Simfit

Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
http://www.simfit.org.uk

One of the features SimFiT provides for EPS PostScript files is the ability to manipulate the metric features of the files as they are created, or retrospectively using program **editps**. The following procedures, for instance, are frequently required.

- Changing the aspect ratio and clipping;

- Scaling, rotating, and shearing;

- Adjusting fonts and line thicknesses if graphs are enlarged or reduced; and

- Stretching the white space between items plotted without altering the aspect ratios of characters or symbols.

This is possible because of two special features in SimFiT PostScript files.

1. The BoundingBox and clipping coordinates, as in the typical header section shown next.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox:    72   252   520   588
%%Creator: Simfit Version 7.2.8 (simfit.org.uk)
%%Title: colours=72/ISOLatin1Encoding/Accents/special/PSfrag/dict=300
%%CreationDate: Saturday, 15 April 2017
%%EndComments
%
%Start of SIMFIT PostScript file
%
save %save current state before clipping, etc.
    70   250   522   250   522   590   70   590%#clipping
newpath moveto lineto lineto lineto closepath clip newpath
   72.00  252.00 translate    0.07    0.07 scale    0.00 rotate%#portrait
   12.00 setlinewidth 0 setlinecap 1 setlinejoin [] 0 setdash
    2.50 setmiterlimit
%
```

The BoundingBox holds the overall dimensions of the EPS file in PostScript units (1/72 inches) while the rest defines the clipping and rotating parameters.

2. The special sections indicated by %#, as in the typical data section shown below.

```
c0
1070 671 59 ce%#2
3514 2443 59 ce%#2
5959 4215 58 ce%#2
```

Text following any % sign are taken as comments in PostScript files, so the parameters following the %# are specific SimFiT stretching factors to alter the preceding coordinates. These stretch white space and alter the starting coordinates for symbols, lines, and character strings but not the inter-word separation, so preserving the aspect ratio of fonts.

Of course the brave may wish to alter these themselves using a text editor but, as this is very tedious and error-prone, the SimFiT package provides functions to perform the editing

Alternative sizes, shapes and clipping

Plots can have horizontal, square or vertical format as in the next figure, and user-defined clipping schemes can be used. After clipping, SᴍFɪT adds a standard BoundingBox so all plots with the same clipping scheme will have the same absolute size but, when **GSview/Ghostscript** transforms ps into eps, it clips individual files to the boundary of white space and the desirable property of equal dimensions will be lost.
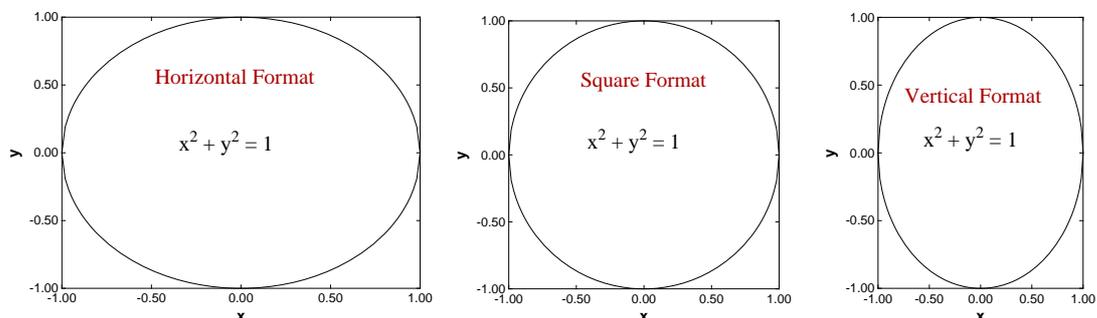
There is also a stretched format for long horizontal ribbon type graphs and a PostScript option to stretch white space without altering symbols and fonts, which is very useful with dense data sets such as dendrograms.

Rotated and re-scaled graphs

PostScript files can be read into **editps** which has options for re-sizing, re-scaling, editing, rotating, making collages, etc. In the next figure the box and whisker plot was turned on its side to generate a side-on barchart. To do this sort of thing you should learn how to browse a SᴍFɪT PostScript file in the SᴍFɪT viewer to read BoundingBox coordinates, in PostScript units of 72 to one inch, and calculate how much to translate, scale, rotate, etc.
