COMPUTERS AND BIOMEDICAL RESEARCH **27**, 83–96 (1994)

# ANALYSIS: Software for Graphical Analysis of Multidimensional Flow Cytometric List Mode Data

TOM C. BAKKER SCHUT, RICHARD M. P. DOORNBOS,
AND BART G. DE GROOTH

*Department of Applied Physics, Applied Optics Group, University of Twente,
7500 AE Enschede, The Netherlands*

Received March 26, 1993

A computer program for graphical analysis of multidimensional flow cytometric list mode data is described. The program offers one-, two-, and three-dimensional inspection of an amount of data that is only limited by disk space. Subpopulations within the original data set can be identified by setting one or more two-dimensional AND gates around them. The order of measurement can be used as a parameter for evaluation of time-dependent processes. Other new parameters can be made by zooming in on a parameter, logarithmic transformation, or division of two parameters. The program is written in Turbo Pascal and it can run on any MS-DOC PC with an EGA/VGA resolution screen. © 1994 Academic Press, Inc.

## 1. INTRODUCTION

This paper addresses the problem of graphical analysis of flow cytometry data. A detailed description of flow cytometers and their use can be found in Shapiro's "Practical Flow Cytometry" (*1*). In a flow cytometer (usually biological) particles are analyzed by illuminating them one by one with a powerful light source and measuring scattered and/or fluorescent light. The number of particles that is measured is typically on the order of 1000–100,000 and the number of parameters is mostly on the order of 2–8. If all information is stored for each individual particle, the resulting data file has a list mode structure.

Even if only one type of particle is measured, not all the measured values of a certain parameter will have exactly the same value. This is due to electronic noise, displacement of the particle with regard to the light spot, etc. If the flow cytometer is optimized, it will be mostly due to variation in the characteristics of the particle.

In most cases more than one type of particle is measured, each type with its own mean and variation for each of the parameters. Flow cytometric data therefore consists of clusters of events that are more or less separated in one or more dimensions. The main objective of FCM data analysis is the separation

0010-4809/94 $6.00

and determination of the relative size and location of these clusters, starting from the not arranged list mode data.

In principle, all kinds of data analysis techniques can be used, such as graphical analysis, parametrical analysis, principal components analysis, and cluster analysis (*2*). List mode data can be analyzed using common spreadsheet programs. However, the limitations on the amount of data and on the analysis methods that can be used often make these programs unsuitable for analysis of flow cytometric data. Therefore dedicated software is being developed by FCM manufacturers and commercial software companies (for an overview see (*2*)). Most of these programs use graphical analysis methods for separating the clusters, like the PAINT-A-GATE program (Becton–Dickinson). For research applications, however, one often wants to have full command over the data manipulation, data analysis, and graphical output, which means that one has to write his own software. To our knowledge, none of the dedicated programs using graphical analysis methods have been published yet, apart from some routines developed for special applications, for example, for calcium determinations (*3*), and for three-dimensional projections (*4*)

In this article we describe a dedicated software program for graphical analysis of flow cytometric data that has been developed in our laboratory and evolved during the period 1987–1993. A multidimensional gating technique is used for selection of subpopulations (or clusters) within the data set. In order to facilitate the process of data inspection, the program offers the possibilities of creating new parameters and various modes of data representation. The program can read FCS 2.0 list mode data files (*5*). For those interested, the ANALYSIS software is available upon request.

## Materials and Methods

All data analysis algorithms, parameter manipulations, and representation routines are embedded in a menu-driven program called ANALYSIS. It is written in Turbo Pascal (Version 6.0, Borland International Inc., Scotts Valley, CA). It can run on any MS-DOS computer that has an EGA/VGA resolution screen. The program supports use of a serial mouse and a HP Paintjet printer.

For internal data storage the program uses TPROF units (Turbo Professional 5.0, TurboPowerSoftware, Colorado Springs, CO), designed for creation and manipulation of large multidimensional arrays. These arrays are not stored in direct memory but on a data drive. This allows the user to investigate an amount of data that is only limited by the space on the disk used; there are no software limitations on the number of events or on the number of parameters. The drive to be used for data manipulation can be selected: for fast manipulation a RAM drive is recommended, and for very large amounts of data a hard disk can be used. Within the program all parameter values are represented as bytes (range, 0–255) and stored in a two-dimensional array called DATA: DATA[$i$, $j$] represents the value for parameter $j$ of the event $i$.

