The Dendrogrammer
A Cross-Browser, Cross-Platform, Web Application to Generate Interactive Dendrograms from Clustering Data

David A. Robb
H00013215
MSc Report

August 2011
Statement of non-plagiarism

DECLARATION:

I, David Allan Robb, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ..................................................................

Date: ......................................................................
Abstract

An interactive web-based application to render dendrograms from clustering data is to be created. The application is to allow interested parties and researchers to visualise and interact with word co-occurrence clustering data from the EPSRC Grants on the Web. This interactive visualisation of the data will allow grouping of the grants in a logical and transparent way such that effective consultation groups can be formed. The application should allow visualisation and interaction with other similar clustering data. The application exploits the latest web technologies, including the Raphael JavaScript graphics library, to allow cross browser and cross platform access including iPad. The first version of the application (The Dendrogrammer) has been installed and used as part the web site in the developing EPSRC project, ICT Perspectives (an ICT Next Decade proposal).

Dendrograms and their role in cluster analysis are described. Some of the background to the analysis that produces the project clustering data, and also the possible technologies that might be used to create the application are investigated. The requirements are analysed, technologies selected, and prototyping by progressive development is used to create the Dendrogrammer. The key stages, problems and solutions in the development are related. The structure and key parts of the code are described. The Dendrogrammer is evaluated and its strengths and shortcomings are exposed. Suggestions for improvement are made.
Acknowledgements

- Professor M. Chantler for his supervisor role, his role as ‘Client’, and for suggesting the method of dendrogram construction.
- F. Halley, of the Heriot-Watt Texture Lab, for his explanation of how MATLAB clustering data is interpreted as a dendrogram, for providing some test data, for advising on the format of the evaluation, and for his role as a client.
- T. Methven, of the Heriot-Watt Texture Lab, for providing some test data and his help when testing on the department’s Android slate.
- Dr J. Wells, my second reader, for advice.
- The participants in the evaluation exercise.
- V. Venkatachalam for showing me the native zoom capability of web browsers.
- A. Robb (my wife) for proofreading.
- PHP.net referred to for PHP file read methods.
- Dimity Barinovsky for his ‘Graffle’ example at http://raphaeljs.com/, which gave me the clues that I needed on how to implement cross-platform drag-able objects, and for creating the Raphael JavaScript graphics library.

Note on References

Some areas investigated in this report are the subject of recent innovation and change. This particularly applies to some of the JavaScript libraries. Also, in researching existing application software, the documentation of these applications is often only published and available online. In these areas I have found it necessary to rely on some unpublished sources (i.e. published only on the World Wide Web). Therefore references are provided in two separate sections, published and unpublished sources. (See References).
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A statement of non-plagiarism</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Acknowledgements</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Note on references</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Contents (this page)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>List of appendices</td>
<td>6</td>
</tr>
<tr>
<td>Section 1</td>
<td>Introduction</td>
<td>7</td>
</tr>
<tr>
<td>Section 2</td>
<td>Existing dendrogram applications</td>
<td>15</td>
</tr>
<tr>
<td>Section 3</td>
<td>Literature review</td>
<td>22</td>
</tr>
<tr>
<td>Section 4</td>
<td>Requirements analysis</td>
<td>38</td>
</tr>
<tr>
<td>Section 5</td>
<td>Tool selection</td>
<td>50</td>
</tr>
<tr>
<td>Section 6</td>
<td>Software development methodology</td>
<td>55</td>
</tr>
<tr>
<td>Section 7</td>
<td>Key stages in the development of the application</td>
<td>57</td>
</tr>
<tr>
<td>Section 8</td>
<td>Problems and solutions. (Approaches adopted.)</td>
<td>67</td>
</tr>
<tr>
<td>Section 9</td>
<td>The structure of the application</td>
<td>69</td>
</tr>
<tr>
<td>Section 10</td>
<td>The code</td>
<td>73</td>
</tr>
<tr>
<td>Section 11</td>
<td>Evaluation</td>
<td>86</td>
</tr>
<tr>
<td>Section 12</td>
<td>Licensing.</td>
<td>105</td>
</tr>
<tr>
<td>Section 13</td>
<td>Conclusion.</td>
<td>106</td>
</tr>
</tbody>
</table>

# References

<table>
<thead>
<tr>
<th>Type</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Published sources</td>
<td>107</td>
</tr>
<tr>
<td>Unpublished sources</td>
<td>110</td>
</tr>
<tr>
<td>Bibliography</td>
<td>112</td>
</tr>
</tbody>
</table>
Appendices

Appendix 1 - The GoW clustering data.

Appendix 2 – Preliminary Investigations.
- Protovis
- SVG
- jQuery UI
- Raphael

Appendix 3 – Planning:
- Professional, legal, and ethical issues
- Planned steps and schedule
- Risk plan

Appendix 4 – Prototype Development – Screen shots of the prototypes by date.

Appendix 5 – Project Diary/Blog – Printed version of the project blog and reference to the full online blog.

Appendix 6 – Code appendix – JavaScript code, PHP code, CSS code.

Appendix 7 – Testing
- Records of formal testing including records of testing in different browsers.
- Example of how large simulation data was constructed in spreadsheets.

Appendix 8 – Documentation
- Technical Guide. (For user guide see help tab in the application)
- Readme file
- Reference to the Code Documentation web pages.

Appendix 9 – Evaluation
- Text of the evaluation task
- Closed question response collation
- Open question response collation
Introduction

(Section 1)

Figure 1.1 - A dendrogram\(^1\) created from a clustering analysis of word co-occurrence in grant applications using MATLAB.

[Created by Tom Methven of the Texture Lab\(^2\)]

In this introduction I will describe
- The motivation for the project.
- How dendrograms are used as an analysis tool.
- The goal of the project.

---

\(^1\) A dendrogram is a tree diagram where all the leaf nodes are arranged adjacent to each other at the base (or one end) of the diagram.

The motivation for the project.

The EPSRC (Engineering and Physical Sciences Research Council) funds research to the value £3.5 thousand million\(^3\). It is interested in planning and prioritising this expenditure looking ahead into the future. To help it do this the EPSRC wishes to develop a better understanding of its grants portfolio (the rationale being if one wishes to plan one’s route it is best to know where one is starting from.)

At the moment the EPSRC publishes its grants portfolio on the Grants on the Web (GoW) web site. This provides a comprehensive “view” of the portfolio. However, as a visualisation, it might be comprehensive but it does not fully aid comprehension of the portfolio. The grant applications are held in a database so, in theory, this can be searched based on the fields in the database. Indeed the GoW site offers a specific search on fields concerned with named researchers allowing grants associated with that researcher to be displayed. It also offers a keyword search. This produces a list of matches ranked by relevance related to the keywords entered. However, the EPSRC wish alternative ways of visualising the portfolio and the relationships between the research projects it funds.

With this in mind EPSRC have established a project, ICT Perspectives (an ICT Next Decade proposal). The work is to be carried out at Heriot-Watt University. It has two main aims\(^4\):

1. To produce overviews of the grant portfolio using text mining and visualisation methods such as word clouds. These visualisations can be used as an alternative method of accessing the GoW.
2. To facilitate “networking”, or brainstorming workshops to inform EPSRC strategic planning.

The work is initially planned to focus in the area of ICT (Information and Communications Technology). However, once the principles have been shown to work it may be applied across the full portfolio. (Indeed, in anticipation of this the web site, at time of writing, in development for the ICT Perspectives project is being termed Research Perspectives.)

---

\(^3\) See the GoW site http://gow.epsrc.ac.uk, accessed 8/3/2011

\(^4\) [EPSRC – ICT Perspectives] Unpublished source.
If effective brain storming sessions are to be organised, representatives of the various research labs need to be formed into groups. The formation of these groups needs to be impartial, transparent and logical. When a research lab applies for a grant it provides a text description of its application. These text descriptions contain descriptions of the work involved in the projects. If the descriptions of two grants share many words in common (ruling out words such as “the”, “and” etc.) then it is likely they represent related areas of research (See Section 3 - Literature Review). As the EPSRC makes details of its grants available on the GoW web site, the text descriptions are readily available for analysis. (See Figure 1.2 below).

![Figure 1.2 – One "Grant on the Web" showing text description (the source of the project data) at the bottom.](image)

Some work on visualising the GoW data has already been done in the Texture Lab at Heriot-Watt University. That work involved analysing the text data, producing clustering data from it and depicting the GoW data in dendrograms produced in MATLAB\(^6\) and in word clouds. (See Section 3, Literature Review).

---


\(^6\) MATLAB is software for analysing numerical data (See Section 3, Literature Review.)
How dendrograms are used as an analysis tool.

Objects can be analysed for similarity to each other. (E.g. different species of flower or gas clouds in astronomy or molecules in chemistry). When objects have been analysed for similarity to each other they can be classified into groups or clusters of those objects which share a certain amount of similarity. The definition of these groups or clusters is often achieved by cluster analysis (See Section 3.2, Showing structure in data sets and Spath’s\(^7\) definition of cluster analysis quoted therein).

Groups thus classified and identified can be singled out for further study.

Examples of the fields of study and type of objects that this has been done for are:

- In numerical taxonomy in biology, to classify groups of organisms\(^8\).
- In astrophysics, to classify and analyse molecular clouds\(^9\).
- In archaeology, to classify finds.\(^10\)
- In information science, to classify web pages\(^11\).
- Another information science example is the classification of documents to facilitate navigation\(^12\).

Dendrograms are a particular form of tree diagram used to visualise the relatedness of the objects and their clusters once the cluster analysis has been done (a cluster being a group of related objects).

Martinez and Martinez\(^13\) have given a good definition of a dendrogram, in which they refer to the level of dissimilarity between objects as the distance between them:

---

\(^7\) [Spath 1980] p7.
\(^8\) [Sokal and Sneath 1973]
\(^9\) [Rosolowsky et al 2008]
\(^10\) [Hartigan 1975] p5
\(^11\) [Vaughan and You 2010]
\(^12\) [Allen et al 1993]
\(^13\) [Martinez and Martinez 2002]p371
“A dendrogram shows the links between objects as inverted U-shaped lines, where the height of the U represents the distance between the objects.”

An example dendrogram is shown in figure 1.3 below. It shows a comparison of seven objects. The objects (often termed *leaves*) are arranged along the horizontal axis. The objects are given numbers for reference. The numbers would key to the names of the objects (e.g. in an archaeological example number 4 might be, say, “Burial 73b, north end of church yard”, while number 5 would be another burial and so on). The vertical axis describes the dissimilarity measure. It is an arbitrary scale that is used to define the relative dissimilarity between the leaves based on the data that was used to compare them.

![Dendrogram Diagram](image)

**Figure 1.3** An example dendrogram showing the relatedness of the objects (or leaves).

In our example dendrogram the two most similar (or least dissimilar) leaves are numbers 4 and 2 as they are joined at the lowest height on the dendrogram. We can
also see that 4 and 2 share more in common with no.1 than they do with 3 and 5. We can also see that 6 and 7 share more in common with each other than they do with all the other leaves on the dendrogram. However, such interpretation is made clearer when a threshold level of dissimilarity is drawn across the dendrogram. In figure 1.4 below we see the same analysis, that 6 and 7 share more in common with each other than they do with the rest of the leaves, is described using a dissimilarity threshold to divide the dendrogram.

![Dendrogram with threshold](image)

**Figure 1.4** The example dendrogram with a threshold applied dividing the dendrogram into two clusters

With the threshold set at an approximate height of 0.8 on our dissimilarity scale we have divided the leaves into two clusters. The resulting clusters consist of leaves 4,2,1,3 and 5 in one cluster and leaves 7 and 6 in the other. These two groupings are

---

14 This use of a dendrogram to suggest separate clusters of the objects being analysed, is described by Everitt and Dunn [Everitt and Dunn 2001] (p138). An early example of the application of a threshold to a dendrogram can be found in Sokal and Sneath [Sokal and Sneath 1973] (p295) where the dendrogram is termed a phenogram and the thresholds are termed phenom lines as they are describing biological phenotypes in numerical taxonomy.
shown graphically in figure 1.5 below. The clusters have been labelled *Cluster A* and *Cluster B*. In terms of the hierarchy of dissimilarity shown in the dendrogram *leaves 1 to 5* are all descendants of *Cluster A* and *leaves 6 and 7* are descendants of *Cluster B*.

So, once cluster analysis has been applied to a group of objects, using a dendrogram those objects can be divided into groups of more or less related objects. Thus, if the objects were the GoW research grants they could be classified or grouped. The kind of analysis and source data for this are discussed in the Section 3, Literature Review and Section 4, Requirements Analysis.

**The goal of the project.**

In this project I hope to add to the work already done here in the Texture lab and contribute to the recently commenced *ICT Perspectives* work by creating a web-based, cross-browser, cross-platform, interactive dendrogram creation application. It
is intended that *The Dendrogrammer* tool will become part of the visualisation tools for the *ICT Perspectives* site. The dendrograms produced using the application will yield information on clusters of GoW (research projects). This information can be used to visualise the clusters of grants (and projects) and also to help form the brainstorming groups described above. It will do this in a way that is less labour intensive and time-consuming than the methods used so far for this.

The Dendrogrammer will be required to be web-based so as to be part of the *ICT Perspectives* site. It needs to be interactive to facilitate analysis of the data and allow such analysis to be done in timely fashion that is not too labour intensive. It should also be cross-browser and cross-platform (i.e. operate on PC, laptop and iPad) so as not to limit access. The fact of these requirements and the graphical nature of the application has posed some challenges. Indeed, the fact that no other cross-browser interactive dendrogram application has been sourced, despite there being an extensive range of computer dendrogram applications (see Section 3.1 Literature Review), is perhaps a clue to the perceived extent of those challenges until now.

The application may also be useful to anyone wishing to visualise clustering data. With the ubiquity of the World Wide Web it should be a particularly accessible tool.
Existing dendrogram and tree diagram applications

(Section 2)

There are many applications that offer a dendrogram creation capability. This survey does not include every single dendrogram application (there appear to be a surprising number of those). Instead it includes examples of various non-web-accessible dendrogram applications. However, it does include all of the web-accessible dendrogram applications that I found.

- JMP\(^{15}\) provides data mining solutions in the form of desktop software which includes a clustering dendrogram capability. JMP is a commercial program, which does the clustering, as well as visualisation in dendrograms that offer a facility to “zoom in to focus on interesting clusters”. It used in the analysis of drug clinical trials. It requires installation on the computer planned for its use. It requires input file formats that are proprietary (JMP and SAS data set. SAS being another product associated with JMP.)

- There are a large number of phylogeny\(^{16}\) programs catalogued at the PHYLIP\(^{17}\) web site. PHYLIP is a suite of programs for use in phylogeny. The PHYLIP suite itself is free for academic use. They all require installation on the operating computer. None produce an interactive dendrogram. Outside the PHYLIP site some of the PHYLIP dendrogramming programs are accessible on web servers. One such is “Phylogenetic Tree Plot” at the Wageningen Bioinformatics Webportal (http://www.bioinformatics.nl/tools/plottree.html ). It presents the “Drawtree” and “Drawgram” programs accessible through a web form. It requires input in Newick format.\(^{18}\)

---


\(^{16}\) Phylogenetics is the study of the relatedness of biological organisms based on molecular sequencing. [Edwards and Cavalli-Sforza 1964]


\(^{18}\) The Newick tree format is a standard for encoding tree diagrams specifically to be computer readable. There is no published source for this definition other than at
I was able to use this over the web by pasting a Newick format, tree definition into the web site form. I used one of Felsenstein’s examples of a tree declared in Newick form in his page on how to use Drawtree and Drawgram\textsuperscript{19}:

```
"((raccoon:19.19959,bear:6.80041):0.84600,(sea_lion:11.99700,
  seal:12.00300):7.52973,(monkey:100.85930,cat:47.14069):20.59201,
```

This resulted in the tree diagram shown in the figure below.

![Phylogenetic Tree Plot](http://www.bioinformatics.nl/tools/plottree.html)

Figure 2.1 shows a “Phylogenetic Tree Plot” produced at the Wageningen Bioinformatics Webportal\textsuperscript{20}. [Screenshot from http://www.bioinformatics.nl/tools/plottree.html]

\textsuperscript{20}[Wageningen Bioinformatics Webportal 2011] Unpublished source
This, while being a tree diagram is not the dendrogram form that is aimed for in this project (despite the site’s web form having a number of options for specifying the shape and labelling style none was the classic dendrogram.). However, I include it here as an example of a specific, tree diagram definition format being used to create a tree diagram over the web.

In addition, the PHYLIP site catalogues phylogeny software that is not part of the PHYLIP suite. Some of the programs catalogued are aimed at producing dendrograms.

- **DendroPy** \(^{21}\) is an example of one of those applications catalogued at the PHYLIP site but not part of the PHYLIP suite itself. Actually, it is described as a library for phylogenetic computing. The output the programs produce, while representing the specified tree structures, provides for no interaction. It requires input in file formats such as NEXUS and Newick. It is open source (with a licence similar to the MIT licence).

- **Dendroscope** \(^{22}\) was also one of those catalogued at PHYLIP. It offers interactive dendrogram creation. Among its tree diagram forms is the classic dendrogram form. It allows labels and nodes to be edited, and sub-trees to be “collapsed and coloured”. It requires input in Newick or NEXUS\(^{23}\) format. It requires to be installed on the operating computer.

- **TreeView** \(^{24}\) Again this is aimed at phylogenetic analysis. It allows editing of the tree diagrams. Clusters can be collapsed and moved. It reads various, tree format files. It is freeware.

- **MIDA** \(^{25}\) (Modular Interactive Dendrogram Analyser) is designed for use with MATLAB. It allows the dendrogram to be interactive in that clusters can be selected and a threshold set to define the clusters.

---

\(^{21}\) [Sukumaran, and Holder. 2010]
\(^{22}\) [Huson et al. 2007]
\(^{23}\) [Maddison et al 1997]
\(^{25}\) [Bifulco et al 2009] Published source and [MIDA 2011] Unpublished source
Figure 2.2 (Existing dendrogram applications) MIDA – showing a dendrogram from MATLAB with a threshold set. [screenshot from http://www.neuronelab.dmi.unisa.it] [MIDA 2011]

MIDA requires to be installed with MATLAB on the operating computer. It is free for academic use. It takes its input directly from MATLAB. Indeed it is called from within MATLAB.

- Hierarchical Clustering Explorer 3.0\textsuperscript{26} is a program that runs on Windows 2000 and upward PCs and is free for academic use. It takes data in the form of data items in rows with several columns representing the multiple variables being measures for each data item. It does the clustering as well as the visualisation. In addition to dendrograms, the software allows histogram and scatter plot views of Multidimensional data sets. The dendrogram feature allows the user to set a “minimum similarity” to form discrete clusters. It also allows removal of detail at the bottom of a dendrogram using a “detail cutoff bar”. It also facilitates copying of dendrogram views as graphics for inclusion in other documents. However, it

does not seem to offer text copy access to the leaf detail. It has been used for optimising signal to noise ratios in gene expression investigation\textsuperscript{27}.

- Discovery Studio is used for pharmacophore modelling\textsuperscript{28} to identify new drugs. Amongst its tools is an interactive dendrogram facility with adjustable threshold. It requires installation of a server and client programs. It is commercial software.

- AutoCardSorter is an automated tool for structuring information.\textsuperscript{29} It produces an interactive dendrogram. As at August 2008 it was still in development but its creators are willing to provide it free on request. I mention it here as an example of the use of dendrograms in analysing information structure. I shall describe its use in more detail because this is a text analysis application and as such is related to the context of this project. AutoCardSorter compares texts based on an algorithm termed Latent Semantic Analysis (LSA)\textsuperscript{30}. It generates a similarity matrix for the texts, applies cluster analysis (see Literature Review section) to them and depicts their semantic relationship as a dendrogram. The dendrogram can be manipulated with a view to devising an information structure for the texts. AutoCardSorter was designed with analysis of web pages in mind. Data input is in the form of text descriptions of pages or the actual page content. The data is provided in a row-numbered grid.

- DendroStar\textsuperscript{31} is an interactive visualisation application designed to enhance dendrogram analysis of molecular line data cubes\textsuperscript{32} used in astronomy. It requires data input in the form of XML files. It runs as a Java applet in a Java-capable web browser. (As such is not compatible with iPad). It appears to have no licence. It is the result of an undergraduate project.

- Mole\textsuperscript{33} is software for classification of “channels, tunnels and pores in molecular structures”. Although it is available as an application to install on a local computer, it has an online web-accessible version. One part of the tools it provides

\textsuperscript{27} [Seo, et al 2004]
\textsuperscript{28} Pharmacophore modelling is used to speed up identification of new drugs.[Purushottamchar et al 2007] Published source and [Accelrys Corporation 2011] Unpublished source.
\textsuperscript{29} [Tselios et al 2007].
\textsuperscript{30} [Landauer and Dumais 1997]
\textsuperscript{31} [Alan 2008] Unpublished source
\textsuperscript{32} [Rosolowsky et al 2008]
\textsuperscript{33} [Petrek et al 2007] Published sources and [Petrek and Otyepka 2005 to 2007] Unpublished sources
is a dendrogram for use with a cluster analysis of tunnels in a molecular structure. It requires data input in the form of a Protein Data Bank (PDB) format file. It used Java applet graphics interface technology (not compatible with iPad).

- WebSort.Net\textsuperscript{34} is a card sort survey analysis site that charges for its services. The dendrogram application that forms part of the analysis features on the site is a Flash application. It features an adjustable similarity threshold. As it is Flash, it is not cross-platform.

**Summary of the tree diagram applications surveyed**

Table 2.1 below summarises the applications described above and compares them to the criteria of the project requirements.

The criteria details are

- Accepts MATLAB clustering output as an input format (Input format).
- Produces its tree diagrams in classic dendrogram form (Form)
- Interactive (ideally, but not necessarily, with adjustable threshold) (Interactive)
- Compatible with a PC (PC)
- Compatible with an iPad (iPad)
- Web accessible (WWW)
- Requires no installation (No install)
- Free-ware, open source, or free for education use (Free)

\textsuperscript{34} [WebSort.net 2011] Unpublished source
<table>
<thead>
<tr>
<th>Requirement Application</th>
<th>Input format</th>
<th>Form Interactive</th>
<th>PC</th>
<th>iPad</th>
<th>WWW</th>
<th>No Install</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMP</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Phylogenetic Tree Plot</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DendroPy</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Dendroscope</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>TreeView</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>MIDA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hierarchical Clustering Explorer 3.0</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Discovery Studio</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>AutoCardSorter</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓*</td>
</tr>
<tr>
<td>DendroStar</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Mole (Online)</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WebSort.net</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 2.1 Tree diagram applications surveyed vs. criteria required.

*AutoCardSorter could probably be obtained free by the University by contacting the authors.

* DendroStar would appear to have no licence.
Existing dendrogram and tree diagram applications

Conclusion
I have been unable to find a web interactive dendrogram application that meets all the criteria of this project. Of all the applications examined, the one which came closest in terms of interactivity over the web was DendroStar (the undergraduate project aimed at astronomic analysis). Although its interactivity did not include an adjustable threshold it did have responsive mouse-over highlighting of clusters. However, being reliant on Java applet technology it was not compatible with iPad. Nor does it accept MATLAB clustering data for input.

If the ICT Research Perspectives web site is to include a web accessible, cross-browser, cross-platform, interactive cluster analysis tool for visualising MATLAB clustering data then one will need to be created.

Literature Review
(Section 3)

The following topics will be addressed:

- Visualisation and Data analysis
- Technologies for rendering visualisations in web browsers
- Note on browser JavaScript performance

Visualisation and Data analysis (Section 3.1)

In this section the following topics will be addressed:

- Visualisation of Science
- Previous unpublished work on visualising GoW data in the Texture Lab.
- Word co-occurrence
- Showing structure in data sets
- MATLAB and Cluster Analysis
**Visualisation of Science**

In a paper which attempts to build a “Map of Science”, Boyack and Klevans\(^{35}\) point out that in recent years there has been significant effort put into the visualisation of “the structure of science”. One of the main motivations for seeking a true and accepted map of science (or a “Consensus Map” as Boyack and Klevans put it) is to bring to light changes in the relationships between the scientific disciplines. In effect they are pointing out that, in visualising science and offering an alternative view of it, the way we think about science might be altered. Visualisation is naturally meant to have the effect of opening ones eyes to some aspect not previously perceived. (Indeed this could be applied to any sub-topic as well as the overview). This aspect of visualising the GoW data is one possible outcome of this project.

In another paper\(^{36}\), Börner, Boyack and Chen, review various methods of visualising scientific domains. They review methods that allow retrieval and classification of information in addition to visualising the domains and their relationships. In their introduction, they make the point that previous manual methods of reviewing and attempting to map and provide an overview of large domains of scientific knowledge is not only time-consuming, but also highly subjective. They also point out that these domains are constantly being changed and added to. It is this aspect of scientific knowledge domains that this project may help to address in the specific area of the EPSRC Computing and ICT related research. The new ICT Research Perspectives’ improved visualisation of the GoW data (of which The Dendrogrammer is to be a part) may facilitate improved planning of the way forward in research in that domain.

**Previous unpublished work on visualising GoW data in the Texture Lab.**

This project is intended to add to and build on previous unpublished work in the Texture Lab here at Heriot Watt University by Tom Methven\(^{37}\). That work involved

---

\(^{35}\) Boyack and Klevans 2009  
\(^{36}\) Börner, et al 2003  
\(^{37}\) Tom Methven 2010 Unpublished available at this URL  
using Wordle\textsuperscript{38} to create word clouds from the top 50 words from the GoW data. The co-occurrence of these words was processed in MATLAB to establish their relatedness and show the clusters in a dendrogram. The dendrogram clustering was then used to produce modified word clouds with the words grouped by clusters and coloured accordingly.

**Word co-occurrence**

It would seem intuitively correct that if two GoW application descriptions share a large proportion of key words, then those research areas, and indeed, the labs engaged on them, are likely to be related. An example of a study which tends to illustrate this is one by Vaughn and You\textsuperscript{39}, “Word co-occurrences on Webpages as a measure of the relatedness of organizations: A new Webometrics concept”. This study contrasted word co-occurrence with hyperlink co-occurrence as guides to relatedness of businesses, in relation to the effectiveness of web search engine methods. The study concluded that word co-occurrence could be as useful as hyperlink co-occurrence in grouping the businesses. This would seem to support the view that word co-occurrence could give a good guide to the relatedness of individual GoWs and clusters of GoWs. (Indeed Vaughn and You speculate in their conclusion that their work might be applied to organisations other than businesses “such as universities”.)

**Showing structure in data sets**

**Cluster Analysis**

**A brief background**

Cluster analysis has been applied to the word co-occurrence data from the EPSRC GoW to produce the data that the application will be required to process.

\textsuperscript{38} Wordle word cloud creation tool: http://www.wordle.net/ (last accessed 16/4/2011)

\textsuperscript{39} [Vaughan and You 2010]
“The objective of Cluster analysis is to separate a set of objects into constituent groups ... so that the members of any one group differ from one another as little as possible...”

[Spath 40, in “Cluster Analysis Algorithms” 1980]

Spath’s definition clearly shows that cluster analysis is a technique suited to the aim of this project (to logically group the EPSRC grants).

The earliest book in this area in English was by Sneath and Sokal 41 and followed ten years of published work by them aimed at biological taxonomy. Indeed it started out in biological taxonomy with Florek et al42 in 1951 (in Poland) with “Single Linkage Clustering” (according to Sneath and Sokal). Numerical taxonomy sought to quantify the differences between different classifications of groups of organisms. Since then it has been extended to other multivariate data. Hartigan43 gives several examples of how clustering was being applied by 1975 including the classification of archaeological objects (both artefacts and human remains), and the clustering of patients in psychiatry. In less scientific spheres, he mentions that clustering had been used in market research and in analysing stock price behaviour.

Interestingly, Hartigan points out the use of clustering in linguistics to quantify how far apart various languages are. He also describes a study in which nouns were categorised by a group of students and grouped according to similarity of meaning44. So here we see early application of clustering to word data.

**Clustering Algorithms**

The choice of clustering method (or algorithm) used to generate the clusters and thus the classification from the data set can have an effect on the eventual dendrogram produced. So below I describe some of these methods.

41 [Sneath and Sokal 1973]
42 [Florek et al 1951]
43 [Hartigan 1975] pp2-5
44 [Miller 1969]
Everitt\textsuperscript{45} identifies 5 categories of clustering techniques: Hierarchical (a classification tree is formed), Optimisation Partitioning (a clustering criterion is chosen and the data are split based on that), Density (concentrations of data form the focus for the clusters), Clumping (clusters may overlap) and “Others”.

As hierarchical techniques are used on the clustering data for this project I will look a little closer at these:

Everitt describes two types of hierarchical clustering technique: Agglomerative and Divisive. The divisive methods repeatedly divide the data into smaller and smaller clusters. The agglomerative methods operate from the other end of the data, so to speak, and repeatedly combine the data items into larger clusters. It is agglomerative methods, which are used to produce the clustering data for this project so I will list those methods, describing some of them based on Everitt’s review:

The agglomerative methods have as their starting point a similarity matrix. This contains the similarity measure between each of the objects being classified.

**Single Linkage method (or nearest neighbour):** When two objects are to be merged into a cluster the distance (or similarity measure) between the cluster and the next object is taken to be that between the next object and whichever member of the cluster is nearest to it.

**Complete Linkage method (or furthest neighbour):** As for single linkage but taking the distance from the furthest member to be the distance that describes the separation between the cluster and the next object.

**Centroid method:** When two objects are combined in a cluster, the cluster itself is given co-ordinates, which take over from the two objects. The co-ordinates are the centroid of the previous two sets of co-ordinates. This can result in a bias when an object relatively separate from a cluster joins that cluster, with the co-ordinates of the

\textsuperscript{45} [Everitt 1974] page 7
old cluster having too great an influence on the formation of the new cluster co-
ordinates.

**Median method:** When two objects or clusters are merged, the size (number of objects
contained in the cluster) of both is taken to be the same and the co-ordinates of the
new cluster calculated accordingly. This prevents a bias as described in the Centroid
method.

**Group Average method:** This method pairs up each member of the joining cluster with
all members of the cluster to be joined and works out the average distance and takes
this as the separation of the two clusters. This calculation is used to determine the next
merge to be done.

**Ward’s method:** This method seeks to minimise the loss of detail that occurs when
objects or clusters merge into clusters. In choosing which objects or clusters to merge
next, the loss of detail that would result is calculated and taken into account. The
merge that results in the least loss of detail is enacted first.

There are two other methods described by Everitt (“McWhitty’s” and “Lance
Williams flexible”). I will not describe these, as they are not among those available in
the tools used to generate the project data.

**The choice of clustering algorithm**

The choice of clustering method (or algorithm) can have an effect on how the clusters
pan out. Figures 3.2.1 and 3.2.2 show two dendrograms each of the same data but
each drawn following clustering with different clustering algorithms\(^46\). If the two sets
of data were to be used to group the leaf nodes based on some similarity threshold, the
groups would contain different leaf nodes.

\(^46\) [Everitt 1974] p79
Figure 3.1.1 A dendrogram from data clustered using the Centroid method

Figure 3.1.2 A dendrogram from data clustered using Single Linkage method\textsuperscript{47}

\textsuperscript{47} [Everitt 1974] p79
Multidimensional Scaling

There is another major method of showing structure in data sets and this is Multidimensional Scaling (MDS). In common with cluster analysis it operates on similarity matrices. However, the result of it is visualisation of the classifications represented in a 3-dimensional space. I have included this brief mention of MDS for completeness but as it is not the method being applied to the data in this project, I will not examine it further.

MATLAB and Cluster Analysis

I will not be required to do any cluster analysis myself. However, this description is included here for completeness and to provide a background for the data to be used in the project.

MATLAB is a programming environment and set of computational tools for data analysis and visualisation. It manipulates arrays or matrices. It was created by MathWorks Inc.

A similarity matrix, such as that made from the word co-occurrence data from the EPSRC Grants on the Web text descriptions, is just the kind of thing MATLAB was designed to help analyse. A similarity matrix has the names (or identifiers) for those items being compared across the top and down the side. Each item’s similarity figure appears where the row and column for those two items meet. The similarity matrix can be loaded into MATLAB and cluster analysis can be applied to it.

According to Martinez and Martinez, MATLAB offers two types of cluster analysis:

- Hierarchical clustering and
- K-Means clustering.

---

48 [Young 1985]
49 [Martinez and Martinez 2002] – Appendix A p511
50 [Martinez and Martinez 2002] – Chapter 9.5
\( K \)-Means clustering aims to divide up the data points into \( K \) groups. That is the number of clusters is specified at the outset.

The other type, Hierarchical clustering, is the type used to process the data for the project’s visualisation. Hierarchical clustering takes the items, or groups of items, and one after the other, “merges” those two items or groups that are most similar. It does this until there is one single group (this would be represented by the “trunk” of the tree in a dendrogram). (See Section 4- Requirements Analysis – Example clustering data in Table 4.1: Explanation of how clustering data relates to its corresponding dendrogram).

The processing is done using the \textit{linkage} function in MATLAB’s \textit{Statistics Toolbox}. The \textit{linkage} function provides clustering algorithms for this:

- Single linkage
- Complete linkage
- Average linkage
- Centroid linkage and
- Ward’s Method

The choice of methods is provided to suit the nature of the data to be clustered. The nature of these various methods is described in the section on Clustering Algorithms in “Showing structure in data sets” above. As has been stated above, the actual clustering work is not within the scope of the project.

MATLAB does include the facility to display clustering data as a dendrogram. However, this is not an interactive one. (While searching for an interactive web dendrogrammer an interactive dendrogram plug-in for MATLAB was found. See “MIDA” in Section 2, Existing dendrogram applications).
Technologies for rendering visualisations in web browsers (Section 3.2)

In this section the following topics are addressed

- SVG (Scalable Vector Graphics)
- JavaScript Libraries
  - jQuery
  - jQuery UI
  - jQuery Plugins and jQTouch
  - Dojo
  - DHTMLX Touch
- JavaScript Graphics Libraries
  - JsDraw2D
  - Raphaël
  - Protovis

SVG (Scalable Vector Graphics)

SVG is presented in XML documents which can be saved with the .svg extension or as XML files. There are tags for various basic shapes such as circle, line (stroke) and rectangle. Tag attributes control properties such as line thickness and fill colour. Custom shapes can be defined with the polygon tag. The shapes can be grouped into objects with the <g> tag and given identifiers. The basic shapes are actually made from the primitive <path> tag which allows any shape or line to be scribed. Shapes, once defined, can be repeated by using the <use> tag with the previously defined identifier as an attribute. The shapes can be transformed (repositioned) or scaled by invoking the <use> tag with an attribute specifying the scaling.\(^{51}\) SVG 1.1 is the current W3C\(^{52}\) specification. SVG 2 is in development\(^{53}\). SVG has been waiting in the

---

\(^{51}\) [Eisenberg 2002]

\(^{52}\) W3C is The World Wide Web Consortium, the international community body established in 1994 to develop standards for the WWW (Tim Berners Lee, director).

wings for a number of years now, but is the graphics standard that is likely to become commonly used for anything that lends itself to vector graphics soon because the latest browsers are capable of rendering graphics using this standard. See Protovis below for an example of how this technology is being put to use now.

**JavaScript Libraries**

There are a number of JavaScript Libraries. Chaffer and Swedberg\(^4\) list three: Prototype, MooTools, and Dojo, in their book on jQuery. I shall focus here mainly on jQuery as a general example of what JavaScript libraries can bring to the development of Rich Internet Applications (RIAs). I will also look at Dojo as being part of one particularly convincing interactive graphic web application. Finally I will briefly mention DHTMLX Touch as an example of a proprietary library.

**jQuery**

The *jQuery* open source JavaScript library is increasingly more widely used. Indeed it might be seen as becoming part of the establishment as Microsoft now bundles jQuery with Visual Studio and ASP.Net\(^5\).

jQuery has a fundamental feature that makes it a major candidate to be used in the solution for this application. It helps deal with issues of cross-browser compatibility.

**Cross-browser compatibility**

This issue can be described as arising out of the fact that different browsers often would render the same page differently were they not provided with pages which are able to take the brand and version of the browser into account and react accordingly. jQuery provides non-browser-specific ways\(^6\) of

---

\(^4\) [Chaffer and Swedberg 2009] (Chapter1)  
\(^5\) Reported in Ajax Magazine March 17\(^{th}\) 2010  
http://ajax.phpmagazine.net/2010/03/microsoft_to_support_actively.html  
\(^6\) [Chaffer and Swedberg 2009] Chapter 1.
1) controlling and altering the appearance of the rendered page and
2) AJAX\textsuperscript{57} – style interaction with served data (where parts of a rendered page are
refreshed without reloading the entire page).