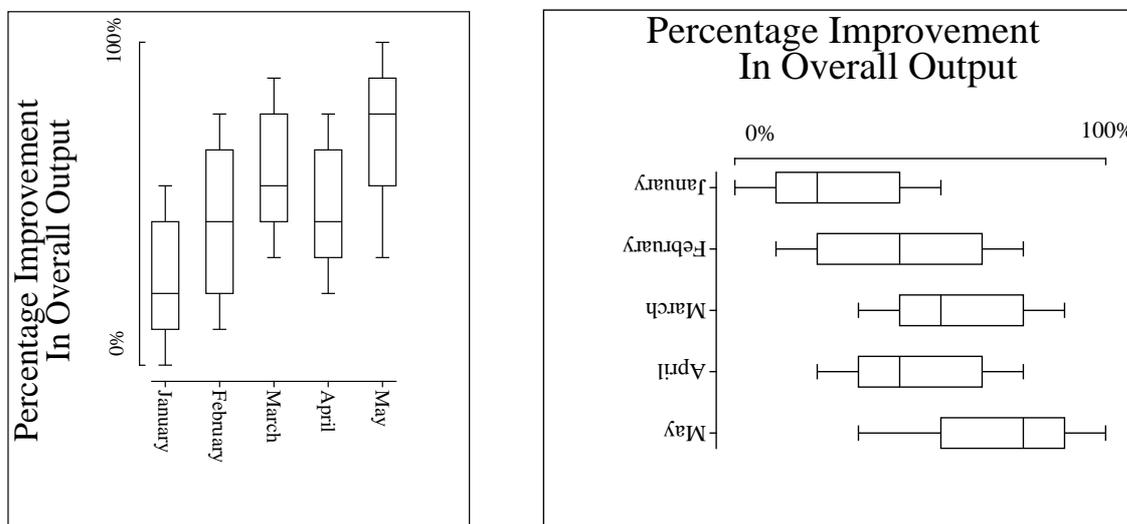

PostScript users should be warned that the special structure of SᴍFɪT PostScript files that allows extensive retrospective editing using **editps**, or more easily if you know how using a simple text editor like **notepad**, is lost if you read such graphs into a graphics editor program like Adobe Illustrator. Such programs start off by redrawing vector graphics files into their own conventions which are only machine readable.

Changed aspect ratios and shear transformations

The barchart in the figure below was scaled to make the X-axis longer than the Y-axis and vice-versa, but note how this type of differential scaling changes the aspect ratio as illustrated. Since rotation and scaling do not commute, the effect created depends on the order of concatenation