When working with multidimensional data and a two-dimensional output

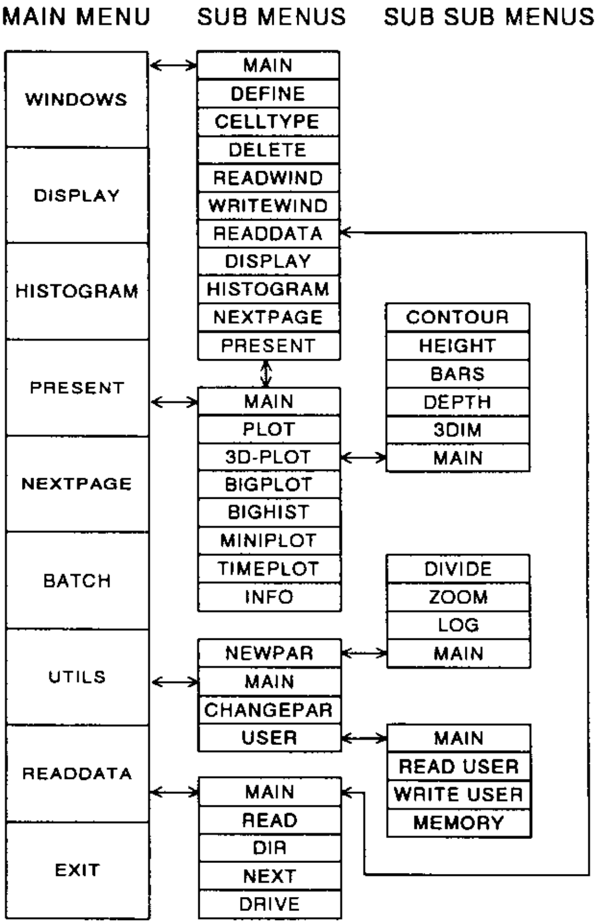

FIG. 1. Command structure of ANALYSIS.

device, dot plots are the most obvious way to represent the data. In a dot plot any event is represented as a single point. The coordinates of that point are the measured values for the parameters along the two axes of the plot. The standard output screen of ANALYSIS shows six dot plots and a menu bar with one line of information about the command that is in the menu selection window; all commands are easily accessible using keyboard/mouse-guided menus.

The command/menu structure of the program is represented in Fig. 1. We use this menu to describe all the data analysis and data representation features of the program.

### *READDATA*

The data is selected using the READDATA command. There is a submenu for selecting different ways and places to search for the data; the data can also be selected using (mouse-oriented) directory menus. The program can read standard FCS 2.0 list mode data files (5). The data are stored as bytes; if Measurement[$i$, $j$] is the measured value of event $i$ for parameter $j$ and Range [$j$] is the range of parameter $j$, then

$$\text{Data}[i, j] = (\text{Measurement}[i, j]/\text{Range}[j]) \cdot 255.$$

When reading data, a new parameter is automatically defined as the order of measurement; if the number of originally measured parameters is $n$, then

$$\text{Data}[i, n + 1] = \text{Round}(i/255).$$

For $n$-dimensional data, there are $n \cdot (n - 1)/2$ different dot plots. This means that our standard output screen can fully represent the data only if the number of parameters is 4 or less.

*NEXTPAGE*

In most cases we have more than four parameters, therefore we implemented the command NEXTPAGE. This command can be used to switch to another output screen that also contains six plots (but with different parameters in the plots).

*WINDOWS*

As mentioned above the main objective of FCM data analysis is the separation of the different clusters present in the data and determination of their relative size and location. In ANALYSIS this separation is done by assigning all events that have parameter values within certain boundaries or "windows" in the two-dimensional plots to a specific cluster. The command WINDOWS opens a submenu with a number of window-oriented commands. Using the DEFINE command, the boundaries for a certain cluster can be set repeatedly until the user is satisfied. For each cluster one can use up to 15 polygonal windows (with a maximum of 10 corners per window) that can be set in any of the two-dimensional plots. Events are only assigned to a cluster if their parameter values are within all boundaries (AND gates). Whether an event falls within a window is determined by counting the number of lines of the window that satisfy the following condition (with (DATA[$i$, $x$],DATA[$i$, $y$]) as the coordinates of the event and ($x_b$, $y_b$) and ($x_e$, $y_e$) the coordinates of the beginning and end of the line):

$$((x_b <= \text{DATA}[i, x]) \text{ AND } (x_e > \text{DATA}[i, x]) \text{ AND}$$

$$(y_b + (y_e - y_b)/(x_e - x_b) \cdot \text{DATA}[i, x] < \text{DATA}[i, y]))$$

$$\text{OR}$$

$$((x_e <= \text{DATA}[i, x]) \text{ AND } (x_b > \text{DATA}[i, x]) \text{ AND}$$

$$(y_e + (y_b - y_e)/(x_b - x_e) \cdot \text{DATA}[i, x] < \text{DATA}[i, y])).$$

It is seen that in fact the number of boundaries between the position of an event and one of the axes is counted. If the number of lines of the window that satisfy this condition is odd, or if the event falls exactly on one of the lines of the window, the event is considered to be inside the window. The maximum number of clusters that can be defined is 14. This number is in principle not limited, but was chosen with regard to the number of colors on the EGA screen; there are 16 colors, and one is used for background and one for the cells that are not assigned to any cluster. An event can in principle belong to more than one cluster. If, for instance, one cluster is defined (main cluster) and a part of that cluster is defined as another cluster (subcluster), all cells of the subcluster will also belong to the main cluster. Within the program, the information about cluster membership of each event is stored in a one-dimensional array of integers called COLOR. Membership of a cluster is expressed by the bit in the integer with the same ordinal number as the cluster. If an event belongs to more than one cluster, the event is assigned to the cluster with the highest ordinal number.