The jQuery project\textsuperscript{58} claims it is cross browser and functions for “\textit{IE 6.0+, FF 2.0+, Safari 3.0+, Opera 9.0+, [and] Chrome}”

\textbf{jQuery UI}

The jQuery UI is a collection of so called “\textit{widgets}” for use in a rich internet
application design in ways often associated with Flash. The widgets allow developers
to create a pleasing user interface without having to build it from the ground up.
These widgets, such as dialogues and slider controls would reduce the amount of
programming required for the application’s user interface.

\textbf{jQuery Plugins and jQTouch}

jQuery supports extensibility using a flexible plugin architecture. \textit{jQTouch}\textsuperscript{59} is a
jQuery plugin designed for creating web applications for mobile devices such as
Android devices, iPhone and iPad. It can allow HTML and JavaScript to appear and
function very like a native Apple iPhone application\textsuperscript{60}. Gmail for the iPad was
developed with jQTouch\textsuperscript{61}.

Firtman\textsuperscript{62} surveys JavaScript UI libraries for compatibility with various browsers and
platforms. The jQuery UI had no incompatibilities exposed in that survey (although
there was no data for jQuery UI with Blackberry.)

See jQuery UI in Preliminary Investigations (Section 6) below.

\textsuperscript{57} AJAX - Asynchronous JavaScript and XML
\textsuperscript{58} http://jquery.com/ accessed on 6\textsuperscript{th} March 2011
\textsuperscript{59} JQTouch was created by David Kadena
\textsuperscript{60} [Stark 2010] Chapter 4
\textsuperscript{61} [Churchill 2010] See Unpublished Sources
\textsuperscript{62} [Firtman 2010]
**Dojo**

While researching interactive mind-mapping tools I found one particularly effective multi-user interactive mind-mapping tool, *Mind42.com*. This used the Dojo JavaScript library. *Mind42.com* allowed users to interactively create mind-map diagrams and edit them. The user can drag elements of the mind-map around the screen to reconnect them to another part of the diagram. The page rendered by browsers running the application contained decorated text elements (underlined text) and also what appeared to be dynamically created PNG graphics closely co-ordinated with the text. This technology is a strong candidate for use in this interactive dendrogram project.

**DHTMLX Touch**

DHTMLX Touch is a JavaScript framework for Mobile and Touch Devices. It is mainly aimed at mobile devices. It can run on standard browsers such as FireFox an WebKit based browsers like Chrome and Safari but not without some problems. It is based on HTML5. DHTMLX Touch is a proprietary piece of software but can be used under the GNU GPL licence. (The company sell support for it).

**Javascript Graphics Libraries**

There are several JavaScript libraries aimed specifically at rendering charts and visualisations in web pages. I focus on two likely candidates for use in this project: Raphael and Protovis.

**JsDraw2D**

JsDraw2D is a JavaScript graphics library. It is designed to enable the drawing of 2-dimensional graphics in web pages. I mention it here as one of the earlier examples of a JavaScript graphics library.

**Raphaël**

A JavaScript library for creating SVG and VML graphics in web pages. It allows creation of DOM addressable graphics and supports the following browsers:

---

63 [DHX 2011] See Unpublished Sources
64 [Baranovskiy 2010] See Unpublished Sources
“Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+.”  

It also appears to operate with the touch user interface on the iPad. (See “Raphael” in Section 6)

**Protovis**

Protovis is described by its authors as “a graphical toolkit for visualisation”. Protovis uses JavaScript and SVG to create graphics that can be viewed in web browsers (assuming the browser is relatively up to date i.e. are SVG compatible). In fact Bostock and Heer state that Protovis supports HTML5 canvas, SVG and Flash.

Bostock et al published an interesting paper, “A Tour Through the Visualization Zoo”. The paper is a very interesting review of methods of computerised visualisation including hierarchical visualisations similar to dendrograms. Illustrations of two such charts made with Protovis are included below to go some way to show its capabilities. What cannot be shown here is that there is some interactivity included with the charts when viewed on the web site that accompanies the paper (http://hci.stanford.edu/jheer/files/zoo/). The leaf nodes link to related pages, in this case each node is a Protovis function and links to its documentation page. Figures 3.2.1. and 3.2.2 show two visualisations produced using Protovis. This shows that Protovis would be well suited to creating dendrograms.

---

65 The W3C DOM is the Document Object Model, a W3C standard for traversing and addressing the elements in a web page, typically by using their assigned Ids, properties, and their hierarchical structure. See [Negrino and Smith 2004 p302]
67 [Bostock and Heer 2009]
68 [Bostock et al 2010]
Figure 3.2.1 – Node-link diagram
(from Bostock et al page 64)

Figure 3.2.2 – Radial node-link diagram of the same data
(from Bostock et al page 64).
Note on browser JavaScript performance

(Section 3.3)

I have added this note about the speed of the JavaScript performance in web browsers because late in the project development browser performance became an issue.

John Resig (inventor of the jQuery JavaScript library) has compared the JavaScript performance of various browsers with a view to gauging the speed of their respective JavaScript engines. He looked at the following browsers and JavaScript engines (See table 3.4.1 below)

<table>
<thead>
<tr>
<th>Browser</th>
<th>JavaScript Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safari/WebKit versions up to 3.1</td>
<td>JavaScriptCore</td>
</tr>
<tr>
<td>Safari 4.0+</td>
<td>Squirrelfish</td>
</tr>
<tr>
<td>Chrome</td>
<td>V8</td>
</tr>
<tr>
<td>Firefox up to version 3.0</td>
<td>SpiderMonkey</td>
</tr>
<tr>
<td>Firefox 3.1+</td>
<td>TraceMonkey</td>
</tr>
<tr>
<td>Opera 9.5+</td>
<td>Futhark</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>JScript</td>
</tr>
</tbody>
</table>

Table 3.3.1 – Browsers and their JavaScript engines

He used the WebKit SunSpider test, which tests only JavaScript performance. It is possible to summary his results by taking Safari 3.1 as a browser performing well for JavaScript. Compared to Safari 3.1 all the other browsers except the Opera and Internet Explorer browsers were faster (some only slightly). Opera 9.5 was only slightly slower. Internet Explorer 8 was less than half the speed of Safari 3.1. and Internet Explorer 7 was over five times as slow as Safari 3.1. (i.e Internet Explorer 7 took over five times as long and Internet Explorer 8 took over twice as long to complete the test compared to Safari 3.1.)

69 [Resig 2008] Unpublished source
More recent tests published at ZDnet.com\textsuperscript{70} show that, looking at more recent browsers (Safari 5, Chrome 10, Firefox 4, Opera 11, and Internet Explorer 9), the JavaScript performance of Internet Explorer 9 is in line with and faster than some of its competitors. (It was noted, however, that the 64 bit version of Internet Explorer 9 was found to be slower by comparison, taking 4 times as long to complete the SunSpider test compared to the 32 bit version.)

\textbf{JavaScript performance: Conclusion}

Users of the older versions of Internet Explorer, version 8 and, particularly, version 7, face significantly slower performance in running JavaScript applications than users of other browsers. This complicates the issue of cross-browser performance. (See Evaluation.)

\textbf{Requirements Analysis}

\textit{(Section 4)}

The following aspects will be addressed here

- The clients
- The users
- The platform
- Dendrogram data display and interactivity
- The data which the application needs to process and represent as a dendrogram.
- Loading the data
- Summary of Requirements – Mandatory and Optional

During the development of the application some of the requirements were clarified altered or enhanced at meetings with The Clients in the light of evaluation of the prototypes. Any such changes are described along side the original specification below.

\textsuperscript{70} [Kingsley-Hughes 2011] Unpublished source
The ‘Clients’

At the start of the project the ‘client’ for the software was Prof. M. Chantler\textsuperscript{71} who also had the role of my MSc project supervisor. Later, F.Halley\textsuperscript{72}, a member of the research staff in the Texture lab, joined the ‘client’ group (hereafter referred to as “The Clients”). He is to be the main user of the software and also is to take forward the development of *The Dendrogrammer*. It will be part of the *ICT Perspectives* web site, which is a project in its early stages at time of writing and which had not started when the original requirements for *The Dendrogrammer* were set out. Other work on the *ICT Perspectives* project must be carried out first to allow analysis of further requirements for *The Dendrogrammer*. Once these are established F.Halley will be developing *The Dendrogrammer* to meet those requirements. This aspect introduced the further ‘requirement’ in that this meant absolute certainty that the software would require being maintainable by another developer. (However, this is not an unusual requirement in a software development).

The users of the application

The application is to be used by EPSRC officials researching the structure of the grant portfolio and those who are tasked with organising group meetings among grant applicants or delegates. Indeed, delegates from the research groups may also use the application as part of any explanation they might receive about the formation of consultation groups. Indeed, it is to be part of the EPSRC *ICT Perspectives* web site. Therefore, any user accessing the EPSRC research portfolio information through the site might use the application.

More generally, in addition it is intended that the application could be used to create interactive dendrograms from any set of MATLAB clustering data output. Therefore it might be used by anyone interested in visualising clustering data and accessing the associated leaf node data via the visualisation. (See “The data which the application needs to represent as a dendrogram” below.)

\textsuperscript{71}See Acknowledgements

\textsuperscript{72}See Acknowledgements
Platform

As it is to be part of the *ICT Perspectives* web site the application needs to use a technology that is compatible with the World Wide Web. It is intended that it be accessible via the *iPad* (in addition to PCs) as these are often in use at meetings and it may be desired to show delegates how the composition of discussion groups was arrived at. An optional requirement is that other tablet computers will be catered for in addition to the *iPad*. In consultation with the client these were defined as being exemplified by the *Asus Transformer Android* slate. (Mobile phones were ruled out as having too small a display for this purpose.)

Dendrogram data display and interactivity

There are three types of interactivity planned for the dendrogram.
- Access to the leaf node data.
- Adjustable highlighting of cluster groupings by a similarity threshold.
- Ability to designate an arbitrary number of cluster groups and have the diagram respond by setting the similarity threshold appropriately.

In addition there is an optional additional type of interactivity
- Some form of zoom facility to allow the alteration of the amount of detail in view.
- Access to word clouds associated with the leaf nodes.

Accessing leaf node data

In addition to the rendition of the dendrogram itself, each leaf node of the dendrogram will be identified by information from the corresponding GoW grant application such as a short title of the grant or a particular key word. Other data associated with that grant application or key word will be available interactively from the dendrogram. The user will be able to interact with the leaf nodes and clusters of the dendrogram and copy the data they contain to the clipboard to be pasted into the collation tool of their choice, e.g. a word processor. (See figure 4.1 below).

During the project this was clarified. It was specified that the data structure should support both short and long descriptions for leaf and cluster nodes. While these might not all be displayed at this stage of The Dendrogrammer development, it would allow
them to be available for later additions to the application in future developments. (See “Meeting with Clients 23/5/2011” in Key Stages section.)

1. User clicks or taps on the leaf node

![Diagram of a dendrogram with a leaf node being interacted with](Image)

2. A pop-up or dropdown opens to give access to the node details, allowing the user to copy to the clipboard.

![Copy and Close buttons with Lorem ipsum text](Image)

**Figure 4.1** – Possible interaction with a leaf node on the dendrogram.
**Adjusting the highlighting or colouring of the clusters**

A second type of interaction is planned for the dendrogram: The ability to control highlighting or colouring of the clusters by adjusting a similarity threshold. The user would drag a slider widget or perhaps even an object representing the similarity threshold itself.

Two groups defined by similarity threshold at a given distance up the dendrogram

**Figure 4.2 Highlighting or colouring of cluster groups by adjusting a similarity threshold**
Figure 4.3 A different highlighting or colouring of cluster groups by adjusting a similarity threshold on the same dendrogram

Ability to designate an arbitrary number of cluster groups
This could be done with a numeric dialler and the similarity threshold would alter accordingly.

Output of a cluster’s leaf list to a search engine
(originally: Access to word clouds associated with the leaf nodes)
Originally it was hoped that this project might be able to implement, as an optional requirement, an interactive visualisation feature that would allow the user to view word clouds by interacting with the leaf nodes. During the project it was realised that the data needed to inform a word cloud was not going to be available. This option was replaced with the facility to output a cluster’s leaf list for analysis in an external...
search engine application (See “Meeting with Clients 13/7/2011” in Key Stages Section) (See Discussion of Word Cloud Feasibility at the end of this section.)

The data which the application needs to represent as a dendrogram

The data on which the application is to operate is clustering data generated by MATLAB. The text descriptions from the Computing Science portion of the EPSRC grants on the web were analysed for word co-occurrence (see Literature Review). This produced a similarity matrix. The similarity matrix was processed through MATLAB using an appropriate clustering algorithm. The resulting data consists of three columns of figures. The full actual data to be used is included as an appendix. This processing was not done by me. The processed data was provided to me by others in the Texture Lab (see Acknowledgements).

Figure 5.4 below is a screenshot of a few rows of the processed data as I received it.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46</td>
<td>64</td>
<td>2.08</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>100</td>
<td>2.13</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>80</td>
<td>2.25</td>
</tr>
<tr>
<td>4</td>
<td>88</td>
<td>93</td>
<td>2.26</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>16</td>
<td>2.20</td>
</tr>
<tr>
<td>6</td>
<td>33</td>
<td>90</td>
<td>2.3</td>
</tr>
<tr>
<td>7</td>
<td>45</td>
<td>87</td>
<td>2.3</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>55</td>
<td>2.31</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
<td>77</td>
<td>2.31</td>
</tr>
<tr>
<td>10</td>
<td>86</td>
<td>86</td>
<td>2.31</td>
</tr>
<tr>
<td>11</td>
<td>24</td>
<td>47</td>
<td>2.34</td>
</tr>
<tr>
<td>12</td>
<td>76</td>
<td>88</td>
<td>2.35</td>
</tr>
</tbody>
</table>

Figure 4.4 – Part of the MATLAB output from the EPSRC ICT GoW data.

Columns A and B contain pairings of data points and/or cluster designations. Column C contains each pair’s corresponding merge height. The lower the merge height the more similarity is shared by the pair concerned. The higher the merge height, the less similar are the two items in the pairing. In terms of an eventual dendrogram, produced
from the data, the merge height is the height up the diagram (from the base) at which the pairing will join. There are 99 rows in the data representing pairings between 100 data points and clusters. The row number represents the order in which the MATLAB processing has imposed on the data.

To accompany this 3-column MATLAB clustering data there will also be a table of leaf node data that will contain the details of each leaf node (each corresponding to one of the actual data points in the MATLAB clustering data). Each designated data point will key to a set of leaf node data. Each set of leaf node data will consist of details from a single GoW grant application. It is a selection of these details e.g. the name of the lab doing the work associated with the grant, that will be displayed interactively on the dendrogram. (See dendrogram data display below).

Although the data described above is the target of this project the application should handle any 3 column MATLAB clustering output along with its associated leaf node data table. The leaf node data table will contain only text data.

In the early stages, to develop the application, I plan to work with a simplified data set. The example set of clustering data below, was provided to me by F. Halley of the Texture Lab. The explanation below is based on the explanation he gave to me of this example data (See Acknowledgements). (See also “MATLAB and Cluster Analysis” in the Literature Review above).

The example data in table 4.1 below represents 6 data points and their clusters.
The data in Table 5.1 is interpreted in the following way:

The middle two columns contain the number designations of data points and the clusters that they are members of. The data points are items 1 to 6. These will be the leaf nodes on the dendrogram. Items 7 to 10 are the designations given to the clusters.

What row one means is that data points (leaf nodes) 1 and 6 share a similarity figure of zero. i.e. they form a cluster that will be joined at the foot of the dendrogram.

What row two means is that data point 4 is paired with item 7 sharing a similarity of zero. Item 7 is not a data point (6 is the highest numbered data point). Item 7 is therefore the first cluster (cluster No.7). Indeed it is the cluster described by row 1. This means 4 is clustered with 1 and 6. i.e. data point 4 is paired with cluster 7 forming the new cluster designated no.8. As the similarity figure (or merge height) of both clusters is zero, they both will be placed on the foot of the dendrogram.

What row three means is that data points 2 and 5 are paired forming cluster No.9, again with similarity of zero.

What row four means is that data point 3 and cluster 8 are paired forming cluster No.10, with a similarity of 1.
And finally the meaning of row 5 is that cluster No.9 and cluster No.10 are paired to form the last cluster. This last cluster would be numbered 11 were a further reference needed. This cluster has a similarity or merge height of 2.

Once interpreted like this the clustering data can be drawn as a dendrogram. (See figure 4.5 below) Note the cluster designation numbers do not appear on the dendrogram. Only the leaf nodes, which are the data items, appear on the dendrogram.

![Dendrogram](image)

**Figure 4.5 – Dendrogram from example clustering data**

The new interactive web dendrogram application will need to contain an algorithm for constructing the dendrogram such that there are no overlapping or crossing cluster connections.

**Loading the data**

The user will need to provide the application with a URL to a file containing the data that is to be visualised. That file will need to contain the clustering data columns and the leaf node data table.

During the project the load data requirement was a subject at two of the Client meetings. In the first instance the level of the specification for the data-load feature was minimised to be only that necessary to allow testing of the other features of the application. It was deemed that, the input format could not yet be fully specified and
this aspect would be developed later in the life of the ICT Perspectives project. (See Key Stages section 25/5/2011). Later, I think in the light of the progress on some of the interactive features, and with a view to handling the data in the form in which it was likely to be produced, the specification was added to. The ability to supply the data in two files with a “manifest” file containing metadata was specified. The manifest was to hold the names of the clustering data and the leaf data files. (See Key Stages section 1/7/11).
## Summary of Requirements – Mandatory and Optional

Table 4.2 summarises the mandatory and optional requirements.

<table>
<thead>
<tr>
<th>Mandatory</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access</strong></td>
<td></td>
</tr>
<tr>
<td>Accessible via the World Wide Web in a web browser</td>
<td></td>
</tr>
<tr>
<td>Support the latest major browsers: Internet Explorer, FireFox, Safari,</td>
<td>Support for earlier versions of the major browsers.</td>
</tr>
<tr>
<td>Chrome.</td>
<td></td>
</tr>
<tr>
<td>Cross platform: accessible on PCs and the iPad.</td>
<td>Accessible on other tablet computers.</td>
</tr>
<tr>
<td></td>
<td>Accessible on other mobile devices such as iPhone and Android.</td>
</tr>
<tr>
<td><strong>Basic functionality</strong></td>
<td></td>
</tr>
<tr>
<td>Load the data from a suitably formatted file on the World Wide Web</td>
<td></td>
</tr>
<tr>
<td>specified by URL</td>
<td></td>
</tr>
<tr>
<td>Draw a dendrogram from clustering data</td>
<td></td>
</tr>
<tr>
<td><strong>Interactivity</strong></td>
<td></td>
</tr>
<tr>
<td>Adjustable similarity threshold to highlight groups of leaf nodes</td>
<td>Ability to specify an arbitrary number of groups and set the similarity</td>
</tr>
<tr>
<td></td>
<td>threshold to fit</td>
</tr>
<tr>
<td>Select a leaf and copy text associated with it</td>
<td>Select a leaf or cluster and view a word cloud associated with it</td>
</tr>
<tr>
<td></td>
<td>Ability to zoom in and out to reveal more or less detail of the dendrogram</td>
</tr>
</tbody>
</table>

Table 4.2 Summary of mandatory and optional requirements
Discussion of Word Cloud Feasibility

Thoughts on the development of the Word cloud capability:

To produce a word cloud of the terms retrieved from a cluster, more data about these terms in the context of their cluster group is required than is available to the application in the MATLAB clustering output. The required data is the relative popularity of the leaf terms in the texts from which they were mined. The leaf list for a cluster needs to be processed through another application that has access to the original texts from which the clusters were generated. At the moment this is thought likely to be a search engine-type application. The infrastructure for output to the search engine is established in *The Dendrogrammer* as it now stands (A search hyperlink in the cluster-click dialog). There is, as yet, no companion infrastructure established to receive back into the JavaScript part of application the results of any processing in response to that output. However, jQuery has established methods, which make use of AJAX techniques to allow a web page to interpret “further data from the server and present it on the page”\(^73\). As jQuery is already part of the application infrastructure this would seem to be a direction worth pursuing to achieve the word cloud functionality.

As the word clouds contain text rather than drawn elements, ways of rotating text might be desirable. However, I have been unable to find any cross-browser way of achieving this that does not rely on images (such as an example by Jonathan Snook\(^74\)). It could be done using Raphael drawn text using the `text()` method. The amount text to be displayed in the word cloud would probably be an issue. The experience with *The Dendrogrammer* and its leaf labels is that the slower browsers (IE7 and IE8) have problems rendering a large amount of Raphael-drawn text. Were this to be a problem then the techniques described by Snook could be used.

\(^73\) [Chaffer and Swedberg 2009] Chapter 6, p115.

\(^74\) [Snook 2011] Unpublished source
Tool Selection  

(Section 5)

This section describes:

- Some preliminary investigations
- The Criteria for selection and the Decisions made.

Some Preliminary Investigations

The following are described in Appendix 2

- Some of the practical investigations of Protovis, Raphael and jQuery UI were undertaken to inform my decisions on the choice of technology to use in the application.
- Some practical basics of creating SVG graphics in a web browser and some of the cross-browser issues with SVG graphics.

The conclusions of these investigations are summarised in the table below:

<table>
<thead>
<tr>
<th>Subject</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protovis</td>
<td>Protovis can be considered cross-platform but not cross-browser. (IE 8 and below not supported). Drag-ability was not demonstrated</td>
</tr>
<tr>
<td>SVG</td>
<td>SVG graphics cross-browser is problematic. The differences in positioning of the graphics was marked</td>
</tr>
<tr>
<td>JQuery UI</td>
<td>The jQuery UI appears to be fairly straightforward to implement. It demonstrates cross platform and cross-browser capability.</td>
</tr>
<tr>
<td>Raphael</td>
<td>Cross-browser and cross-platform. Drag-ability was demonstrated. Raphael would seem a good candidate to use in this project.</td>
</tr>
</tbody>
</table>

Table 5.1 Conclusions from Preliminary Investigations
Tool Selection: Criteria and Decisions

Areas of decision

There were two main decisions to be made on the technologies to use to implement the application:

- The drawing technology
- The supporting user interface technology

The drawing technology

For the drawing technology the following are to be the criteria for the decision:

- Render uniformly cross-browser (including recent older browsers)
- Render cross-platform: in PC and in iPad
- Good documentation available
- A body of user support in web forums
- Coding challenges and risks
- Drag-ability: can object be made drag-able?
- Drag-able cross-platform, PC and iPad

Considering these criteria SVG on its own is eliminated on the following grounds:

To render cross browser would require programming for individual browser versions and brands. The coding challenges and risk associated with SVG at this point in time with the browsers that the application might be expected to meet would be too great. However, both Raphael and Protovis employ SVG.

Raphael and Protovis, as graphics frameworks, which aim to overcome cross-browser issues, are compared against the criteria in the table below:
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Technology</th>
<th>Protovis</th>
<th>Raphael</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-browser</td>
<td></td>
<td>Not fully. Internet Explorer 8 (IE8) and lower are not supported.</td>
<td>Yes. All the recent browsers are supported including IE8.</td>
</tr>
<tr>
<td>Cross-platform</td>
<td></td>
<td>Yes. Examples render similarly on PC and iPad</td>
<td>Yes. Examples render similarly on PC and iPad</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td>Very good, simple documentation. Clearly presented. Many and varied good examples.</td>
<td>Good clear documentation. Some good simple examples. Some of the examples are complex.</td>
</tr>
<tr>
<td>Support in web forums</td>
<td></td>
<td>Well supported and popular.</td>
<td>It is clearly well, regarded. However, the issue of incompatibility with IE8 seems like the “elephant in the room”. Developers trying to address this issue get scant response.</td>
</tr>
<tr>
<td>Coding risks</td>
<td></td>
<td>Medium risk. Examples use a declarative style rather different to that normally used in JavaScript.</td>
<td>Low risk. Documentation reveals Raphael examples are coded in a common JavaScript style.</td>
</tr>
<tr>
<td>Drag-ability</td>
<td></td>
<td>Yes. Graphic objects can be dragged.</td>
<td>Yes. Graphic objects can be dragged.</td>
</tr>
<tr>
<td>Drag-able cross-platform</td>
<td></td>
<td>Not demonstrated. It would be risky to assume drag-ability was possible on iPad.</td>
<td>Yes. Drag-ability is demonstrated on PC and iPad with the same example.</td>
</tr>
</tbody>
</table>

Table 5.2 Raphael and Protovis vs. selection criteria
**Decision on Drawing Technology**

Protovis’ incompatibility with IE8 and its lack of demonstrated drag-ability on iPad are what is ruling it out. IE8 is not an uncommon browser, as IE9 cannot be used with Windows XP.

Raphael meets all the criteria. As such it is the lower risk choice. Raphael will be proceeded with in the first prototype.

**Supporting user interface technology**

The two frameworks under consideration are jQuery and Dojo.

The following are be the criteria for the decision:

- Documentation and community support
- User Interface widget library
- Compatibility with mobile devices

<table>
<thead>
<tr>
<th>Criteria \ Technology</th>
<th>jQuery</th>
<th>Dojo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation and community support</td>
<td>Extensively written about. A long list of industry supporters. Bundled as a development tool with .Net. No shortage of forum support</td>
<td>There are books, but they seem less ubiquitous than for jQuery. A good list of industry supporters. No shortage of forum support.</td>
</tr>
<tr>
<td>User Interface widget library</td>
<td>Yes. It seems easy to use after a short initial investigation.</td>
<td>Yes. However, I have not tried it out yet.</td>
</tr>
<tr>
<td>Compatibility with mobile devices</td>
<td>jQTouch has been very well received. The possibility of introducing jQTouch to help with the iPad compatibility offers reduced risk.</td>
<td>Dojo as yet does not appear to have a mobile device partner framework.</td>
</tr>
</tbody>
</table>

Table 5.3 jQuery and Dojo vs. selection criteria
**Decision on user interface technology**

This is less clear cut than for the drawing technology. I have decided to proceed with jQuery for the first prototype because a) it is very well supported and b) the existence of jQTouch mobile partner framework offers reduced risk in creating an interface compatible with mobile devices.

**Software Development Methodology**

*(Section 6)*

The methodology I chose to use was motivated by the conditions that existed for this project.

1) The personnel available. There was just a single developer (myself) and although I did have some prior experience with JavaScript programming it was not extensive. In addition I had never worked with jQuery or Raphael libraries before.

2) The relative novelty and speciality of the desired product and also the requirement that it be cross-platform i.e. work on PC and iPad. Prior to the project I had no experience of the iPad nor iPad programming.

3) The good availability of the “client”. I was not limited to referring to formally specified requirements on paper. I would be able to get feedback on prototypes readily.

Using a formal analyse-design-implement-test-document-evaluate-maintain such as the Waterfall method would be far too risky. Designing a full, end product with no real idea of how to implement it is unlikely to work.

My chosen methodology would need to rely on prototyping. Indeed there are some established methodologies which can do so.

There were three that could be considered:

- Prototyping
- Iterative and Incremental Development
- Agile software development
Prototyping, per se, is not a full software development methodology in itself. It is an activity that might form part of a project development. However, in describing the “Evolving system development life cycle”, David King, describes two approaches to prototyping. One involves developing prototypes separately as disposable items. The other, termed, Prototyping by progressive development, involves developing each prototype as a deliverable stage. Once each development is completed that prototype is the starting point for the next. In this way of working the prototype is not discarded and the final development is the production system. This is the way I chose to work on this project.

Cockburn, in his book on Agile software development describes iterative and incremental development as taking account of the fact that mistakes will be made. As the steps are incremental there is less to lose by failure in one of the increments.

Much of Agile practice is concerned with development teams and as this project involved a single developer that aspect of it is not relevant. However, the Agile approach also advocates frequent communication with the client and this is open to me to help reduce the risk of developing the wrong end product.

The aspect of the Incremental approach to development that applied in particular here is that of being able to demonstrate progress to my “client” and project supervisor early and often.

In practice, this project consisted of the development of successive prototypes each addressing and solving a particular aspect. Each incremental development was shown to the client. Finally, the last issue was addressed and the prototype became the finished product. The chosen methodology was Prototyping by progressive development.

---

[King 1984]

[Cockburn 2002]p48
Key stages in the development of the application

(Section 7)

In this section I describe important stages in the development of The Dendrogrammer in chronological order. I describe these in this way to show how I applied the incremental development method, how the prototypes developed, and also to show where particularly important client meetings took place in the chronology of the development.

- All have a date reference to the Project Diary appendix. Those diary entries referring to prototypes contain URLs to the prototypes, which are all available via the World Wide Web.
- Some of the entries below are illustrated with partial screen shots. The full screenshots can be found (keyed by date) in the Prototype Development Appendix.

A Project schedule was produced

It was based on the list of tasks I had discussed with my supervisor (See Diary 7/3/2011). (See also the Planning Appendix).

Example data obtained

I obtained a copy of the example clustering data and was given an explanation of how to interpret it as a dendrogram (See Acknowledgements, F. Halley and T. Methven) (See Diary 13/3/2011).

Clarified requirements

I met with my supervisor and clarified some of the requirements. Specifically: the type of interactivity, that the data was to be loaded from a file chosen by the user, and that there would be a need for an additional leaf data file in which the MATLAB leaf designation number was keyed to the leaf data (such as name and description). (See Diary 14/3/2011)
Raphael graphics framework chosen

I decided to pursue Raphael as the framework to use for the drawn elements in the first prototype. Early work produced drawn axes with labels that rendered cross-browser. See Figure below. (See Diary and Development Appendix 26/4/2011).

![Canvas div comes below here](image)

Figure 7.1 – Screenshot from early work on 1st prototype showing the portion of the screen with labelled axes.

Data structure designed

An initial data structure based around two tables was designed. It consisted of

- The Cluster array, an array of objects. Each object storing the data from one row of the MATLAB cluster data.
- The Dendrogram element array. Each object storing values used in the drawing of the elements in the dendrogram, and also the drawn elements.
- In addition many other variables holding various items form axes and canvas to scale factors and margins.

However the number of variables was so great that I felt the need to create a data dictionary to keep a handle on. This unwieldy data structure was to be superseded later. (See Diary and Development Appendix 30/4/2011)
Prototype drawing dendrogram from hard coded data

A prototype using the data structure and some simple hard coded data was working. This prototype animates the dendrogram with a bounce “easing” as it is drawn. (The animation easing was fun, but discarded in later prototypes as it added processing load). Early on I decided that the dendrogram should be drawn on its side. As it was intended for visualising test data, I wanted leaf labels to be horizontal for reading. See Figure below. (See Diary and Development Appendix May 9th)

![Dendrogram Diagram](image)

Figure 7.2 – Screenshot from prototype 9th May showing a portion of the screen. Dendrogram drawn from a small hard-coded data set.

Evaluated this first prototype with my supervisor

My supervisor suggested a tree data structure to help in dendrogram construction, pointing out that a depth-first search of the tree would reveal the order in which the leaves should be drawn (See Acknowledgements). He also suggested that without classes it was going to be difficult to manage the code.
**Code rewritten to be properly object oriented**

After some research into JavaScript I found that classes can be implemented in JavaScript\(^ {77}\) (I knew about JavaScript objects but I found out how to associate methods with the objects). I rewrote the code with proper classes. I discarded about 90\% of the old code. I redesigned the data structure around an array\(^ {78}\) to hold the incoming cluster data and a binary tree to represent the dendrogram. (See The Code section, “How the tree structure is built” and “How the dendrogram is drawn”. That is one part of the code that was produced during this rewrite.) See the figure below. Although it may look similar to that of 9\(^ {th}\) May, most of the change in code was behind the scenes. The same Raphael drawing methods were being used but with some styling changes such as line thickness. However, the leaf order is different resulting in the “singleton” leaf being drawn in the middle. This is as a result of the new drawing sequence. (See Diary and Development Appendix 15/5/2011)

![Figure 7.3 – Screenshot from prototype 16\(^ {th}\) May showing a portion of the screen. Dendrogram drawn from a small hard-coded data set.](image)

\(^{77}\) [Flanagan 2002, page 119]

\(^{78}\) JavaScript arrays do not require to have their length declared when they are established. In this respect they are similar to Java ArrayLists.
Cluster bounding boxes and Events added. Also larger data set.

See figure below. (See Diary and Development Appendix 23/5/11)

![Figure 7.4 Mouse-over and Mouse-out events were added to the cluster bounding boxes. A 100 leaf data set was hard coded.](image)

Meeting with The Clients (23/5/2011)

I showed the new version of the prototype to The Clients (See Requirements Analysis, “The Clients”). It was decided that on the loading data feature I was to spend only the minimum of time needed to develop loading of data sufficient to test the application. It was deemed likely that a new specification for the data load feature would be devised once The Dendrogrammer became part of the ICT Perspectives site and the analysis of that could not yet be done. I was to focus on the following:

1) The interactivity aspect (adjustable groupings via a similarity threshold slider)
2) Addition of a short and long description property to each node
3) Display and User-interaction with a cluster’s leaf list and short description and, following that:
4) The scaling aspect: to do with coping with the larger data sets.

(See Diary 23/5/2011)
Prototype reads data from csv file

The file could be selected from a table of links which invoked PHP scripts. The PHP scripts formed a sequence of two pages

- `csvfileread_buildform.php` read the data file and allowed it to be submitted as POST data. Or it allowed different data to be loaded via a link to itself with a query string specifying the file to load (as GET data).
- `accept_post_data.php` rendered the data into the page as hidden form elements for the JavaScript to access. It passed the data to the application page via POST data. It also had a link

(See Diary and Development Appendix 29/5/2011)

Clicking a cluster reports descendant leaf data

Clicking a cluster caused a list of the cluster’s descendant leaves to be displayed in a div panel above the dendrogram. (See Diary and Development Appendix 3/6/2011)

Threshold adjusted and groups formed by using slider widget

A jQuery UI slider widget was used to adjust the threshold at which to form groups from the clusters. Methods manipulating new `Group` and `GroupList` classes populated the group list using the set threshold and the data held in the `ClusterTree`. The list of groups was displayed in a panel above the dendrogram.

(See Diary and Development Appendix 5/6/2011).

Native Browser Zoom found to be of use

A friend showed me that browsers have a native zoom facility with the control key and plus or minus (see Acknowledgements). When I showed this facility to my Client he agreed that this did what was required by way of zooming in and out.

(See Diary 7/6/2011).
**Introduced jQuery UI tabs to rationalise data display**

The use of the jQuery UI tabs effect allowed the groups listings and various other data displays to be rationalised into the tab sections. This freed up vertical space for the dendrogram. See figure below.

![Diagram](image-url)

**Figure 7.5 Prototype at 8/6/2011**

(See Diary and Development Appendix 8/6/2011).
Cluster-click and group threshold report to jQuery dialogs

See figure below. (See Diary and Development Appendix 12/6/2011).

Figure 7.6 Prototype at 12/6/2011 showing dialog.

20/6/2011 Implemented drag-able threshold adjuster.

(See Diary and Development Appendix 20/6/2011).

Figure 7.7 Prototype at 20/6/2011 showing drag-able threshold

Cluster descendants reports to dialog

Drag-able threshold adjuster bar
Groups formed by specifying number of groups.

Button toolbars were added and the methods for specifying the number of groups and automatically generating a matching threshold were implemented.
(See Diary and Development Appendix 26/6/2011).

Meeting with The Clients (1/7/2011)

We established a list of desired features not yet in the application.
1) Leaf labels down the y-axis
2) Ability to make a “summary” dendrogram. Effectively truncating the dendrogram up from 0 ht to a level at which the structure can be discerned and then the clusters interrogated, rather than having hundreds of leaves along the bottom.
3) Read in of leaf data, in addition to clustering data (which is already read). Leaf data was to be read from a separate file. The names of the two files to be specified in a third “manifest” file, which is chosen by the user, and cause the two data files to be read. (See Diary 1/7/2011).

File read rationalised.

The manifest file and leaf data file read was implemented. The file reading was rationalised and brought within the application page eliminating the need for additional PHP scripts posting data between scripts. All the data read commands and display are now in the data tab.

Also a no-data-found default was coded to use a small hard coded demo data set.
(See Diary and Development Appendix 6/7/2011).