of the transformation matrices. For instance, scaling then rotation cause shearing which can be used to generate 3-dimensional perspective effects as in the last sub-figure.

Bar Chart Overlaps, Groups and Stacks
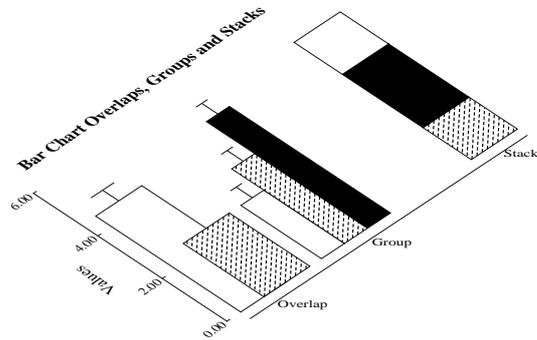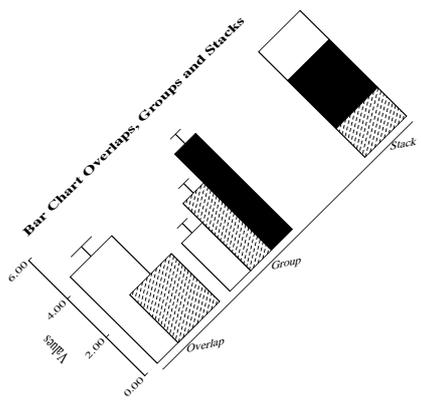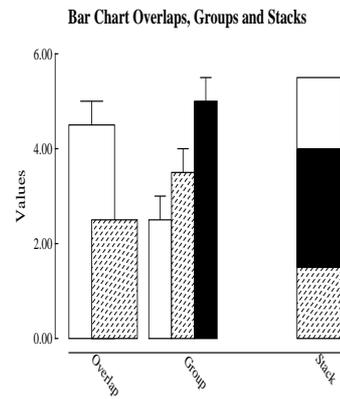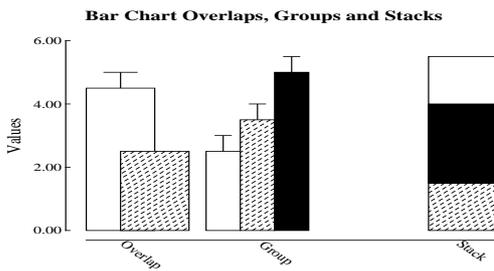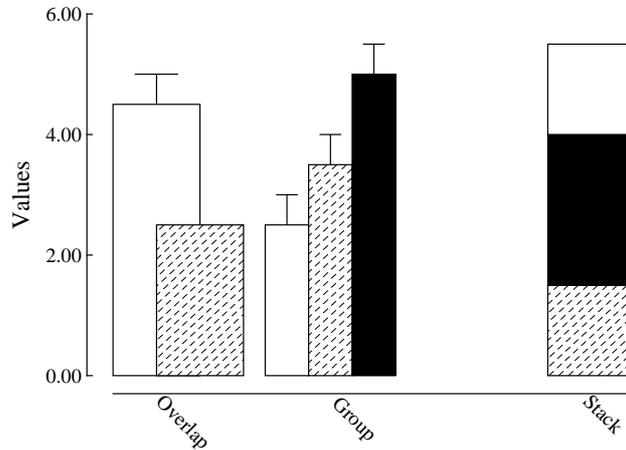
Reduced or enlarged graphs
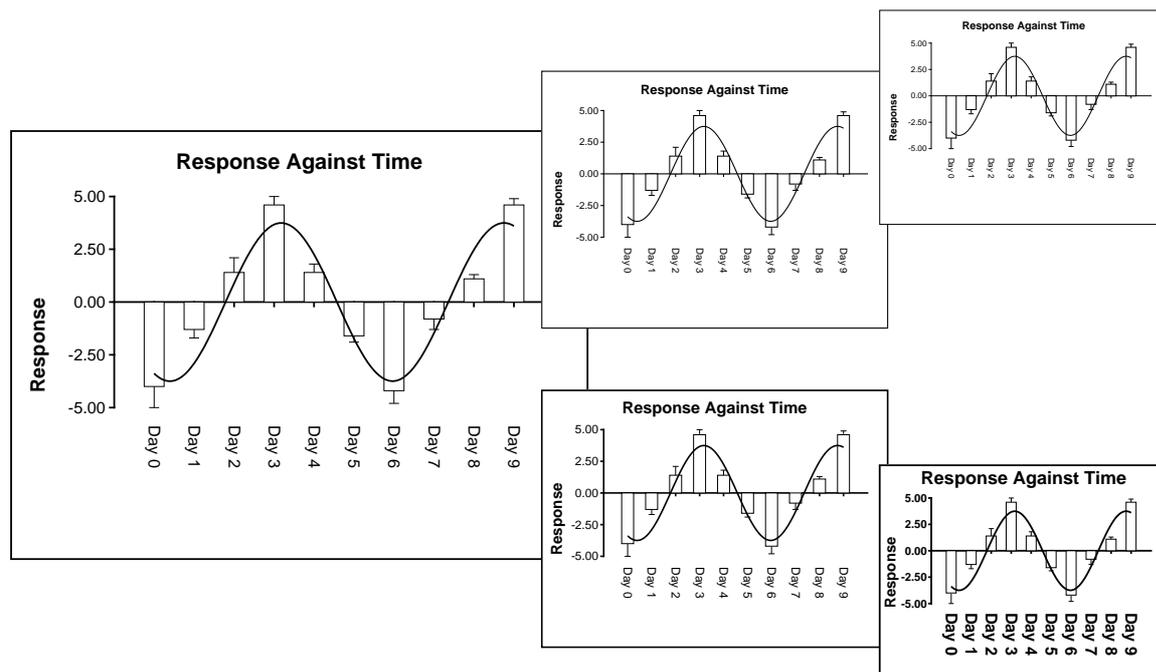
It is always valuable to be able to edit a graph retrospectively, to change line or symbol types, eliminate unwanted data, suppress error bars, change the title, and so on. SɪᴍFɪT PostScript files are designed for just this sort of thing, and a typical example would be altering line widths and font sizes as a figure is re-sized.