Within the WINDOWS submenu there are a number of other window-related commands. READWIND and WRITEWIND are commands to read and write window sets from and to disk. The window sets are stored as a collection of points together with the information about the parameters used in the plots they were set in. This option can be used to color a series of similar measurements with the same window settings. CELLTYPE shows all events plotted in the color of the cluster, that they are assigned to, together with information of the relative numbers of cells in all clusters. If the command READDATA is used in the windows menu, the data will be automatically divided into clusters according to the windows that are active at that moment. The DELETE command can be used to exclude certain clusters from being plotted and taken into account when calculating the relative numbers of cells in all clusters. With this option, hidden clusters can be revealed.

### *DISPLAY*

The command DISPLAY refreshes the screen. If there are any windows active, this command also deactivates all windows.

### *HISTOGRAM*

The command HISTOGRAM offers the possibility of showing one to four histograms (for an example, see Fig. 7). If there are windows active, it shows the histograms of the defined clusters in their respective colors. The histograms of the different clusters can be shown relative to each other (with respect to the maxima in the histograms) or unscaled. The histograms can be filtered using one or more second-order filters, with the same adjustable filter constant $\tau$, in line. The algorithm for one filter is as follows, with Hist the original histogram array, HistS the smoothed histogram array, and Histl and Histr help arrays:

$$\begin{aligned}
\text{Histl}[0] &= \text{Hist}[0]; \text{Histl}[1] = (\tau \cdot \text{Hist}[1] + \text{Histl}[0])/(\tau + 1),\\
\text{Histr}[255] &= \text{Hist}[255];\\
\text{Histl}[254] &= (\tau \cdot \text{Hist}[254] + \text{Histr}[255])/(\tau + 1); \qquad x = 2\text{–}255:\\
\text{Histl}[x] &= (\tau^2 \cdot \text{Hist}[x] + (2 + \tau) \cdot \text{Histl}[x - 1] - \text{Histl}[x - 2])/(\tau^2 + \tau + 1);\\
x &= 253\text{–}0:\\
\text{Histr}[x] &= (\tau^2 \cdot \text{Hist}[x] + (2 + \tau) \cdot \text{Histr}[x + 1] - \text{Histr}[x + 2])/(\tau^2 + \tau + 1);\\
\text{HistS}[x] &= (\text{Histl}[x] + \text{Histr}[x])/2.
\end{aligned}$$

This process is repeated with the smoothed histogram when more than one filter is used.

### *UTILS*

Special utilities are available under the UTILS command. The command CHANGEPAR can be used to change the parameters in the dot plots of the output screen. In NEWPAR three types of parameters may be defined. One can select a subrange of a parameter, $p$, as a new parameter, $n$, using the ZOOM function:

$$n = \text{ZOOM}(p)\text{: DATA}[i, n] = ((\text{DATA}[i, p] - \text{ll})/(\text{hl} - \text{ll})) \cdot 255,$$

with ll and hl the lower and higher zoom levels, respectively. In order see all the information of linearly measured parameters, having a large dynamic range, the new parameter, $n$, can also be defined as a logarithmic transformation of a parameter, $p$, with the LOG command:

$$n = \text{LnTr}(p)\text{: Data}[i, n] = 58.02775 \cdot \ln(1 + 80 \cdot \text{DATA}[i, p]/\text{range}[p]).$$

This transformation was chosen because it transforms 0 into 0, full range into 255, and a tenth of the range into 127.5. The third kind of parameter can be defined using the command DIVIDE. This parameter is especially useful for ratio measurements. The new parameter, $n$, is then defined as the logarithmically transformed ratio of the parameters $p1$ and $p2$:

$$\begin{aligned}
n = \text{LnTr}(3/4)\text{: DATA}[i, n] &= 58.02775\\
&\cdot \ln(1 + 8 \cdot \text{DATA}[i, p1]/\text{DATA}[i, p2]).
\end{aligned}$$

We transformed this parameter in such a way that it gives a good resolution between $p1/p2 = 0$ and $p1/p2 = 10$. After defining a new parameter the data file is read again, the DATA array is adjusted to the right size, and the new parameters are calculated using the total range of the original parameters.

The last command in the utilities menu is the USER command. In the USER menu there are two commands for reading and writing of so-called user files to disk. User files contain information about the configuration used, i.e., the drive used for storage of the DATA and COLOR arrays, the work directory, the parameters that are used in the dot plots of the standard output screen, and the number of self-defined parameters (with their definitions). When the program is started, it searches in the current directory for a user file called default.anu. If this file is not found, the drive used for data storage has to be defined first. The third command in the USER menu is WORK MEMORY, which lets you change the drive, used for storage of the DATA and COLOR arrays, within the program.