Leaf Labels implemented and tested with very large data files

The leaf labels were implemented. They highlighted in response to mouse-over/tap on the clusters. Large data files (200 to 500 leaves) were constructed from simulated data made in spreadsheets. The leaf labels added too great a processing load for Internet Explorer 8. It slowed significantly in responding to events. In other browsers and in IE9 it still worked well. (See Diary and Development Appendix 8/7/2011).
Meeting with The Clients (13/7/2011)

It was desired that The Dendrogrammer “interface” with one the ICT Perspectives site components, a search engine script. What was wanted was that the node descendants dialogue have a hyperlink to the search script to carry the leaf list into that script. This was related to the Word cloud optional requirement (which it had been decided could not be implemented for now as data to inform Word cloud construction was not available). However the hyperlink call to the search engine would establish one half of that. Later the application will require modified to accept a response to the leaf data query (which will contain the data for word cloud construction).

(See Diary 13/7/2011).

Summary dendrogram feature completed

The truncate/restore features were completed. The creation of this feature had two revisions following pursuing earlier wrong directions. (See Diary 14/7 to 17/7/2011 and Development Appendix 17/7/2011).

Code Documentation pages established

Using YUIDoc Yahoo API documentation engine and a customised template and batch file The Dendrogrammer Code Documentation web pages were produced. They reside in a directory (Docs) beside the application and libraries directories http://www.macs.hw.ac.uk/~dar14/project/dendrogrammer/doc/index.html

(See Diary 30/7/2011 and Development Appendix 13/8/2011)

Production version established

This includes a Readme file to guide installation.

(See Diary 2/8 to 13/8/2011 and Development Appendix 13/8/2011)

Usability evaluation conducted

Although this was intended as a “summative” evaluation I felt a couple of items raised in answers to the open questions pointed to omissions in the application. Therefore those were addressed through some minor changes.
(See Diary 10/8/2011 and 13/8/2011)

Problems and solutions (Section 8)

Table 8.1 below describes some of the specific difficulties encountered during the project and how these were overcome. This is intended to give a flavour of the kind of problems faced and varying approaches to solving them. Sometimes a problem was tackled head-on, such as when the early prototype code became unmanageable and was completely rewritten. Sometimes a problem was side-stepped such as when difficulty was encountered assigning events to Raphael composite objects, so the composite object was redesigned to create it from one single drawn object.

<table>
<thead>
<tr>
<th>Problem encountered</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early code was not properly object oriented and difficult to manage. (I had been</td>
<td>Researched the implementing of classes in JavaScript. Code extensively rewritten using classes with methods.</td>
</tr>
<tr>
<td>held back by my previous limited experience of JavaScript which had coloured my</td>
<td></td>
</tr>
<tr>
<td>view of the language. I was still only viewing it as a relatively simple language.</td>
<td></td>
</tr>
<tr>
<td>)</td>
<td></td>
</tr>
<tr>
<td>The algorithm for building the dendrogram worked for the small example data set but</td>
<td>New way of drawing the tree implemented based on the order that leaves were encountered on a depth-first search of the data.</td>
</tr>
<tr>
<td>used an iterative redraw method working from the leaves up. It would become slow for</td>
<td></td>
</tr>
<tr>
<td>large data sets</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1a – Some Problems and their solutions.
<table>
<thead>
<tr>
<th><strong>Problem encountered</strong></th>
<th><strong>Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Difficulty in assigning events to Raphael shapes made from a composite of 2 or more</td>
<td>Redesigned the leaf object as a single drawn object.</td>
</tr>
<tr>
<td>drawn objects:</td>
<td></td>
</tr>
<tr>
<td>IE9 slipping into compatibility view.</td>
<td>Researched IE9 behaviour. Added a meta tag to force IE9 out of compatibility view.</td>
</tr>
<tr>
<td>The overlay of group bands obscured elements under them from receiving events.</td>
<td>Tried using Raphael to send to back, but this was not effective across all browsers. Finally went with assigning the events to the overlay bands themselves instead.</td>
</tr>
<tr>
<td>Classes in JavaScript make extensive use of the keyword, <em>this</em>, to refer to their</td>
<td>Overcome by passing a named reference to the class instance as an argument whenever events were being handled.</td>
</tr>
<tr>
<td>associated methods and properties from within their scope.</td>
<td></td>
</tr>
<tr>
<td>However during events the keyword, <em>this</em>, is taken over by the event and refers to</td>
<td></td>
</tr>
<tr>
<td>the event object.</td>
<td></td>
</tr>
<tr>
<td>The dendrogram was rendering as narrower on iPad.</td>
<td>First detected iPad by interrogating the browser user agent string and adjusted the scale factor specifically for iPad. Eventually this was overtaken when scaling was linked to window width rather than screen width.</td>
</tr>
<tr>
<td>The use of a jQuery slider widget to set the threshold led to a lack of precision in</td>
<td>Went for a more bespoke solution. Replaced the widget with a drag-able Raphael element on the graph itself.</td>
</tr>
<tr>
<td>threshold placement due to the slider being separate from the graph.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 8.1b – Some Problems and their solutions.**
The Structure of the application (Section 9)

This section describes

- The directory structure of the application and the purpose of the JavaScript Libraries.
- How data flows in the application (how PHP is used to load data files from web space.). Including how the data is accessed by the JavaScript part of the application.
- How the application JavaScript classes interrelate.

The Files

The application files must reside in web space served by a PHP-enabled web server. They consist of the following elements:

- The Application directory containing:
  - A PHP / HTML page – This links in all the CSS and JavaScript files from the Application directory and the Libraries directory
  - A CSS style sheet
  - The application JavaScript file
  - The Data directory containing the data files
- A Libraries directory containing:
  - The Raphael JavaScript Library file provides the SVG Graphics methods.
  - The jQuery JavaScript Library file provides methods for cross-browser Dynamic HTML
  - Portions of the jQuery UI JavaScript Library. This consists of a number of JavaScript files and accompanying CSS style sheets. jQuery UI provides methods for cross-browser dynamic user interface elements such as the tabs effect.

See Figure below.
Figure 9.1 The Directory Structure
How data is loaded by the application

The figure below shows how the data is made available to the JavaScript part of the application.

**PHP pre-processor**
- Data in CSV files residing on web server
- PHP reads in a set of data files
- PHP pre-processor embeds data in HTML form elements

**Web server**
- CSS
- JavaScript
- HTML page. Data in hidden form elements.
- Application files served to the browser

**Web browser**
- JavaScript accesses data embedded in the HTML page elements.
- Application page rendered
- Hyperlink requests PHP page again. GET data is passed in a URL query string specifying data file to load.

Figure 9.2 How data is loaded.
The JavaScript Classes

The figure below shows the dependencies of the JavaScript classes. This view excludes the jQuery and Raphael functions. Those are made available globally to the rest of the application.

Dendrogrammer Class Dependencies showing Multiplicities.
The arrows indicate that a class knows about another class. i.e. it is passed a reference to an instance of the pointed to class in its constructor parameters or at some point it makes use of that class.
The number 1 indicates that there is only one instance of the class in existence at any given time.
0..* indicates that there will be multiple instances of that class.

Figure 9.3 - Dendrogrammer Class Dependency Diagram.
The Code (Section 10)

The reader may wish to refer to the Code Appendix or to The Dendrogrammer Code Documentation web pages (http://www.macs.hw.ac.uk/~darl4/project/dendrogrammer/doc/) while reading this section. If referring to the Documentation web pages, note that, in addition to the Class links on the left hand navigation panel there is also a “Files” link, which displays the full JavaScript code with code highlighting.

This section describes

- the approach to commenting the code
- some of the code techniques used
- some key parts of the code

Comments

The code is liberally commented. One reason for this is coding style (I tend to write comments before I write a block of code.) However, one of the project requirements is that it be highly maintainable as it is to be adapted to as yet unknown requirements by the developer working on the ICT Perspectives project. To this end I investigated, installed, and configured YUIDoc\(^{80}\) and used it to create an automated code documentation site customised for the project code. This was a time-consuming exercise however it had two very tangible benefits

1) It forced me to formalise my comments. I had to apply complete consistency in style to the documentation comments. This line-by-line re-visiting of the comments has resulted in a high level of consistency.

2) I have found the documentation site itself very useful. Since creating it I find it is my first port of call when addressing the code.

\(^{80}\) [Yahoo Inc 2011]
The Structuring of the JavaScript Classes

The following code excerpt shows the ClusterTree class with some of its comments edited out. It is annotated here as an example of how a class is declared in the code.

```javascript
//######################################################
//ClusterTree class
/**
 * The ClusterTree class holds a tree of clusters. Actually...
 * @class ClusterTree
 */
//Define functions for the methods
/**
 * @method getRoot
 * @return {node} the root node in the tree
 */
function ClusterTree_getRoot(){
    return this.root;
}
/**
 * Recursively traverses the tree setting the yPos attribute...
 * @method setClusterY
 * @param node {ClusterNode object}
 * @return {boolean} value not used.
 */
function ClusterTree_setClusterY(node){
    //if the yPos if childA is null then set its yPos
    if (node.childA.yPos==null){
        this.setClusterY(node.childA);
    }
    //if the yPos of childB is null then set its ypos
    if (node.childB.yPos==null){
        this.setClusterY(node.childB);
    }
    //finally set the yPos based on yPoses of children
    if (node.childA.yPos<node.childB.yPos){
        //set ypos to that of child A + difference between A and B
        node.yPos=node.childA.yPos+(Math.abs(node.childA.yPos-node.childB.yPos)/2);
    } else {
        //set ypos to that of child B + difference between A and B
        node.yPos=node.childB.yPos+(Math.abs(node.childA.yPos-node.childB.yPos)/2);
    }
    return true;
}
/**
 * Calls the new ClusterNode constructor method which in turn calls the...
 * @constructor ClusterTree
 * @param cTableIn {ClusterTable} the instance of ClusterTable
 * @param stylesIn {style object} carries the style info for local access
 */
function ClusterTree(cTableIn,stylesIn){
    //initialise attributes
    this.cTable=cTableIn;
    this.styles=stylesIn;
    //the root node is the node from the end of the clusterArray
    //i.e. that cluster with the highest cluster height
    var rootIdNo=this.cTable.getLast().idNo;
    //This line causes a recursive all on the ClusterNode Constructor
    //in conjunction with the data in the cluster table which fill up
    //the tree
    this.root= new ClusterNode(this.cTable,rootIdNo,null,0,this.styles);
    //associate methods
    this.getRoot = ClusterTree_getRoot;
    this.setClusterY = ClusterTree_setClusterY;
    //processing to be done by the constructor
    //set the yPos values for all the clusters
    this.setClusterY(this.root);
}
//End of ClusterTree class
//######################################################
```

Documentation comments with YUIDoc @labels to facilitate automated documentation.

Comment indicating the start of method declarations

A method declaration. This is simply a JavaScript named function declaration.

Another method declaration.

Constructor declaration. The constructor is simply a function with the same name as the class which, when called with the “new” keyword, instantiates the object.

The “this” keyword is used to assign the attributes which are JavaScript object properties.

A section of the constructor must associate the method function names as properties of the object.
One thing to note about a JavaScript class is that it consists of a group of function declarations all at the same level of scope. They are connected together by the assignment of the method function references to object properties inside the class’s Constructor function. It is a feature of JavaScript that it allows a function to be assigned to a variable name.

**Coding style**

The coding style I have used is more verbose than it would be were the aims purely to produce the fastest running most efficient solution. For example I have made relatively little use of the anonymous function and so there will be some objects in memory that perhaps do not need to be there. However, I have known, from early on in the project that the code was expected to be maintained and developed by someone else. So an abbreviated JavaScript style such as would be appropriate for a JavaScript API framework, say, would not be appropriate here.

**Key parts of the code**

I describe four key parts of the application code:

- How the tree data structure is built.
- How the dendrogram is built
- How shapes are drawn
- How the data is read

**How the tree structure is built.**

The tree structure is built of `ClusterNode` objects. In fact the `ClusterTree` is defined purely by reference to its root `ClusterNode`. All other `ClusterNodes` in the `ClusterTree` are descended from the root.

Some of a `ClusterNode`’s attributes are:

- `nType`: 1 indicates a cluster which has children. 2 indicates a leaf (no children)
- `parent`: a reference to another `ClusterNode` above it on the tree (or null for the root node).
• childA: a reference to another ClusterNode below it on the tree (or null for a leaf).
• childB: a reference to another ClusterNode below it on the tree (or null for a leaf).
• element: a drawn element on the chart
• Other attributes include the node’s height on the dendrogram (xHt), and the drawn graphic element representing it on the chart (element). See the Code Section for a more complete description.

See figure below.

A ClusterNode’s attributes:
nType, parent, childA, childB, xHt, element…..

Figure 10.1– A ClusterNode object. Listing some attributes and depicting the ClusterNode graphically

To build the ClusterTree all that need be done is to supply the ClusterNode constructor with the ClusterTable and the ID number of the root node. The ClusterNode constructor then does the following
1) retrieves the root node details from the ClusterTable (by ID number),
2) assembles the root ClusterNode and then
3) calls the ClusterNode.grow() method with the root ClusterNode as the parameter.
   This “grows” the branches first down childA and then down childB, by calling the ClusterNode constructor with that new cluster’s ID. A branch is not “grown” any further when a leaf node is encountered. When the ClusterNode constructor is asked to construct a leaf node, it does not call the grow() method. Instead it simply sets the leaf node’s childA and childB to be null.

In the table below is the example clustering data described in the Requirements Analysis Section, but labelled in accordance with the ClusterTable attributes.
Columns ItemA and ItemB hold the cluster ID numbers. It describes leaf clusters 1 to 6, clusters 7 to 10 and the root cluster. (The root cluster ID of 11 is implicit as being 1 over the last explicit cluster ID. It is always described by the last row, which has the highest $xHt$). See Requirements Analysis Section for a full explanation of this data.

<table>
<thead>
<tr>
<th>ClusterTable index</th>
<th>ItemA</th>
<th>ItemB</th>
<th>xHt</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 10.1 – Example clustering data**

This ClusterTable results in the construction of a tree as shown in the figure below. In turn the method calls and objects involved in the construction of the emboldened part of the tree shown below (Figure 10.2) are described in the sequence diagram (Figure 10.3) below that.
Figure 10.2 – Representation of the ClusterTree built from the example data in the table above.
(The portion of the tree in bold is that encompassed by the method calls shown in the sequence diagram in the next figure.)

The construction is effectively a depth-first search of the data. (This can be likened to the method sometimes used to solve a maze puzzle by turning right whenever possible.) In the next part of this section “How the dendrogram is built” there is a figure which illustrates this.
The ClusterTree, and ClusterNodes exist for the life of the application (until it is exited or reloaded). Likewise the ClusterTable, except after the tree is constructed it is not referred to again.

Further new ClusterNodes are created until the tree is complete.

Figure 10.3 – Sequence diagram showing the key method calls involved in constructing the ClusterTree from the data in the ClusterTable.
How the Dendrogram is built

One can consider that in representing each cluster on the dendrogram there are three crucial aspects that must be known for each cluster:

1) The dissimilarity or merge height of the cluster (0 for leaves at the bottom, or maximum for the root cluster at the top.) Due to the dendrogram being built on its side in the application, this is termed the xHt.
2) The clusters it connects to (i.e. its children and parent).
3) Its position along the leaf axis. Its yPos.

The xHt is straightforward. It is read in with the data and stored as a property of each ClusterNode.

The tree structure is key to items 2 and 3 above. Not only do the parent/child links represent the connections of the clusters, but the order in which the leaves “sprout” during ClusterTree construction holds the key to the yPos for each cluster.

Figure 10.4 – The order in which the ClusterTree is constructed. The “leaf-sprout” order is shown by the bold numbers in boxes.
The order in which the leaves are encountered during the construction of the tree from the ClusterTable is stored as a property of each leaf’s ClusterNode. It is used to plot each leaf’s position on the leaf axis (its yPos). The figure above shows the order in which the tree is constructed and leaves “sprouted”, for the example data. The yPos for each non-leaf cluster is derived from the yPos of the cluster’s children. In our example data, cluster ID9 has children ID2 and ID5, both of which are leaves. Cluster ID9’s yPos is half way between the yPos of ID2 and ID5. The yPos of leaf cluster ID2 is 1 (its leaf sprout order). The yPos of leaf cluster ID5 is 2 (its leaf sprout order). Therefore the yPos of cluster ID9 is 1.5 (half way between 1 and 2). So the leaf sprout order of the leaves determines their own and all their ancestors’ yPos’s all the way back up the tree. (Remembering that the tree is constructed on its side and yPos is the distance along the leaf axis.)

The Dendrogram class contains most of the drawing methods. e.g. drawLeaf() which draws the graphical element for the leaf. However, some are ClusterNode methods. (see Code Appendix).

**How shapes are drawn**

The shapes are drawn using methods of the Raphael object which is assigned to the Canvas.paper attribute. In the case of the Dendrogram. drawLeaf() method the Raphael path() method is used. The path() method draws out a path given a starting point and relative way-points along the path.

An example of a leaf graphic is shown on the figure below. It is made of a path which traces a box around the leaf point and then continues as a single line on to the xHt (distance up the dendrogram) of the leaf’s parent cluster.
Figure 10.5 – A leaf graphic with its leaf point (top) and the way-points visited when it is drawn by the Raphael `path()` method.

Below is the code for a simpler path which traces out a “U” shape. (The origin is top left of the canvas the X direction is left to right and the Y-direction is top to bottom.)

`paper.path("M 0 0 L 0 4 L 3 0 L 0 -4")`

The English translation of this would be:

- Move (M) to the point (0,0). ("M 0 0")
- Trace a line moving zero in the X-direction and 4 in the Y-direction (“L 0 4”)
- Trace a line moving 3 in the X-direction and 0 in the Y-direction (“L 3 0”)
- Trace a line moving zero in the X-direction and -4 in the Y-direction (“L 0 -4”)

This would trace the path shown in the figure below.
Figure 10.6 – A simple U shaped path.

The path parameter is just a string and can be concatenated from numeric variable values and characters. The Raphael \texttt{path()} method is very similar to the SVG path command in that it requires a string parameter of the same syntax.

The \texttt{path()} method is used for the axes, the leaves, and the clusters. The \texttt{rectangle()} method is used for the bounding boxes (which describe the “live” area for a cluster to sense events) and for the group colour bands. The \texttt{text()} method is used to draw the leaf and axis labels.

How the data is read

The data is read from CSV files using the standard PHP file handling functions \texttt{fopen()}, \texttt{fgetcsv()}, and \texttt{fclose()}. I based my code on examples at PHP.net\textsuperscript{81}

Once the data item is read in, it is echoed into the HTML as hidden form elements. Each element is given a name property based on its originating table, the row and column. An excerpt from one such line is shown below.

```php
echo '<input name="node'.$row.'col'.$c.'" type="hidden" value="'.$data[$c].'">';
```

Here, \texttt{$data[$c]} is the data item being written to the HTML; \texttt{node} is the originating table (this is the table containing the data for each cluster); \texttt{$row} and \texttt{$c} are the row and column numbers. If, say, the data item were 7.34, the row and column were 5 and 3 respectively, then the following HTML input element would be echoed to the HTML page output:

\hspace{4cm}

Inside the JavaScript code in the ClusterTable.getData() method the data is extracted from the HTML elements using jQuery techniques as shown in the annotated code below. jQuery is used here as there are cross-browser problems associated with accessing data held within the DOM on a web page. Different code is often needed depending on the browser. However, jQuery takes care of that so my code is cross-browser.

```javascript
// gather input elements in a jQuery object
var $allInputElements = $(':input');

// create an associative array of all the input elements
var inputValues = {};

$allInputElements.each(function() {
    inputValues[this.name] = $(this).val();
});
```

The result is an associative array with all the input element names keyed to their values. So the array entry associated with our example input element would be:

`'node5col3' => '7.34'`

Then the JavaScript reconstructs the tables as JavaScript arrays by iterating through the associative array. There are 3 arrays

- `theClusterTable.datArray`: An array for the information from the `dat` file. Each element holding the string item from one file row.
- `theClusterTable.cArray`: An array for the `node` table. Each item in the array is an object representing a row in the `node` table. (The `c` standing for cluster).

---

82 [Chaffer and Swedberg 2011]
theClusterTable.leafArray: An array of objects corresponding to each row from the ‘leaf’ table.

The three arrays are attributes of theClusterTable, the single instance of the ClusterTable class.
Evaluation (Section 11)

I will describe the evaluation in three sub-sections:

- Formative evaluation (during the project)
- Summative evaluation (at the end of the project)
- A critique of the application including suggestions for future alterations or additions.

Formative evaluation (during the project)

“Formative” evaluation was explicitly built into the schedule in the project plan as phases of prototype “testing and evaluation” (See Planning Appendix). However, such evaluation was less regimented and indeed much more frequent than the schedule suggested. My supervisor who was the client for the project was shown each increment to each prototype by emailing the latest prototype URL, sometimes on a daily basis. There were several meetings where the prototypes were evaluated, some involving the intended user and onward developer of the application. Some aspects of the requirements e.g. the data input feature, varied slightly during the project (as is natural in most software developments) such evolution of the requirements arising from the meetings.

In addition, some feedback from the Evaluation task (which was originally intended as a “Summative” Evaluation) was used in a formative way in that some minor additions were made to the “Production” version following that. (See below).

Summative evaluation (at the end of the project)

In this sub-section the following is addressed:

- The provision of features as assessed by the checklist of functionality as laid out at the start of the project, modified in line with changes agreed with The Clients.
- The quality of the usability as assessed by the evaluation task.
Checklist of functionality

The tables below contain the checklists of functionality produced at the start of the project (in the first two columns). A third column now details whether or not a given criteria has been met.

Access

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Criteria</th>
<th>Criteria met?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory Accessible via the World Wide Web in a web browser</td>
<td>Can the application be accessed via the WWW?</td>
<td>Yes</td>
</tr>
<tr>
<td>Mandatory Support the latest major browsers: Internet Explorer, FireFox, Safari, Chrome.</td>
<td>A survey of the browsers that the application functions on noting any degradation in function.</td>
<td>The latest major browsers are supported. See details below this table</td>
</tr>
<tr>
<td>Optional Support for earlier versions of the major browsers.</td>
<td>As above</td>
<td>Some earlier versions are supported. However, performance is slow in IE7 and 8 with large data sets. See details below this table</td>
</tr>
<tr>
<td>Mandatory Cross platform: accessible on PCs and the iPad.</td>
<td>Does the application function on PC and the iPad for all mandatory functions?</td>
<td>It functions well in Safari on iPad</td>
</tr>
<tr>
<td>Optional Accessible on other tablet computers. Accessible on other mobile devices such as iPhone and Android.</td>
<td>A survey of the devices that the application functions on noting any degradation in function.</td>
<td>It renders well on the Android slate but performance has noticeable delays. See detail below.</td>
</tr>
</tbody>
</table>

Table 11.1 Access functionality checklist
Details of browser and Android slate performance test

Appearance
It renders correctly in Firefox 3.6+, Safari 5.0+, Chrome 5.0+, Opera 11.5+ and Internet Explorer 7+

The jQuery toolbar buttons render acceptably in Internet Explorer 7

(The Raphael documentation claims that the graphics should render properly in Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+. 83

jQuery UI claims to render properly in “IE 6.0+, Firefox 3+, Safari 3.1+, Opera 9.6+ and Google Chrome” 84. There is the slight discrepancy here compared with my results for IE7)

Function
It functions well, without noticeable processing delays, in all the non-Internet Explorer tested browsers and indeed also in Internet Explorer 9.

It is slow in IE 7 and 8. It is particularly slow, indeed unacceptably so with large data sets, on less modern equipment with those browsers.

Mobile devices other than iPad:
Android Slate (Asus Transformer running Android Honeycomb v3.1)
Everything renders correctly. With the 100 leaf data there is a noticeable delay in responding to button taps. There is a noticeable lag in threshold drag response.

---

83 http://raphaeljs.com/
84 http://jquery.com/
Basic functionality

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Criteria</th>
<th>Criteria met?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Load the data from a suitably formatted file on the World Wide Web specified by URI</td>
<td>Can the data be loaded over the WWW?</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Draw a dendrogram from clustering data</td>
<td>Does the application draw a dendrogram to accurately portray the clustering data?</td>
</tr>
</tbody>
</table>

Table 11.2 Basic functionality checklist

Additional requirements

Two requirements added during the project:

1) See Key Stages section “Meeting with The Clients (1/7/2011)”

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Criteria</th>
<th>Criteria met?</th>
</tr>
</thead>
<tbody>
<tr>
<td>File loading handled by manifest file and separate node and leaf data files.</td>
<td>Data files can be specified in a manifest file.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 11.3 Additional requirements (1)

2) See Key Stages section “Meeting with The Clients (23/5/2011)”

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Criteria</th>
<th>Criteria met?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide for Short and Long description for leaves and clusters within the data structure.</td>
<td>Is such provision in the data structure?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 11.4 Additional requirements (2)
### Interactivity

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Criteria</th>
<th>Criteria met?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjustable similarity threshold to highlight groups of leaf nodes</td>
<td>Can cluster groups of leaf nodes be highlighted using an adjustable threshold?</td>
<td>Yes</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to specify an arbitrary number of groups and set the similarity threshold to fit</td>
<td>Can the desired number of groups be specified by the user, and that specified number of groups are highlighted?</td>
<td>Yes</td>
</tr>
<tr>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select a leaf and copy text associated with it</td>
<td>Can the text associated with a leaf node be copied to the clipboard?</td>
<td>Yes</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select a leaf or cluster and view a word cloud associated with it</td>
<td>Can a word cloud associated with a leaf be viewed? Can a word cloud associated with a cluster be viewed?</td>
<td>No. This original optional requirement was replaced by that detailed on the next row.</td>
</tr>
<tr>
<td>Ability to pass the leaf list short descriptions from a cluster to a search engine.</td>
<td>Is a search engine link provided with a cluster's leaf list?</td>
<td>Yes See details below.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to zoom in and out to reveal more or less detail of the dendrogram</td>
<td>Can the dendrogram view be zoomed in and out?</td>
<td>Yes. There are two ways. 1) Native browser zoom. 2) Summary dendrogram feature.</td>
</tr>
</tbody>
</table>

**Table 11.5 Interactivity checklist**
**Detail of search engine hyperlink**

See Key Stages section “Meeting with The Clients (13/7/2011)”

When a cluster is clicked, a list of the short descriptions from its descendant leaves is presented in a dialog allowing copy/paste along with a hyperlink that invokes a search engine to search for the terms in the leaf list. The URL for use with this can be specified in the manifest file (and is intended to be set to a specialised ICT Perspectives search engine currently in development) else it defaults to Google.

**Conclusion (of evaluation by feature checklist)**

The evaluation by checklist shows that all the requirements were met with the following exceptions:

1) The word cloud optional requirement was replaced during the project by the requirement to provide a cluster’s leaf list to a search engine by a hyperlink. This alternative requirement was met.

2) With large data sets Internet Explorer versions 7 and 8 cannot cope with the JavaScript processing load. It would be reasonable to say that this rules out calling The Dendrographer truly cross-browser (at this time).

**Usability as assessed by the Evaluation Task**

- What the Evaluation was designed to do
- What the evaluation was not designed to do
- The users doing the evaluation
- The Evaluation conditions
- The Results and Conclusions from the Evaluation

**What the Evaluation task was designed to do**

An evaluation task was designed\(^{85}\) to assess usability in terms of

1) Are users able to carry out the tasks for which this application was designed

2) The opinion of the users as to the ease of use of given features

\(^{85}\) Advice was given by F.Halley. See Acknowledgements
3) The users were given the opportunity to give their opinion on 3 aspects of the application:

- Navigation,
- Responsiveness of the dendrogram and,
- Presentation of the commands.

4) The users were given the opportunity to comment on any other aspect they wished.

The text of the task is in the Evaluation Appendix.

The various sections of the evaluation task were aimed at establishing if users could carry out tasks related to specific aspects of the requirements specification. The tables below map the task sections to those requirements:

<table>
<thead>
<tr>
<th>Section</th>
<th>Purpose: To…</th>
<th>Related requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section A</td>
<td>Establish that the user can load the application</td>
<td>Accessible via the World Wide Web in a web browser</td>
</tr>
<tr>
<td>Section B</td>
<td>Establish that the user can load a new data set.</td>
<td>Load the data from a suitably formatted file on the World Wide Web specified by URL</td>
</tr>
<tr>
<td></td>
<td>Introduce the user to the data so that the user begins to understand that the leaf items share a level of dissimilarity and how this is stored within the application.</td>
<td></td>
</tr>
<tr>
<td>Section C</td>
<td>Establish that the user can navigate around the dendrogram and interpret the data.</td>
<td>Draw a dendrogram from clustering data</td>
</tr>
</tbody>
</table>
|           | Establish that the user can select a dendrogram element and copy text associated with it. | Select a leaf and copy text associated with it                                        

Table 11.6a Evaluation task mapped to requirements
<table>
<thead>
<tr>
<th>Section</th>
<th>Purpose: To…</th>
<th>Related requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section D</td>
<td>Establish that the user can adjust the similarity threshold</td>
<td>Adjustable similarity threshold to highlight groups of leaf nodes</td>
</tr>
<tr>
<td>Section E</td>
<td>Establish that the user can adjust the similarity threshold and can create groups using that threshold setting. Establish that the presentation of the group list allows the user to analyse the list and select details from within it. Establish that the user can create groups using a precise threshold setting by entering the threshold numerically at the keyboard.</td>
<td>Adjustable similarity threshold to highlight groups of leaf nodes</td>
</tr>
<tr>
<td>Section F</td>
<td>Establish that the user can form groups by specifying the desired number of groups.</td>
<td>Ability to specify an arbitrary number of groups and set the similarity threshold to fit</td>
</tr>
<tr>
<td>Section G</td>
<td>Establish that the user can create a summary dendrogram and manipulate it to view the data.</td>
<td>Ability to zoom in and out to reveal more or less detail of the dendrogram</td>
</tr>
</tbody>
</table>

Table 11.6b Evaluation task mapped to requirements

Answers to questions about ease of use of a given feature was restricted to a Likert scale consisting of the choices, “very easy”, “easy”, “neither easy nor difficult”, “difficult”, and “very difficult”. The purpose of this was two-fold. Firstly to produce a standardisation of response while still offering an ordinal range. Secondly, to help reduce the time spent on the task for the respondents.
**What the evaluation task was not designed to do**

The evaluation task does not assess usability on the iPad. I was limited in time available to do the evaluation. So I limited the scope of the evaluation. I would have had to have found and organised several experienced iPad users. From my own experience with iPad I know that using an iPad is an acquired skill. I could not have simply handed some volunteer non-iPad users my iPad and got a meaningful usability assessment from them. I decided to make do with my own testing on the iPad which has showed all The Dendrogrammer functions can be carried out on the iPad.

**The users doing the evaluation**

The target users were considered before seeking volunteers for the task. i.e. EPSRC researchers (who could be expected to have a good knowledge of cluster analysis) and also delegates, attending consultations, not necessarily possessing cluster analysis knowledge.

Volunteers were sought from among

a) MSc computing students as likely to represent users who do not necessarily have a great deal of knowledge about cluster analysis.

b) University researchers, some of whom will represent users with the same level of cluster analysis knowledge as the MSc students, but a proportion of them are known to have a good knowledge of cluster analysis and had used dendrogram tools for cluster analysis before.

In total 8 volunteers completed the evaluation task.

**The Evaluation conditions:**

For consistency the task was completed using Firefox browser. Similar computer equipment was used. All were done on PCs.

The task involved locating information on the dendrogram, copying it and pasting it into boxes on the electronic task document. The lab researchers were able to use two screens, one for the task document and on for the Dendrogrammer. The MSc students
had single screen equipment but were provided with a printed copy of the task on which they ticked off steps as they worked.

The Results and Conclusions from the Evaluation

The task questions results.
Were the users able to perform the tasks with The Dendrogrammer?
1) No user found any of the tasks was impossible to complete.
2) All of the tasks involving locating information on the dendrogram, copying it and pasting it were done accurately.
3) Some tasks involved examining information presented in the dendrogram and interpreting it. Those were: finding a cluster of between 4 and 6 leaves, and picking out the largest and smallest groups from the groups list. Of these tasks there was only one single error. That was in identifying the smallest group.
4) One task involved placement of the threshold at a particular value (5.6) with 5 significant figures displayed. The users were not asked to be especially accurate and all were sufficient to achieve the same cluster group construction. The achieved threshold adjustment values ranged from 5.5045 to 5.6549, a range of 0.1504 on a scale that ranged from 0 to 9.7272. The threshold value furthest from the target was 5.5045 which is 0.0955 too low, out by less than 1% (0.98%).

Task questions conclusion
The Dendrogrammer can be used successfully to access its information interactively. With the exception of one error all of the tasks were completed accurately. The users set the drag-able threshold with a good level of accuracy even when not urged to be particularly accurate.
The Closed response opinion questions results and conclusions

The results and initial conclusions are dealt with facility by facility:

Loading Data

Loading Data Results:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Response</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>loading a new data set</td>
<td>very easy</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>5</td>
</tr>
<tr>
<td>accessing the information in the data tab</td>
<td>very easy</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>neither easy nor difficult</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 11.7 Loading Data Results

Loading Data Conclusions:

Loading data is found to be easy. With 3 neutral responses, accessing the information in the data tab was not found to be emphatically easy. I think this is probably to be due to the need to scroll down to find information in the tables displayed for data confirmation.

Interactive Nodes

Interactive Nodes Results:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Response</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessing a node’s information</td>
<td>very easy</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>neither easy nor difficult</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 11.8 Interactive Nodes Results

Interactive Nodes Conclusions:

Most found accessing a node’s information to be easy.
Threshold and groups

Threshold and groups Results:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Response</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjusting the threshold</td>
<td>very easy</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>3</td>
</tr>
<tr>
<td>accessing the information about the groups</td>
<td>very easy</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>3</td>
</tr>
<tr>
<td>creating groups by specifying a particular number of groups</td>
<td>very easy</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 11.9 Threshold and groups Results

Threshold and groups Conclusions:

Adjusting the threshold is found to be easy. This ties in with the reasonably small range of threshold figures achieved in the task.

Accessing the information in the Groups tab is found to be easy.

Creating groups by specifying a particular number of groups using the Set no of groups toolbar is predominantly found to be very easy. (By that stage in the tasks the users would have the experience of setting the threshold manually to compare this with.)

Summary Dendrogram

Summary Dendrogram Results:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Response</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>creating a summary dendrogram</td>
<td>very easy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>easy</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>neither easy nor difficult</td>
<td>1</td>
</tr>
<tr>
<td>restoring the original dendrogram</td>
<td>very easy</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 11.10 Summary Dendrogram Results
Summary Dendrogram Conclusions:
Creating a summary dendrogram is predominantly found to be easy and restoring the original dendrogram is found to be very easy. (Not surprising as it only needs one button-click).

The open response opinion questions: Results

Positives

Navigation
All the users were positive about navigation (which employed jQuery UI tabs), e.g. “intuitive and simple”.

Responsiveness of the dendrogram
All but one of the users was positive about this, e.g. “very responsive”. One user stated it was “a little bit slow”. Equally another stated simply, “Quick”. Users were asked to use Firefox. However, the tasks were not supervised so there is a possibility the user who found it “slow” had used Internet Explorer 8 or a slow browser.

Presentation of commands
Most comments were positive here, some highly so, e.g. “clear and exactly what I would have expected to see”.

Other comments
This section mainly contained useful suggestions. There were some generalised positive observations e.g. “Professional looking application”.

Negatives

1. One user noticed that the error message on requesting a number of groups that was not possible was confusing.
2. Most of the negative comments were to do with the leaf labels being too small and one wished the button labels to be bigger.
3. One comment stated that the purpose of the search link in the Node Descendants dialog was unclear.
4. Two suggested the command buttons could be spaced out and the labels expanded e.g. “Remove grouping” instead of “Remove”. One user suggesting that the draggable “threshold works really well” and the +/- buttons were not necessary.

5. One suggested that 4 decimal places were not needed for the threshold value.

6. There were two requests for formatting in the Groups tab to make the information easier to interpret.

**The open response opinion questions: Conclusions**

Having considered the user feedback I have decided to approach it in three categories.

- Feedback pointing to specific omissions in the application as it stood at the time of the evaluation.
- Feedback which highlights areas where compromises were deliberately part of the design. These aspects will also be addressed in the Critique sub-section of the Evaluation.
- Positive feedback confirming that the aims of the application were achieved.

**Feedback pointing to specific omissions in the application as it stood at the time of the evaluation.**

In item no. 2 in the “negatives” above about the leaf labels being too small, users were making a valid point about the leaf size (See section on compromises below). The evaluation task included no pointers on how to zoom using the browser’s own native zoom. I decided that information in The Dendrogrammer on zooming was lacking.