In the figure above the upper sub-figures are derived from the large figure by reduction, so the text becomes progressively more difficult to read as the figures scale down.

In the lower sub-figures, however, line thicknesses and font sizes have been increased as the figure is reduced, maintaining legibility. Such editing can be done interactively, but SɪᴍFɪT PostScript files are designed to make such retrospective editing easy as described in the w_readme.* files and now summarized.

- Line thickness: Changing 11.00 setlinewidth to 22 setlinewidth doubles, while, e.g. 5.5 setlinewidth halves all line thicknesses, etc. Relative thicknesses are set by **simplot**.

- Fonts: Times-Roman, Times-Bold, Helvetica, Helvetica-Bold (set by **simplot**), or, in fact, any of the fonts installed on your printer.

- Texts: ti(title), xl(x legend), yl(y legend), tc(centered for x axis numbers), tl(left to right), tr(right to left), td(rotated down), ty(centered for y axis numbers).

- Lines: pl(polyline), li(line), da(dashed line), do(dotted line), dd(dashed dotted).

- Symbols: ce(i.e. circle-empty), ch(circle-half- filled), cf(circle-filled), and similarly for triangles(te, th, tf), squares(se, sh, sf) and diamonds(de, dh, df). Coordinates and sizes are next to the abbreviations to move, enlarge, etc.

If files do not print after editing you have probably added text to a string without padding out the key. Find the fault using the **GSview/Ghostscript** packages then try again.

Plotting combined meta analysis results
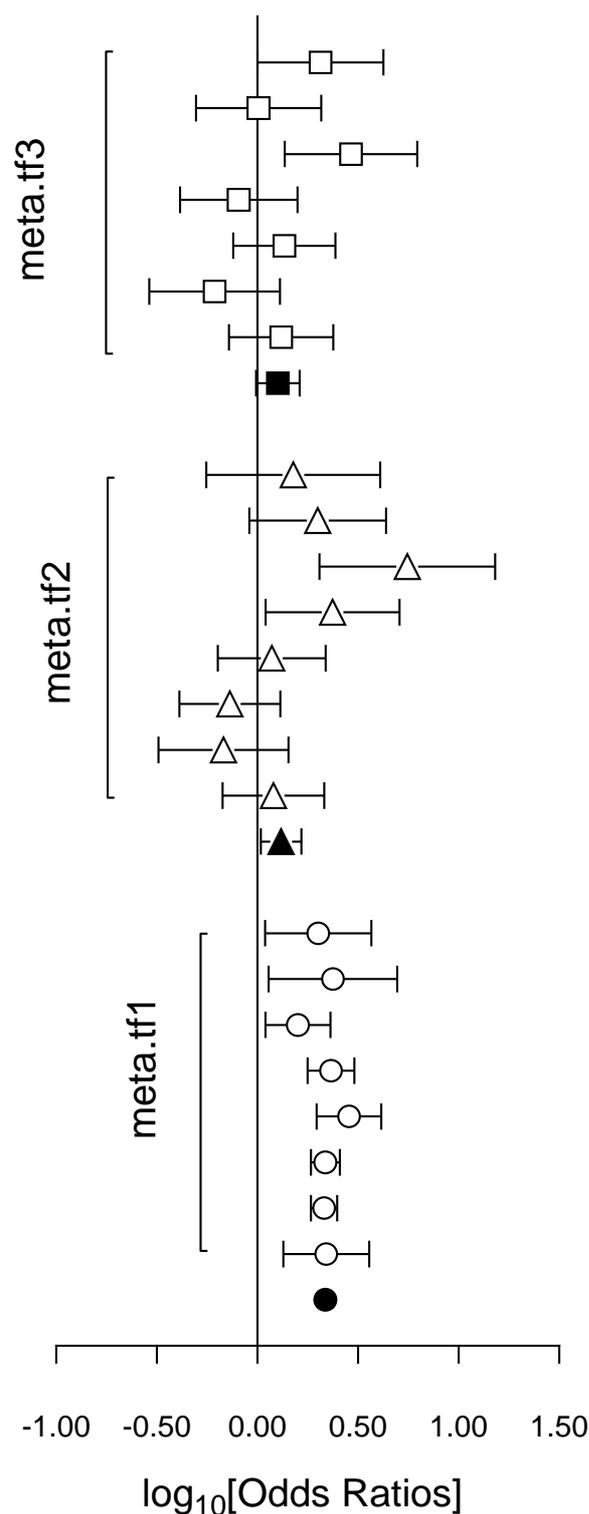
$$\log_{10}[\text{Odds Ratios}]$$

It is often useful to plot Log-Odds-Ratios, so the creation of the adjacent figure will be outlined.

