first, like TRACE\_DIFF, compares the traces from any pair of overlapping readings and creates a new trace display of their differences. This display, as seen in Figure 1, results in three sets of traces appearing on the screen. The difference trace is calculated 'on the fly' and the three sets of traces can be horizontally scrolled in register using the cursor in the contig editor.

The second mode is more complicated in that the trace of a selected reading is compared with a 'consensus trace' calculated from the readings it overlaps. The consensus trace and its differences from the traces of the selected reading are displayed as shown in Figure 1. The consensus trace is the average trace found by summing and scaling selected traces at each point. Optionally, the segments of the trace that correspond to a position where the called base for a reading does not match the consensus sequence are not included in the consensus trace calculation. In this way, the expected trace for the wild-type can be created from a set of mutant sequences. The consensus traces can be saved to a file in SCF format.

If, prior to assembly into the GAP4 database, the readings have been analysed with TRACE\_DIFF to produce mutation tags, the contig editor search function can step from tag to tag whilst simultaneously scrolling traces and their difference trace. In this way, the user can check each possible mutation assignment and delete the tags for those that look false.

### Obtaining an overview using the GAP4 template display

The GAP4 template display can provide an overview of all the mutations in a set of readings. Figure 3 shows readings as red

arrows and tags as small coloured rectangles. Along the base of the display is a scale marked in 100s of bases and the vertical black line is a cross-hair. Tags are shown both on the individual readings and on the scale at the bottom. We were particularly interested in the overall distribution of mutations and this display clearly shows that the majority are found to the left of the cross-hair. This display can also be used to immediately identify polymorphic residues in population studies.

#### Choosing a representative wild-type sequence

A suitable wild-type trace for use by TRACE\_DIFF and GAP4 can be a single high quality reading or (and this has given the best results for our data) the consensus from several less good readings processed by GAP4. As mentioned above, GAP4 can even create one by, at each point, only including the components of traces that produce the consensus base. Alternatively, it may be beneficial to always include the wild-type DNA on the gel used to determine the suspected mutant sequences. For consistent results it is probably best to use GAP4 to create a single consensus trace for use in all experiments.

# RESULTS

The mutation detection program TRACE\_DIFF is being used to study somatic hypermutation in immunoglobulin genes. In Table 1, column A, we show the results of applying TRACE\_DIFF to three sets of readings (214 in total) determined as part of the somatic hypermutation study. The sequences were obtained from mice containing immunoglobulin  $\kappa$  light chain transgenes. The transgenic lines were Lk6 (19), Lk[ $\Delta$ Li] (20) and LK045 (C.Rada, unpublished results). The sequences were obtained by PCR and cloning in M13mp18 as previously described (21). Sequencing was performed using fluorescent dye terminators (Thermosequenase cycle sequencing pre-mix kit from Amersham Life or ABI Prism Dye terminator cycle sequencing ready reaction kit with AmpliTaq DNA polymerase FS from Applied Biosystems) in an ABI 377 sequencer.



Figure 3. Here the GAP4 template display shows readings as red arrows and tags as small coloured rectangles. Along the base of the display is a scale marked in 100s of bases and the vertical black line is a cross-hair. In this example tags automatically generated by TRACE\_DIFF are shown in green, false negatives have been manually edited to red and false positives to yellow.

Table 1. Comparison of automated versus visual detection of mutations

	А	В	С	
Number of readings	214	65	65	
Number of bases	108 497	43 741	36 400	
Average analysed length	503	673	560	
Base differences	1 232	274	162	
Real mutations	353	165	132	
TRACE_DIFF false positives	28	15	0	
TRACE_DIFF false negatives	36	5	0	

The automatic mutation detection program TRACE\_DIFF was applied to two sets of sequence readings. Column A, dye terminator data; column B, BigDye terminator data loosely clipped for quality; column C, the same readings as column B, but quality clipped more severely. For column A, threshold n = 4.0; for columns B and C, n = 3.0.

For each of the three sets of sequences the consensus trace used by TRACE\_DIFF was generated from five of the better quality readings. Prior to mutation analysis the readings were automatically clipped of poor quality data at either end using TRACE\_CLIP (R.Staden, unpublished results). The threshold n was set at 4.0 and searches were performed for bases 10–600 within the clipped data. The value (4.0) used for n was derived from work on a separate set of data (not included in the analysis) and was not tuned to give the best results for the three test sets. The results from TRACE\_DIFF were compared with those obtained from scanning the complete traces by eye after the readings had been assembled into a GAP4 database.

The test data consisted of 108 497 bases called using the standard ABI software. After the readings had been aligned with their consensus sequences they contained 1232 differences, of which 392 were bases called as unknown (N) and a further 166 were padding characters introduced during alignment. Visual inspection showed that there were 353 real mutations, and with n = 4.0 TRACE\_DIFF missed 36 of them and found 28 false positives. The false positives tended to be at the two ends of the readings, where the data were less reliable, and the false negatives were almost entirely due to the weak G after A problem that is found in the chemistry used.

While this manuscript was in preparation we received the materials to use the new ABI BigDye terminators and found a marked improvement in the sequences obtained: their lengths were increased and the weak G problem was almost non-existent. The results from one batch of data are shown in Table 1, columns B and C. Column B contains results from sequences that were loosely clipped for quality, giving an average analysis length per reading of 673 bases, and column C has the results when the readings were clipped more severely, to leave only high quality data of an average analysed length of 560 bases. As can be seen, there are far fewer base calling errors or uncertainties for both ranges. Using the threshold parameter n set to 3.0 for the extended range set, TRACE\_DIFF missed five mutations and found 15 false positives and for the narrower range, it missed no mutations and gave no false positives.