Item no. 3 in the “negatives” above, about the unknown purpose of the search link in the Node Descendants dialog was, again, a valid point. I decided that here too, information within The Dendrogrammer was lacking.

Although the Evaluation task study was intended as a “summative” evaluation I decided that, because only online documentation and minor presentational changes
would address these issues, they should be addressed in the final product. The following changes were made:

- An additional tab, “Zoom Help”, was added, to ensure that users are aware of how to zoom in and out using the native browser controls, and also that a refresh, would re-scale the dendrogram following a zoom.
- A title attribute was added to the search link. Appropriately configured browsers will now show a tool tip describing the action of the search link.

Were I creating the Evaluation task again I might include an early step introducing the browser zoom to the user. That said, though, the evaluation was already long enough and I would have felt obliged to remove some other aspect of it to compensate.

Item 1 about the confusing “error message” on attempt to specify an impossible number of groups is a valid criticism. This error message was coded as a bit of a “catch all”. There were three circumstances in which the user might find themselves specifying an impossible number of groups. It could be broken down with some conditional statements. The error messages could then apply to the specific circumstances of the erroneous input request. I have decided that this fix can be left for a future revision.

**Feedback which highlights areas where compromises were deliberately part of the design**

Item no. 5 in the “negatives” about the number of decimal places in the threshold value field addressed a point on which a design decision had to be made. It was a compromise. Too few significant figures and in some data sets the user might find it not possible to separate two very similar but none-the-less discrete clusters. Too many significant figures and the field would occupy too much space. I had two, “real-world” word co-occurrence MATLAB clustering, data files to work with. One of them had some clusters that were very close to each other on the dissimilarity scale. Five significant figures was what I chose so as not to limit functionality.

Item no. 6 in the “negatives” was about formatting in the groups tab. It was a deliberate design decision to have no formatting in the Groups tab, other than splitting
the display of each group onto separate lines. This was to allow the groups to be pasted into the collation tool of the users choice with minimal effect on the formatting in the destination document. However, formatting would not be difficult to add to the Groups tab were it desired.

In item no. 4 in the “negatives” about the necessity of the +/- buttons, and the labelling on buttons, bears on the design decision in response to the requirements to try to make the application cross-platform. This aspect is discussed in the critique below.

Positive feedback confirming that the aims of the application were achieved.

All of the comments about Navigation were positive. The use of the jQuery UI tabs effect was clearly a success. The “pulsate” effect employed on the Groups tab (with the accompanying event attached to the tab to halt the pulse when the tab was clicked) was mentioned specifically in one of the feedback comments as being useful.

The responsiveness and interactivity of the dendrogram was also well received by the users. Add to that the good task performance by the users and I think the evaluation does show that, on PC in Firefox, the application is a good tool with which to access clustering data interactively.

A critique of the application including suggestions for future alterations or additions.

I will address the following issues

- The pursuit of the cross platform solution. (The main focus of the critique).
- The data load feature specification
- Performance in Internet Explorer 7 and 8

This sub-section ends with a

- Summary of suggested improvements
The pursuit of the cross platform solution

The requirements called for the application to be cross-platform. Specifically, it was to work on the iPad as well as the PC. Several aspects of the application are the way they are with the aim of either rendering similarly on these two platforms or allowing a similar mode of operation on the two platforms.

Why did I pursue the goal of keeping operation and appearance similar? The rationale for having the application operate on the iPad at all, was to allow iPad users present at meetings to participate in presentations of how group formations were arrived at. They might even be following along with a presentation that was being given from a PC or a laptop. I envisioned any significant differences in what a user sees displayed on the two platforms would get in the way of such “participatory” use. If the PC user told the gathering “click this button here” and that button was presented differently on the iPad then that way of working would likely break down if the iPad user was new to the application.

The result of this pursuit of uniformity was a simple user interface, but one that may not be seen as totally ideal, perhaps particularly by the PC or laptop user. One such aspect was the additional buttons in the toolbar for moving the threshold. These were not needed for mouse interface, which had very good “drag-ability” for the threshold. However, there were times when initiating threshold drag on the iPad was “fiddly” but was eased by bringing the threshold in with button taps and then dragging could proceed more smoothly.

I did have to “give in” on one aspect of how The Dendrogrammer rendered on mobile devices with touch interfaces compared to the PC. That was in displaying dialogs containing a large amount of information. On the PC, large dialog are restricted in height within the window height and set to scroll if need be. Scrolling dialogs were not suitable on the mobile devices if the text in the dialog was to be selected and copied (which was a requirement). So on mobile devices large dialogs extend beyond the bottom of the screen and the user gestures to scroll the entire screen down. This way the dialog text is selectable quite simply. To do this, mobile OS detection had to be used. This is perhaps a better direction to follow.
Differing presentations of the controls could be coded, such as with and without the threshold adjuster buttons. jQuery JavaScript/CSS techniques can be used to hide and show the buttons. The different views could be displayed automatically in response to mobile OS detection, which is present in the applications’ `Gui.mobile` property value (`true` if a mobile OS is detected). These views could also be chosen at the user’s behest using show/hide toggle controls. There could be a “Work-along” or “Participation” view which could be selected to allow users working together cross-platform to face a uniform presentation of the controls.

These view preferences could be stored in a cookie to prevent the user continually have to reselect them.

The data load feature specification
The user opinions about “accessing the information in the data tab” contained three “neither easy nor difficult” responses. This made it the lowest rating of the evaluated features in terms of perceived ease of use. The current presentation of the data load feature owes some of its current format to the client specification (see Key Stages section “Meeting with The Clients (23/5/2011)”). In early stages of the project the specification of the data load feature was minimised and it was foreseen that different ways of loading the data would be explored after further developments in the ICT Perspectives project. The current implementation does go well beyond that minimal specification. However, as two of the user feedback comments pointed out it is not finished as well as the dendrogram is. Indeed one of the user feedback comments focussed on adding navigation to the presentation of the file information. These comments chime with my own ideas I had about adding jQuery show/hide toggle controls to collapse and expand the data tables to allow the user to view them but otherwise have them collapsed for initial neatness.

Performance in Internet Explorer 7 and 8
When leaf labels were added the performance of The Dendrogrammer in Internet Explorer 8 (IE8) became unacceptably slow on my single processor three and a half
year old PC. A mouse-over event might take 2 seconds to be processed as style changes on a 100-leaf dendrogram. The leaf labels are SVG drawn objects in Raphael. The Raphael SVG `text()` method was used. The processing involved was clearly too much for IE8. IE9 handles the dendrogram with labels with no problems. IE 7 and 8 use a JavaScript interpreter which is known to be slow with IE8 being half the speed of its contemporary competitors and IE 7 being over 5 times slower than them. By contrast IE 9 performs better than some of its competitors (See Literature Review, “Browser JavaScript performance”.) So why not just ask everyone to upgrade to IE 9? Unfortunately IE 9 is not compatible with Windows XP. (It works with Windows Vista and above).

Windows XP users could use Firefox, however the goal of cross-browser web application construction is usually to avoid forcing users to change their browser software just to view one application. There are steps that can be taken to reduce the processing load in JavaScript applications. Such as reducing the number of named objects by making use of the anonymous function wherever possible. However these have been applied already in the Raphael library. It was the leaf labels, which introduced the extra processing load and those are created using the Raphael `text()` method. Also they have to be named objects to allow access to them. I do not see a way around the slow processing in IE 8 other than using one of the other browsers. As such it is hard to claim The Dendrographmer is properly cross-browser.

Users could be offered the option to forgo leaf labels and this would speed up response in IE8.

**Summary of suggested improvements**

- Provision of different views on the controls dependent on mobile OS detection and user choices
- Any user choices of control views could be stored in a cookie
- Data load tab improved to include expand/collapse of the table displays
- Offer the option to turn off leaf labels to improve performance on slower browsers
• Improve the “Too many groups requested” feedback message to present three more targeted feedback messages.

• If user choice in control views were to be offered and stored in a cookie, perhaps users could also be offered a choice of style “skins” so they do not have to make do with what one user described as a “conservative but helpful colour scheme”.

• To improve download times the JavaScript code could be compressed with a tool such as YUICompressor\textsuperscript{86}

• In relation to the implementation of a word cloud feature possible techniques for this are suggested at the end of Section 4, Requirements Analysis (See “Discussion of Word Cloud Feasibility”).

Licensing (Section 12)

While the project work of undergraduate students and PhD students is usually the intellectual property of the University, as I understand it this is not the case with an MSc. As the purpose of the project is to provide an analysis tool which can be further developed it would be better that there be nothing standing in the way of such further development.

Most of the application is JavaScript and it is the nature of JavaScript in a web application that, whether or not the code is “open source”, it is open to be viewed. As such, even if it were commercially exploitable it would be difficult to protect.

The Raphael and jQuery libraries used in the application, both use the MIT licence. (See the Readme file in the Documentation Appendix). It is a very simple licence, which protects the author, any distributor, and any future developer. It allows changes to be made to the code.

In view of these facts, the MIT licence is attached to the Dendrogrammer.

\textsuperscript{86} [Yahoo Inc YUI Compressor] Unpublished sources
Conclusion (Section 13)

The Dendrogrammer has been created. It think it would be fair to say it does meet the requirements set out for it with the one major drawback of not functioning well with large data sets in Internet Explorer 8. I would say it makes good job of meeting the iPad/PC cross-platform requirement. However, the reservation with that aspect is in my strategy of pursuing uniformity of appearance in the two platforms, at the expense of usability on the PC. The requirements could perhaps still be met by providing a set of differing control views avoiding any sacrifice of PC usability.

Indeed the “Summary of suggested improvements” at the end of the Evaluation section describes what I would change in the Dendrogrammer.

On the plus side, The Dendrogrammer has been delivered to the client. It has been installed in the developing Research Perspectives site and has already been used to analyse clustering data based on the grants portfolio.

My own computing experience has grown considerably during the project. At the start there were many unknowns in terms of satisfying myself it would be feasible. I had no idea JavaScript was such a powerful language. I had not used it properly before this. Working with jQuery has also been an eye-opener for me. I can’t imagine I will create a web application without it again (in the foreseeable future). In addition the Raphael JavaScript graphics library was great fun to work with. Its creator has made vector graphics truly achievable cross-browser.

---

At time of writing this is available at http://www.researchperspectives.org/ and the Dendrogrammer is invoked from the “ResPers Apps” menu. The Research Perspectives site is being developed by F.Halley of the Texture Lab.
References

Published Sources


Unpublished Sources


[Last accessed 24th June 2011]


[Last accessed 17/8/2011]

[Seo and Shneiderman 2002-2007] Pages at the University of Maryland Human Computer Interaction Lab, Hierarchical Clustering Explorer for Interactive Exploration of Multidimensional Data. 2002 to 2007


http://developer.yahoo.com/yui/yuidoc/

Bibliography

The following were consulted but not referred directly to in the report.


Appendix 1

The GoW clustering data

- The original 100 leaf test data from MATLAB
## Appendix 1

The GoW clustering data.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46</td>
<td>64</td>
<td>2.03</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>100</td>
<td>2.13</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>80</td>
<td>2.25</td>
</tr>
<tr>
<td>4</td>
<td>88</td>
<td>93</td>
<td>2.26</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>16</td>
<td>2.28</td>
</tr>
<tr>
<td>6</td>
<td>33</td>
<td>90</td>
<td>2.3</td>
</tr>
<tr>
<td>7</td>
<td>45</td>
<td>87</td>
<td>2.3</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>55</td>
<td>2.31</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
<td>77</td>
<td>2.31</td>
</tr>
<tr>
<td>10</td>
<td>85</td>
<td>86</td>
<td>2.31</td>
</tr>
<tr>
<td>11</td>
<td>24</td>
<td>47</td>
<td>2.34</td>
</tr>
<tr>
<td>12</td>
<td>72</td>
<td>88</td>
<td>2.35</td>
</tr>
<tr>
<td>13</td>
<td>28</td>
<td>63</td>
<td>2.35</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>35</td>
<td>2.35</td>
</tr>
<tr>
<td>15</td>
<td>84</td>
<td>99</td>
<td>2.36</td>
</tr>
<tr>
<td>16</td>
<td>61</td>
<td>94</td>
<td>2.36</td>
</tr>
<tr>
<td>17</td>
<td>101</td>
<td>102</td>
<td>2.36</td>
</tr>
<tr>
<td>18</td>
<td>36</td>
<td>95</td>
<td>2.37</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>111</td>
<td>2.37</td>
</tr>
<tr>
<td>20</td>
<td>48</td>
<td>71</td>
<td>2.37</td>
</tr>
<tr>
<td>21</td>
<td>106</td>
<td>107</td>
<td>2.37</td>
</tr>
<tr>
<td>22</td>
<td>69</td>
<td>98</td>
<td>2.38</td>
</tr>
<tr>
<td>23</td>
<td>103</td>
<td>104</td>
<td>2.38</td>
</tr>
<tr>
<td>24</td>
<td>108</td>
<td>121</td>
<td>2.38</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>42</td>
<td>2.39</td>
</tr>
<tr>
<td>26</td>
<td>8</td>
<td>117</td>
<td>2.4</td>
</tr>
<tr>
<td>27</td>
<td>13</td>
<td>113</td>
<td>2.41</td>
</tr>
<tr>
<td>28</td>
<td>97</td>
<td>115</td>
<td>2.41</td>
</tr>
<tr>
<td>29</td>
<td>65</td>
<td>68</td>
<td>2.41</td>
</tr>
<tr>
<td>30</td>
<td>115</td>
<td>122</td>
<td>2.42</td>
</tr>
<tr>
<td>31</td>
<td>62</td>
<td>75</td>
<td>2.42</td>
</tr>
<tr>
<td>32</td>
<td>78</td>
<td>110</td>
<td>2.42</td>
</tr>
<tr>
<td>33</td>
<td>26</td>
<td>73</td>
<td>2.43</td>
</tr>
<tr>
<td>34</td>
<td>10</td>
<td>76</td>
<td>2.44</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>123</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>27</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>120</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>119</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>112</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>81</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>12</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>78</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>15</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>124</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>4</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>129</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>135</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>91</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>19</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>67</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>51</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>118</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>30</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>141</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>9</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>5</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>43</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>140</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>38</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>58</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>63</td>
<td>40</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>37</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>32</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>66</td>
<td>125</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>25</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>39</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>69</td>
<td>11</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>52</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>134</td>
<td>158</td>
</tr>
<tr>
<td></td>
<td>72</td>
<td>152</td>
<td>167</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>139</td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>34</td>
<td>161</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>162</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>76</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>40</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>62</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>79</td>
<td>105</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>31</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>150</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>82</td>
<td>171</td>
<td>178</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>148</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>84</td>
<td>142</td>
<td>172</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>179</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>159</td>
<td>165</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>173</td>
<td>183</td>
</tr>
<tr>
<td></td>
<td>88</td>
<td>68</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>89</td>
<td>169</td>
<td>182</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>184</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>181</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>163</td>
<td>186</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>189</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>94</td>
<td>3</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>20</td>
<td>187</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>191</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>193</td>
<td>195</td>
</tr>
<tr>
<td></td>
<td>98</td>
<td>194</td>
<td>195</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>197</td>
<td>198</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix 2

Preliminary Investigations

- Protovis
- SVG
- jQuery UI
- Raphael
Protovis Investigation

Protovis appears to be a good candidate as my choice for a JavaScript library with which to build the application.

Cross-browser capability

Here I do a short evaluation of its performance in terms of cross-browser capability. On the face of it, Protovis’ use of JavaScript and the SVG standard bodes well for it working in the major browsers and in particular in WebKit browsers.

Method

I tried to view the same Protovis dendrogram page in different web browsers. Screenshots were taken of the rendered diagram and cropped down to show the same area of the diagram.

Results

Figures Protovis.1 to Protovis.4 show how the Protovis dendrogram page rendered in various browsers

---

Figure Protovis.1 – part of a dendrogram using Protovis in Firefox v3.5 [screenshot of the Protovis web site]

---

1 [Protovis web site 2010]
Figure Protovis.2 – part of a dendrogram using Protovis in Opera v11 [screenshot of the Protovis web site]

Figure Protovis.3 – part of a dendrogram using Protovis in Safari on iPad [screenshot of the Protovis web site]
I looked into why Internet Explorer does not render the Protovis diagram. I found that some fairly simple steps could be taken to allow it to work in Internet Explorer.

I added an HTML 5 document type declaration at the top

```html
<!DOCTYPE html>
```

and also a `meta` tag to force IE9 to avoid using any compatibility mode

```html
<meta http-equiv="X-UA-Compatible" content="IE=9">
```

See Figure Protovis.5 below showing how the Stanford Visualisation Group’s dendrogram code renders in IE9 after these additions.
**Figure Protovis.5 – Dendrogram rendered in IE9 after additional html **DOCTYPE** and **meta** tags were added to the Stanford dendrogram code.**

(The screenshot was taken on a different computer)

**Conclusion:**
Protovis renders a complex diagram identically in Safari on the iPad, FireFox 3.5, Chrome 11 and IE 9 but not at all in IE8. This means it can be considered cross-platform in terms of being runnable on PCs and tablet computers, but not cross-browser in as much as only IE9 and above support it.

However, the Bostock and Heer paper states that Protovis supports HTML 5 canvas, SVG, and Flash.
Drag-ability, Cross-platform with Protovis

Method
I loaded the protovis Spline Editor example from the Protovis site\(^2\) in a browser on my PC and on the iPad. I tried to drag the graphic as the example was supposed to be used.

Results
On the PC, the spline object could be dragged around. However, on the iPad the object would not drag.

Conclusion
While there is a possibility that there is some other event that the drag-able code for the example might be successfully linked to, this was not promising. There was a strong possibility that dragging a graphic object in the application would be desired.

\(^2\) [Protovis website 2010] Unpublished source, Spline Editor Example
SVG investigation

Prompted by the impressive rendering of graphics done by Protovis which uses JavaScript and SVG to render graphics in web browsers, I decided to look further into SVG graphics. I decided to see what, at its simplest level needs to be done to render SVG graphics in a web browser. Referring to Eisenberg and W3Schools I created a simple pair of pages to try out SVG graphics with various browsers.

Method

First I created the SVG file which is a valid XML document.

```xml
<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
 "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="100%" height="100%" version="1.1"
 xmlns="http://www.w3.org/2000/svg">
  <circle cx="70" cy="95" r="50" stroke="black"
 stroke-width="2" fill="none"/>
</svg>
```

The `<svg>` tag declares the SVG canvas. The `<circle>` tag is the SVG command to draw a circle. (In this case, with no fill and a black outline.)

Next: the html. There are various methods of including SVG graphics in html for rendition in a browser. These are

1) using an object tag (not valid xhtml)
2) using an image tag
3) using an embed tag

3 [Eisenberg 2002] (See references) Chapter 1 and pages 205 to 209.
4) in-line method (which I did not get working)
To find out how the various browser would cope with these I included all three on the page. (This will not validate as xhtml due to the object tag not being in any of the W3C specifications):

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="UTF-8"/>
  <title>SVG Included with <object> tag</title>
</head>
<body>
<h3>SVG circle (via object tag)</h3>
<object type="image/svg+xml" data="circle3.svg">
</object>

<h3>SVG circle (via img tag)</h3>
<img src="circle3.svg" alt="a circle"/>

<h3>SVG circle (via embed tag)</h3>
<embed src="circle3.svg" alt="a circle"/>

</body>
</html>
```

I tested how the page rendered in various browsers

**Results**

See the results on the following page. The results are summarised in table $SVG.1$ following the screenshots.
Results from SVG investigation.

1) Safari and iPad
   SVG circle (via object tag)
   SVG circle (via img tag)
   SVG circle (via embed tag)

2) Firefox
   SVG circle (via object tag)
   SVG circle (via img tag)
   a circle
   SVG circle (via embed tag)

3) Chrome v11
   SVG circle (via object tag)
   SVG circle (via img tag)
   SVG circle (via embed tag)

4) IE8
   SVG circle (via object tag)

5) IE9
   SVG circle (via img tag)
   a circle
   SVG circle (via embed tag)

6) Firefox 3.6
   No plugin.
Browser | Object tag | Image tag | Embed tag | Comment
--- | --- | --- | --- | ---
Safari on iPad | ✓ | x | ✓ | Circles to right of heading text
Firefox 3.5 | ✓ | x | ✓ |  
Chrome 11 | ✓ | ✓ | ✓ |  
IE 8 | ✓ | x | x |  
IE 9 | ✓ | x | ✓ | Scroll bars around the circle canvases
Firefox 3.6 (on Win7 laptop) | ✓ | x | x | Instead indicated a plugin was missing for the Embedded item

Table SVG.1 – Results of SVG investigation.

**Conclusion**

1) This did give me a better feel for the SVG graphics after having created the files to test in the browsers.

2) The object tag gives the best cross-browser performance.

3) Interestingly, IE8 can handle the SVG graphics if it is not handed them inside a canvas tag.

4) The differences in positioning of the graphics was marked. In Internet Explorer the circles were rendered in significantly different positions relative to the associated heading text. The scroll bars around the graphics in IE9 were an obvious sign of this. In Safari on the iPad the graphics rendered beside the headings rather than below them.

It is item 4 in particular that highlights the kind of issue that the Protovis JavaScript library is handling seamlessly. Presumably this is in a similar manner to which jQuery takes care of cross-browser differences.
JQuery UI Investigation

Method

I tried out one of the jQuery UI widgets. Creating a page to support the widget was very quick. When downloading the jQuery UI library one is asked to choose a theme and a set of styles are provided along with the library to fit that theme. It was simply a matter of examining the demonstration code\(^5\) to find the name of the method to place the widget, placing a div with an id, then declaring and calling an anonymous function using the jQuery function keyword. The $ sign is short for jQuery. The two places where the $ sign appears indicate where jQuery constructs are being accessed.

```html
<script>
  $(function(){
    // Datepicker
    $('#datepicker').datepicker({
      inline: true
    });
  });
</script>

<!-- Datepicker -->
<h2>Datepicker</h2>
<div id="datepicker"></div>
```

The code tells the browser to apply the datepicker method (with the inline parameter set to true) to the div element identified by the #datepicker jQuery CSS selector. (See figure JqueryUI.6)

The widget allows the user to navigate back and forward through the calendar and choose a date.

Figure JQueryUI.1 – Rendering of the simple jQuery UI script to display the Datepicker widget (Chrome v11).

The page rendered similarly in Safari on the iPad. The widget responded to touches in the same way as it did to mouse clicks.

**Conclusion**

The jQuery UI appears to be fairly straightforward to implement. It demonstrates cross platform and cross-browser capability.
Raphael Investigation

**Method**

I loaded one of the Raphael demonstrations in IE8 and in Safari on iPad.

**Results**

Figures 6.7 shows that an object on the demonstration chart was able to be dragged around. This could be done equally well on the iPad in Safari as in IE on a PC.

![Figure Raphael.1](https://example.com/figure.png)

Figure Raphael.1– Raphael demonstration graphic rendered in IE8. The interactive graphic responded to dragging.

**Conclusion**

Raphael has demonstrated both cross-browser rendering and cross-platform drag-ability. Raphael would seem a good candidate for a JavaScript graphics library to use in this project.

---

Appendix 3
Planning

- Professional, legal, and ethical issues
- Planned steps and schedule
- Risk plan
Planning Appendix

The project plan consists of 4 sub-sections

- Professional, legal, and ethical issues considered (See also Section on Licensing in main body of report)
- Tasks or steps in the project
- Project schedule Gantt chart
- Risk Analysis

Professional, legal, and ethical issues.

The Data to be visualised

The data that will be held during the development of the project and that will be made available through the application is not data I personally will collect during the project. It is data that is already in the public domain via the EPSRC GoW web site. It is made available there as part of a transparency policy by the EPSRC. The processed data will be being provided to me by the Texture Lab.

The clustering data contains nothing that could be considered personal data.

The leaf node data may contain a key field that can be used to identify an particular Grant on the Web.

This visualisation project is part of the visualisation work of the Texture Lab for the EPSRC. If my project supervisor deems it necessary to obtain additional permission to cover this alternative visualisation of already publicly available data then such permission will be sought.

An optional usability evaluation

I have included the option of carrying out a usability evaluation if time permits. Anonymous human subjects would be used. This would require adherence to the rules for conducting interviews/focus groups.
Tasks or steps in the project

1. a) Investigate how to use jQuery and jQuery UI widget library - Evaluate
   b) Investigate how to create images (for diagram railroad tracks) dynamically (See Protovis and Raphael) – Evaluate.
   c) Investigate how to place such images accurately in a browser window in relation to text labelling – Evaluate.
   d) Investigate how to do the graphic placement cross browser – Evaluate

Steps c) and d) will include an evaluation of Raphael and Protovis.

2. Create a prototype which can produce a very simple dendrogram-type diagram with hard coded data.

3. Test the prototype in the browser test group. Tune/amend to render acceptably cross-browser. Evaluate cross-browser issues that arise at this stage.

4. Investigate how to dynamically adjust the diagram parameters to give different cluster groups.

5. Test/Modify the prototype to render cross-browser.

6. Evaluate the prototype
   Criteria:
   - Cross browser rendering similarity
   - Cross platform similarity (different hardware)

7. Investigate how to import the clustering data into the application

8. Create a prototype that imports the clustering data

9. Create a prototype, which imports the clustering data, creates a dendrogram for it and allows it to be adjusted to give different cluster groups.

10. a) Test/Modify this prototype to render cross-browser.
    b) Evaluate the prototype (criteria as above).

11 a) Settle on the final methods to use in the production application.

12 Design the final user interface.

13 Evaluate the design.

14 Implement the user interface in the production application.

15 Test and Evaluate (might include a usability comparison with the Flash application)

16 Project write-up.
17. Poster preparation.
16. Test and Evaluate
15. Implement the Application
14. Evaluate Design
13. Design User Interface
12. Decide Final Techniques
11. Test and Evaluate
10. Build Prototype
 8. Build and Prototype (with data)
 7. Investigate data import
 6. Test and Evaluate
 5. Modify Prototype
 4. Investigate dynamic drawing
 3. Test and Evaluate
 2. Build 1st Prototype
 1. Investigate Techniques
Project Risk Management

Project Risk Matrix with Mitigation/Contingency plans.

Using the Risk Matrix below I identified the 3 most important risks. (See rows marked with an asterisk). These are:

1) Learning JQuerie may take longer than expected
2) The method of dynamic generation of graphical elements for the diagrams takes longer to develop than expected.
3) The iPad interface might not work with JQuery vis-à-vis drag-and-drop.

These are addressed up front in the project during a phase of investigation followed by evaluation addressing each of the 3 main issues.

My original risk matrix was drawn up prior to my reading about Protovis and Raphael. So in fact for “jQuery” we can read “choice of JavaScript libraries”. Step 2 above will include an evaluation of whether or not Protovis or Raphael will be practical and acceptable to use for this project.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Impact</th>
<th>Likelihood</th>
<th>Mitigation/Contingency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming skills might not be up to the job</td>
<td>High</td>
<td>Medium</td>
<td>Implement fewer of the requirements and so increase the time available to program the remaining features. Sacrifice lower priority requirements first.</td>
</tr>
<tr>
<td>Learning JQuery may take longer than expected</td>
<td>Medium</td>
<td>Medium</td>
<td>Start investigating JQuery in April before I start the scheduled project.</td>
</tr>
<tr>
<td>JQuery might not be the answer to solving this problem. Another technology may be required,</td>
<td>Medium</td>
<td>Low</td>
<td>Gain experience with JQuery early on by prototyping and building time into the plan for this learning</td>
</tr>
<tr>
<td>Problem</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>The clustering data may be unexpectedly not available</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Illness resulting in inability to work on the project for a period of time.</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project supervisor illness.</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May not have access to an ipad to test the application</td>
<td>High</td>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td>The normally available ipad that has been planned to be used breaks down and is unavailable</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The method of dynamic generation of graphical elements for the diagrams takes longer to develop than expected</td>
<td>Medium</td>
<td></td>
<td>High</td>
</tr>
<tr>
<td>The iPad interface might not work with JQuery vis-à-vis drag-and-drop</td>
<td>High</td>
<td></td>
<td>Medium</td>
</tr>
</tbody>
</table>
Appendix 4

Prototype Development

• Screen shots of the prototypes by date
Prototype development Appendix

1/5/2011 Early work showing axes drawn and labelled in Raphael

Prototype 1 dendrogrammer

What this page does

1. creates clusterArray, an array of objects to store the cluster data. Each object has the following properties: (implicit is row number: the array index less 1) itemA, itemB, mergeH (short for mergeHeight)
2. hard codes in the data shown below
3. uses the largest mergeH to help decide the size of the canvas in the X axis and no of leaf nodes to decide the Y axis
4. sets up a Raphael canvas in the div element below here (the div has an id = canvas_container)
5. draws a dendrogram ... (OK so I am working on it :P

<table>
<thead>
<tr>
<th>Row</th>
<th>First Data point/cluster in pair</th>
<th>Second Data point/cluster in pair</th>
<th>Merge Height (Or similarity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

The largest mergeH is 2. There are 6 leaf nodes. The canvas X setting is 300 and Y is 190. (Don't let the proportions fool you. A Raphael canvas seems to have some minimum dimension in the X, maybe around 500, but no minimum in the Y)

Canvas div comes below here

30/4/2011 Early data structure

Data dictionary for Prototype 01

clusterArray

An array, an array of objects to store the cluster data. Each object has the following properties:

1. (implicit is row number: the array index less 1)
2. itemA first leaf or cluster in the cluster
3. itemB second leaf or cluster in the cluster
4. mergeH (short for mergeHeight)
5. redrawn - boolean, true if it was redrawn, else false

e.g. from our data table The first cluster goes in clusterArray[0] so clusterArray[0].itemA = 1;
clusterArray[0].itemB = 6; clusterArray[0].mergeH = 0; clusterArray[0] redrawn = 0;
30/4/2011 Early data structure (continued)

`clusterArrayPtr`
Indicates the current row of the clusterArray being processed. Set the array pointer to -1 to indicate empty dendrogram.

`lastDataRow`
Indicates the size of the data set for loop control, e.g.
`lastDataRow=5`

`iterationCount`
To watch for infinite looping in the drawing iterations.

`backtrackCount`
To watch for drawing backtracking.

`largestHt`
Largest merge height used to decide the size of the canvas, e.g.
`largestHt=0;`

`noOfLeaves`
Used to help dimension the canvas and graph, e.g.
`noOfLeaves=lastDataRow+1;`

`leafSpace`
Used to help dimension the canvas and graph, e.g.
`leafSpace=15`

`startingSlot`
Sets the slot to place the first leaf.
`startingSlot=noOfLeaves/2` rounded down to nearest whole number.

`htScaleFactor`
No of pixels per merge height unit. Used to help dimension the canvas and graph, e.g.
`htScaleFactor=100`

`margin`
Gap between graph and edge of canvas. Used to help dimension the canvas and graph, e.g.
`margin=50`

`canvasX`
Width of the canvas, e.g.
`canvasX=(largestHt*htScaleFactor)+(margin*2);`

`canvasY`
Ht of the canvas, e.g.
`canvasY=(noOfLeaves)*leafSpace)+(margin*2);`

`paper`
To declare and draw the canvas, e.g.
`var paper = new Raphael(document.getElementById("canvas_container"), canvasX, canvasY);`

`origin`
Used in drawing the graph, e.g.
`origin=makePt(margin,margin);// gives us origin x and origin y`
30/4/2011 Early data structure (continued)

xAxis

the xaxis of the graph e.g.
var xAxis = drawXaxis(paper, origin, canvasX-margin-margin, htScaleFactor/2);

tyAxis

the yaxis of the graph e.g.
var yAxis = paperPath("M"+origin.x+""+origin.y+"0"+(canvasY-margin-margin));

xAxisLabelMax

label for xaxis of the graph e.g.
var xAxisLabelMax = paperText((largestHt*htScaleFactor+margin), margin-10, 'Max ht = '+largestHt);

originLabel

label for origin of the graph e.g.
var originLabel = paperText((margin), margin-10, 0);

yAxisLabel

label for yaxis of the graph e.g.
var yAxisLabel = paperText(15, Math.round(canvasY/2), 'Leaves');

deArray

dendrogramElementArray or deArray for short. There are 3 types of element:

- Leaf has a leafplot number (its Y pos)
- Cluster has a topBoundY, bottomBoundY and nx (its X pos)
- Stem has a child node (leaf or cluster). The child node's Y is the stem's Y
  The child node's X is the stem's childHt.
  The stem also has a parent cluster. The Ht of the parent cluster is the stem's parentHt

Each deArray object has the following properties:

- eType
  - for Leaves this is 1
  - for Clusters this is 2
  - for Stems this is 3
  - e.g. deArray[0].eType=1

- nodeNo
  - for Leaves this is the node item number from the clusterArray
  - for Clusters this is the node item number from the clusterArray
  - for Stems this is irrelevant (Set to zero)
  - e.g. deArray[3].nodeNo=2

- clusterArrayInd
  - the index of the clusterArray row that caused it to be drawn
  - e.g. deArray[3].clusterArrayInd=4

- topBoundY
  - for Leaves this is the leaf plot number (Y pos)
  - for Clusters this is topBoundY
  - for Stems this is its child node's Y
  - e.g. deArray[3].topBoundY=1
30/4/2011 Early Data Structure (Continued)

- `bottomBoundY`
  - for Leaves this is the same as `topBoundY`
  - for Clusters this is its `bottomBoundY`
  - for Stems this is the same as its `topBoundY`
  - e.g. `deArray[3].bottomBoundY` = 2

- `htX`
  - for Leaves this is 0
  - for Clusters this is its height
  - for Stems this is the height of its child node
  - e.g. `deArray[3].htX` = 2

- `parenthtX`
  - for Leaves this is (irrelevant, set to same as `htX`)
  - for Clusters this is (irrelevant, set to same as `htX`)
  - for Stems this is the height of its child node
  - e.g. `deArray[3].htX` = 2, 5

- `deArrayPointer`
  - a pointer to the most recent successfully drawn element e.g.
  - `deArrayPointer = -1.0.1` indicates empty dendrogram

9/5/2011 Prototype drawing dendrogram from hard coded data

**Prototype 1 dendrogrammer**

<table>
<thead>
<tr>
<th>Cluster Data</th>
<th>Program variable values</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Data point/cluster in pair</td>
<td>The largest mergeHt is 2. There are 6 leaf nodes. The canvas X setting is 380 and Y is 350. The iterationCount is 7. The deArrayPtr is 20.</td>
</tr>
<tr>
<td>Second Data point/cluster in pair</td>
<td>Merge Height (or similarity)</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Canvas div comes below here:

**Application message Panel**
The dendrogram is complete.

<table>
<thead>
<tr>
<th>deArrayPt</th>
<th>Node color</th>
<th>Node Size</th>
<th>X Position</th>
<th>Y Position</th>
<th>dendrogram height</th>
<th>dendrogram width</th>
<th>parentHtX</th>
<th>height</th>
<th>iterationCount</th>
<th>redraw</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>5.5</td>
<td>6.875</td>
<td>3</td>
<td>N.A.</td>
<td>4</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
16/5/2011 Screenshot from prototype showing Dendrogram drawn from a small hard-coded data set

Prototype 1.3 dendrogrammer

Cluster Data

<table>
<thead>
<tr>
<th>Pool</th>
<th>First Data part/Cluster in part</th>
<th>Second Data part/Cluster in part</th>
<th>Merge Height (or similarity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Program variable values
The largest merge is 2. There are 6 leaf nodes.
The canvas X setting is 390 and Y is 360. The iterationCount is 0, My Test Value is 5.

Canvas div comes below here

Application message Panel
Messages from the application may appear here

23/5/11 Prototype
Partial Screenshot from zoomed browser window.
Dendrogram drawn from a large hard-coded data set with mouse-over events and cluster bounding boxes
29/5/2011 Prototype reads data from csv file + slider widget trial

The file could be selected from a table of links:

File read for dendrogram

Choose a different file to load by clicking link.

- anexample6leafdata.csv
- natastabellefdataanform.csv
- anhundredD7ran12wsopy.csv

OR

Check the data below and click the submit button to feed it to the application:

Submit

File name = anexample6leafdata.csv

It included a trial of the jQuery UI slider widget (not affecting the dendrogram yet).
3/6/2011 Clicking a cluster reports leaf data to a display div above the dendrogram.

5/6/2011 Groups formed by using slider. Groups list reported to display div.
6/6/2011 Group banding on chart when groups formed

8/6/2011 Introduced jQuery UI tabs to rationalise data display
12/6/2011 Cluster click and group threshold report to jQuery dialogs

20/6/2011 Implemented drag-able threshold adjuster.

Groups tab flashes on application of threshold
26/6/2011 Groups formed by specifying no of groups. JQuery UI buttons styled in toolbars added.

4/7/2011 manifest file and leaf data file read added

**File read for dendrogrammer**

Choose a different file to load by clicking link

- anexample6leafdata.dat
- default.dat
- matlablink.mat
- outputfromtom.dat
- anexample6leafdatanode.csv
- anexample6leafdataleaf.csv

OR

Check the data below and click the submit button to feed it to the application

**Dat file name** = anexample6leafdata.dat

It contains the following data file names and description text:

<table>
<thead>
<tr>
<th>Line No</th>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>anexample6leafdata.node.csv</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>anexample6leafdata.leaf.csv</td>
<td></td>
</tr>
</tbody>
</table>

**Node file name** = anexample6leafdata.node.csv

<table>
<thead>
<tr>
<th>Line No</th>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

**Leaf file name** = anexample6leafdata.node.csv

<table>
<thead>
<tr>
<th>Line No</th>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Thing One</td>
<td>Thing One long description</td>
</tr>
<tr>
<td>2</td>
<td>Thing Two</td>
<td>Thing Two long description</td>
</tr>
<tr>
<td>3</td>
<td>Thing Three</td>
<td>Thing Three long description</td>
</tr>
<tr>
<td>4</td>
<td>Thing Four</td>
<td>Thing Four long description</td>
</tr>
<tr>
<td>5</td>
<td>Thing Five</td>
<td>Thing Five long description</td>
</tr>
<tr>
<td>6</td>
<td>Thing Six</td>
<td>Thing Six long description</td>
</tr>
</tbody>
</table>

Manifest file (.dat) NEW

Node file (.csv) as before

Leaf file (.csv) NEW
5/6/2011 File read rationalised and integrated into application page.

Prototype 3.0.2 dendrogrammer

Load different data
Load a different data set by clicking one of the links in the table below.
```
assonehundredFraser.dat
anexample6leafdata.dat
default.dat:
matlabupdateFraser.dat
assonehundredFraser.dat:
```

The current data
Below are details of the currently loaded data set:
The largest merge-Ht is 2.
There are 6 leaf nodes.
Dat file name = anexample6leafdata.dat

It contains the following data file names and description text:

<table>
<thead>
<tr>
<th>Line No</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>anexample6leafdata.node.csv</td>
</tr>
<tr>
<td>2</td>
<td>anexample6leafdata.leaf.csv</td>
</tr>
</tbody>
</table>

Node file name = anexample6leafdata.node.csv

<table>
<thead>
<tr>
<th>Line No</th>
<th>col1</th>
<th>col2</th>
<th>col3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Leaf file name = anexample6leafdata.leaf.csv

<table>
<thead>
<tr>
<th>Line No</th>
<th>col1</th>
<th>col2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Thing One Thing One long description</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Thing Two Thing Two long description</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Thing Three Thing Three long description</td>
</tr>
</tbody>
</table>

Also, when no data is found it defaults to hard coded demo data

Prototype 3.0.2 dendrogrammer

Load different data
Load a different data set by clicking one of the links in the table below.

No .dat file in the data directory
Regrettably there is no loadable data in the data directory. A .dat file containing the names of the node and leaf data files needs to be uploaded to a loadable data in the application directory by the before this application can be useful. (Someone with write access to the application directory would need to do this.) The .csv files named in the .dat file also need to be present in the directory. The node data needs to be a 3-column .csv file containing clustering data and the leaf data should be a 2-column .csv file containing leaf description data.

The current data
Below are details of the currently loaded data set:
The largest merge-Ht is 2.
There are 6 leaf nodes.

Hard coded example data loaded
Some hard coded example data has been loaded for the time being, for demonstration purposes.
Leaf Labels implemented and tested with very large data files

Leaf labels respond with change in font-style when cluster moused-over

300-leaf data set loaded and displayed. Zoomed back with native browser zoom.
17/7/2011 “Truncate” feature completed.

1) Set threshold

2) Click “Truncate”.

3) “Restore” returns the dendrogram to normal. (See screen for step 1)
13/8/2011 - Final ‘Production’ version

With additional Zoom Help tab

Also the Documentation site

Index page
One of the class pages