(1) The data
Test files `meta.tf1`, `meta.tf2`, and `meta.tf3` were analyzed in sequence using the SIMFIT Meta Analysis procedure. Note that, in these files, column 3 contains spacing coordinates so that data will be plotted consecutively.

(2) The ASCII coordinate files
During Meta Analysis, $100(1-\alpha)\%$ confidence limits on the Log-Odds-Ratio resulting from a 2 by 2 contingency tables with cell frequencies n_{ij} can be constructed from the approximation $\hat{e}$ where

$$\hat{e} = Z_{\alpha/2}\sqrt{\frac{1}{n_{11}} + \frac{1}{n_{12}} + \frac{1}{n_{21}} + \frac{1}{n_{22}}}.$$

When Log-Odds-Ratios with error bars are displayed, the overall values (shown as filled symbols) with error bars are also plotted with a x coordinate one less than smallest x value on the input file. For this figure, error bar coordinates were transferred into the project archive using the [Advanced] option to save ASCII coordinate files.

(3) Creating the composite plot
Program **simplot** was opened and the six error bar coordinate files were retrieved from the project archive. Experienced users would do this more easily using a library file of course. Reverse y-semilog transformation was selected, symbols were chosen, axes, title, and legends were edited, then half bracket hooks identifying the data were added as arrows and extra text.

(4) Creating the PostScript file
Vertical format was chosen then, using the option to stretch PostScript files, the y coordinate was stretched by a factor of two.

(5) Editing the PostScript file
To create the final PostScript file for LaTeX a tighter bounding box was calculated using **gsview** then, using **notepad**, clipping coordinates at the top of the file were set equal to the BoundingBox coordinates, to suppress excess white space. This can also be done using the [Style] option to omit painting a white background, so that PostScript files are created with transparent backgrounds, i.e. no white space, and clipping is irrelevant.

Plotting dendrograms: standard format

Dendrogram shape is arbitrary in two ways; the x axis order is arbitrary as clusters can be rotated around any clustering distance leading to 2^{n-1} different orders, and the distance matrix depends on the settings used. For instance, a square root transformation, Bray-Curtis similarity, and a group average link generates the second dendrogram in this figure from the first. The y plotted are dissimilarities, while labels are $100 - y$, which should be remembered when changing the y axis range.

Users should not manipulate dendrogram parameters to create a dendrogram supporting some preconceived clustering scheme. You can set a label threshold and translation distance from the [X-axis] menu so that, if the number of labels exceeds the threshold, even numbered labels are translated, and font size is decreased.