### *PRESENT*

The command PRESENT opens a submenu with all kinds of plot-oriented commands designed for a HP Paintjet. The command PLOT plots the upper part of the output screen containing filenames and dot plots or histograms (for an example, see Figs. 2 and 7). The command INFO can be used to plot the parameter names and any other comments. The command BIGPLOT is used to plot one or two large plots in which all legends, etc., can be set (for presentation purposes). With BIGHIST one can plot one big histogram and all data such as peaks, scans, cv's, and areas; it offers the possibility of reading more data files into the same plot (for comparison of the histograms of different files) (for an example, see Fig. 8). TIMEPLOT gives a screen-size plot of the values of one of the parameters or the numbers of events in each cluster as a function of the measuring order with the option of averaging over a number of consecutive events belonging to the same cluster (for an example, see Fig. 10). This can be very useful for analysis of time-dependent processes (*4*). The command MINIPLOT was made for noncolor presentations of a large number of clusters. For each defined cluster, it plots a very small black and white representation of the six plots of the output screen; the cluster is shown in black, and the rest of the data in gray (for an example, see Fig. 9). The command 3DPLOT offers a number of 3D representations of the data. The commands CONTOURS, BARS, and HEIGHT are three different ways for showing two-dimensional histograms (for an example, see Fig. 4). The command DEPTH gives a perspective front view of one of the plots with any of the parameters for depth information. The command 3-DIM gives a rotating image of the data within a cube (for an example, see Fig. 5). The moving image can be stopped and then plotted.

### *BATCH*

A BATCH command was added for analysis and plotting of large amounts of data files using predefined windows. This option was added because the plotting of many data similar data files is a tedious and time-consuming task (about 40 sec per plot).

### *EXIT*

Finally, the command EXIT can be given to leave the program.

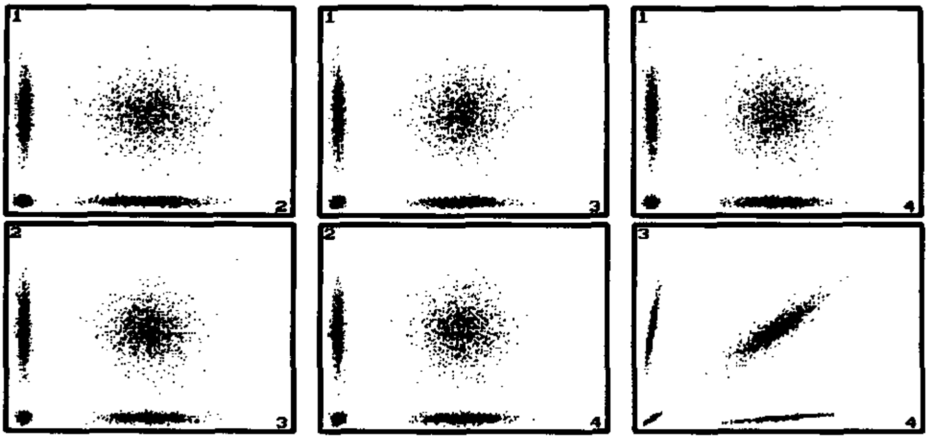

FIG. 2. The six possible dot plots for four-dimensional data. The data displayed are artificial, made with a Gaussian random generator.

## RESULTS

A number of the analysis and plotting routines are shown using an artificially made data file containing eight four-dimensional clusters. The data were made using a Gaussian random generator. All clusters contain the same number of events (512) and in all parameters they have either a mean of 128 and a FWHM (full-width at half-maximum) of 32, or a mean of 16 and a FWHM of 4. Normally all plots are made in color, which reveals the relative position of the clusters if they are hidden or overlapping. For this paper we have created a data file in which the clusters are clearly separated so that the black and white presentations