```plaintext
### Classes
- ClusterTable

### Methods
-Constructor:
  ```
  ClusterTable()
  ```

-Getters:
  ```
  getClustFlow()
  getClusterFlow()
  getClusterFlowList()
  getClusterFlowStats()
  getEmptyClusterFlow()
  getFlows()
  getFlowsList()
  getFlowStats()
  getNormalizedFlows()
  getNormalizeFlows()
  getPullDendrogram()
  getPullDendrogramList()
  getWeightedFlows()
  getWeightedFlowsList()
  ```

-Private:
  ```
  private static void main() {
      // Code goes here
  }
  ```

### Constructor

The constructor initializes the cluster table, as described in the class.

```java
ClusterTable() {
    // Code goes here
}
```

### Getters

- `getClustFlow()`: Returns the cluster flow.
- `getClusterFlow()`: Returns the cluster flow.
- `getClusterFlowList()`: Returns the cluster flow list.
- `getClusterFlowStats()`: Returns the cluster flow stats.
- `getEmptyClusterFlow()`: Returns the empty cluster flow.
- `getFlows()`: Returns the flows.
- `getFlowsList()`: Returns the flows list.
- `getFlowStats()`: Returns the flow stats.
- `getNormalizedFlows()`: Returns the normalized flows.
- `getNormalizeFlows()`: Returns the normalize flows.
- `getPullDendrogram()`: Returns the pull dendrogram.
- `getPullDendrogramList()`: Returns the pull dendrogram list.
- `getWeightedFlows()`: Returns the weighted flows.
- `getWeightedFlowsList()`: Returns the weighted flows list.

### Private

- `private static void main()`: Contains the main method for the class.
```java
private static void main() {
    // Code goes here
}
```
Appendix 5

Project Diary/Blog

- Printed version of the project blog
- The full online blog can be accessed at http://dar14.wordpress.com/
Project Diary/Blog

From early on in the project I kept a diary in blog form. This allowed me to update it from anywhere and also allowed my supervisor to see it if desired. The text of the entries is listed in this appendix. The entries are shown as is but sometimes with the addition of URLs that were originally hidden in hyperlinks. The blog can be accessed at this URL: http://dar14.wordpress.com.

The online blog starts on February 17th when I was working on a different topic (a web-based 3D multi-user mind mapping application). The topic then changed (see the entry dated March 1st) to be the Dendrogrammer. To save space here I have omitted the entries on the earlier topic.

1/3/2011 - Generating interactive dendrograms
During the meeting my supervisor explored ideas inspired by the neat way the Mind42.com app created mind maps based on what looks like maybe jQuery and generating interconnecting line graphics on the fly. Tonight I turned the ideas into a mind map…

How to Implement using jQuery

Methods of generating diagrams

Flash

HTML5

Have to have a canvas set up

Interactivity

How maintainable are the diagrams

Speed of response

Download speed

Content identified in different browsers

Flash OK in different browsers

Including Select on canvas

Subplot.js can be done in Flash

Comparison of methods

Interactive Tree diagrams

Whether or not run on browser

Flash cannot run on iPad

An Alternative app to use as demonstration

To create a dendrogram

Preliminary comparison Flash vs JavaScript

Flash app exists

So focus on jQuery

1/3/11-JQuery
jQuery central http://jquery.com/ it is very small 200 k JavaScript library js file. There is a squished file 20 odd k

Tried a bit of the tutorial. So far it looks no different from normal JavaScript… but I am no expert. It will come back to me soon I imagine.

2/3/11 Dendrograms from two similarity algorithms on same data

4/3/11 Project presentation advice from Mike
1) 6 marking criteria = 2 slides per point as only 10 minutes (was ok if some short screenshot slides to show so they don’t count. I said a dozen slides…. Mike though that might be too many
2) Keeping it cut down prevents seeming rushed
3) in project planning make sure of risk register
   - highlight the 3 most important risks and place addressing these early in the project schedule.
   - explain that there will be 1 week each for review and evaluating after each of these 3 risk tackling phases
   - assuming all these 3 main risks are resolved on time then this is what the schedule will look like…..

7/3/11 Websort: The Flash application
…. that can turn clustering data into dendrograms http://websort.net/

7/3/11 The Jquery API site

7/3/11 Short Meeting to show drafts
… of Risk register, List of tasks and, Important previous work list.
My supervisor OKed them but suggested I add one risk: That the iPad interface may not be compatible with Jquery, and added 1 item to the literature review (important previous work) “showing structure in data sets”.
Well.. I had no joy on a search with “showing structure in data sets”. However I am interpreting this as getting at How to get clustering data from data sets. Clustering data and Clustering analysis brought some results and from there I have tracked down 5 books from the Library
Cluster analysis / by Brian Everitt. Publisher:Heinemann, for SSRC, 1974.
Cluster analysis algorithms for data reduction and classification of objects / by H. Spath. Publisher:Ellis Horwood, 1980.

12/3/11 Presentation
On Wed 9th seemed to go ok. present for my talk were: Mike and Dr HW Loidl Also Mike’s other MSc students: Suhayb, Douglas, and Celine. Spectators: Monika of my netaps group, Suhayb’s pair programming mate Jose, Akis, and several of Dr HW Loidl’s students who had all piled in around 4pm. Due to time pressure I gave my 10 min presentation in 8 mins. Perhaps fortunately there was no time for questions on my project at the end.

10/4/11 Started learning jQuery
Doing some of the exercises from Learning jQuery 1.3 by Chaffer and Swedberg.

13/4/11 Data from matlab
Got the matlab data from Tom on Monday. Fraser explained it to me today.
14/4/11 Clarified Requirements
Met with Mike today and clarified some of the requirements:
1) type of interactivity
2) data to be loaded from a file (probably a csv file). User to designate the file.
3) The need for a leaf node data table keyed to the MATLAB leaf node numbers was noted. (In addition to the 3-column clustering data table)
I showed him the Protovis JavaScript Graphics site http://vis.stanford.edu/protovis/ex/dendrogram.html
I later made a IE 9 compatible version of that page
http://www.macs.hw.ac.uk/~dar14/project/protovis_expts/flare_dendrogram.html

21/4/11 Raphael Tutorial

22/4/11 Made this interactive cross-browser ipad compatible page
http://www.macs.hw.ac.uk/~dar14/project/raphael_investigation/Tutorial01/dr_animation_event.html

26/4/11 Installed Ajax Tools Framework for Eclipse
On my desktop pc eclipse is now set up to function as an IDE for JavaScript with Raphael. It paid a dividend immediately.
I spotted 2 missed semicolons. Then in about 10 mins I debugged the axis code for the prototype dendrogrammer so now
the axes are nicely drawn and labelled. I tweaked the lengths, and the intervals for notches on the x axis
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/prototype_1.html

30/4/11 Designed data structure
I am working on the algorithm to draw the dendrogram. After a day or two mulling over my ideas on paper and in my head
I have designed the data structure and written a fair bit of code. So far the code draws the axes and labels them. The
structure to draw the elements is about done. So is the structure to initiate backtracking and loop until the diagram is drawn.
It has a build in iteration counter and will bomb out at 5000 steps if it loops too much. It reports parameters to the page
using DHTML. The data dictionary as at May_01_0054hrs:
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May01_0054hrs/data_dictionary_prot_01.html)
The dendrogrammer as at May_01_0054hrs
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May01_0054hrs/prototype_1.html)

1/5/11 Referred to this web page
About arrays of JavaScript objects
http://www.slis.indiana.edu/faculty/hrosenba/www/Demo/javascript2/arrays.html

1/5/11 Prototype now includes a leaf node
Current state of prototype includes a leaf node
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendroger/as_at_May01_1539hrs/prototype_1.html

11/5/11 Prototype as at Monday 9th May
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendroger/as_at_May10_1108hrs/prototype_1.html
Since then I have done a simple bug fix which was causing a name clash in IE between some JavaScript identifiers and HTML IDs. IE does not allow you to have a Document ID the same as one of the JavaScript variable names.

11/5/11 Met with my supervisor on Monday
and showed him the prototype He suggested I implement a tree data structure. He showed me a Flash application he had been working on. So I am going to do the data structure change. The aim is two-fold
1) allow access to leaf data by selecting clusters further up the tree
2) make the code more manageable

11/5/11 Continuing Learning jQuery
Today I am back on Learning jQuery so that the supporting structure for the dendrogram will be cross-browser by using jQuery for that.

11/5/11 Classes in JavaScript
I have discovered that classes can be defined in JavaScript
I knew that you could define a custom object with new Object()
Today I have read that methods can be associated with an object via its constructor. The methods are defined as named functions outside the constructor function. Then in the constructor function the methods are tied to the object
e.g
//creating a class in JavaScript
//Method
function Square_area(){ return side*side; }
//Constructor function
function Square(s){
  //intitialise properties
  this.side = s;
  //define methods
  this.area = Square_area;
}

//use these
//create a square
var mySquare=new Square(3);
//assign the area to a variable... a trivial use
var theArea=mySquare.area();

My simple example based on one in this Reference:
12/5/11 JavaScript functions

3 ways:
1) standard named function
   function myNamedFunction(){
   //do stuff
   }
   i.e. using the function statement
2a) Anonymous, usually as only to be called once
   window.onload = function(){
   //do stuff
   }
   Flannigan terms these “function literals”, Chaffer and Sweberg say they are also called “Lambda functions”
2b) Can also be used to assign the function to a variable name
   var myFuncVar = function(){
   //do stuff
   }
   The variable name can then be used to refer to it
3) Using the new Function constructor
   var myFuncVar = new Function(){
   //do stuff
   }
   Compiled every time it is used… so not good to be called in a loop

——

Functions can be referenced, passed around without executing the code, through their names as long as you dont use the brackets.
   e.g.
   mmOtherFunc=myNamedFunction;
   Flanagan p104-105, p 38
   Chaffer and Sweberg, Learning jQuery 1.3 PackT Publishing 2009, p16

12/5/11 Raphael with jQuery

I wanted to be sure they would work together. I found this good discussion thread on Raphael vs other graphics libraries

13/5/11 Began implementing classes in the prototype.

First class is a simple display class. I took a group of lines which output program variables to the page and packaged them into a class with no attributes and one method. The values to be displayed are passed in as parameters when the method is called. In this case there are a lot more lines of code than my original 4 lines. However, in Eclipse, the class is now listed and I get intellisense after typing the name of the object and dot.
15/5/11 Don’t use return with a Constructor function
If you do the returned result becomes the value of the “new” expression instead of the value of any new object created in
the constructor. The created object is discarded. Flanagan p117

15/5/11 Good progress on the recursive tree construction
New classes: clusterTable, clusterTree, clusterNode On Friday I had the clusterTable class up. The constructor reads in the
data. Today I got the new ClusterTree class up and running. It builds a tree of cluster nodes starting with the last node in the
clusterTable. Starting with the Root node it recursively builds the tree from the table It starts it off with this call
this.root= new ClusterNode(cTableIn,rootIdNo,null,0);
and the ClusterNode class responds with
//if this a leaf then set the child nodes to null
//else grow(this), causing a recursive call on the new Cluster Node constructor.
It is fabulous. I have sytematically tested it using methods to output the tested values. It tracks the order the leafs sprout in
as Mike suggested and this is indeed the order to lay out a valid dendrogram. I drew it by hand to test.
The constructor in ClusterTree then calls this
this.setClusterY(this.root)
which recursively traverses the tree setting all the cluster node yPositions based on the yPosof the children, at the start of
the call which are all set to be the same as their sprouting order number. I drew the tree on paper and worked out what the
yPos of all the clusters should be. I ran a test with the predicted yPoses. It exposed one bug: I had to separate the if, elsif in
the setClusterY into two successive ifs and it worked fine.

18/5/11 Added Styling and events to the dendrogram elements
I ran into a problem assigning events to the leaf nodes as I had made them from a composite of a circle and straight line
path for the leaf stem. Raphael allows you to push these into a set and treat them as one object. However this does not
extend to assigning events to the set. It appears you have to go inside the set and assign the events to each shape in there.
So I side-stepped that by redesigning the leaf as a single shape made using the path command. The events catered for are
mouse over and mouse out on the dendrogram elements. They change style on mouse over and revert on mouse out. It
works on iPad too: tap = mouse over and tap something else = mouse out on the first one and mouse over on the newly
tapped one. This version of the prototype is here
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May17_2035hrs/prototype_1_3.html)

23/5/11 Finished the first prototype
It interactively responds to mouseover (or tap on ipad) by highlighting the cluster and the branch(es) that descend from it.
Also each cluster has a bounding box to define a larger area for interaction. 6 leaf data set version
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May20_1145hrs/prototype_1_3.html)
100 leaf data set version
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May20_1127hrs100d/prototype_1_3_1.html)

23/5/11 Showed the prototypes to my supervisor
Mike liked the prototype and I met him today to talk about the next steps. Fraser sat in for some of it as he will be involved in taking the project forward after I am finished the MSc. He also is going to be providing me with the test data. I am to spend as little time as possible on the data import aspect. Only enough to allow designation of different test data files. No user friendliness needed on that aspect as Fraser will be handling that later.

I am to focus on
1) the interactivity aspect (adjustable groupings via a similarity threshold slider)
2) addition of a short and long description property to each node
3) display and user interaction with a cluster’s Leaf list and short description and following that
4) the scaling aspect: to do with coping with the larger data sets.

26/5/11 IE9 and compatibility view
I noticed the ap was not responding properly to events in IE9.
1) mouse over made the elements disappear
2) the bounding boxes were not clearly visible.
I went looking on the Raphael forums for IE9 problems. But All I found was Dmitry saying back in 2010 that he was saving up for windows 7. So I figured he must have fixed any problems by now. When I tried turning on the script error tracer I got as far as displaying the “developer tools”. I noticed it said I was in compatibility view. I changed to IE9 mode and everything was fine. I think I may need to put a document declaration to force IE9 to not use compatibility mode.
I think I will put in one of these meta tags
<meta http-equiv="X-UA-Compatible" content="IE=9" >
to cover IE9 compatibility. I am using an html5 doctype declaration already
<!DOCTYPE html>
but maybe there is something else in the page in my ad hoc supporting structure for the prototype which is causing IE9 to slip into compatibility view mode.

29/5/11 ap now reads csv files
The application is now fed data by some php scripts.
csvfileread_buildform.php displays links allowing each csv file in the application directory to be selected. It loads either the first one it comes to or that one selected by the user if they choose to change the loaded file. The user can view the data and submit it to the application or choose another csv to load accept_post_data.php is included at the start of the application page. It receives the data from the file read page and renders it into the application page as a form with hidden elements. The application JavaScript can then parse the form fields reading those into the ClusterTable. The application page redirects to the file read page if it detects no post data in its header. The application page file name now has a .php suffix and the application must run on a server with a php pre-processor. An additional script, set_path.php, is included to set the application path. This must be edited if the application is moved. See here (http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_May28_1055hrs_fileread/csvfileread_buildform.php?status=nodata)
3/6/11 Descendants display

You can now display descendants of a cluster by double-clicking on it. This is a stepping stone to the adjustable threshold feature. The double click will not feature in the final ap as it is not cross platform: No double-click on iPad. The displayed descendants list consists of the short description of the chosen cluster and a list of all the descendant leaves’ short descriptions. See here

http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun02_2024hrs_descendants_report/csvfileread_buildform.php

In addition I have rationalised the php structure which deals with the file read and redirects to enforce page sequence for the file read:

csvfileread_buildform.php – file read
prototype_1_4_3.php – the ap itself
ap_lib.php – functions supporting the php
set_path.php – supports redirects, Must be edited if the ap is moved.

3/6/11 Got rid of processing delay for Descendants display

It turns out the processing delay for my Descendants display code was not the code exactly but down to using double-click to trigger it. I switched to single click to trigger it and it is very quick even when selecting the root and it needs to traverse the entire tree. I was experimenting with a jQuery UI progress bar to mitigate the delay for the user but now I have removed that and just gone with single click. Single click suits iPad better anyway. See here

http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun03_2204hrs/csvfileread_buildform.php

3/6/11 Narrower on iPad

When testing on iPad I see that the graph does not span the page I found this useful post about iPad and screen width called, iPad web development tips.

5/6/11 Adjustable similarity threshold for group formation

A draft version of this feature is now working. The user adjusts a slider to change the similarity threshold then clicks a button to form the groups. At the moment the groups are listed as text in the output div. I have done a test and compared the output groups with those expected and the groups as formed are accurate. See here

http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun05_2245hrs_adjust_threshold/csvfileread_buildform.php

Currently the slider works like this: On PC it can be dragged by mouse OR moved by clicking anywhere on the slider OR once the slider is selected the left/right arrows adjust it OR the orange input field numbers can be selected by mouse and changed by keyboard input. On iPad: Tap anywhere on the slider and that is the new setting. Select the orange input field and change with keyboard that pops up.

Next step:

Implement the dendrogram group banding display

Currently each group is made up of an array of the leaf node short description attributes (text). I need to make the group arrays hold the actual leaf nodes now.
Stage 1) do that change and make the above text display come out the same but accessing through the nodes. Not tricky just need to do it carefully.

Stage 2) That will allow access to the first and last nodes in each group’s yPos attribute which is the top and bottom bounds of the group band for the graph.

6/6/11 Group banding now working
1) User adjusts Similarity threshold with slider widget 2) User clicks/taps button Bands appear on chart with text listed in output div. See here
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun06_1250hrs_groupbanding/prototype_1_4_3.php

7/6/11 Screen width optimisation again
I did some work this morning to address the graph rendering as narrower on the iPad. I introduced iPad detection to boost the htScaleFactor*1.33 in iPad. That makes it render about the same width relative to the window in iPad as it does on a PC: Non-iPad detect version: See here
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun06_1250hrs_groupbanding/csvfileread_buildform.php

With-iPad detect version: see here
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun05_2245hrs_workinprogress/csvfileread_buildform.php

However, I thought I had introduced a bug when I saw other users access it today at HW. The graph was hanging off the edge of the page. But it was not a bug. After some testing again here at home I see the iPad detection version is working just fine on my PC here. The issue is I am using screen width not window width. The people I showed it to today all are working with browser window not maximised… On a PC in a non maximised window the graph hangs of the edge of the page (tested in IE8, Firefox 4 and Safari 5), while on the iPad the graph resizes when you rotate from landscape to portrait. Maybe I need to be detecting window size, but it looks like on the PC if I do that I will have to detect the window change event and redraw the canvas on resizing. I guess I had better look into that.

7/6/11 gRaphael
Discovered that Dmitry has made an API for graphs on top of Raphael See here http://g.raphaeljs.com/basic.html
I might find a use for some of the labels maybe?

7/6/11 Zoom
I had thought I might have to do something special to zoom into the chart on a PC. On the iPad you can zoom with a pinch. My mate Vinoth showed me how to zoom in a browser on a PC today: Control and mouse wheel.
How could I have lived so long without finding this out before?
Zooming the chart on the PC like this has convinced me, finally, to completely get rid of the grey background with centered div for content in the page. For that zoom to look good with the chart, there can be no defined edge to the page.

7/6/11 by reference and by value in JavaScript
At last I stumbled across what I had been hoping to find in Flanagan confirming what I had found about how JavaScript treats function parameters. i.e. function parameters which are objects such as arrays or custom objects (i.e. not strings, and primitives) are passed by reference. Alterations to them are visible outside the function. (unless overwritten by another variable). See pages 166-170 (especially p168) in Flanagan.

When I looked for it before I looked under “Reference” and “Value”… it was listed under “*By* reference”… I had just missed it a few days ago: It is on the pages just preceding the section on garbage collection which I read then.

8/6/11 Rationalising the supporting structure of the application

I am trying out the jQuery UI tabs widget system to shift some of the descriptive stuff out of the way reduce the amount of data that lands on the page when forming groups. Instead it will be directed to another tab. See that work in progress here http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun08_2330hrs_tabstrial/csvfileread_buildform.php

9/6/11 Cross platform issues

I am starting to feel the fundamental difference between the interaction on the tablet with tap and on the PC with the mouse. tablet: has tap, pinch and swipe. My ap only uses tap. Mouse-over works if you tap. (I suppose pinch is working for native zoom) PC with mouse: has click, double-click, click-drag, mouse-wheel, right-click, mouse-over. Should the app be sensing the tablet/mobile device and presenting a customised interface for that perhaps? Could the ap be still considered cross-platform? What has got me worked up on this is that jQuery-UI has a fabulous drag-able interaction that works with the mouse and I want to use it with the group listings. But the drag-able interaction does not work on the tablet. Maybe there is something in the literature or on the web about this…

9/6/11 Nielsen and friend sum up the mobile device UI as I see it too

See Gestural Interfaces: A Step Backwards In Usability also specific to iPad more recently http://www.jnd.org/dn.mss/gestural_interfaces_a_step_backwards_in_usability_6.html
also specific to iPad more recently: http://www.useit.com/alertbox/ipad.html

11/6/11 JsDoc Toolkit

Found a documentation toolkit for JavaScript that looks as if it will be useful. (i.e. to do the same job for a JavaScript application as can be done by Javadoc for Java code. See here http://code.google.com/p/jsdoc-toolkit/

12/6/11 Tabs and Dialogs

Prototype v2.0.0 with tabs and dialogs See here still to do: Arrange the groups on the Groups tab, into separate divs. http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun12_2350hrs_dialogs/prototype_2_0_0.php

13/6/11 Now with groups listed in separate divs in the groups tab
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun13_1515hrs_groups/csvfileread_buildform.php)
Also: Tested this out on the iPad vis-à-vis copying and pasting the data from the formed group lists. The data in the dialogs and the group lists in the groups tab all copy easily on the iPad. The groups list pasted into an iPad spreadsheet a treat. See this word doc of testing.

13/6/11 Outstanding issues at the moment

1) a) Screen width optimisation of the chart should be working on window width rather than screen width. This should be do-able I think b) Ideally it should respond to window resizing. This will probably require a chart redraw after the resizing operation ends.

2) On Webkit browsers (Chrome, Safari) and Firefox. A click on a cluster “underneath” a group band is not sensed despite the group band being sent to back in Raphael. (The same clusters do respond to click in IE8). I think I see a way around this. I can tie that functionality to a click on the band instead. The click could highlight the cluster at the top of the band which is probably what the user would want anyway.

3) … in case this was not what they wanted maybe it needs a “Remove group bands” command.

4) slider widget, being separate from the graph is not perfect. I am looking at having a draggable element on the graph. (See this Raphael draggable example (http://raphaeljs.com/graffle.html) which works on iPad and PC) [Dealt with. See “Dragable threshold adjuster with limits” above.

14/6/11 Alternative syntax for “if” in JavaScript

```javascript
condition ? val_if_true : val_if_false
```

more alternate syntax:- Colon notation

```javascript
var o = { r: ‘some value’, t: ‘some other value’ };
```

is equivalent to

```javascript
var o = new Object(); o.r = ‘some value’; o.t = ‘some other value’;
```

14/6/11 safari() method in Raphael

Is used to force a redraw.

15/6/11 SVG attribute list

for use with Raphael elements See the W3C SVG attribute list (http://www.w3.org/TR/SVG/attindex.html)

15/6/11 Dragable Raphael element

I have implemented the drag-able threshold adjuster. The green line is drag-able. It does not properly set the threshold yet. I am working on that. While doing a trace on the bug holding that back I got side-tracked onto some comment rewording/refactoring with a view to trying out the documenter toolkit. It is nice occupational therapy while my subconscious chews over the bug. The drag-able adjuster version is here.

(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun15_1300hrs_dragline/csvfileread_buildform.php)

17/6/11 JavaScript reference on formatting numbers

See here (http://www.javascriptkit.com/javatutors/formatnumber.shtml)
20/6/11 Dragable threshold adjuster with limits
Before I went away for the weekend I made the drag-able threshold adjuster have limits on its travel. See here (http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun20_1007hrs_dragthresh/csvfileread_buildform.php)
It is not quite perfect for iPad yet as the iPad drag is just a bit too coarse. If you want to adjust it very close to the maximum threshold it can be hard to get all the way to the end. So I am working on some fine adjustment buttons to go with the ability to type in the similarity number.

24/6/11 window.width
I altered the screen width optimisation to work on window.width instead. (This renders the chart just a tad too wide iPad so I think I need to do some adjustment to the iPad scaling) This version also introduces the toolbar. The – and + buttons are not coded yet but the zero, less, more, and max buttons work. When the input field is alters manually and “apply threshold” button is clicked the threshold jumps to the set value. See here (http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun24_1732hrs_toolbar2/csvfileread_buildform.php)
while testing this I exposed a bug in group formation. When threshold is set to zero, if there are any clusters at zero hieght one of the leaves is omitted from the group that should be formed at zero ht. I am heading to track it down now. It must be somewhere in the code to build the individual groups I imagine. The bug can be seen in the above link by using the Fraser100D or 6 leaf example data and setting the threshold to zero. Leaf 14 and 30 should be in a group together at zero ht in the Fraser100D data…. They are on the graph ok but the group list only shows Leaf 30 on its own.

26/6/11 Specifying the number of groups to form
I implemented this feature today. The user can enter a number in a field and click a button to specify the number of groups they wish to form. See here (http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jun26_2336hrs_noofgroups/csvfileread_buildform.php)
I still need to code some validation into those user entry fields.

1/7/11 Features still required
Met with Mike and Fraser yesterday
A) The following are the features still desired in the ap:
1) Leaf labels down the y axis
2) Ability to make a “summary” dendrogram. For use in dendrograms with a very large number of leaves. Effectively truncating the dendrogram up from 0 ht to a level at which the structure can be discerned and then the clusters interrogated, rather than having hundreds of leaves along the bottom.
3) Read in of Leaf data, in addition to clustering data (which is already read). Thinking about how the data would eventually be produced, Fraser asked that the leaf data be read from a separate file and perhaps the names of the two files could be specified in a “manifest” file which is chosen by the user and causes the two data files to be read.
B) Mike also suggested I should seek out my second reader (Dr Wells) and get his view on whether my chosen method of evaluation (by feature/function checklist) was going to be appropriate.
C) As one of the original optional requirements is to establish whether it works on mobile devices other than iPad, I asked about this point. Mike said there were Android slate in the department. I should see Tom or Stefano.

1/7/11 priority of the features in previous post
I am tackling them in order of priority
1) Leaf data read. This is really a core feature. I knew it was coming.
2) Leaf labels. May be considered core as part of the dendrogram. I think it will need to be switched on and off.
  2) b) This on/off of leaf labels leads me to consider introducing an Settings feature perhaps using cookie functionality.
3) Summary dendrogram. I am going to count this as optional and tackle it last.

1/7/11 YUI Doc
http://developer.yahoo.com/yui/yuidoc/

4/7/11 Met with 2nd reader
My meeting with Dr Joe Wells, my second reader, was to seek his opinion on my methods of evaluation (in my plan this was to be by feature checklist). I asked if I should be getting groups of people to try performing tasks with the application. He said I should be asking myself what I would learn from any evaluation I might do. I suggested that Mike and Fraser are really my clients for the project. The discussion brought up the fact that I had been showing the prototypes to Mike and Fraser as they were developed. I had got feedback along the way. Some of that feedback was suggestions for features/functions they would like to see in the application. Joe thought that using this formative feedback/evaluation in the development was good. He also reminded me I had stated in my report that the application could be used for visualising any MATLAB clustering data and might be of general use.

4/7/11 Met with Mike
Dropped in to see Mike. I recounted my understanding of Joe’s view on my evaluation methods. We agreed that I should take care to document the feedback on prototypes I had from meetings with Mike and Fraser (in view of seeing Mike and Fraser as “clients” for the application). I ran the idea of maybe keeping any user settings (such as font size for leaf labels) in a cookie past Mike and he liked that. As he is going to be away he asked me to draft a report structure for a week from now. No content needed yet but each important section is to have 3 bullet points stating what the reader will get from that section.

4/7/11 Leaf data read
Implemented Fraser’s wished for leaf data file read using a manifest file. So now each data set consists of
  1) .dat file which contains the file names of the cluster data file (node) and the leaf descriptions (leaf) 2) the node data csv
  3) the leaf data csv See here (best to choose the onehundredDfromFraser.dat when testing)
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogrammer/as_at_Jul04_2200hrs_leafdataread/csvfileread_buildform.php

5/7/11 Integration of file read into application page
I am embarking on a rationalisation of the file read code to bring it within the application page
Achieved so far: a) The full data read from two files and a manifest file b) In addition I have introduced a data directory next to the application pages and the data files reside there to keep the ap directory tidy. The code is altered to accommodate the new path At the moment the csvfileread.php page handles file reading and builds its own hidden form containing the data items each input element having a unique name which identifies that field. The Application page PHP accepts the POST data in the HTTP request and renders it into the page as a hidden form with input elements having the same names as those items in the POST data. The JavaScript then reads that data out of those fields based on a jQuery grab of all the input elements in the page into an associative array of fieldname => value.

The csvfileread page ALSO displays the available data files and provides links which call the page again with a different GET parameter to instruct a different data set to be loaded. My aim is to integrate csvfileread.php and the main php page by some incremental steps which I hope will keep bugs to a minimum and readily traceable. The task is to integrate these functions.

1) The application page must now be able to accept the GET data instructing it what file to load OR in the absence of such GET data it will assume a default
2) The application page must load either the requested OR default data and render it into the hidden form
3) The application page must display the currently loaded data
4) The application page must display the available data sets and links to allow them to be requested.
5) All the JavaScript is executed on document.ready which means that it waits until all the tags on the page have loaded before it executes. So it does not matter at what point I place the PHP to render the hidden form code in relation to the rest of the page. The entire form with all it tags will be rendered before the JavaScript goes looking for those input elements.

I have written this note to get clear in my head what the new app page must do and how I am going to go about coding it. I had been worried about merging the two pages and having to accommodate two hidden forms (with input elements having clashing names). But that won’t be needed. The change of data set info will be carried to the page using links, which will carry GET data to itself if used. There will be no SUBMIT button required any more. These links and the currently loaded data will be displayed in the data tab on the application page in a similar way as they were displayed on the file read page.

5/7/11 Tested on Andriod slate

After my meeting with Joe yesterday I asked Tom if he had the Android slate Mike had mentioned. He did and I was able to test the ap there and then on the slate. All the functions worked fine just like on the iPad. I forgot to test the selecting text from the dialogs and tabs. I need to create a test plan so that next time I test it in the slate I don’t miss anything out. I can put the test checklist here on the blog and it will be to hand during testing.

5/7/11 File read integration completed

I have rationalised the application pages down from 6 files (js, css, 4 PHP) down to just 3 files: The PHP page, the js file, and the style sheet. There is currently a place holder html file representing the Documentation pages for the JavaScript which will change and will be a separate directory beside the application directory. There is also the library directory also beside the application directory. (It holds the Raphael file and all the jQuery and jQuery UI files). The data can now be viewed and changed from within the application page via the data tab. If no data set is present in the data directory then the application now loads some hard coded data and displays appropriate information in the data tab about the situation. I am calling this Prototype 3.0.2. See here:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul05_2048hrs_integrated2/prototype_3_0_2.php

and the no data demo is here:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul06_1000hrs_nodatademo/prototype_3_0_2.php

5/7/11 While I remember
The following items still need attention

1) The zero ht cluster group formation bug DONE [Actually the bug was affecting the display of all groups. Every group display was missing the last but one leaf from its leaf list. Only the display was affected not group formation. Now fixed 13/7/11] 2) attach show descendants to the on-click event for the group bands DONE 3) add a “remove group bands” button (it should reset the threshold to the top and the two input fields too)DONE 4) add text description field to the manifest data file, read, and display

6/7/11 Group banding colour intensity
While trying out the Android slate on Monday, Tom (see link to “Tom Methven’s Word Analysis"http://www.macs.hw.ac.uk/texturelab/EPSRC/clusters.html ) remarked how he could see that the groups were faintly banded. I could see he had difficulty making out the bands on the slate screen from an angle. So I have boosted the opacity number on the banding to make the colour more intense. I altered the banding colour to be blue (from green) to prevent it merging with the green threshold bar. I also added these properties Class attributes in the Dendrogram class for easier future tuning. (groupBandColour, groupBandOpacity, groupBandAlternateColour)

6/7/11 Licensing
My meeting with Joe touched on licensing. Which reminded me that this area will need addressed.

6/7/11 Dialog resize
A friend asked me about the jQuery dialogs today and while I was showing him how it can be done with jQuery UI and what the .dialog() method has in the way of dependencies I spotted and was reminded that I have not deployed the resizable aspect of the dialogs. But I had come to realise that sometimes the dialogs are the wrong shape so I think I will add that in and let the user resize the dialogs… except now I remember why I left it out: because the jQuery UI drag does not work on ipad. So it would add to the differences between the ipad and PC interface.