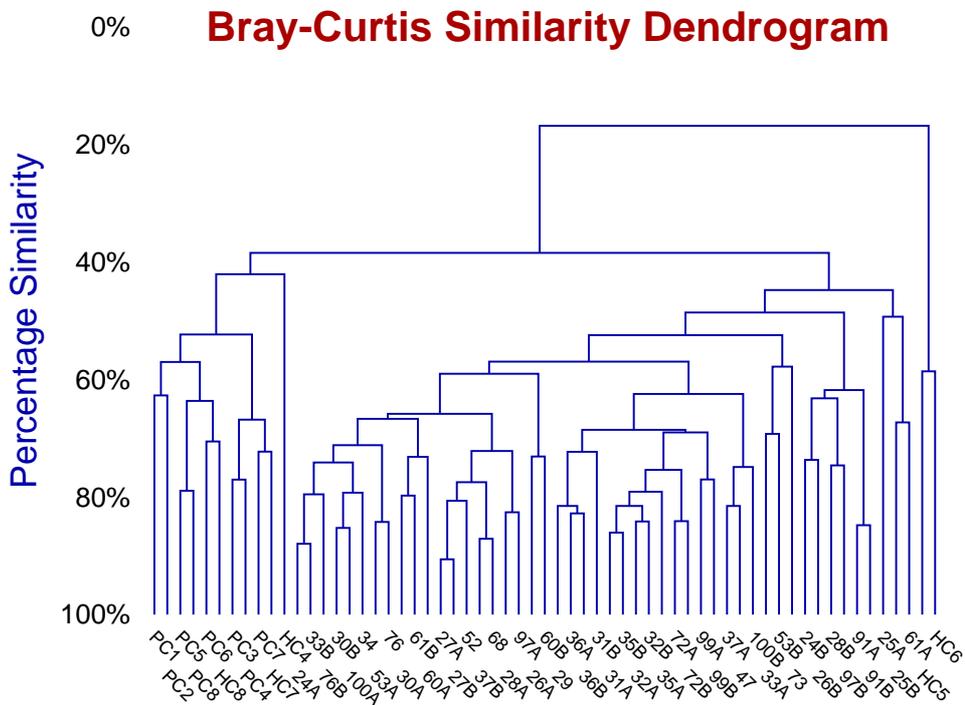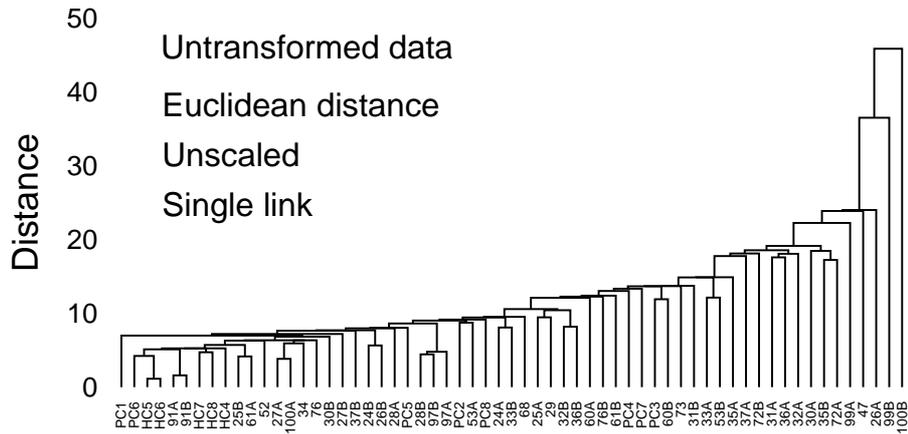
Plotting dendrograms: stretched format

Sometimes dendrograms are more readable if the white space is stretched without distorting the labels.