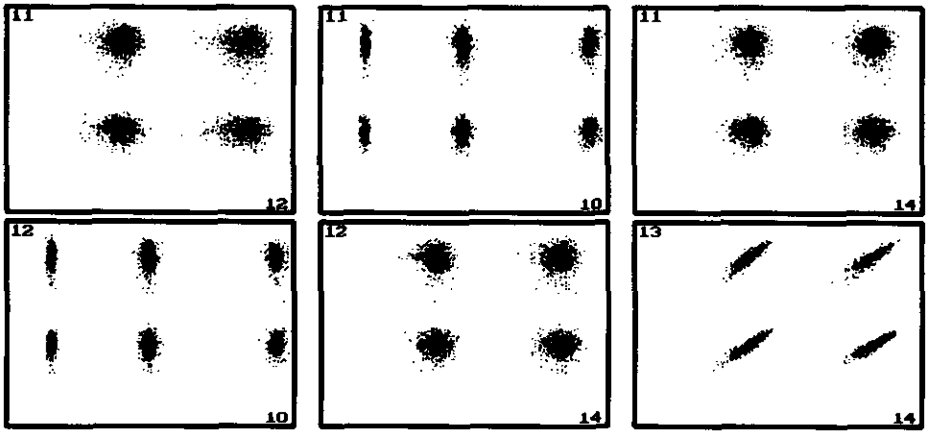

FIG. 3. Six dot plots of new parameters, constructed from the original four: parameter 10 is the logarithmic transformation of parameter 3 divided by parameter 4, and parameters 11–14 are the logarithmic transformations of parameters 1–4, respectively (made with LOG).

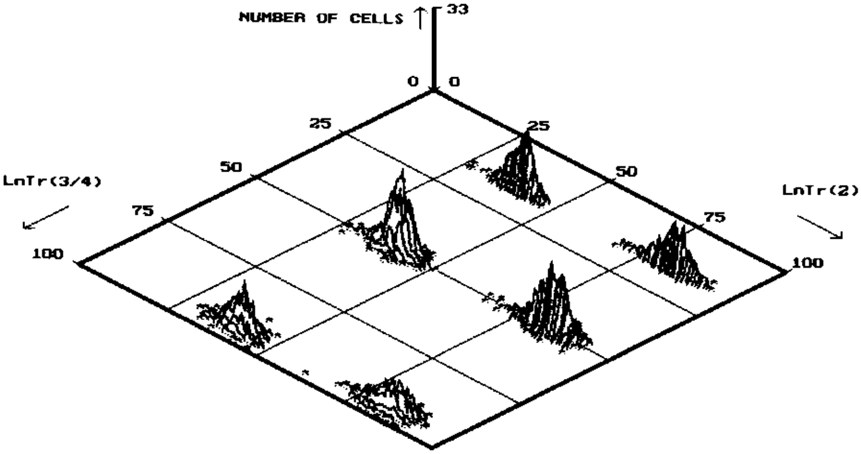

FIG. 4. Two-dimensional histogram (contour plot) of parameters 3 and 4.

are more understandable. As an example of the TIMEPLOT option we used a data file in which the fluorescence spectrum of cells, loaded with 100 $\mu M$ THF-DOX, was measured as function of the measurement order.

Figure 2 shows the six dot plots of the standard output screen of the program ANALYSIS with all possible two-dimensional representations of the artificial data using the four original parameters. The parameters of the plots are given as numbers along the axes.

The dot plots of Fig. 2 show that parameters 3 and 4 are to some extent correlated. Using the DIVIDE command we constructed a new parameter, 10, as parameter 4 divided by parameter three. To enhance the resolution of the low regions of parameters 1–4 we used LOG to construct parameters 11–14 as

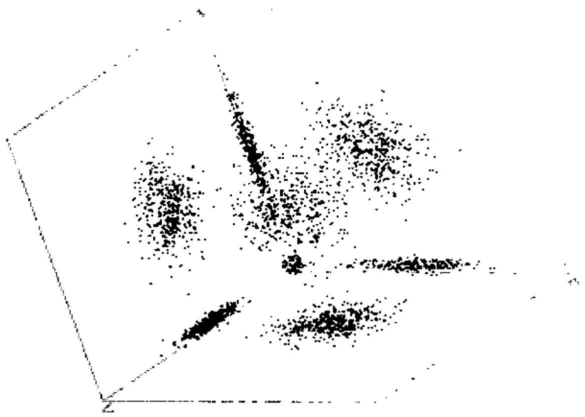

FIG. 5. Three-dimensional representation of the data: $x$ axis, parameter 2; $y$ axis, parameter 3; $z$ axis, parameter 4.