7/7/11 Further refinement of file read
A possible addition might be to make the data tables in the data tab expandable/collapsible using a jQuery show/hide toggle.

7/7/11 Working on the leaf labels
Spent a couple of hours this morning on this. I have prepared space at the left hand end of the graph by adding a new canvas attribute canvas.leafLabelSpace and also canvas.marginLeft (this.marginLeft= this.margin+this.leafLabelSpace). This extra space has had ramifications in several methods in Canvas and Dendrogram classes. This is down to me having
had a margin property which defined both the left and right margins in one number. Anyway to cut a long story short I tracked down all the uses of canvas.margin and factored in the new attribute as appropriate.

7/7/11 Fixed the band-hiding cluster problem

While I was avoiding thinking about what to do next in creating the leaf labels, I sorted out that issue (see While I remember … item 2 ). I made a version of Dendrogram_addevents, called it Dendrogram_addeventsToBand. I made it take a group instead of a node as it is the group object that holds the band. The group has an attribute, group.ancestorNode, which is the node spanned by the group for that band. That ancestor node gives access to all the cluster data for that group/band. And Bob’s your uncle: clicking a band gets you the node data cross-browser See here:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul07_1430hrs_bandevents/prototype_3_0_2.php

I removed the line I had added before to send each band to the back

this.groupList.getCurrent().band.toBack();

as it had only worked in IE and not other browsers. I also made the mouse over text for the band relect the ancestorNode’s shortDescription attribute. Isn’t avoidance behaviour a wonderful thing? :P

… so now back to figuring out the leaf labels code…

7/7/11 Leaf labels are on but need some work

I have the labels on but they are centred and need re-aligned. See here

http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul07_1739hrs_labels1/prototype_3_0_2.php

A bit of experimentation has shown me that 1) the label point is the centre of the final label. 2) if the label object will try to protrude out of the left hand edge of the Raphael paper then the left most part of the label is truncated.

What needs done:
1) Aligned right (against the vertical axis)
   a) requires some gauging of the length. No of chars won’t do as the font is proportional
   b) then needs some translation left or right
2) Some limiting of the amount of text. I think this should be done on a ball park max character length basis before the label is drawn.
3) part of the positioning code will need to gauge if the final label is too long and successively shorten its text attribute until it fits.
4) The labels need to be included in the reaction to the events. i.e. they should turn red when their leaf turns red in response to the mouse over/tap. Not sure yet if I want the labels themselves to be live…

7/7/11 Had an idea about word clouding

If this was to be added then the leaf data could include a column for the weight of each leaf. i.e. this would represent the height of the word were it to be included in a word cloud. The more I think about it the more I think it might be worth writing word clouding into this application. Raphael can draw the text and scale it and rotate it. The Cluster tree has all the leaf data ready to jump in response to events. When I click a group/cluster the dialog has all the words ready. How hard would it be to squeeze them into a word-clouding routine? Indeed if I was happy for all the words to be horizontal the cloud
could maybe just be done in DHTML with jQuery and not on a Raphael canvas. The possibilities are now buzzing around my brain but I had better put this on the back burner while I deal with the leaf labels and the summary dendrogram.

7/7/11 Evaluation
I’ve been reading the MSc Handbook and it looks like maybe I do need to run an evaluation with real users. But Fraser already suggested that that might be able to be handled within the Texture lab. Maybe 4 or 5 Texture lab personnel (as they are acquainted already with Tom’s and Fraser’s work) and also one or two peers not yet familiar with it. Maybe get them to do a task for which it was designed i.e. load a given data file, set a threshold to give 10 groups, compile a list of the groups in a word processor, and print it out. Then they comment on whether it worked and how simple the task was. I could maybe ask Tom to comment further in relation to how it compares to the manual cut and paste job he did before the ap existed.

7/7/11 text anchor property
I had a look in the Raphael discussion group and found “text anchor” mentioned found this post as a result http://stackoverflow.com/questions/2124763/raphael-js-and-text-positioning I think I can use this feature to overcome my text positioning problem

7/7/11 Leaf labels results
In reference to the earlier post about adjustment to leaf labels I have decided to go with: 1) right justified 2) responding to mouse over ancestor node by going bold and italic (when I altered the stroke colour it altered the stroke width in an irreversible way so I left out the colour change) 3) on mouse out the revert to normal and normal (i.e. font-weight and font-style attributes) Not decided on length limit yet… thinking of maybe 17-20 characters.

8/7/11 Size of dialog
I have decided against making the jQueryUI dialogs resizable as this which won’t help on a tablet due them responding to the wrong event for draging. Instead I am going to adjust the width and limit the ht dynamically based on the message length. Two Gui class attributes

this.maxDialogHt=$(window).height()*0.75;//dialog is max ht 75% of window
this.maxMessageLengthToScroll=2000;//beyond this and dialogs are ht limited

and some code in the showMessage() method which sets the default width the usual 300 but sets it to this.maxMessageLengthToScroll/4 if message length gets bigger than this.maxMessageLengthToScroll/5. Also if message length gets beyond this.maxMessageLengthToScroll then it sets the ht of the dialog to this.maxDialogHt. This brings scrolling into play.

I have tested that the maxDialogHt code works but to do it I had to reduce the message text limit unrealistically. What I really need to test this is some really big data set. I dont have one so I think I am going to have to create some. I think I can get a spreadsheet to make some valid cluster data for any number of leaves. So that is what I am going to do next.

8/7/11 Detecting ipad and android tablets
Found this article on how to do this reliably.

Script: http://www.ignitebyte.net/scrwidth.htm
I think I am going to have to do this. I need to selectively turn off the scrolling set-up for dialogs on the tablets. Really large dialogs will just need to make the pages longer on these systems and the user scroll the window.

I think I will use the “modern mobile OS” test

“// If the screen orientation is defined we are in a modern mobile OS
var mobileOS = typeof orientation != 'undefined' ? true : false;”

This should be enough to detect ipad and android

See also: Firtman p229

8/7/11 Javascript engines and browsers
A comparison of “IE9 vs Chrome 10 vs Firefox 4 RC vs Opera 11.01 vs Safari 5 – The BIG browser benchmark!” an ZNet review article by By Adrian Kingsley-Hughes, March 15, 2011

concluded IE9 32bit had the fastest JavaScript performance but all the others were similar (except IE9 64 bit which was 3 times slower). No figures on IE 8 in that one. I am finding IE8 on my desktop is terribly slow now a) since I introduced the leaf labels and b) very much so in the 200+ leaf test data sets

Here is a survey by John Resig (from 2008) which includes IE8 against the competition then:
http://ejohn.org/blog/javascript-performance-rundown/
He shows IE8 (8b2) to be 2x slower than Firefox and Safari, and 3 x slower than Chrome (b1). Note IE7 was 3x slower than IE8.

8/7/11 Large data test files
I did make that spreadsheet to create large test data files. It had some parameters at the top, the chief one being no of leaves, the others being to do with merge Ht range and increment. The data rows start with leafs 1 and 2 paired on the first row and then the next and subsequent rows pair the next leaf with the next cluster so the early rows for the 500D data are thus

1, 2, 0.1
501, 3, 0.1198
502, 4, 0.1396

and so on. The second and subsequent rows were all generated by formula so I was able to just replicate down the number of rows I needed. I made 4 new node files: 200,300,400, and 500. I made leaf files simply of repeats of the 100D leaf data I made up to go with Fraser and Tom’s cluster data. I made appropriate dat files for the 4 new sets.

See here for a directory containing the xls files I used to make the data sets.
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/prototype_testing/v3_0_2/
The Dendrograms it produces are all just gradual rising clusters to the root at the bottom. But they have allowed me to 
a) test how large data is handled by my new dialog code
b) and to see how that works on the ipad with the new “mobile” detection property in the Gui class.
c) test that large dendrogram construction works
d) test how the different browsers cope with very large dendrograms in terms of responding to events when there are so many elements with events to monitor and so much more processing to be done in response to those events.

IE 8 is now overcome by the processing load and just can barely handle the 100D data set since adding text labels. The processing to draw all those text labels seems to have tipped it over the edge. And now in responding to the mouse over event which alters the text labels en masse it grinds to a halt.

Firefox, Safari, and Chrome are all still happy enough.

Prototype with justified labels and labels altering with events is here:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul08_1940hrs_labels2/prototype_3_0_2.php

Compare it for processing speed with the labels version with no events affecting the labels:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul07_1739hrs_labels1/prototype_3_0_2.php

And also with the version before labels were added:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul07_1430hrs_bandevents/prototype_3_0_2.php

9/7/11 idea for summary dendrogram
Rather than introduce another drag-able threshold the current one could be used for setting the summary threshold. Just use a different button to enact the summary based on that threshold. I think to do that summary it would need any banding to be removed first. So I will code the “remove-group-bands-” function first.

9/7/11 Useful forum post about stopping jQuery UI pulsate pulsating

9/13/11 Max url length vis a vis query strings
RFC 2068 states: Servers should be cautious about depending on URI lengths above 255 bytes, because some older client or proxy implementations may not properly support these lengths. The spec for URL length does not dictate a minimum or maximum URL length, but implementation varies by browser. On Windows: Opera supports ~4050 characters, IE 4.0+ supports exactly 2083 characters, Netscape 3 -> 4.78 support up to 8192 characters before causing errors on shut-down, and Netscape 6 supports ~2000 before causing errors on start-up. we could use a hidden form instead and then we would only be limited by the HTTP request limit. Not sure what that is yet…

13/7/11 Meeting with Fraser Tuesday
While waiting to see Mike I showed Fraser the latest version (remove-bands as at Jul10) He like the interface and access to the cluster data by dialog and to the groups data in the tab. He liked the tab interface and said it was “really quite a good way of doing it”. He said it was “excellent”, “a really good design”. We discussed the possibilities of how to go about running an evaluation.

13/7/11 Meeting with Mike, Tuesday
We discussed the draft report outline. By the end of the meeting there were several changes discussed and agreed.
Fraser joined us and we discussed:
a) some of the requirements. In particular part of the interfacing with one of his components developed as part of hi EPSRC ICT Perspectives work, a search engine script. What was wanted was a the node descendants dialogue to link out to the search script to carry the leaf list into that script.

b) further on the evaluation exercise.
A meeting for me and Fraser was agreed for Wednesday for me to get more background on the EPSRC ICT Perspectives project and where the denrogrammer fits in that. We would also firm up the plan for the evaluation.

13/7/11 Linking out to Fraser’s search script
I coded in a link to call a php stub script with a cluster’s leaf list in GET data, in the morning before going to meet with Fraser. I have not coded any stripping of illegal characters from the lead data. Also following the meeting I adjusted the application style sheet to turn down the font size in the toolbars to 75%. This has bought a significant amount of extra room to add a further button to the toolbars (which will be needed for the “truncate” feature).

Also Bug fixes:
1) the loosing-one-leaf-from-groups-displayed bug. (Each group larger than 1 leaf reported to the groups tab no longer is minus it’s second last leaf). Fixed.
2) the error when dragging the threshold off the max ht end of the dendrogram. Fixed. see here: http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogrammer/as_at_Jul13_2200_linkout/prototype_3_0_2.php?read=onehundredDfromFraser.dat

14/7/11 Embarked on coding the drawSummary() method for Dendrogram
So far I have added the “truncate” button, and done the code to remove the axes, drawn node elements and threshold bar. Along the way I have done a bit of refactoring

1) moved all direct references to page Div ids into the Gui class. Accessed those as gui attributes or via gui set methods from outside the gui class.
2) made the No of Groups field display the actual no of groups after a threshold is applied
3) rationalised the parameters for GraphAxes class constructor. It now only requires one parameter, the canvas.
4) added target="_blank" to the link code in the link out feature so that the link opens in a new tab or window, thus preserving the dendrogram as is in the original window.

14/7/11 Change of tactic
I have decided to change tactic on the summary dendrogram. Instead of erase and redraw, I am going to have a go at morphing the already drawn dendrogram. Some of the erasing code will still be used e.g. removing the leaf labels.

Ok. Finished for today. I used the group formation mechanism to assign an alternative yPos for each leaf node (called it yPosSummary). And now I have the truncate button

1) Applying the threshold as one would for groups
2) Removing the leaf elements and drawing them as rectangle objects this gives them an easily animated x property which they do not have as a path object.
3) remove leaf labels (with a fade of opacity)
4) assigning the new yPosSummary to each leafNode based on their original graph position and also the group they are a member of
5) an animated collapse up in the y direction in moving to the new yPosSummary
Tomorrow I will work on assigning new elements and yPosSummary to the cluster nodes.
I need to tune the thickness of the summarised leafs.
I will link here to the work in progress directory but I am not freezing this one. Here:
http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul14_1212hrs_sumworking/prototype_3_1_0.php?read=onehundredDfromFraser.dat

16/7/11 Another change of tack
…on the summary dendrogram feature. I junked the morphing code. Not because of the morphing angle but because of the complications arising from the way I was plotting the new yPos for each element. I had been basing the redraw on groups and placing a gap between the groups. This was going to mean a lot of extra searching in the tree. The yPos of every element was going to be dependent on the chosen cut-off height (ie on what groups would be formed).
This final method (I know it is final as I have coded it) is based purely on the tree as it already exists.
1) The graph yPos of each element remains the same. There is no extra gap between groups.
2) What changes is the leafSpace (scaling in y direction)
3) and the amount of x axis displayed.
4) Elements are either omitted, partially drawn, or fully drawn depending on whether the are below, straddling, or above the cut-off Ht.
5) a new scaling is applied to the x axis
6) the old chart is erased right down to removing the Raphael paper element and a new one drawn.
So far I have completed
1) the new summary elements are drawn.
Working on
2) Modification/Refactor of GraphAxes to allow for different axes draw depending on arguments
3) Modification/Refactoring of Canvas to deal with the new way of making the axes. (I took the opportunity to refactor the direct reference to the ‘canvas_container’ div out of Canvas and into Gui where these belong.
4) New Events adding code will need done for the Summary chart.
5) I am considering adding another element beyond the y axis (where the leaf labels would be on the normal chart)… for the purpose of conveying the “wieght” or quantity of the leaves hanging off an given cluster (or indeed leaf) stem.

16/7/11 Getting there on Summary feature
Now new summary chart with axes is drawn when Truncate button is clicked
This link below is to this intermediate stage (the code still contains some fade out animating code that is no longer needed)
Summary chart with axes
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul16_2124hrs_summary_axes/prototype_3_1_0.php?read=onehundredDfromFraser.dat)
After this I will refactor it out. (worth noting its presence here in case I want to refer back to it though.)
After the refactoring the next stage is to amend the add events code to take into account the fact that not all the ClusterNodes have a drawn element on the graph, and so we must not attempt to assign events to those items. Events are added in a couple of places. Dendrogram_addevents, and also when creating groups. So that is where I am off to next.
NB: Actually it is the ClusterNode method `showBranch()` that requires amending to terminate with the last drawn element in a branch.

17/7/11 Removed the yPosSummary code
So I removed all the yPosSummary setting code from Group and Grouplist, and the yPosSummary attribute from ClusterNode, and the removeNode and transformNode methods from Dendrogram.

16/7/11 Working on getting the threshold bar
…reporting properly on the summary graph. some weird bug holding me up on the value display during drag. Better sleep on it.

17/7/11 Dynamic type conversion bug
At last I tracked down the bug I had been chasing in the threshold value display for the summary chart. Maybe 1.5 hours last night and 1 hour today. Finally I remembered about js dynamic type conversion. I have not been affected by it until now. The new canvas.originHt property had been getting converted to a string and was counting as zero in my calculations, resulting in the threshold bar displaying the wrong value during drag on the summary chart.
EDIT: and I eventually found the line that was allowing a string into the works
```
var cutoff=this.gui.getThresholdVal();
```
…the first line in Dendrogram_drawSummary, almost my first line of code to create the summary graph. It just too the input text field value and assigned it to the cutoff making the cutoff a string, and it all rolled on from there.
It is now changed to:
```
var cutoff=this.gui.getThresholdVal()-0;//forces a number
```

17/7/11 The truncate feature is working well
With that bug out the way the truncate feature is working well. The dendrogram can be successively truncated over and over to make for a less and less detailed dendrogram. Really quite slick.
The threshold is working properly on drag. I now need to adapt the code for the +, -, ff, fr, start, and end threshold adjuster buttons. [DONE] After that the group display code needs adjusted for the summary graph. [DONE] After that is the coding of restore to return to the original dendrogram.[DONE] Then coding will be over (I have decided not to work on an undo function)

17/7/11 Truncate feature finished
… apart from the issue of whether or not some form of leaf stubs are wanted to display at the leaf end of the summary. (http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/as_at_Jul17_2350hrs_fullsum/prototype_3_1_0.php ?read=onehundredDfromFraser.dat)

20/7/11 Evaluation task
I have finished working up the evaluation task. Fraser thinks it is well designed. It follows his advice on methodology vis a vis including some steps which promote understanding and so, engagement, in the participants. Just one amendment to do to it. Now I am working on adding an event to the groups tab to halt the pulsate on group formation. I already have coded
the halt function (for the remove button). I just need to invoke it when the tab is clicked. [DONE - now I might extend the pulse time on the pulse] Also need to get back to making YUIDoc work.

30/7/11 Got YUIDoc working :)
I had been working on the survey part of the report and it had been, as they say, ‘doing my head in’. So today I took a break from that. Yesterday I had sorted out Python and the YUIDoc installation on my lappy. Today I got down to figuring out YUIDoc. I worked up my own version of the Dos batch file to run it. Made a modified template to a) simplify and b) customise the YUI template. (I removed the form elements which are for searching the API which I will not require my docs to do. Mine is just to document the one js file with my classes). Then I set to converting my doc comments to work with YUIDoc. I was doing it class by class as I had run into a blockage when feeding it the whole file. I finally found out what the blockage was. It was bombing out on one character: the accented e in ‘Raphaël’. It had got in there with a reference I had pasted in a comment. The doc pages look great. I’ll post a link to it when I am done. I have done down to Dendrogrammer. Just GroupList, Group and Gui classes still to do. (It was such a relief to get back in touch with the code again.)

30/7/11 The JavaScript Code Documentation pages
The JavaScript Code Documentation
(http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/doc/module_dendrogrammer.html)
I have put a link to it on the About tab in the ap

2/8/11 Found a bug and fixed it
I found a bug that was only showing up in IE and fixed it. I had missed it as I had stopped testing in IE8 on my PC due to slow JScript engine. In the summary dendrogram the node leaf label style was still being addressed during mouse-over even when it did not exist. The other browsers were forgiving this. I introduced the isSummary attribute for Dendrogram (previously this was being detected via Canvas.originHt being greater than 0). I passed this into the showBranch method and avoided label reference with a condition. I have modified the comments concerned.

2/8/11 Made a final “production” directory.
(http://www.macs.hw.ac.uk/~dar14/project/dendrogrammer/ )
Named the scripts/files dendrogrammer rather than prototype. I added a version variable at the top of the php file to make it obvious how to change the version number display. I also polished off the last cross-browser testing. I can show it works for Firefox 3.6 (Raphael documentation claims from 3.0). It works fine in Chrome, Safari, Opera, and IE9. (Slow in IE 7 and 8). For the jQuery bits: the toolbar buttons rendered with differing heights in IE7, fine in everything else.
Appendix 6

Code appendix

- JavaScript code
- PHP code
- CSS code
//JavaScript Document
$(document).ready(function() {
    //##########################################################################
    //##########################################################################
    //PROCESSING STARTS HERE
    //##########################################################################
    //##########################################################################
    //Constants. Placed here to allow easy location for tuning
    //to help dimension the Raphael canvas
    var leafSpace=8; var margin=45; var leafLabelSpace=50;
    //standard styles for elements
    var standardStrokeWidth= 1.5;
    var standardLeafRadius= 2;
    var nodeHighlightColour="red";
    //END OF Constants.
    //###########################################################################
    //###########################################################################
    //The data structure
    //ClusterTable made of read in
    // - clusterRows (node data),
    // - datRows (filename data or manifest)
    // - and leafRows (leaf description data)
    //and
    //ClusterTree made of ClusterNodes (made from the ClusterTable data).
    //###########################################################################
    //set up the ClusterTable reading in the data
    var theClusterTable= new ClusterTable();
    //###########################################################################
    //instantiate the Gui class for user interaction
    //###########################################################################
    var theGui=new Gui(theClusterTable);
    //Global count of sprouted leaf nodes. Used when creating the tree
    //incremented by the global function, getNextLeafSproutNo()
    var leafSproutCounter=0;
    var growCounter=0; // used in limiting the growth (see top of code).
    //create the Styles object
    var theStyles= new Styles(standardStrokeWidth,standardLeafRadius, nodeHighlightColour);
    //set up the ClusterTree using data from the ClusterTable
    var theClusterTree= new ClusterTree(theClusterTable,theStyles);
    //set up the canvas
    var theCanvas = new Canvas(theClusterTree,theClusterTable,margin, leafSpace,leafLabelSpace, theGui);
    //draw the Axes using the GraphAxes class
    var theAxes= new GraphAxes(theCanvas,0,true);//x axis from 0 and have notches
    //###########################################################################
    //draw the dendrogram
    var theDendrogram = new Dendrogram(theClusterTree,theCanvas,theStyles,theGui,theAxes);
    //###########################################################################
    //Bind function to buttons
    theGui.bindButtonFunctions(theDendrogram,theGui);
    //###########################################################################
    //output values to web page
    theGui.showDataDisplay(theCanvas.largestHt,theCanvas.noOfLeafs);
    //End OF output values to web page
    //###########################################################################
    //###########################################################################
});
dendrogrammer.js

//############################################################################
//PROCESSING ENDS HERE
//############################################################################

//############################################################################
//Global Functions and objects
//############################################################################

/*
* A global method to access and increment the leafSproutCounter
* @method getNextLeafSproutNo
* increments the global leafSproutCounter
* @return the value of that counter
*/
function getNextLeafSproutNo(){
    //increment count
    leafSproutCounter++;
    return leafSproutCounter;
}

/*
* A global object constructor to create point objects
* @constructor point
* Point object constructor
* @return an object with x and y properties as per parameters
*/
function point(x,y){
    this.x=x;
    this.y=y;
}

/*
* For iPad detection.
* ref: Nicholas C. Zakas
* http://www.nczonline.net/blog/2010/04/06/ipad-web-development-tips/
* This is not made use of now. However I have left it in in case it may be
* needed in future. Instead a more generic mobile OS detection is done
* elsewhere
* @method isIPad
* @return boolean
*/
function isIPad(){
    return navigator.platform == "iPad";
}

//END OF Global Functions and objects
//############################################################################

//############################################################################
//Class Definitions
//############################################################################

//Below here are all the class definitions.

//GraphAxes class
/**
* For defining, drawing, and redrawing the dendrogram axes<br /><br />
* The constructor creates x and y axis graphic and axis label objects
* on the given Raphael canvas.
*
* @class GraphAxes
*/
//Define functions for the methods
/**
* Styles a given axis
* @method style
* @param axis (graphic object) the axis object to be styled
* @param title (string) title text for the axis mouse-over
* @return (boolean) true. Return value not used.
*/
function GraphAxes_style(axis,title){
    //use the attr method to style the axis object
}
dendrogrammer.js

```javascript
axis.attr(
{
  stroke: '#b2b2b2',
  'stroke-width': 2,
  title: title
});
return true; // not used
```

```javascript
/**
 * Draws an horizontal axis with notches every so often according to parameters
 * @method drawXAxis
 * @param paper {Raphael canvas}
 * @param startPt {point object} canvas point to start drawing
 * @param length {number} length to be drawn
 * @param notchInterval {number} interval at which to place notches
 * @return {graphic element} the drawn X axis
 */
function GraphAxes_drawXAxis(paper, startPt, length, notchInterval ){
  var notchSize=3;

  var myPathString="M "+startPt.x+" "+startPt.y+" ";
  currentLength=length;
  currentX=startPt.x;
  while ((currentLength-notchInterval)>=0){
    myPathString+="l 0 -"+notchSize+" l 0 "+notchSize;// add a notch
    // draw an interval
    myPathString+="l"+notchInterval+" 0 ";
    // chop interval off length
    currentLength=currentLength-notchInterval;
  }
  // one more notch
  myPathString+="l 0 -"+notchSize+" l 0 "+notchSize;// add a notch
  // complete the length
  myPathString+="l "+currentLength+" 0 z ";
  return paper.path(myPathString);
}
/**
 * Draws a smooth horizontal axis with no notches except at start and end
 * @method drawXAxisSmooth
 * @param paper {Raphael canvas}
 * @param startPt {Point}
 * @param length {number}
 * @return {graphic element} the drawn X axis
 */
function GraphAxes_drawXAxisSmooth(paper, startPt, length ){
  var notchSize=3;

  var myPathString="M "+startPt.x+" "+startPt.y+" ";
  myPathString+="l 0 -"+notchSize+" l 0 "+notchSize;// add a notch
  myPathString+="l1 0 "+length+" ";
  myPathString+="l 0 -"+notchSize+" l 0 "+notchSize;// add a final notch
  myPathString+="l1 "+currentLength+" 0 z ";
  return paper.path(myPathString);
}
/**
 * Removes drawn elements of the axes.
 * @method remove
 */
function GraphAxes_remove(){
  this.xAxis.remove();
  this.yAxis.remove();
  this.xAxisLabel.remove();
  this.xAxisLabelB.remove();
  this.originLabel.remove();
}
/**
 * Creates x and y axis graphic objects on the given Raphael canvas
 * @constructor GraphAxes
 * @param canvasIn - the Canvas ( a Canvas object)
 * @param originHtIn - the Ht at the origin (a number)
 * @param isNormalGraph - indicites if axes are for normal or summary.
 */
```
dendrogrammer.js

* Affects labelling whether or not the X axis is to have notches (boolean)
* @return {GraphAxes}
*/

//Constructor function
function GraphAxes(canvasIn, originHtIn, isNormalGraph ){
  //initialise attributes
  //none read straight in

  //associate methods
  this.style = GraphAxes_style;
  this.drawXAxis= GraphAxes_drawXAxis;
  this.drawXAxisSmooth= GraphAxes_drawXAxisSmooth;
  this.remove = GraphAxes_remove;

  //derived attributes
  // set up origin.x and origin.y
  this.origin= new point(canvasIn.margin+canvasIn.leafLabelSpace,canvasIn.margin);
  if (isNormalGraph){
    //draw an x-axis with notches
    this.xAxis= this.drawXAxis(canvasIn.paper, this.origin, canvasIn.xWidth-(canvasIn.margin*2)-canvasIn.leafLabelSpace, canvasIn.htScaleFactor/2);
  } else {
    //draw a smooth x-axis
    this.xAxis= this.drawXAxisSmooth(canvasIn.paper, this.origin, canvasIn.xWidth-(canvasIn.margin*2)-canvasIn.leafLabelSpace);
  }
  this.yAxis= canvasIn.paper.path("M " + this.origin.x + " " + this.origin.y + "l 0 0 " + (canvasIn.yHt-(canvasIn.margin*2)));

  //label the axes
  //create some drawn text to show the max value of ht. Call it xAxisLabel.
  this.xAxisLabel = canvasIn.paper.text( (canvasIn.largestHt*canvasIn.htScaleFactor+canvasIn.margin) +canvasIn.leafLabelSpace, canvasIn.margin-10, 'Max ht = ' +canvasIn.largestHt);
  this.xAxisLabelB = canvasIn.paper.text( ((canvasIn.largestHt*canvasIn.htScaleFactor/2)+canvasIn.margin +canvasIn.leafLabelSpace), canvasIn.margin-20, 'Dissimilarity');

  //label the origin
  this.originLabel = canvasIn.paper.text((canvasIn.margin +canvasIn.leafLabelSpace), canvasIn.margin-10, 'Origin. Click clusters for leaf lists.');

  //actions to be done by the constructor
  this.style(this.xAxis,"X-axis"); //style the xAxis
  this.style(this.yAxis,"Y-axis"); //style the yAxis
}

//End of GraphAxes class

//ClusterTable class
/**
 * Reads the data (held in the page as hidden elements of an HTML form) into
 * a table. The 'table' is actually three tables: an array of clusterRows,
 * an array of datRows, and an array of nodeRows.<br />
 * The constructor calls the getData() method to fill the arrays. It also
 * calls assignIdNos() method.<br />
 * ClusterTree is generated from the ClusterTable and once this has been done
 * the job of the ClusterTable is complete and it is not referred to again.
 * @class ClusterTable
 */
/**
* Defined functions for the methods
*/
//Fills the table with data.
//Gathers data from the hidden form elements and insert in the table
//1) Creates an associative array of all the input elements in the page
//2) Dat file data: loop through the array picking out the fields for each Dat row (manifest file)
//3) Node file data: as above but for the main cluster data
//4) Leaf file data: as above but for the leaf data

* Encompasses object constructor functions for clusterRow, datRow and leafRow objects.
* @method getData
* @return {string} the filename of the originating Dat file.
*/
function ClusterTable_getData(){ //fills the table with data

    //object constructor
    function clusterRow(itemA, itemB, mergeHt){
        this.itemA=itemA;
        this.itemB=itemB;
        this.mergeHt=mergeHt;
        this.idNo=0;//set to zero for now
    }

    //object constructor
    function datRow(fileNameIn){
        this.fileName=fileNameIn;
    }

    //object constructor
    function leafRow(shortIn, longIn){
        this.shortDescription=shortIn;
        this.longDescription=longIn;
    }

    //gather data from the hidden form elements and insert in the table
    //create an associative array of all the input elements in the page
    var $allInputElements=$(':input');//gather input elements in a jQuery object
    //make an array
    var inputValues = {};
    $allInputElements.each(function() {
        inputValues[this.name] = $(this).val();
    });
    //extract the file name
    var fileName=inputValues['fileName'];
    //$( "#traceDump" ).append("fileName="+fileName+"</br>");

    //First the dat file data
    //loop through the array picking out the fields for each row
    var row=1;
    field = "dat"+row+"col0";
    while (inputValues[field]!=null){
        fileNameIn=inputValues[field];
        this.datArray[row-1]=new datRow(fileNameIn);
        row++;//next row
        field="dat"+row+"col0";
    }

    //Second the node file data
    //loop through the array picking out the fields for each row
    row=1;
    field = "node"+row+"col0";
    //$( "#traceDump" ).append("!value="+inputValues[field]+"!");
    while (inputValues[field]!=null){
        itemA=inputValues[field];
        field="node"+row+"col1";
        itemB=inputValues[field];
        field="node"+row+"col2";
dendrogrammer.js

mergeHt=inputValues[field];
this.cArray[row-1]=new clusterRow(itemA, itemB, mergeHt);
row++; //next row
field="node"+row+"col0";
}

// Third the leaf file data
// loop through the array picking out the fields for each row
var row=1;
field = "leaf"+row+"col0";
while (inputValues[field]!=null){
    shortDesc=inputValues[field];
    field="leaf"+row+"col1";
    longDesc=inputValues[field];
    this.leafArray[row-1]=new leafRow(shortDesc, longDesc);
    row++; // next row
    field="leaf"+row+"col0";
}

return fileName; // returns the name of the file originating the data
}
/**
 * Returns a row from the cArray by index
 * @method getRowByIndex
 * @param indexIn {number}
 * @return {clusterRow object}
 */
function ClusterTable_getRowByIndex(indexIn){
    return this.cArray[indexIn];
}
/**
 * Returns the row indexed by the current pointer value
 * @method getCurrentRow
 * @return {clusterRow object}
 */
function ClusterTable_getCurrentRow(){
    return this.cArray[this.pointer];
}
/**
 * Returns the current pointer value
 * @method getPtr
 * @return {integer}
 */
function ClusterTable_getPtr(){
    return this.pointer;
}
/**
 * Increments the pointer then returns the newly pointed to row.
 * Should only be called after checking with .nextRowExists
 * @method getNextRow
 * @return {clusterRow object}
 */
function ClusterTable_getNextRow(){
    this.pointer++;
    return this.cArray[this.pointer];
}
/**
 * Returns the last row in the table
 * @method getLast
 * @return {clusterRow object}
 */
function ClusterTable_getLast(){
    return this.cArray[this.cArray.length-1];
}
/**
 * Detects whether or not there is another row beyond that pointed to
 * by the pointer
 * @method nextRowExists
 * @return {boolean}
 */
function ClusterTable_nextRowExists(){
    if(this.pointer+1<this.cArray.length){
        return true;
    }
    return false;
}
/**
 * Loops through the rows assigning an ID to the clusters
 * the first is lastLeaf number + 1 the next is lastLeaf+2 and so on
 * @method assignIdNos
 */
function ClusterTable_assignIdNos(){
    nextClusterNo=this.lastDataRow+3;//e.g. 0 to 4 rows = 5 data rows = 6 leafs
    //= the first cluster is number 7
    for (i=0;i<this.cArray.length;i++){
        this.cArray[i].idNo=nextClusterNo;
        nextClusterNo++;
    }
}
/**
 * Detects whether a cluster IdNo represents a leaf.
 * It will be a leaf if the idNo <= the number of leaves
 * @method isLeafByIdNo
 * @param idNoIn {integer}
 * @return {boolean}
 */
function ClusterTable_isLeafByIdNo(idNoIn){
    if(idNoIn<=this.getNoOfLeaves()&&idNoIn>0){
        return true;
    }
    return false;
}
/**
 * Loops through the rows for the given IdNo and returns the array index
 * for that row
 * @method getRowIndexByIdNo
 * @param idNoIn {integer}
 * @return {integer}the array index for that row
 */
function ClusterTable_getRowIndexByIdNo(IdNoIn){
    for (i=0;i<=this.lastDataRow;i++){
        if(this.getRowByIndex(i).IdNo==IdNoIn){
            return i;
        }
    }
    return false;
}
/**
 * Loops through the rows for the given IdNo and returns the itemA for that row
 * @method getItemAByIdNo
 * @param idNoIn {integer}
 * @return {number} the itemA for that row
 */
```javascript
function ClusterTable_getItemAByIdNo(idNoIn){
    for (i=0; i<this.cArray.length; i++) {
        if (this.getRowByIndex(i).idNo==idNoIn) {
            return this.getRowByIndex(i).itemA;
        }
    }
    return false;
}
/**
 * Loops through the rows for the given IdNo and returns the itemB for that row
 * @method getItemBByIdNo
 * @param idNoIn {integer}
 * @return {number} the itemB for that row
 */
function ClusterTable_getItemBByIdNo(idNoIn){
    for (i=0; i<=this.lastDataRow; i++) {
        if (this.getRowByIndex(i).idNo==idNoIn) {
            return this.getRowByIndex(i).itemB;
        }
    }
    return false;
}
/**
 * Loops through the rows for the given IdNo and returns
 * the mergeHt for that row with that given IdNo
 * @method getHtByIdNo
 * @param idNoIn {integer}
 * @return {number} the itemB for that row
 */
function ClusterTable_getHtByIdNo(idNoIn){
    //returns the mergeHt for that row with that given IdNo
    //if it is a leaf id then return zero a leaf's ht is zero
    if (this.isLeafByIdNo(idNoIn)) {
        return 0;
    }
    //loops through the rows for the given IdNo
    //and returns the mergeHt for that row
    for (i=0; i<=this.lastDataRow; i++) {
        if (this.getRowByIndex(i).idNo==idNoIn) {
            return this.getRowByIndex(i).mergeHt;
        }
    }
    return false;
}
/**
 * Returns the number of rows in the cArray table
 * @method getNoOfLeaves
 * @param idNoIn {integer}
 * @return {number} the itemB for that row
 */
function ClusterTable_getNoOfLeaves(){
    return this.cArray.length+1;
}
/**
 * Returns the search link text which is the 4th row in the dat table
 * @method getSearchLink
 * @return {string} the 4th item in the dat table
 */
function ClusterTable_getSearchLink(){
    if (this.datArray.length>3){
```
dendrogrammer.js

    return this.datArray[3].fileName;
}
return null;
}

//Constructor function
/**
* Calls the getData() method to fill the tables. Calls assignIdNos() method.
* @constructor ClusterTable
*/
function ClusterTable()
{
    //initialise attributes
    this.cArray = new Array(); //to hold the table rows
    this.datArray = new Array(); //to hold the dat table rows
    this.leafArray = new Array(); //to hold the leaf table rows

    //set the array pointer to -1 to indicate empty dendrogram
    this.pointer = -1;