## DISCUSSION

We have described simple yet very useful methods for the detection and identification of mutations when sequences are determined using fluorescence-based sequencing instruments. As we have demonstrated, numerical treatment of the peaks is an effective method of automatic mutation detection and, second, the way that the differences are displayed within GAP4 makes it very much easier for the results to be checked visually.

We have demonstrated the reliability of automatic mutation detection for dye terminators and, for a more limited dataset, the new BigDye terminators. Given the wide choice of instruments and protocols in use it is not possible for us to cover them all. Nevertheless, we believe that those using the programs will quickly be able to establish suitable threshold values for TRACE\_DIFF appropriate to the sequencing method of their choice. Obviously, the choice of threshold value n also depends on the type of project being undertaken: for some work an error rate similar to that obtained for our test data would be acceptable and no visual checking within GAP4 would be required, but for other projects the threshold would need to be set low enough to give a high chance of finding all possible mutations and visual inspection using the tag search routine would be essential in order to rule out false positives. For this latter type of project we have considered a modification of the tag searching routines that would locate the mutation tags in descending order of trace difference peak height.

We would like to emphasize the convenience of the overall environment provided by our methods. The readings can be automatically processed and searched for mutations in batches of any size and then assembled into a GAP4 database. With the contig editor, its search facilities and trace difference displays the data can then be checked rapidly by eye. The methods have been in routine use here for 1 year.

We have concentrated on the use of the new methods for mutation detection, but believe that they may have a role to play in genome sequencing projects. In very repetitive regions of genomes, where very long repeats differ at only a few positions, assembly engines can easily create contigs that contain segments in which almost identical but unrelated readings have been aligned together. These repetitive regions are problematical, as it is difficult to determine how many copies of each repeated element there are and then to fit them together in the right order. The new facility in the GAP4 contig editor to compare a trace against the consensus trace can be used to find misplaced readings and this procedure could be automated if it proved useful.

Information about obtaining the programs (for UNIX systems only) is available from the WWW address given in the abstract.

# ACKNOWLEDGEMENTS

We thank Cesar Milstein for suggestions, Caroline Napper for help with sequencing and Kathryn Beal and Tony Crowther for critical reading of the manuscript. The work was supported by the UK Medical Research Council.

## REFERENCES

- Orita, M., Iwahana, H., Kanazawa, H., Hayashi, K. and Sekiya, T. (1989) Proc. Natl. Acad. Sci. USA, 86, 2766–2770.
- 2 Myers, R.M., Maniatis, T. and Lerman, L.S. (1987) Methods Enzymol., 155, 501–527.
- 3 Perry, D.J. and Carrell, R.W. (1992) J. Clin. Pathol., 45, 158–160.
- 4 White, M.B., Carvalho, M., Derse, D., O'Brien, S.J. and Dean, M. (1992) *Genomics*, **12**, 301–306.
- 5 Cotton, R.G., Rodrigues, N.R. and Campbell, R.D. (1988) Proc. Natl. Acad. Sci. USA, 85, 4397–4401.
- 6 Grompe, M. (1993) Nature Genet., 5, 111-117.
- 7 Tamary,H., Surrey,S., Kirschmann,H., Shalmon,L., Zaizov,R., Schwartz,E. and Rappaport,E.F. (1994) Am. J. Hematol., 46, 127–133.
- 8 Jonsson, J.J., Aronovich, E.L., Braun, S.E. and Whitley, C.B. (1995) Am. J. Hum. Genet., 56, 597–607.
- 9 Phelps,R.S., Chadwick,R.B., Conrad,M.P., Kronick,M.N. and Kamb,A. (1995) *BioTechniques*, **19**, 984–989.
- 10 Kwok,P.-Y., Carlson,C., Yager,T.D., Ankener,W. and Nickerson,D.A. (1994) *Genomics*, 23, 138–144.
- Nickerson, D.A., Tobe, V.O. and Taylor, S.L. (1997) Nucleic Acids Res., 25, 2745–2751.
- 12 Bonfield,J.K., Smith,K.F. and Staden,R. (1995) Nucleic Acids Res., 23, 4992–4999.
- 13 Dear, S. and Staden, R. (1992) DNA Sequence, **3**, 107–110.
- 14 Bonfield, J.K. and Staden, R. (1996) DNA Sequence, 6, 109–117.
- 15 Parker, L.T., Deng, Q., Zakeri, H., Carloson, C., Nickerson, D.A. and Quok, P.Y. (1995) *BioTechniques*, **19**, 116–121.
- 16 PE Applied Biosystems (1996) User Bulletin P/N 904357, stock no. 237827-001.
- 17 NC-IUB (1985) Eur. J. Biochem., 150, 1-5.
- 18 Bonfield, J.K. and Staden, R. (1995) Nucleic Acids Res., 23, 1406–1410.
- 19 Sharpe, M.J., Milstein, C., Jarvis, J.M. and Neuberger, M.S. (1991) EMBO J., 10, 2139–2145.
- 20 Rada, C., Yelamos, J. Dean, W. and Milstein, C. (1997) Eur. J. Immunol., 27, 3115–3120.
- 21 Gonzalez-Fernandez, A. and Milstein, C. (1993) Proc. Natl. Acad. Sci. USA, 90, 9862–9866.