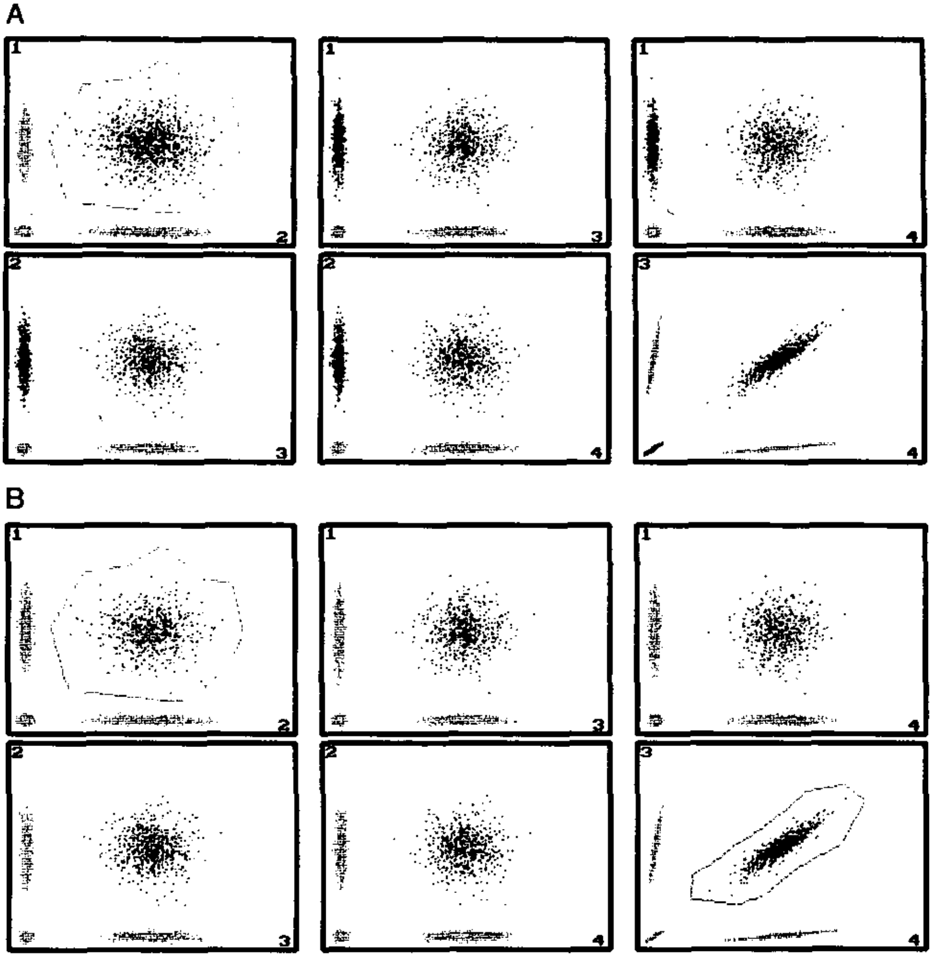

FIG. 6. Selection of a cluster using two windows. (A) Gating with one window. The colored events are not yet a single cluster. (B) Adding a second gate gives a well-separated cluster. Only the events that fall within both windows are selected and colored black.

LnTr(1)–LnTr(4). The dot plots of the new parameters are shown in Fig. 3. In the dot plots of the LOG transformed parameters it is seen that all the clusters have an equal spread in these parameters (because they had an equal relative spread in the original parameters). It is also seen from these plots that parameter 10 now clearly shows three ratios with very small spreads compared to the spread in the original parameters.

The data can also be displayed in three types of two-dimensional histograms. Figure 4 shows one type, a contour plot, of parameters 10 and 12 (bottom left plot in Fig. 3).

A three-dimensional plot is shown in Fig. 5. Parameter 2 is used for the *x*

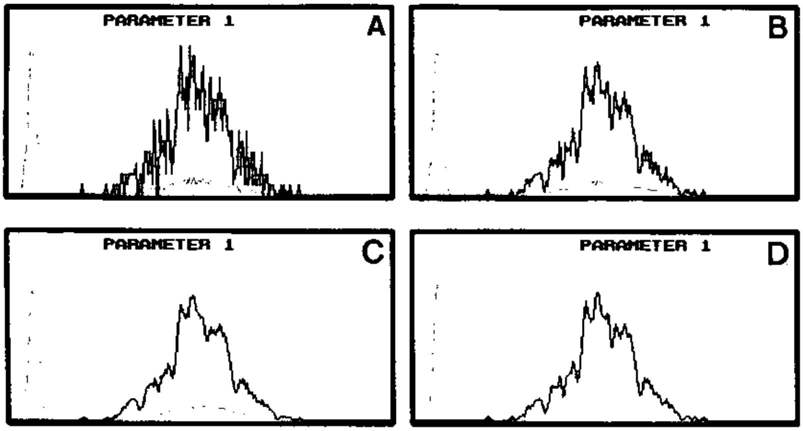

FIG. 7. Histograms of parameter 1 of the data shown in Fig. 1 using different filter constants and numbers of filters in line; the maxima of the histograms of the two clusters are not scaled to each other. (A) No filters; (B) one filter with filter constant 2; (C) on filter with filter constant 4; (D) two filters with filter constant 2.

axis, parameter 3 for the $y$ axis, and parameter 4 for the $z$ axis; in this plot one can see all eight clusters almost separately.

The process of separation of the clusters is shown in Fig. 6. First we gated one population in the first plot (see Fig. 6A). All events that are within this gate are colored black, and the rest are colored gray. It is seen from the other plots that the events in this gate still form two clusters. Therefore we placed a second gate in the sixth plot (see Fig. 6B); the events that lie within both windows now clearly form one cluster. We now have subdivision into two populations; one is truly a single cluster, and the second one consists of the seven other clusters.