    //associate methods
    this.getData = ClusterTable_getData;
    this.getRowByIndex = ClusterTable_getRowByIndex;
    this.getCurrentRow = ClusterTable_getCurrentRow;
    this.getPtr = ClusterTable_getPtr;
    this.getNextRow = ClusterTable_getNextRow;
    this.nextRowExists = ClusterTable_nextRowExists;
    this.assignIdNos = ClusterTable_assignIdNos;
    this.getLast = ClusterTable_getLast;
    this.getNoOfLeaves = ClusterTable_getNoOfLeaves;
    this.getRowIndexByIdNo = ClusterTable_getRowIndexByIdNo;
    this.getItemAByIdNo = ClusterTable_getItemAByIdNo;
    this.getItemBByIdNo = ClusterTable_getItemBByIdNo;
    this.isLeafByIdNo = ClusterTable_isLeafByIdNo;
    this.getHtByIdNo = ClusterTable_getHtByIdNo;
    this.getSearchLink = ClusterTable_getSearchLink;

    //actions to be done by the Constructor
    this.getData(); //fill the table
    this.lastDataRow = this.cArray.length - 1;
    this.assignIdNos();
}

//End of ClusterTable class

//######################################################
//######################################################
//ClusterTree class
/**
* The ClusterTree class holds a tree of clusters. Actually it is one
* root node leading to all the other nodes in the tree.
* The tree is a single root node which is the overall ancestor of all
* nodes in the tree. Each node is an instance of the ClusterNode class.
* Each node represents a cluster from the ClusterTable.
* The root node is the cluster that has the highest merge height.<br />
* Each node has<br />
* - nType: either leaf 2, cluster 1, or root 0<br />
* - childA: one of the two child nodes (that are other node objects)<br />
* - childB: see above<br />
* - xHt: the height of the cluster OR zero for a leaf<br />
* - parent: a parent node (another node object)<br />
* - yPos: for a leaf this is the numbered leaf slot on the leaf
*   axis (y axis). For a cluster or the root this is the point midway
*   (in y) between the yPos of its two child nodes.<br />
* For the purpose of the algorithm the leaf axis is the y axis because the
* default orientation for the tree is on its side with root at the right
* and leaves at the left arranged down the y axis.<br />
* - element: a visible drawn graphical object representing the cluster
*   on the graph<br />
* - bBox: a bounding box, a transparent graphical object representing
*   the live area on the graph associated with the cluster<br />
*/
dendrogrammer.js

* - rowIndex: the index of the row on the clusterTable from which it was generated<br />
* - idNo: the number given it by MATLAB in creating the clustering data<br />
* e.g. in our example data the cluster on row 0 (Item A =1, ItemB=6 which are two leaves are part of the first cluster. 5 rows (0 to 4) => 6 leafs => first cluster is no.7 so the idNo for that cluster will be 7.<br />

* The ClusterTree constructor calls the new ClusterNode constructor method <br />
* on the root node which in turn calls the ClusterNode grow() method which recursively calls itself to build the tree from the data in ClusterTable table. The constructor also sets the graph yPos for each node.<br />
* Important ClusterTree attribute: <br />
* this.root. However, this.root is accessed via the getRoot() method.<br />
*<br />
* @class ClusterTree<br />
*/

//Define functions for the methods<br />
/**<br />
* @method getRoot<br />
* @return {node} the root node in the tree<br />
*/
function ClusterTree_getRoot(){
    return this.root;
}
/**<br />
* Recursively traverses the tree setting the yPos attribute.<br />
* Based on the yPos of the leaf nodes.<br />
* The yPos for each leaf is already set (it is the sproutCount property)<br />
* The yPos for a cluster is half way between the childA.yPos and childB.yPos <br />
* but the yPos of the clusters start off null.<br />
* @method setClusterY<br />
* @param node {ClusterNode object}<br />
* @return {boolean} value not used.<br />
*/
function ClusterTree_setClusterY(node){
    //if the yPos if childA is null then set its yPos<br />
    //if the yPos of childB is null then set its yPos<br />
    if (node.childA.yPos==null){<br />
        this.setClusterY(node.childA);<br />
    }<br />
    if(node.childB.yPos==null){<br />
        this.setClusterY(node.childB);<br />
    }<br />
    //finally set the yPos based on yPoses of children<br />
    if (node.childA.yPos<node.childB.yPos){<br />
        //set yPos to that of child A + difference between A and B<br />
        node.yPos=node.childA.yPos+(Math.abs(node.childA.yPos-node.childB.yPos)/2);<br />
    }else{<br />
        //set yPos to that of child B + difference between A and B<br />
        node.yPos=node.childB.yPos+(Math.abs(node.childA.yPos-node.childB.yPos)/2);<br />
    }
    return true;
}
/**<br />
* Calls the new ClusterNode constructor method which in turn calls the ClusterNode<br />
* grow() method which recursively calls itself to build the tre from the<br />
* data in ClusterTable table. It also sets the graph yPos for each cluster.<br />
* @constructor ClusterTree<br />
* @param cTableIn {ClusterTable} the instance of ClusterTable containing the input data.<br />
* @param stylesIn {style object} carries the style info for local access<br />
*/
function ClusterTree(cTableIn,stylesIn){
    //initialise attributes<br />
    this.cTable=cTableIn;<br />
    this.styles=stylesIn;<br />
    //the root node is the node from the end of the clusterArray
// i.e. that cluster with the highest cluster height
var rootIdNo = this.cTable.getLast().idNo;

// This line causes a recursive all on the ClusterNode Constructor
// in conjunction with the data in the cluster table which fill up
// the tree
this.root = new ClusterNode(this.cTable, rootIdNo, null, 0, this.styles);

// associate methods
this.getRoot = ClusterTree_getRoot;
this.setClusterY = ClusterTree_setClusterY;

// processing to be done by the constructor
// Set the yPos values for all the clusters
this.setClusterY(this.root);

// End of ClusterTree class

// ClusterNode class

/**
 * The ClusterTree is made up of ClusterNodes. See description of
 * ClusterTree class for further details. A ClusterTree is a tree
 * structure of these ClusterNodes.
 * 
 * @method isLeaf
 * @return {boolean} true if the node is a leaf
 */
function ClusterNode_isLeaf(){
    if (this.nType==2){
        return true;
    }
    return false;
}
function ClusterNode_isRoot(){
    if(this.nType==0){
        return true;
    }
    return false;
}

/**
 * Grows the tree by calling itself recursively. The recursion is via the ClusterNode constructor which calls this grow() method. First grow down the childA branch. Then grow down the childB branch.
 */
function ClusterNode_grow(nodeIn){
    //retrieve the ID numbers of the node's ChildA and ChildB //from the ClusterTable
    var childIdA=this.cTable.getItemIdAByIdNo(nodeIn.idNo);
    var childIdB=this.cTable.getItemIdBByIdNo(nodeIn.idNo);

    //First grow down the childA branch. Then grow down the childB branch. //Pass the each CLuster node the table, the id of the new node, the //current node as the parent, and 4 as unknown type for the new node //(i.e. don't know if cluster or leaf yet)
    this.childA=new ClusterNode(this.cTable,childIdA,nodeIn,4,nodeIn.styles);
    //Next grow down the childB branch
    this.childB=new ClusterNode(this.cTable,childIdB,nodeIn,4,nodeIn.styles);
    return true; //return value not used
}

/**
 * Turns on or off the highlighting of a tree branch<br/>
 * The reason that "this" is not used in this block is due to it being involved in event handling. When an event fires "this" becomes the object associated with the event rather than the current in scope class object.<br/>
 *<br/>
 * Uses a recursive call to highlight itself and all its children.
 */
function ClusterNode_showBranch(nodeIn,on,isSummaryIn){
    //check for nodeIn==null is so return false
    if(nodeIn==null){return false;}//attempt no processing on a null node
    //check for element null incase element is undrawn in a summary graph
    //attempt no styling if there is no element
    if(nodeIn.element==null){return true;};
    //if on = true then show highlighting else revert from highlighting
    if(on){
        //show highlighting
        nodeIn.element.attr(
        {
            'stroke':'red',
            'stroke-width':(nodeIn.styles.strokeWidth*1.5)
        });

        //if it is a leaf AND NOT a summary chart then highlight the label
        if((nodeIn.isLeaf())&&(!isSummaryIn)){
            nodeIn.label.attr(
            {
                'font-style':'italic',
                'font-weight':'bold'
            });
        }
    }
}
else {
  // revert from highlighting
  nodeIn.element.attr(
    {
      'stroke': 'black',
      'stroke-width': nodeIn.styles.strokeWidth
    });

  // if it is a leaf AND NOT a summary chart then turn of label highlight
  if((nodeIn.isLeaf())&&!isSummaryIn)
    nodeIn.label.attr(
      {
        'font-weight': 'normal',
        'font-style': 'normal'
      });
}

// do same for children
nodeIn.showBranch(nodeIn.childA,on,isSummaryIn);
nodeIn.showBranch(nodeIn.childB,on,isSummaryIn);
return true; // has no effect
}

/**
* Draws a transparent rectangular bounding box around the node element
* This will be used as a larger area to be sensitive for interacting with the
* cluster.
* *
* @method drawBoundingBox
*
* @param paperIn {Raphael canvas}
*
*/
function ClusterNode_drawBoundingBox(paperIn){
  // make the bounding box
  this.bBox=paperIn.rect(
    this.element.getBBox().x,
    this.element.getBBox().y,
    this.element.getBBox().width,
    this.element.getBBox().height);

  // style it transparent
  this.bBox.attr{
    'fill': 'blue',
    'stroke-opacity':0,
    'fill-opacity':0.1,
    'title':this.shortDescription
  });

  return true; // has no effect
}

/**
* Called by the ClusterNode_getDescendants method.
* It does this by calling itself recursively on the node's two
* children until it comes to a leaf, then pushing the leaf onto the array.
* *
* @method appendDescendants
*
* @param nodeIn {a ClusterNode} the node for which the list is sought.
* @param arrayIn {array} the array to which any leaves are to be appended
*   (an array of ClusterNodes all members of which will be type leaf)
* @return {array of leaf nodes} an array of leaf nodes descended from
*   the given node.
*
*/
function ClusterNode_appendDescendants(nodeIn,arrayIn){

  // check for nodeIn==null if so return arrayIn
  if(nodeIn==null)
    arrayIn.push("null node");
  return arrayIn;
}

// /attempt no processing on a null node. Should not be needed.
// if the node is a leaf append to array and return
if(nodeIn.nType==2){
  // now appending actual node itself
}
```javascript
arrayIn.push(nodeIn);
return arrayIn;
}
//arrayIn.push("NodeID="+nodeIn.idNo);//just for tracing
arrayIn=nodeIn.childA.appendDescendants(nodeIn.childA,arrayIn);
arrayIn=nodeIn.childB.appendDescendants(nodeIn.childB,arrayIn);
return arrayIn;
}

/**
  * Given a cluster node this returns an array of the leaves descended from
  * that cluster node. Calls the appendDescendants method to fill the array.
  * @method getDescendants
  * @param {node} the cluster node for which we want to list the descendant leaves
  * @return {array} An array of descendant leaves OR if the node itself is a leaf
  * then it will return the node itself.
  */
function ClusterNode_getDescendants(nodeIn){
  var resultArray=
  new Array();
  //if this is a leaf then append push node and return
  if (nodeIn.nType==2){
    resultArray.push(nodeIn);
    return resultArray;
  }
  resultArray=nodeIn.appendDescendants(nodeIn,resultArray);
  return resultArray;
}

/**
  * Outputs a string listing the descendant leaf nodes of the given node
  * Calls the node's getDescendants() method to make an array of
  * descendant leaves from which it compiles the string.
  * If the node is a leaf it outputs the leaf's short and long description
  * It also includes a search link in the dialog to send the descendant list to
  * a search engine.
  * @method displayDescendants
  * @param nodeIn {a ClusterNode} the node for which the list is sought
  * @param guiIn {a Gui} the gui
  * @return boolean. Value not used.
  */
function ClusterNode_displayDescendants(nodeIn,guiIn){
  var resultsString="";
  var urlString="";
  //if the node is a leaf report its short and long description
  //else report its descendant leaves short descriptions
  if(nodeIn.isLeaf()){ resultsString="Leaf: "+nodeIn.shortDescription;
    resultsString+=", "+nodeIn.longDescription;
    urlString=nodeIn.shortDescription;
  }else{
    var resultsArray=nodeIn.getDescendants(nodeIn);//get the descendants
    //loop through the array appending them to a string
    resultsString="Descendants of "+nodeIn.shortDescription+": ";
    for(var i=0;i<resultsArray.length-1;i++){
      resultsString+=resultsArray[i].shortDescription+", ";
      urlString+=resultsArray[i].shortDescription+"%20";
    }
    resultsString+=resultsArray[resultsArray.length-1].shortDescription;
    urlString=resultsArray[resultsArray.length-1].shortDescription;
  }
  guiIn.showMessageWithLink(resultsString,'Node Descendants',urlString );
  return true;//has no effect
}

//Constructor function
/**
 * If this node is a leaf
 */
```
* then it sets the child nodes to null and set default leaf descriptions and then returns.
* else it calls grow(), causing recursive calls on the new Cluster Node constructor to spawn its childA and its childB child nodes. (and also sets default cluster descriptions)

@constructor ClusterNode

@constructor
ClusterNode
cTableIn {ClusterTable}
@constructor
idNoIn {number}
@constructor
parentNodeIn {ClusterNode}
@constructor
nTypeIn {number}
@constructor
stylesIn {the Styles object}

function ClusterNode(cTableIn,idNoIn,parentNodeIn,nTypeIn,stylesIn){
  //initialise attributes
  this.growLimit=2000;
  this.cTable=cTableIn;
  this.idNo=idNoIn;
  this.nType=nTypeIn;
  this.styles=stylesIn;
  this.parent=parentNodeIn;
  this.xHt=cTableIn.getHtByIdNo(idNoIn);
  this.rowIndex=cTableIn.getRowIndexByIdNo(idNoIn);
  this.leafSproutNo=null;//default for any node. Set later for a leaf
  this.currentGroup=1;//default starting group for any node.
  this.yPos=null;//graph position on the vertical axis
  this.element=null;//the drawn object represented on the graph
  this.label=null;//graphical Raphael label drawn to left of the y axis
  this.bBox=null;//the bounding box for event sensing yPos, element, //label, and bBox will be added later after further processing

  //associate methods
  this.grow = ClusterNode_grow;
  this.isLeaf = ClusterNode_isLeaf;
  this.isRoot = ClusterNode_isRoot;
  this.showBranch=ClusterNode_showBranch;
  this.drawBoundingBox=ClusterNode_drawBoundingBox;
  this.getDescendants=ClusterNode_getDescendants;
  this.appendDescendants=ClusterNode_appendDescendants;
  this.displayDescendants=ClusterNode_displayDescendants;

  //if this a leaf then set the child nodes to null and set default
  //leaf descriptions
  //else grow(), causing a recursive call on the new Cluster Node constructor.
  //also set default cluster descriptions)
  if(this.cTable.isLeafByIdNo(this.idNo)){
    this.nType=2;
    this.shortDescription=cTableIn.leafArray[this.idNo-1].shortDescription;
    this.longDescription=cTableIn.leafArray[this.idNo-1].longDescription;
    this.leafSproutNo=getNextLeafSproutNo();
    this.yPos=this.leafSproutNo;
    this.childA=null;
    this.childB=null;
  }else{
    //set default. Detail added later
    this.shortDescription="Cluster ID"+this.idNo;
    this.longDescription=
      "No long description of cluster ID "+this.idNo+" has been set."
    //check grow counter
    //if growCounter>some limit don't grow
    if (growCounter>this.growLimit){
      growCounter++;
      this.grow(this);//recursively calls the new ClusterNode
    }
  }
}

//End of ClusterNode class
The Canvas class defines the area on which the dendrogram is drawn.

It holds some methods key to drawing the elements:

- A drawn element effectively has two kinds of coordinates or position. There is its graph position (the logical position on the graph in relation to its mergelt (on the X axis) and its yPos along the leaf axis). Then there are its Raphael coordinates determining where it is drawn on the canvas, taking into account margins and scaling factors.

- A number of the Canvas methods deal with conversion between the graph (logical) coordinates and the Raphael canvas coordinates. (e.g. cvY() takes a logical yPos and returns the Raphael canvas yPos coordinate.)

- Important Canvas attributes used often by other classes:

- margin
- marginLeft
- leafSpace
- leafSpaceSummary
- originHt
- htScaleFactor
- htScaleFactorSummary
- leafLabelSpace
- noOfLeafs
- largestHt
- paper (the Raphael canvas on which all elements are drawn).

@class Canvas

//Methods

** Takes a diagram Y and converts it to a canvas Y, taking into account margins and scale factors

* @method cvY
  * @param yIn {point object} a graph point yPos
  * @return {number} the canvas yPos
*/
function Canvas_cvY(yIn){
    return this.margin+(yIn*this.leafSpace);
}

** Does the same job as cvY but for a summary dendrogram

* @method cvYSummary
  * @param yIn {number} a graph point yPos
  * @return {number} the canvas yPos
*/
function Canvas_cvYSummary(yIn){
    return this.margin+(yIn*this.leafSpaceSummary);
}
function Canvas_cvX(xIn) {
    return this.margin + this.leafLabelSpace + (xIn * this.htScaleFactor);
}

function Canvas_cvXSummary(xIn) {
    return this.margin + this.leafLabelSpace + (xIn * this.htScaleFactorSummary);
}

function Canvas_getGraphXfromCvX(xIn) {
    return (xIn - this.margin - this.leafLabelSpace) / this.htScaleFactor;
}

function Canvas_cVPt(ptIn) {
    ptIn.x = this.cvX(ptIn.x);
    ptIn.y = this.cvY(ptIn.y);
    return ptIn;
}

function Canvas_cVPtSummary(ptIn) {
    ptIn.x = this.cvXSummary(ptIn.x);
    ptIn.y = this.cvYSummary(ptIn.y);
    return ptIn;
}

function Canvas_getLeafPt(slot) {
    var y = this.cvY(slot);
    return new point(this.cvX(0), y);
}
```javascript
/**
 * Returns the canvas coords (in the form of a point used for the centre of the leaf) given a leaf slot number, in the summary graph.<br />
 * e.g. if slot is 2 and leafspaceSummary is 4 then feed 8 to cvY and return that as y coord and feed 0 to cvX and return that as x coord
 * @method getLeafPtSummary
 * @param slot {number} the leaf slot number
 * @return {point object} a canvas point
 */
function Canvas_getLeafPtSummary(slot){
    var y = this.cvYSummary(slot);
    return new point(this.cvX(0),y);
}

/**
 * Calculates what the scaleFactor will be for this diagram taking into account canvas size and largest merge Ht. This gets assigned to the Canvas.htScalefactor attribute
 * @method calculateScaleFactor
 * @param largestHt {number} the max in the x axis
 * @param widthForCanvas {number} available width
 * @param margin {number} to leave round the sides outside the axes
 * @param leafLabelSpace {number} space on LH side
 * @return {number} the scale factor
 */
function Canvas_calculateScaleFactor(largestHt, widthForCanvas,margin,leafLabelSpace){
    var result=widthForCanvas-margin-margin-leafLabelSpace;
    result=result/(largestHt*1.05);
    return result;
}

/**
 * Sets new values for xWidth and yHt. Creates a new Raphael canvas in the container
 * @method createPaper
 * @param xWidthIn {number} the new width
 * @param yHtIn {number} the new ht
 */
function Canvas_createPaper(xWidthIn,yHtIn){
    this.xWidth=xWidthIn;
    this.yHt=yHtIn;
    this.paper = new Raphael(document.getElementById(this.gui.canvasContainer), xWidthIn, yHtIn);
}

//Constructor function
/**
 * @constructor Canvas
 * @param cTreeIn {the ClusterTree object} the cluster tree to be represented.
 * @param cTableIn {the ClusterTable object} the cluster table.
 * @param htScaleFactorIn (number) the scale factor in the x direction.
 * @param marginIn {number} the size of the margin around the chart within the canvas.
 * @param leafSpaceIn {number} the amount of space allowed for each leaf on the leaf (Y) axis.
 * @param leafLabelSpaceIn {number} the amount of extra space down the vertical axis to leave for leaf labels.
 */
function Canvas(cTreeIn,cTableIn,marginIn,leafSpaceIn,leafLabelSpaceIn, guiIn){
    //initialise attributes
    //passed in
    this.cTree = cTreeIn;
    this.cTable = cTableIn;
```

Page 18
```javascript
this.gui = guiIn;
this.margin = marginIn;
this.leafSpace=leafSpaceIn;
this.leafSpaceSummary=leafSpaceIn/2;
this.originHt=0; //set to higher during summary graph
this.htScaleFactorSummary=null; //set later
this.leafLabelSpace=leafLabelSpaceIn;
this.marginLeft= this.margin+this.leafLabelSpace;

//associate methods
this.cvY=Canvas_cvY;
this.cvYSummary=Canvas_cvYSummary;
this.cvX=Canvas_cvX;
this.cvXSummary=Canvas_cvXSummary;
this.cVPt=Canvas_cVPt;
this.cVPtSummary=Canvas_cVPtSummary;
this.getLeafPt=Canvas_getLeafPt;
this.getLeafPtSummary=Canvas_getLeafPtSummary;
this.calculateScaleFactor=Canvas_calculateScaleFactor;
this.getGraphXfromCvX=Canvas_getGraphXfromCvX;
this.createPaper=Canvas_createPaper;

//derived attributes
this.screenWidth=$(window).width();
$(
  "#traceDump"
).append("Window.width="+this.screenWidth+", ");
this.widthForCanvas=this.screenWidth-50; // allowing for scroll bars etc
this.noOfLeafs = this.cTable.getNoOfLeaves();
this.largestHt = this.cTree.getRoot().xHt;
this.htScaleFactor= this.calculateScaleFactor(
  this.largestHt, this.widthForCanvas, this.margin, this.leafLabelSpace);
$(
  "#traceDump"
).append("isIPad()=+(navigator.platform == "iPad")+, ");

//declare and draw the canvas
this.createPaper(
  (this.largestHt*this.htScaleFactor)+(this.margin*2)+this.leafLabelSpace,
  ((this.noOfLeafs+1)*this.leafSpace)+(this.margin*2)
);
```

---

**The Dendrogram class encapsulates most of the methods for drawing, removing, and redrawing the dendrogram and summary dendrogram (Some drawing methods are within the ClusterNode class.)**

* <br />

* The constructor draws a dendrogram from the given ClusterTree.<br />
* It calls drawNode() to recursively draw the tree. <br />
* It calls setDragParams() to set up some functions as Raphael drag parameters for the threshold bar.<br />
* It calls drawThresholdBar() to draw the threshold bar onto the graph<br />
* It calls this.gui.showThreshold() to display the threshold value<br />
* It calls this.gui.showNoOfGroups(1) to display the no of Groups value initially as 1<br />

* Important Dendrogram attributes:<br />
  * thresholdBar - the draggable bar graphic object. <br />
  * groupList - the list of groups formed in response to a user command. <br />
  * isSummary - default false. Set to true if displaying summary dendrogram.<br />

* Some attributes are set in the constructor for tuning:<br />
* leafLabelLength - the number of characters allowed for the leaf labels<br />
* some thresholdBar dimensions<br />
* some group band styling attributes<br />

* @class Dendrogram class
function Dendrogram_drawCluster(nodeIn)
{
    var parentHt=null;// to hold ht of parent or notional parent ht for root
    //if nodeIn is the root set parentHt to nodeIn.xHt*1.05 i.e. 5% higher
    //than max ht
    //this will give an area to interact with above the root node
    if (nodeIn.isRoot()){
        parentHt = nodeIn.xHt*1.05;
    }else{
        parentHt=nodeIn.parent.xHt;
    }

    //draw the cluster by starting at the cluster point node.xHt, node.yPos
    //scribing over the crossbar section returning to the cluster
    //point and finishing by scribing up the stem section creating
    //one shape from a single path call
    //create a diagram point for the cluster
    var clusterPt=new point(nodeIn.xHt,nodeIn.yPos);

    //define the graph points for the cluster with no scaling factors
    var clusterPt=new point(nodeIn.xHt,nodeIn.yPos);
    var boundA=new point(nodeIn.xHt,nodeIn.childA.yPos);
    var boundB=new point(nodeIn.xHt,nodeIn.childB.yPos);
    var cEnd=new point(parentHt,nodeIn.yPos);

    //now convert them to canvas coordinates incorporating
    //scale factors and margins
    clusterPtCv=this.canvas.cVPt(clusterPt);
    boundACv=this.canvas.cVPt(boundA);
    boundBCv=this.canvas.cVPt(boundB);
    cEndCv=this.canvas.cVPt(cEnd);

    //now create the required lengths for the path
    var cToBoundA= clusterPtCv.y-boundACv.y;
    var cToBoundB= clusterPtCv.y-boundBCv.y;
    var cToHt= cEnd.x-clusterPtCv.x;

    //draw the element as a path on the raphael canvas and assign it to
    //the node element
    nodeIn.element=this.canvas.paper.path{
        "M "+clusterPtCv.x+" "+clusterPtCv.y+
        " l 0 "+cToBoundA+" 1 0 "+(-1)*cToBoundA+
        " l 0 "+cToBoundB+" 1 0 "+(-1)*cToBoundB+
        " l "+cToHt+" 0"
    };
    nodeIn.drawBoundingBox(this.canvas.paper);//draw its bounding box
    this.styleCluster(nodeIn);
    return true;//has no effect
}

/**
  * Draws a given cluster node onto the summary dendrogram.
  * @method drawCluster
  * @param nodeIn {ClusterNode object} The cluster node to be drawn
  * @returns {boolean} but value not used
*/

//Methods

*/

/*

Dendrogrammer.js
*/
then set element and bounding box to null and return
* else<br />
* set element and bounding box to null<br />
* set the left hand end to be the cutoffIn Ht and draw the new cluster
* @method drawClusterSummary
* @param cutoffIn {number} the Ht at which the graph is truncated
* @param nodeIn {ClusterNode object} The cluster node to be drawn.
* @returns {boolean} but value not used
*/

function Dendrogram_drawClusterSummary(nodeIn,cutoffIn){
    if(!nodeIn.isRoot()){
        if(nodeIn.parent.xHt<cutoffIn){
            nodeIn.element=null;
            nodeIn.bBox=null;
            return true;
        }
    }
    var parentHt=null;// to hold ht of parent or notional parent ht for root
    //if nodeIn is the root set parentHt to nodeIn.xHt*1.05
    //i.e. 5% higher than max ht
    //this will give an area to interact with above the root node
    if (nodeIn.isRoot()){
        parentHt = nodeIn.xHt*1.05;
    }else{
        parentHt=nodeIn.parent.xHt;
    }
    //draw the cluster by starting at the cluster point node.xHt, node.yPos
    //scribing over the crossbar section returning to the cluster point
    //and finishing by scribning up the stem section creating one shape
    //from a single path call adjust all Hts by subtracting the cutoffIn Ht
    //define the graph points for the cluster with no scaling factors
    //first check cluster Ht and adust to cutoff if below cutoff Ht
    if(nodeIn.xHt<cutoffIn){
        var clusterSummaryHt=cutoffIn;
    }else{
        clusterSummaryHt=nodeIn.xHt;
    }
    var clusterPt=new point(clusterSummaryHt-cutoffIn,nodeIn.yPos);
    var boundA=new point(clusterSummaryHt-cutoffIn,nodeIn.childA.yPos);
    var boundB=new point(clusterSummaryHt-cutoffIn,nodeIn.childB.yPos);
    var cEnd=new point(parentHt-cutoffIn,nodeIn.yPos);
    //now convert them to canvas coordinates incorporating
    //scale factors and margins
    clusterPtCv=this.canvas.cVPtSummary(clusterPt);
    boundACv=this.canvas.cVPtSummary(boundA);
    boundBCv=this.canvas.cVPtSummary(boundB);
    cEndCv=this.canvas.cVPtSummary(cEnd);
    //now create the required lengths for the path
    var cToBoundA= clusterPtCv.y-boundACv.y;
    var cToBoundB= clusterPtCv.y-boundBCv.y;
    var cToHt= cEnd.x-clusterPtCv.x;
    //draw the element as a path on the raphael canvas and assign it
    //to the node element
    nodeIn.element=this.canvas.paper.path(
        "M " +clusterPtCv.x+ "l 0 " +cToBoundA+ " 0 " +(-1*cToBoundA)+
        " 1 0 " +cToBoundB+ " 0 " +(-1*cToBoundB)+
        " 1 " +cToHt+ " 0"
    );
    nodeIn.drawBoundingBox(this.canvas.paper);//draw its bounding box
dendrogrammer.js

```javascript
this.styleCluster(nodeIn);
return true; //has no effect
}
/**
* Styles a given cluster
* @method styleCluster
* @param nodeIn {ClusterNode object} The cluster node to style (a Node)
*/
function Dendrogram_styleCluster(nodeIn){
  //use the attr method to style the axis object
  nodeIn.element.attr(
    {
      'stroke-width':this.strokeWidth,
      'stroke-linecap':'round',
      title: nodeIn.shortDescription
    }));
  this.addEvents(nodeIn,this.gui,this); //add events to the new node
}
/**
* Styles a given leaf
* @method styleLeaf
* @param nodeIn {ClusterNode object} The leaf node to style
*/
function Dendrogram_styleLeaf(nodeIn){
  //use the attr method to style the axis object
  nodeIn.element.attr(
    {
      'stroke-width':this.strokeWidth,
      'stroke-linecap':'round',
      //title: "Leaf "+nodeIn.idNo
      title: nodeIn.shortDescription
    }));
  this.addEvents(nodeIn,this.gui,this); //add events to the new node
}
/**
* Draws a leaf graphic element
* @method drawLeaf
* @param nodeIn {ClusterNode object} The node describing the leaf
* @return {boolean} but value not used
*/
function Dendrogram_drawLeaf(nodeIn){
  //generate a Raphael canvas pixel coordinate
  var leafPt=this.canvas.getLeafPt(nodeIn.yPos);

  //points and dimension for the stem part
  var parentHt=nodeIn.parent.xHt;
  var cEnd=new point(parentHt,nodeIn.yPos);
cEndCv=this.canvas.cVPt(cEnd);
  var cToHt= cEnd.x-leafPt.x;

  //scribe out from the leaf pt to a box around the leaf pt. The box to
  //enclose an imaginary circle radius of this.leafRadius.
  //Then return to the point on the edge of the box in line with the stem.
  //Then scribe the stem up to the parent height (or down to the parent
  //height if the parent height is really low or zero.

  nodeIn.element=this.canvas.paper.path(
    "M "+leafPt.x+" "+leafPt.y+
    "l "+this.leafRadius+" 0 "+
    "0 "+(-1*this.leafRadius)+
    "l "+(-2*this.leafRadius)+" 0"+
    "0 "+(2*this.leafRadius)+
    "l "+(2*this.leafRadius)+" 0"+
    "0 "+(-1*this.leafRadius)+
    "l "+(cToHt-this.leafRadius)+" 0"
  );

  nodeIn.drawBoundingBox(this.canvas.paper); //draw its bounding box
```
Dendrogrammer.js

```
this.styleLeaf(nodeIn);//and add events
return true; //has no effect
```

**/* Draws a leaf graphic element for the Summary dendrogram. Summary leaves are* different to those on the normal chart. They have no base and vary* in length if they are to be drawn at all.<br />*/
* If the xHt of the parent < cutoffIn then set element to null and return<br />
else set the left hand end to be the cutoffIn Ht and draw the leaf<br />
*/
* @method drawLeafSummary
* *
* @param nodeIn {ClusterNode object} The node describing the leaf (a Node)
* @param cutoffIn {number} the Ht at which the graph is truncated
* @returns {boolean} but value not used.
*/

```
function Dendrogram_drawLeafSummary(nodeIn,cutoffIn){
  if(nodeIn.parent.xHt<cutoffIn){
    nodeIn.element=null;
    nodeIn.bBox=null;
    return true;
  }
  //generate a Raphael canvas pixel coordinate at the truncated Ht
  var leafPt= this.canvas.getLeafPtSummary(nodeIn.yPos);
  //points and dimension leaf summary line
  var parentHt=nodeIn.parent.xHt;
  var cEnd= new point(parentHt-cutoffIn,nodeIn.yPos);
  var cEndCv=this.canvas.cVPtSummary(cEnd);
  var cToHt= cEndCv.x-leafPt.x;
  //represent the truncated leaf
  //Scribe a T shape as for a cluster but the head of the T is
  //2/3rds the width of a leafSpaceSummary. The leg of the T runs
  //from the parent ht down to the cutoff ht, i.e. the base line.
  //The T head is needed to give the leaf some body for events
  //on its bounding box.
  nodeIn.element= this.canvas.paper.path("M "+leafPt.x+" "+leafPt.y+" l 0 "+(-this.canvas.leafSpaceSummary*1/3)+" l 0 "+(this.canvas.leafSpaceSummary*2/3)+" l 0 "+(this.canvas.leafSpaceSummary*1/3)+" l 0 "+cToHt+" 0"");
  nodeIn.drawBoundingBox(this.canvas.paper);//draw its bounding box
  this.styleLeaf(nodeIn);
  return true; //has no effect
}
```
dendrogrammer.js

/**
 * Declares event handlers for the shape using the anonymous function.<br />
 * Events for:<br />
 * - mouseover, or tap in iPad<br />
 * - mouseout, or tap something else in iPad<br />
 * - mouse click, or tap in iPad<br />
 */

* The parameters are passed here rather than referred to by "this" even
* when they are attributes local to the class because the "this" keyword
* has its scope hijacked by the event. The keyword, "this",
* becomes the event object during an event.
* @method addEvents
* @param nodeIn {ClusterNode object} The cluster tree node concerned.
* @param guiIn {Gui object} the gui.
* @param dendrogramIn {Dendrogram object} the dendrogram. Needed to
* avoid "this" keyword for event handling
 */

function Dendrogram_addEvents(nodeIn,guiIn,dendrogramIn){
    //show branch
    //aim is to produce a highlighting of the given cluster and all its childen
    nodeIn.bBox.node.onmouseover =
    function(){
        nodeIn.showBranch(nodeIn, true, dendrogramIn.isSummary);
        this.style.cursor = 'pointer';
    };
    nodeIn.bBox.node.onmouseout =
    function(){
        nodeIn.showBranch(nodeIn, false, dendrogramIn.isSummary);
        this.style.cursor = 'pointer';
    };
    //display descendants reports details of descendant nodes
    nodeIn.bBox.node.onclick =
    function(){
        nodeIn.displayDescendants(nodeIn,guiIn);
    };
}

/**
 * Declares event handlers for the shape using the anonymous function.<br />
 * Events for:<br />
 * - mouseover, or tap in iPad<br />
 * - mouseout, or tap something else in iPad<br />
 * - mouse click, or tap in iPad<br />
 */

* The parameters are passed here rather than referred to by "this" even
* when they are attributes local to the class because the "this" keyword
* has its scope hijacked by the event. The keyword, "this",
* becomes the event object during an event.
* @method addEventsToBand
* @param groupIn {Group object} group which contains the band for the events
* @param guiIn {Gui object} the gui
* @param dendrogramIn {Dendrogram object} the dendrogram. Needed to
* avoid "this" keyword for event handling
 */

function Dendrogram_addEventsToBand(groupIn,guiIn,dendrogramIn){
    //show branch
    //aim is to produce a highlighting of the cluster represented
    //by the given group and all its childen.
    groupIn.band.node.onmouseover =
    function(){
        groupIn.ancestorNode.showBranch(groupIn.ancestorNode, true, dendrogramIn.isSummary);
        this.style.cursor = 'pointer';
    };
    groupIn.band.node.onmouseout =
    function(){
        groupIn.ancestorNode.showBranch(groupIn.ancestorNode, true, dendrogramIn.isSummary);
    };
}
dendrogrammer.js

```javascript
false, dendrogramIn.isSummary);
}
//display descendants reports details of descendant nodes
groupIn.band.node.onclick =
  function(){
    groupIn.ancestorNode.displayDescendants(groupIn.ancestorNode,guiIn);
  };
}

 /**
 * Appends cluster groups to a list
 * if the nodeIn is above the threshold then call itself recursively on childA and child B
 * else make a group from its descendants, append it to the list and return
 */
* @method appendGroups
* @param nodeIn {Group object} the node being examined.
* @param groupListIn {GroupList object} the groupList to hold the resulting groups.
* @param thresholdIn {number} the threshold value used to define the groups.
*/

function Dendrogram_appendGroups(nodeIn,groupListIn,thresholdIn){
  //check for nodeIn==null is so return groupList unchanged
  if(nodeIn==null){
    return groupListIn;
  } //attempt no processing on a null node
  //if the node is above the threshold then call itself on childA and child B
  if(nodeIn.xHt>thresholdIn){
    groupListIn=this.appendGroups(nodeIn.childA,groupListIn,thresholdIn);
    groupListIn=this.appendGroups(nodeIn.childB,groupListIn,thresholdIn);
    return groupListIn;
  }else{
    //make a group from this node's descendants and return
    var foundGroup=new Group(nodeIn.getDescendants(nodeIn),nodeIn);
    //append to group list
    groupListIn.addToEnd(foundGroup);
    return groupListIn;
  }
}

 /**
 * Dendrogram_applyGroupingThreshold
 * 1) Defines groups of leaves based on a threshold mergeHt.<br />
 * 2) Overlays rectangles on the dendrogram to highlight. Those clusters below the threshold mergeHt form groups differentiated by colour or shading of the overlayed rectangles.<br />
 * 3) Display the details of the groups under the Groups tab.<br />
 * 4) Feedback to the user via a dialogue that all this has occured.<br />
 */
* @method applyGroupingThreshold
* @param thresholdIn {number} the threshold to be applied.
*/

function Dendrogram_applyGroupingThreshold(thresholdIn){
  //This is a depth first search to find the first cluster below the threshold (with mergeHt < thresholdIn). The depth first search is carried out by the appendGroups() method which calls itself recursively. The search is started here by calling appendGroups with the tree root node.
* @method applyGroupingThreshold
* @param thresholdIn {number} the threshold to be applied.
*/

Page 25
```javascript
validThreshold = 0;

if (validThreshold < this.canvas.originHt) {
    validThreshold = this.canvas.originHt;
}

if (validThreshold > this.canvas.largestHt) {
    validThreshold = this.canvas.largestHt;
}

var messageString = "";

this.groupList.reset(); // empty any existing group list
this.groupList.threshold = validThreshold; // store the threshold