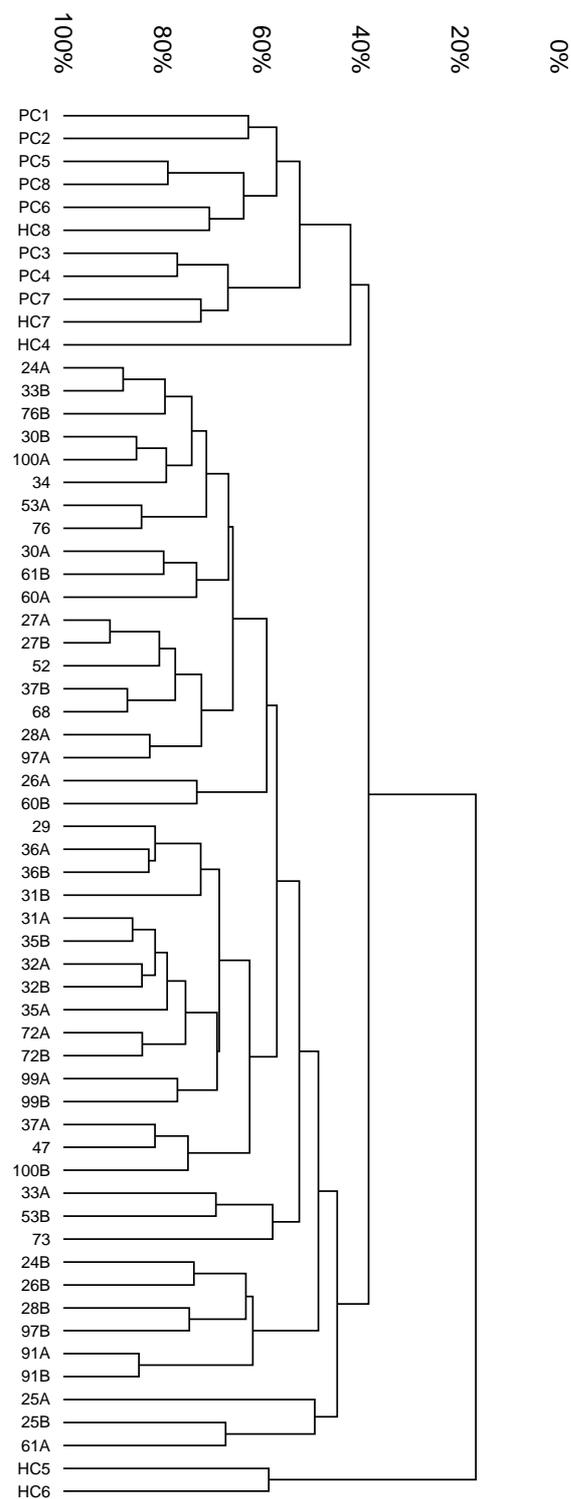
So SimFIT PostScript graphs have a very useful feature: you can stretch or compress the white space between plotted lines and symbols without changing the line thickness, symbol size, or font size and aspect ratio. For instance, stretching, clipping and sliding procedures are valuable in graphs which are crowded due to overlapping symbols or labels, as in previous figures. If such dendrograms are stretched retrospectively using **editps**, the labels will not separate as the fonts will also be stretched so letters become ugly due to altered aspect ratios. SimFIT can increase white space between symbols and labels while maintaining correct aspect ratios for the fonts in PostScript hardcopy and, to explain this, the creation of this stretrched figure will be described.

The title, legend and double x labeling were suppressed, and landscape mode with stretching, clipping and sliding was selected from the PostScript control using the [Shape] then [Landscape +] options, with an x stretching factor of two. Stretching increases the space between each symbol, or the start of each character string, arrow or other graphical object, but does not turn circles into ellipses or distort letters. As graphs are often stretched to print on several sheets of paper, sub-sections of the graph can be clipped out, then the clipped sub-sections can be slid to the start of the original coordinate system to facilitate printing.

If stretch factors greater than two are used, legends tend to become detached from axes, and empty white space round the graph increases. To remedy the former complication, the default legends should be suppressed or replaced by more closely positioned legends while, to cure the later effect, GSview can be used to calculate new BoundingBox coordinates (by transforming .ps to .eps). If you select the option to plot an opaque background even when white (by mistake), you may then find it necessary to edit the resulting .eps file in a text editor to adjust the clipping coordinates (identified by %#clip in the .eps file) and background polygon filling coordinates (identified by %#pf in the .ps file) to trim away unwanted white background borders that are ignored by GSview when calculating BoundingBox coordinates. Another example of this technique is with meta analysis plots, where it is also pointed out that creating transparent backgrounds by suppressing the painting of a white background obviates the need to clip away extraneous white space.

The procedures described can also be used to improve the readability of dendrograms where subgroups have been assigned by partial clustering. The next figure shows a graph from `iris.tf1` when three subgroups are requested, or a threshold is set corresponding to the horizontal dotted line. The figure was created by these steps.

First the title was suppressed, the y-axis range was changed to $(0, 4.25)$ with 18 tick marks, the (x, y) offset was canceled as this suppresses axis moving, the label font size was increased from 1 to 3, and the x-axis was translated to 0.8.