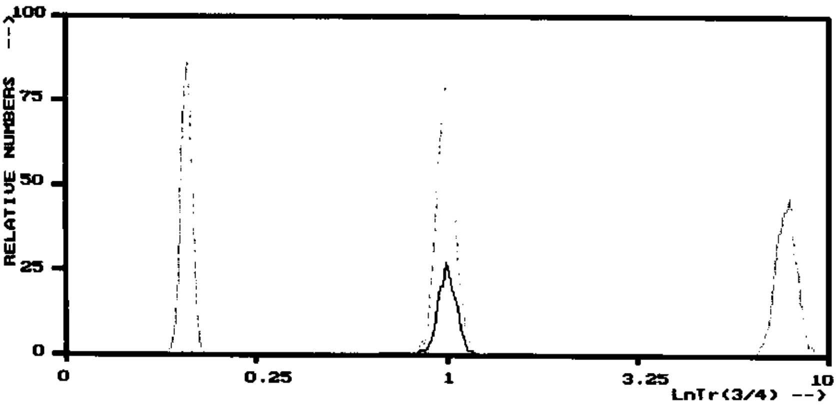

FIG. 8. BIGHIST of parameter 10, the ratio between parameter 3 and parameter 4; the maxima of the histograms of the two clusters are scaled to each other.

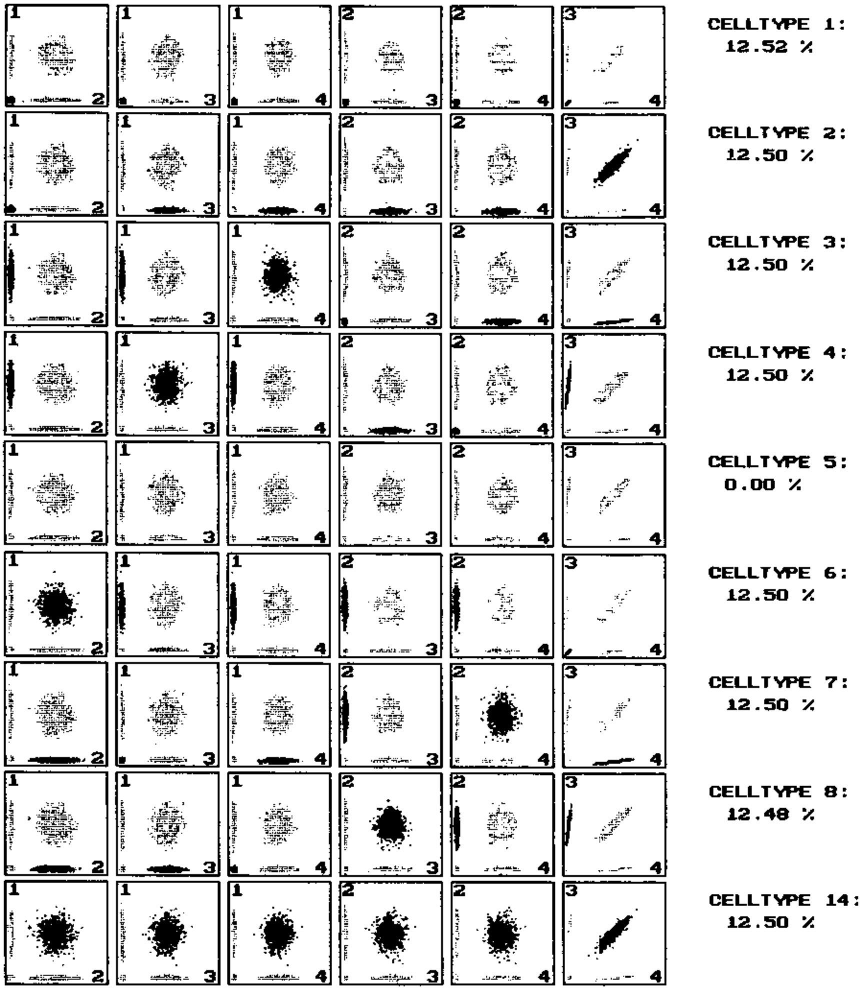

FIG. 9. The eight different clusters that can be discriminated in the data, plotted using miniplots.

We take this data partition to illustrate the single-parameter histogram options. The histograms of parameter 1 of the data shown in Fig. 6B, using different filter constants and number of filters in line, are shown in Fig. 7. We did not scale the two populations with respect to their maxima. Figure 7A shows the histograms of the unfiltered data. Figure 7B shows the two histograms, smoothed with one filter with filter constant 2. In Fig. 7C the histograms are smoothed with two filters with filter constant 2. In Fig. 7D the histograms are smoothed with one filter with a filter constant of 1. It is seen that, in this case, one filter with a filter constant of 2 already gives a good smoothing of the histogram (Fig. 7b).

Figure 8 gives the big histogram for parameter 10, the logarithmical trans-

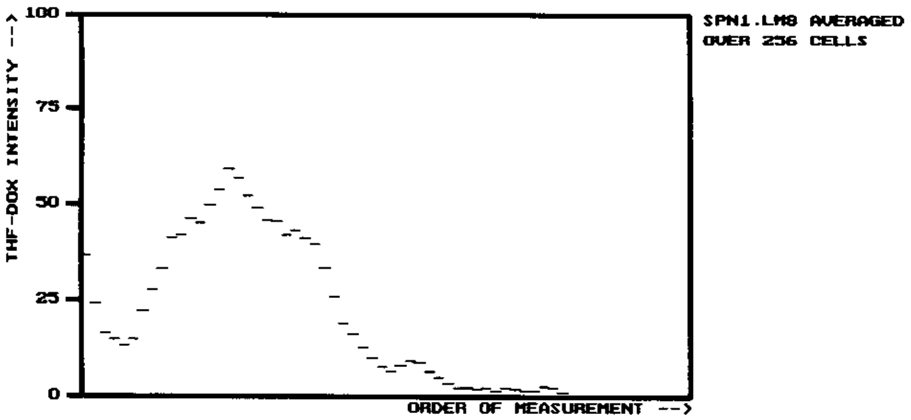

FIG. 10. Time plot of fluorescence peak heights of THF-DOX measured with a monochromator and a detector. The order of measurement is proportional to the wavelength measured. For each wavelength interval 256 cells were measured. The range of the spectrum was 505–836 nm, measured in 50 intervals.

formed ratio of parameters 3 and 4. In this histogram the populations are scaled with respect to maxima. It is seen from this histogram that the cluster, colored black, has a ratio parameter of 1 with a very small spread and that it contains less events than the gray-colored population.

The data, colored gray in Fig. 6B, can be further subdivided into separate clusters, resulting in a partition, shown in Fig. 9 using the MINIPLOTS. From these plots the eight clusters can be clearly discriminated.

The order of measurement can be used for studies of time-dependent processes like dye uptake, dye loss, aging, etc. It can also be used if one wants to measure the spectrum of different cells in a mixed population using a automatically adjustable monochromator and a detector. During the measurements one can adjust the monochromator after a specific number of cells, measure again, and so on. The information gathered in the other parameters can be used afterward to select a certain cell type and only look at the spectrum of that specific cell type. As an example the spectrum of THF-DOX-loaded K562 cells is shown. We measured the spectrum in the range 505–836 nm by measuring 50 times 256 cells and adjusting the monochromator in between. The time plot of the monochromator parameter is shown in Fig. 10. The measured fluorescence peak heights are averaged over 256 cells; the order of measurement is proportional to the wavelength measured.

The speed of the different routines in the program is dependent on the amount of data, the medium used for internal data storage, and the complexity of the windows used. For an indication, the average time to assign 4096 events to 1 of 15 clusters (each window defined using 3 windows) and plot all events on the screen takes about 10 sec on a 40-MHz 80486 IBM-compatible personal computer.

## Conclusions

The program, described in this article, is a very useful tool for the analysis of FCM list mode data. The data structure of the program is created dynamically so that there are no limitations on the amount of events or number of parameters that can be analyzed (minimum of one event and one parameter). Every event takes $n + 2$ bytes memory (2 bytes for cluster information), with $n$ the number of parameters. This means that on a RAM drive of 2 megabytes, one could store more than 200,000 events with eight parameters, which is sufficient in most cases. The separation method implemented in the program is simple and effective. Polygonal windows are easy to use and in principle one can even select separate events (using windows in ZOOM parameters). The creation of new parameters can be useful for compressing and enhancement of the information in the data; division of two parameters has proven to be very useful in ratio measurements of two fluorescent probes (*6*).

## References

*1*. Shapiro, H. M. "Practical Flow Cytometry." Liss, New York, 1985.

*2*. Dean, P. N., Data processing. *In* "Flow Cytometry and Sorting" (M. R. Melamed, P. F. Mullaney, and M. L. Mendelsohn, Eds.), 2nd ed., pp. 415–444. Wiley, New York, 1990.

*3*. Keij, J. F., and Griffioen, A. J. The T. H. and Rijkers G. T. INCA: Software for Consort 30 analysis of flow cytometric calcium determinations. *Cytometry* **10,** 814 (1989).

*4*. Greimers, R., Rongy, A. M., Schaaf-Lafointaine, N., and Boniver, J. CUBIC: A three-dimensional colored projection of Consort 30 generated trivariate flow cytometric data. *Cytometry* **12,** 570 (1991).

*5*. Data file standard for flow cytometry. *Cytometry* **11,** 323 (1990).

*6*. Graft, M. van, Kraan, Y. M., Segers, G. M. J., Radošević, K., de Grooth, B. G., and Greve, J. Flow cytometric measurement of $[Ca^{2+}]_i$ and $pH_i$ in conjugated natural killer cells and K562 target cells during the cytotoxic process. *Cytometry* **14,** 257 (1993).