Then the PostScript stretch/slide/clip procedure was used with these parameters

$$x_{\text{stretch}} = 1.5$$
$$y_{\text{stretch}} = 2.0$$
$$x_{\text{clip}} = 0.15, 0.95$$
$$y_{\text{clip}} = 0.10, 0.60.$$
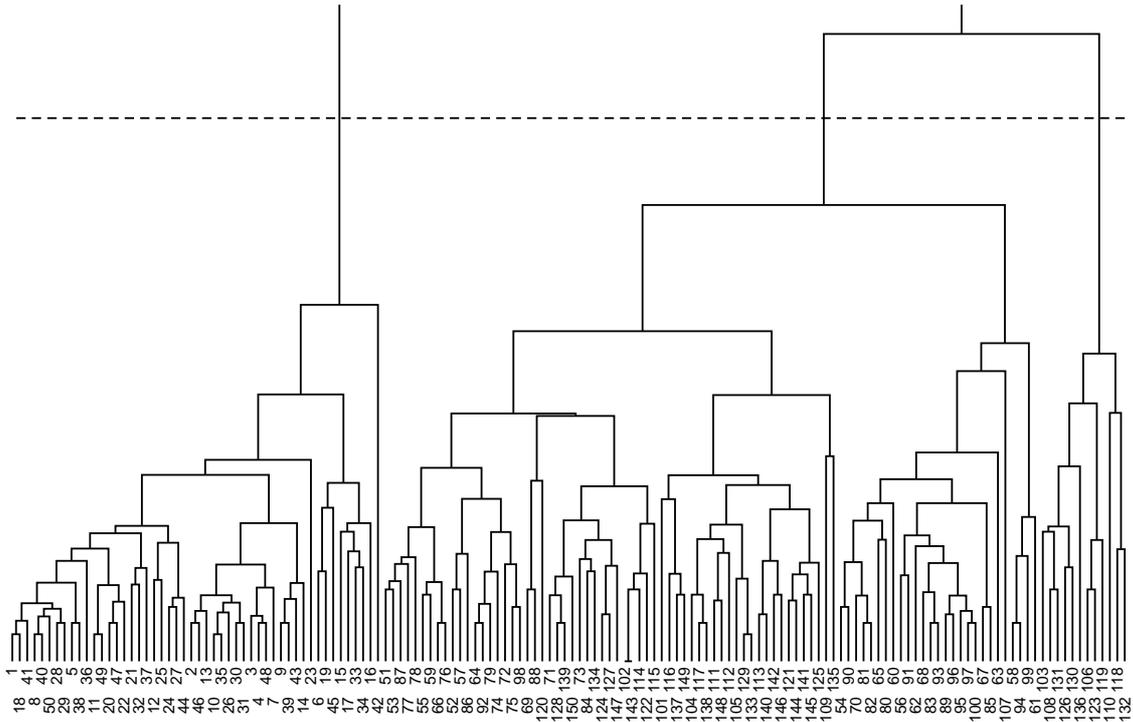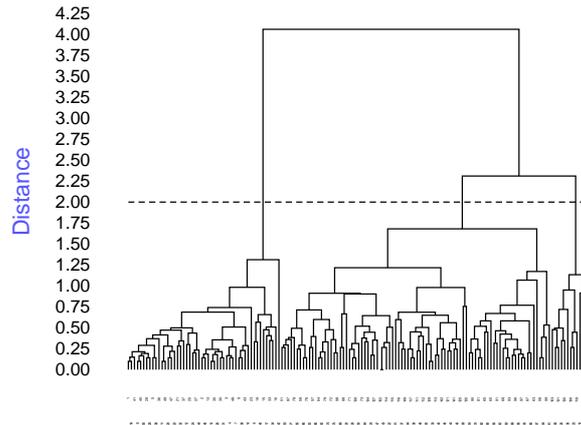
Windows users without PostScript printing facilities must create a `*.eps` file using this technique, then use the SimFiT procedures to create a graphics file they can use, e.g. `*.jpg`. Use of a larger font and increased x-stretching would be required to read the labels, of course.

8