// first build the group list
this.groupList = this.appendGroups(this.cTree.getRoot(),
                                    this.groupList, validThreshold);

// output the groups list to the div
this.displayGroupList(this.groupList, messageString);
this.drawGroupBands(this.groupList); // draw the band size();
// in case an invalid threshold was entered
this.gui.showThreshold(validThreshold);
this.gui.showNoOfGroups(this.groupList.getSize());

/**
 * Displays a single group in a div on the groups tab
 * by appending it to the current contents of the groups tab
 *
 * @method displayOneGroup
 *
 * @param {Array of ClusterNode objects} leafArrayIn the leaf array which
 * represents the group
 *
 * @param {string} groupIdIn a string identifier or label for the group
 *
 */
function Dendrogram_displayOneGroup(leafArrayIn, groupIdIn) {
    var outputString = groupIdIn + " : <br/>
    for (var i = 0; i < leafArrayIn.length - 1; i++) {
        outputString += leafArrayIn[i].shortDescription + " , ";
    }
    outputString += leafArrayIn[leafArrayIn.length - 1].shortDescription;

    var outputHtml = '<div class="groupDiv" title="' + groupIdIn + '" id="' + groupIdIn + '">' + outputString + '</div>';
    outputHtml += outputString + '</div>'; // append the group div to the current content of the groups tab

    $(this.gui.groupsTab).html($(previousContent + outputHtml));
}

/**
 * Outputs details of the designated group list.
 * Makes calls to this.displayOneGroup() and this.gui.notifyGroupFormation().
 *
 * @method displayGroupList
 *
 * @param {GroupList object} groupListIn an array of groups.
 *
 * @param {string} messageIn String to preceed the display of the list
 *
 * Each group is an array of leaves.
 *
 */
function Dendrogram_displayGroupList(groupListIn, messageIn) {
    var tempLeafArray = new Array();
    var resultsSummary = messageIn + " groups were formed using";
    resultsSummary += " the threshold set at " + groupListIn.threshold;
    resultsSummary += ". See the 'Groups' tab for details."
```
dendrogrammer.js
//var resultsString="Contents of Group List : ";
var resultsString="";
resultsString+="Threshold: "+groupListIn.threshold+"<br/>";
if(groupListIn.isEmpty()){//if the list is empty say so
resultsString+="The list is empty";
return;
}
//add the list introduction into the output div clearing what was there
var groupsIntroHtml='<div class="groupDiv" title="Intro" id="Intro">';
groupsIntroHtml+=resultsString+'</div>';
$(this.gui.groupsTab).html(groupsIntroHtml);
//now trundle through the list displaying the groups
this.displayOneGroup(groupListIn.getFirst().leafList,'Group-1');
var limit=groupListIn.getSize();
for (i=2;i<=limit;i++){
this.displayOneGroup(groupListIn.getNext().leafList,'Group-'+i);
}
//inform user
this.gui.notifyGroupFormation(resultsSummary,groupListIn);
}
/**
* Takes a group and draws an overlay on the graph to delineate it.
* The band is a rectangle of length (the X dimension), thresholdIn.
* Its top and bottom (the Y dimension) defined by groupIn.topBound and
* groupIn.bottomBound<br /><br />
*
* Makes a call on this.addEventsToBand().
*
* @method drawOverlayBand
* @param groupListIn {Group object} an array of leaves.
* @param {number} the threshold defining the groups
*/
function Dendrogram_drawOverlayBand(groupIn,thresholdIn){
//draw the band assigning it to the group's band attribute;
//First gather the coordinates and dimensions
// a raphael rectangle needs x,y,width,and ht.
//Dimensions without scaling or margins://create an origin point for the band rectangle and the top left corner.
//(At the base of the dendrogram)
var bandPt=new point(0,groupIn.topBound-0.5);
// now the rentangle width
//check for summary graph and make allowances
if (this.canvas.originHt>0){
//it is a summary graph
var bandLength=thresholdIn-this.canvas.originHt;
var bandHt=(groupIn.bottomBound-groupIn.topBound)+1;
//now convert them to canvas coordinates incorporating
//scale factors and margins
bandPt=this.canvas.cVPtSummary(bandPt);
bandLength=bandLength*this.canvas.htScaleFactorSummary;
bandHt=bandHt*this.canvas.leafSpaceSummary;
}else{
//it is a normal graph
var bandLength=thresholdIn;
var bandHt=(groupIn.bottomBound-groupIn.topBound)+1;
//now convert them to canvas coordinates incorporating
//scale factors and margins
bandPt=this.canvas.cVPt(bandPt);
bandLength=bandLength*this.canvas.htScaleFactor;
bandHt=bandHt*this.canvas.leafSpace;
}
//draw it
groupIn.band=this.canvas.paper.rect(
bandPt.x,
bandPt.y,
bandLength,
bandHt);
//style it transparent
groupIn.band.attr(
Page 27


```javascript
function Dendrogram_drawGroupBands(groupListIn) {
  // loop through the list drawing bands on the graph
  if (groupListIn.isEmpty()) {
    return;
  }

  var noOfGroups = groupListIn.getSize();
  for (var i = 2; i <= noOfGroups; i++) {
    // do the drawing
    this.drawOverlayBand(groupListIn.getNext(), groupListIn.threshold);
    // and alternate styling
    if (i % 2 == 0) {
      groupListIn.getCurrent().band.attr({
        'fill': this.groupBandAlternateColour,
        'stroke-opacity': 0,
        'fill-opacity': this.groupBandOpacity
      });
    }
  }

  // record that the groups have been drawn
  groupListIn.drawn = true;
}

function Dendrogram_drawNode(nodeIn) {
  // draws the tree by starting with the root node and recursively calling itself
  // and recursively calling itself
  var childA = nodeIn.childA;
  var childB = nodeIn.childB;

  // if the node is a leaf then draw the leaf and return
  if (nodeIn.isLeaf()) {
    this.drawLeaf(nodeIn);
    this.drawLeafLabel(nodeIn);
    return true;
  }
```
```javascript
//draw the node
this.drawCluster(nodeIn);

//draw the child nodes
this.drawNode(childA);
this.drawNode(childB);

return true; //has no effect
}

/**
 * Draws the summary tree by starting with the root node and recursively
 * calling itself<br /><br />
 * Makes calls on this.drawClusterSummary() and this.drawLeafSummary().
 *
 * @method drawNodeSummary
 * @param nodeIn {ClusterNode object} the node to be drawn
 * @param cutoffIn {number} the mergeHt below which nothing is to be drawn
 * @return {Boolean} but value not used.
 */
function Dendrogram_drawNodeSummary(nodeIn,cutoffIn){
    var childA=nodeIn.childA;
    var childB=nodeIn.childB;

    //if the node is a leaf then draw the leaf and return
    //else draw the cluster
    if(nodeIn.isLeaf()){
        this.drawLeafSummary(nodeIn,cutoffIn);
        return true;
    }
    this.drawClusterSummary(nodeIn,cutoffIn);

    //draw the node
    //draw the child nodes
    this.drawNodeSummary(childA,cutoffIn);
    this.drawNodeSummary(childB,cutoffIn);

    return true; //has no effect
}

/**
 * Draws the dragable threshold adjuster bar
 *
 * @method drawThresholdBar
 *
 * @return {Boolean} but value not used.
 */
function Dendrogram_drawThresholdBar(){
    //set the graph point for the top left of the bar
    //It is set a number of leaf spaces above the top so it should protrude up
    var barPt=new point(this.canvas.largestHt,this.thresholdBarOverlapTop*-1);

    //generate a Raphael canvas pixel coordinate
    var barPtCv=this.canvas.cVPt(barPt);
    //set the bar length (Ht)
    var leafSpace=this.canvas.leafSpace;
    if (this.canvas.originHt>0) {//it is a summary graph
        leafSpace=this.canvas.leafSpaceSummary;
    }
    var barLength=(this.canvas.noOfLeafs+this.thresholdBarLengthPlus)*leafSpace;

    //draw it
    this.thresholdBar=this.canvas.paper.rect(
        barPtCv.x,
        barPtCv.y,
        barLength,
        barLength);

    //style it
    this.thresholdBar.attr(
        
        {'stroke':'green',
        'fill':'green',
        'stroke-opacity':0.5,
    
```
dendrogrammer.js

'fill-opacity':0.5,
'title': 'Threshold adjuster bar'};
//make it draggable
this.thresholdBar.drag(this.move, this.dragger, this.up);
return true; //has no effect

/**
 * Detects the threshold bar position and returns that in terms of the xaxis
 * merge Ht.<br /><br />
 * Makes a call on this.canvas.getGraphXfromCvX().
 * @method getThresholdPos
 * @param barIn {graphic object} The threshold bar whose pos is to be detected
 * @return {number} the xPos of the bar
 */
function Dendrogram_getThresholdPos(barIn){
  //var barX=this.thresholdBar.x;//the shape x coord
  return this.canvas.getGraphXfromCvX(barIn.x);
}

/**
 * Sets Dendrogram properties to serve as parameters for the
 * Raphael drag() method.
 * Based on Dmitry's Graffle example http://raphaeljs.com/graffle.html
 * <br /><br />
 * Makes a call on guiIn.showThreshold().
 * @method setDragParams
 * @param canvasIn {Canvas object} The canvas
 * @param guiIn {Gui object} The gui
 */
function Dendrogram_setDragParams(canvasIn, guiIn){

  var originHtNum=canvasIn.originHt-0;//the -0 forces a number type conversion
  //##############################################################
  // Set up drag method vars for use in drag() method
  // -dragger is the storage of the original start pos
  // -move is the acceptance of the new coord(s) and shape redraw
  // -up is the action to do on release
  //###############################################################
  this.dragger = function () {
    this.ox = this.attr("x");
    //this.oy = this.attr("y"); //uncomment to unlock y movement
  };
  this.move = function (dx, dy) {
    var att = new Object();
    var limit=canvasIn.largestHt;
    //limit the movement to with the x axis bounds
    if (((this.ox + dx)>=canvasIn.marginLeft) & &
      ((this.ox + dx)<=(canvasIn.cvX(limit)))){
      att.x=this.ox + dx;
    }
    else{
      //limit the move
      if((this.ox + dx)<canvasIn.marginLeft){//at low end
        att.x=canvasIn.marginLeft;
      }else if ((this.ox + dx)>canvasIn.cvX(limit)){//at high end
        att.x=canvasIn.cvX(limit);
      }
    }
    //att.y=this.oy + dy;//uncomment to unlock y movement
    this.attr(att);
    //thePaper.safari();//forces a redraw
    //calculate value to display depending on originHt (affected
    //by summary graph)
    if (originHtNum>0){//it is a summary graph
      var displayVal=
        (((att.x-canvasIn.marginLeft)/canvasIn.htScaleFactorSummary));
    }
    return true;
  };
  // -up is the action to do on release
  this.up = function () { return true; };
}
```javascript
// Sets the threshold bar position in response to some other control

function Dendrogram_setThresholdBarToMatch(thresholdIn) {
    // Check for this being a summary graph and allow for altered threshold position
    if (this.canvas.originHt > 0) {
        // it is a summary graph
        var endX = (thresholdIn - this.canvas.originHt) * this.canvas.htScaleFactorSummary * this.canvas.marginLeft;
    } else { // just a normal graph
        var endX = (thresholdIn * this.canvas.htScaleFactor) + this.canvas.marginLeft;
    }
    var startX = this.thresholdBar.attr('x');
    var dx = endX - startX;
    this.thresholdBar.translate(dx, 0); // translate the bar by the x difference
    return thresholdIn; // the merge ht
}

// Sets the threshold bar to the maximum merge Ht.

function Dendrogram_setThresholdBarToMax() {
    var startX = this.thresholdBar.attr('x');
    // set the end X to be that of the max Ht
    var endX = this.canvas.largestHt * this.canvas.htScaleFactor * this.canvas.marginLeft;
    var dx = endX - startX;
    this.thresholdBar.translate(dx, 0);
    return this.canvas.largestHt; // the merge ht
}

// Sets the threshold bar to zero (or the origin Ht in the case of Summary graph)

function Dendrogram_setThresholdBarToZero() {
    var startX = this.thresholdBar.attr('x');
    var endX = this.canvas.marginLeft;
    var dx = endX - startX;
}
```

```javascript
this.thresholdBar.translate(dX,0);
return this.canvas.originHt;//the merge ht at the orgin
}
/** *
* Sets the threshold bar to one step down the
* merge Ht.
* *
* @method setThresholdBarToLess
* *
* @param fractionIn {number} the fraction of the full travel to make each step
* e.g. 20 means each step is 1/20th of the full travel
* *
* @return {number} the Merge Ht value of the new threshold position
*/
function Dendrogram_setThresholdBarToLess(fractionIn){
  var startX=this.thresholdBar.attr('x');
  //Check for this being a summary graph and allow for altered threshold
  //position
  if (this.canvas.originHt>0){
    //it is a summary graph
    var step=((this.canvas.largestHt-this.canvas.originHt)/fractionIn)*
      this.canvas.htScaleFactorSummary;
  }
  else{
    //just a normal graph
    var step=(this.canvas.largestHt/fractionIn)*this.canvas.htScaleFactor;
  }
  //set the end X to be one step down
  var endX=startX-step;
  if(endX<this.canvas.marginLeft){ //check not off end of chart
    endX=this.canvas.marginLeft;
  }
  var dX=endX-startX;
  this.thresholdBar.translate(dX,0);
  //Check for this being a summary graph and allow for altered return pos
  if (this.canvas.originHt>0){
    //it is a summary graph
    return ((endX-this.canvas.marginLeft)/this.canvas.htScaleFactorSummary)
      +this.canvas.originHt;//the merge ht
  }
  //just a normal graph
  return (endX-this.canvas.marginLeft)/this.canvas.htScaleFactor; //the merge ht
}
/** *
* Sets the threshold bar to one step up the
* merge Ht.
* *
* @method setThresholdBarToMore
* *
* @param fractionIn {number} the fraction of the full travel to make each step
* e.g. 20 means each step is 1/20th of the full travel
* *
* @return {number} the Merge Ht value of the new threshold position
*/
function Dendrogram_setThresholdBarToMore(fractionIn){
  var startX=this.thresholdBar.attr('x');
  //Check for this being a summary graph and allow for altered threshold
  //position
  if (this.canvas.originHt>0){
    //it is a summary graph
    var step=((this.canvas.largestHt-this.canvas.originHt)/fractionIn)
      *this.canvas.htScaleFactorSummary;
  }
  else{
    //just a normal graph
    var step=(this.canvas.largestHt/fractionIn)*this.canvas.htScaleFactor;
  }
  //set the end X to be one step up
  var endX=startX+step;
```
}
dendrogrammer.js

//this next line works for both normal and summary chart ok
if(endX>((this.canvas.largestHt*this.canvas.htScaleFactor)+this.canvas.marginLeft)){//check not off end of chart
    endX=(this.canvas.largestHt*this.canvas.htScaleFactor)+this.canvas.marginLeft;
}
var dX=endX-startX;
this.thresholdBar.translate(dX,0);
//Check for this being a summary graph and allow for altered return pos
if((this.canvas.originHt>0){
    //it is a summary graph
    return ((endX-this.canvas.marginLeft)/this.canvas.htScaleFactorSummary)+this.canvas.originHt;//the merge ht
}
//normal graph
return (endX-this.canvas.marginLeft)/this.canvas.htScaleFactor;//the merge ht
}

/**
 * Sets the threshold to be one which will give the required no of groups.
 * Takes the given no of groups and by a binary search sucessively applies a 
 * new threshold until it finds one that fits the required no of groups and 
 * leaves the threshold bar set there.<br />

 * The binary search is limited to 75 iterations if it does not find the right 
 * spot by then it stops and returns the nearest no of groups it could get. 
 *<br />
 *
 * Group formation can fail to achieve the number of requested groups. There 
 * are 4 ways this can occur:<br />
 */

* 1) after 75 binary search iterations if some data set had many many 
   clusters at very similar hieghts, it might be that the requested 
   no. of groups was not reachable, due to running out of iterations. 
   I recon this is highly unlikely at 75 iterations.<br />
* 2) An impossible number of groups was requested (maybe a decimal, or 
   less than 1, or greater than the number of leaves<br />
* 3) if there are any leaves which merge at Ht zero then these cannot be 
   allocated to separate groups. They must share a group with their 
   zero Ht cluster mate. <br />
* 4) if creating groups on a summary graph then if the threshold would 
   need to be below the cutoff to create the requested number of groups 
   then it won't go below the cutoff Ht.<br />
*
* Makes a call on:
*  this.displayGroupList()<br />
*  this.drawGroupBands()<br />
*  this.setThresholdBarToMatch()<br />
*  this.gui.showThreshold()<br />
*  and<br />
*  this.gui.showNoOfGroups()<br />
* Also makes calls on a number of GroupList methods
* 
* @method setThresholdForNoOfGroups
*
* @param noOfGroupsIn (number) the required no of groups
*
*/

function Dendrogram_setThresholdForNoOfGroups(noOfGroupsIn){
    //eliminate less than 1 and also iron out any erroneous data entered
    //by defaulting to 0 if entry is dodgy
    var desiredGroups=noOfGroupsIn-0;//Enforces a number
    //now check for in being NaN, if so then set it to zero
    if(isNaN(desiredGroups)){
        desiredGroups=0;
    }
    //only allow minimum of 1
    if(desiredGroups<1){
        desiredGroups=1;
    }
    var currentHigh=this.canvas.largestHt;
var currentLow = this.canvas.originHt;
// var currentGroups = this.canvas.noOfLeafs;
var currentGroups = -1;
var loopCount = 0;
var trialThreshold = currentHigh; // set in loop
var message = ""; // user feedback
// loop while currentGroups > noOfGroupsIn and loopCount < 50
// increment loopCount
// set trialThreshold to half way between high and low
// apply trialThreshold.
// set currentThreshold to the generated no of groups
// end loop
// if currentGroups > noOfGroupsIn then output alternative groups chose diag
// else indicate success
// show the current threshold anyway.
while ( (currentGroups != desiredGroups) && (loopCount < 75) ) {
    loopCount++;
    if (desiredGroups != 1) { // don't try a new threshold of we want just 1 grp
        trialThreshold = currentLow + (currentHigh - currentLow) / 2;
    }
    // if the groupList has been drawn to canvas then undraw
    if (this.groupList.drawn) {
        this.groupList.reset();
    } else {
        this.groupList.resetButLeaveCanvasAlone();
    }
    this.groupList.threshold = trialThreshold;
    this.groupList = this.appendGroups(this.cTree.getRoot(),
        this.groupList, trialThreshold);
    currentGroups = this.groupList.getSize();
    if (currentGroups < desiredGroups) { // threshold is too high
        currentHigh = trialThreshold;
    } else {
        currentLow = trialThreshold;
    }
}
if (currentGroups != noOfGroupsIn) {
    // output message about unable to match up. This message is tagged on
    // the front of the normal dialog about formed groups.
    // check for summary chart and output a message as appropriate
    if (this.canvas.originHt > 0) {
        message = "The desired no. of groups, " + noOfGroupsIn + " was not possible. Maybe you did not enter a number or " + "or you entered a decimal or your number" + " was outside the number of leaves. Or, as you are working on a " + "truncated summary " + "dendrogram at the moment, it is possible that the cutoff " + "is higher than the threshold would need to be to achieve " + "the required number of groups. In that case you will " + "need to restore the full dendrogram and try again. Instead, ";
    } else { // this is not a summary chart
        message = "The desired no. of groups, " + noOfGroupsIn + " was not possible. (The minimum no. of groups that can " + "be formed is 1. There is a theoretical maximum of " + "the number of leaves but if there are any leaf " + "pairs that cluster at a dissimilarity ht. of zero" + " then these must share a group). Instead, ";
    }
} else {
    // indicate total success ... by mentioning nothing extra
    message = "";
}
// output the groups list to the div
this.displayGroupList(this.groupList, message);
this.drawGroupBands(this.groupList); // draw the bands
// set the threshold bar to match
this.setThresholdBarToMatch(trialThreshold);
this.gui.showThreshold(trialThreshold); // display the threshold value
// display the final no of groups achieved
dendrogrammer.js

```
this.gui.showNoOfGroups(currentGroups);

return true; // no effect
```

```javascript
/**
* Removes the groups from the dendrogram <br />
* 1) reset the group list<br />
* 2) reset the threshold to max and if its field<br />
* 3) reset group no field to 1<br />
* 4) undraw the bands<br />
*<br />
* Makes calls on gui methods and the this.groupList.reset() method.
*<br />
* @method removeGroups
*/

function Dendrogram_removeGroups(){

  this.groupList.reset(); // empty the group list
  this.gui.showThreshold(this.setThresholdBarToMax()); // reset the threshold
  // clear groups tab
  $(this.gui.groupsTab).html("The groups have been removed.");
  this.gui.clearGroupNotifier(); // clear the notifier tab
  this.gui.showNoOfGroups(1); // reset the display in the no of Groups field
}

/**
* Removes all of the drawn elements including the axes
* @method removeAll
*/

function Dendrogram_removeAll(){

  // remove any groups
  this.removeGroups();
  // remove the axes and labels
  this.axes.remove();
  // remove nodes
  // this.removeNode(this.cTree.getRoot());
  // remove the threshold bar
  this.thresholdBar.remove();
}

/**
* Draws a summary dendrogram truncated at the current threshold ht.<br />
* 1) record the cutoff Ht from the threshold pos<br />
* 2) clear the canvas by removing the paper object<br />
* 3) set a new x axis scale factor<br />
* 4) create a new paper object of appropriate size<br />
* 5) draw new axes<br />
* 6) draw the summary dendrogram taking into account the cutoff<br />
* @method drawSummary
*/

function Dendrogram_drawSummary(){

  this.isSummary=true;
  var cutoff=this.gui.getThresholdVal()-0; // forces a number
  this.setThresholdBarToMax(cutoff); // threshold input may have been keyed

  this.canvas.paper.remove(); // erase the paper and all the drawn elements
  // remove any groups
  this.removeGroups();
  // set the summary scale factor in the canvas
  this.canvas.htScaleFactorSummary=this.canvas.calculateScaleFactor{
    this.canvas.largestHt-cutoff,
    this.canvas.widthForCanvas, this.canvas.margin,
    this.canvas.leafLabelSpace};
  // create a new paper for the new chart.
  // This requires that the new dimensions be calculated. x (width) is
  // to be the same as for original dendrogram
  this.canvas.createPaper{
    this.canvas.xWidth,
    ((this.canvas.noOfLeafs+1)
    *this.canvas.leafSpaceSummary)
    +this.canvas.margin*2)
```
dendrogrammer.js

// draw new axes
this.axes=new GraphAxes(theCanvas,cutoff,false);
// draw the tree in summary
this.drawNodeSummary(this.cTree.getRoot(),cutoff);
// set the canvas origin Ht to cutoff
this.canvas.originHt=cutoff;
// set up some functions as Raphael drag parameters
this.setDragParams(this.canvas,this.gui);
this.drawThresholdBar(); // draw the threshold tool onto the graph
this.gui.showThreshold(this.canvas.largestHt); // display the threshold value
this.gui.showNoOfGroups(1); // display the no of Groups value
}
/**
 * Redraws the dendrogram after displaying truncated graph
 * 1) clear the canvas by removing the paper object
 * 2) resets the canvas originHt to 0
 * 4) create a new paper object of appropriate size
 * 5) draw new axes
 * 6) draw the dendrogram
 *
 * @method restore
 */
function Dendrogram_restore(){
    this.isSummary=false;
    this.canvas.paper.remove(); // erase the paper and all the drawn elements
    // remove any groups
    this.removeGroups();
    // set the canvas origin Ht to 0
    this.canvas.originHt=0;
    // create a new paper for the new chart.
    // This requires that the new dimensions be calculated. x (width) is
    // to be the same as for original dendrogram
    this.canvas.createPaper((this.canvas.xWidth,
        (this.canvas.noOfLeafs+1)*this.canvas.leafSpace)+(this.canvas.margin*2)
    );
    // draw new axes
    this.axes=new GraphAxes(theCanvas,0,true);
    this.drawNode(this.cTree.getRoot()); // draw the tree;
    // set up some functions as Raphael drag parameters
    this.setDragParams(this.canvas,this.gui);
    this.drawThresholdBar(); // draw the threshold tool onto the graph
    this.gui.showThreshold(this.canvas.largestHt); // display the threshold value
    this.gui.showNoOfGroups(1); // display the no of Groups value
}
// Constructor function
/**
 * Some attributes are set here for tuning
 *
 * @constructor Dendrogram
 *
 * @param cTreeIn {ClusterTree object } the filled cluster tree root node
 * @param canvasIn {Raphael canvas object} the Raphael canvas on which to
draw the chart
 * @param stylesIn {Styles object } various styles used in the dendrogram
 * such as line thickness.
 * @param guiIn {Gui object } an object giving access to user interface methods
 * @param axesIn {GraphAxes object} the axes for the graph
 */
function Dendrogram(cTreeIn, canvasIn, stylesIn, guiIn, axesIn){
    // initialise attributes
    // For Tuning
    this.leafLabelLength=12;// number of characters allowed for the leaf labels
    this.thresholdBarWidth=20; // number of units wide
    this.thresholdBarLengthPlus=9;
    this.thresholdBarOverlapTop=4;
    // number of leafSpaces it protrudes above the chart.
```javascript
this.groupBandColour = 'purple';
this.groupBandAlternateColour = 'yellow';
this.groupBandOpacity = 0.4;

// other attributes
this.isSummary = false; // default. Set true if displaying summary dendrogram
this.styles = stylesIn;
this.cTree = cTreeIn;
this.canvas = canvasIn;
this.leafRadius = stylesIn.leafRadius;
this.strokeWidth = stylesIn.strokeWidth;
this.gui = guiIn;
this.axes = axesIn;
this.thresholdBar = null; // will be the draggable bar but null for now
this.dragger = null; // set later by setDragParams
this.move = null; // set later by setDragParams
this.up = null; // set later by setDragParams

// initialise the GroupList using the current value of the slider
this.groupList = new GroupList(this.gui.getThresholdVal());

// associate methods
this.drawNode = Dendrogram_drawNode;
this.drawNodeSummary = Dendrogram_drawNodeSummary;
this.drawLeaf = Dendrogram_drawLeaf;
this.drawLeafSummary = Dendrogram_drawLeafSummary;
this.drawLeafLabel = Dendrogram_drawLeafLabel;
this.drawCluster = Dendrogram_drawCluster;
this.drawClusterSummary = Dendrogram_drawClusterSummary;
this.applyGroupingThreshold = Dendrogram_applyGroupingThreshold;
this.appendGroups = Dendrogram_appendGroups;
this.displayGroupList = Dendrogram_displayGroupList;
this.drawGroupBands = Dendrogram_drawGroupBands;
this.drawOverlayBand = Dendrogram_drawOverlayBand;
this.displayOneGroup = Dendrogram_displayOneGroup;
this.drawThresholdBar = Dendrogram_drawThresholdBar;
this.setDragParams = Dendrogram_setDragParams;
this.getThresholdPos = Dendrogram_getThresholdPos;
this.setThresholdBarToMatch = Dendrogram_setThresholdBarToMatch;
this.setThresholdBarToMax = Dendrogram_setThresholdBarToMax;
this.setThresholdBarToZero = Dendrogram_setThresholdBarToZero;
this.setThresholdBarToLess = Dendrogram_setThresholdBarToLess;
this.setThresholdBarToMore = Dendrogram_setThresholdBarToMore;
this.setThresholdForNoOfGroups = Dendrogram_setThresholdForNoOfGroups;
this.removeGroups = Dendrogram_removeGroups;
this.drawSummary = Dendrogram_drawSummary;
this.removeAll = Dendrogram_removeAll;
this.restore = Dendrogram_restore;

this.styleCluster = Dendrogram_styleCluster;
this.styleLeaf = Dendrogram_styleLeaf;
this.addEvents = Dendrogram_addEvents;
this.addEventsToBand = Dendrogram_addEventsToBand;

// processing to be done by the Constructor
this.drawNode(this.cTree.getRoot()); // draw the tree
// set up some functions as Raphael drag parameters
this.setDragParams(this.canvas, this.gui);
this.drawThresholdBar(); // draw the threshold tool onto the graph
this.gui.showThreshold(this.canvas.largestHt); // display the threshold value
this.gui.showNoOfGroups(1); // display the no of Groups value
}

// End of Dendrogram class
This class has no methods

@class Styles

//Define functions for the methods
//none for now

//Constructor function
/**
 * @constructor Styles
 *
 * @param {number} strokeWidth
 * @param {number} leafRadius
 * @param {number} nodeHighlightColour
 */
function Styles(swIn, lrIn, cIn){
    //initialise attributes
    //e.g.
    this.strokeWidth=swIn;
    this.leafRadius=lrIn;
    this.nodeHighlightColour=cIn;
    this.animateTime=2000;//used for timing fades etc
    //associate methods
    //none for now
}

//End of Styles class

//GroupList class
/**
 * The group list contains a list of the groups formed from drawing a
 * similarity threshold on the dendrogram.
 * The list is an array of Groups. The GroupList methods are for interrogating,
 * traversing, adding to and removing from the list.
 */
@class GroupList

//Define functions for the methods
/**
 * Returns the length of the list
 *
 * @method getSize
 *
 * @return {number}
 */
function GroupList_getSize(){
    return this.gArray.length;
}

/**
 * Returns the first group in the list
 *
 * If the list is not empty set pointer to first item and return the first item
 * else null
 *
 * @method getFirst
 *
 * @return {Group object}
 */
function GroupList_getFirst(){
    if(!this.isEmpty()){
        this.pointer=0;
        return this.gArray[0];
    }
    return null;
}
/**
 * Returns the last group in the list if the list is not empty set pointer to last item and return the last item
 * else null
 * @method getLast
 * @return {Group object}
 */
function GroupList_getLast(){
    if(!this.isEmpty()){
        this.pointer=this.gArray.length-1;
        return this.gArray[this.gArray.length-1];
    }
    return null;
}
/**
 * Returns the next group in the list if pointer is less then the list length-1 then increments pointer and returns the item
 * else returns null
 * @method getNext
 * @return {Group object}
 */
function GroupList_getNext(){
    if(this.pointer<this.gArray.length-1){
        this.pointer++;
        return this.gArray[this.pointer];
    }
    return null;
}
/**
 * Returns the currently pointed group in the list
 * @method getCurrent
 * @return {Group object}
 */
function GroupList_getCurrent(){
    return this.gArray[this.pointer];
}
/**
 * Removes a group from the list
 * @method removeFromEnd
 * @return {Group object} the popped group
 */
function GroupList_removeFromEnd(){
    return this.gArray.pop();
}
/**
 * Adds a group to the list
 * @method addToEnd
 * @param groupIn {Group object} the group to be added
 */
function GroupList_addToEnd(groupIn){
    this.gArray.push(groupIn);
}
/**
 * Returns true if there are no items in the list
 * @method isEmpty
 */
dendrogrammer.js

* @return {boolean}
*/

function GroupList_isEmpty(){
    if (this.gArray.length==0){
        return true;
    }
    return false;
}
/**
 * Resets the group list to be empty and undraws the bands
 * @method reset
 * @return {boolean} but value not used.
*/
function GroupList_reset(){
    while (!this.isEmpty()){
        //empty the array and erase from canvas
        this.removeFromEnd().band.remove();
    }
    this.pointer=-1;
    this.drawn=false;
    return true;//has no effect
}
/**
 * Resets the group list to be empty but leaves the canvas alone.
 * Used if no bands have been drawn
 * @method resetButLeaveCanvasAlone
 * @return {boolean} but value not used.
*/
function GroupList_resetButLeaveCanvasAlone(){
    while (!this.isEmpty()){
        //empty the array
        this.removeFromEnd();
    }
    this.pointer=-1;
    return true;//has no effect
}
/**
 * @constructor GroupList
 *
 * @param thresholdIn {number} the mergeHt set to define the groups
 */
function GroupList(thresholdIn){
    //initialise attributes
    this.gArray= new Array();///<to hold the Groups
    this.pointer=-1;///<index of current group
    this.threshold=thresholdIn;
    this.drawn=false;//records whether or not the group list has been drawn
        //onto the graph

    //associate methods with defined functions
    this.isEmpty=GroupList_isEmpty;
    this.addToEnd=GroupList_addToEnd;
    this.removeFromEnd=GroupList_removeFromEnd;
    this.getCurrent=GroupList_getCurrent;
    this.getNext=GroupList_getNext;
    this.getSize=GroupList_getSize;
    this.getFirst=GroupList_getFirst;
    this.getLast=GroupList_getLast;
    this.reset=GroupList_reset;
    this.resetButLeaveCanvasAlone=GroupList_resetButLeaveCanvasAlone;
}

//End of GroupList class
//######################################################
//######################################################
//Group class
dendrogrammer.js

/**
 * A group contains a list of leaf nodes (such leaf nodes sharing a common ancestor). Groups are collected in a GroupList. <br /><br />
 * Important attributes:<br />
 * A group has attributes topBound and bottomBound which are the yPos of the topmost (nearest the origin) and bottommost (furthest from the origin) leaves in the group. These are used in drawing the group bands.
 * @class Group
 */

//Define functions for the methods
/**
 * Returns the first item. (A group will never be empty)
 * @method getFirst
 * @return {ClusterNode object} a leaf node
 */
function Group_getFirst(){
    return this.leafList[0];
}

/**
 * Return the last item. (A group will never be empty)
 * @method getLast
 * @return {ClusterNode object} a leaf node
 */
function Group_getLast(){
    return this.leafList[this.leafList.length-1];
}

//Constructor function
/**
 * @constructor Group
 * @param leafListIn {Array of ClusterNode objects} an array of leaf nodes
 * @param ancestorNodeIn {ClusterNode} the node which is the ancestor of all the other nodes in the list
 */
function Group(leafListIn, ancestorNodeIn){
    //initialise attributes
    this.leafList=leafListIn;
    this.ancestorNode=ancestorNodeIn;
    this.topBound=this.getFirst().yPos;
    this.bottomBound=leafListIn[0].yPos;
    this.topBound=leafListIn[leafListIn.length-1].yPos;
    this.bottomBound=this.getLast().yPos;
    this.band=null; //a drawn object to be assigned later by the dendrogram
    //associate methods
    this.getFirst=Group_getFirst;
    this.getLast=Group_getLast;
}

//End of Group class

//Gui class
/**
 * The Gui class.
 * A reference to the instance of this class can be passed to any class that needs to interact with the user or output onto the page.
 * It has methods for displaying various dialogs and outputing messages.
 * Also I have tried to gather in here any references to the DOM page elements such as specific divs or buttons. This is so as too isolate the rest of the code from these references. Instead

Page 41
dendrogrammer.js

* the rest of the code accesses the buttons/divs via the
* gui attribute to which they are assigned.<br /><br />
* Important Gui attributes:<br />
* - leafSearchURL - the destination for the search link in the groups dialogs.
* Obtained from the clusterTable passed into the constructor<br />
* - mobile - true if a mobile OS is detected. This affects the display
* behaviour of dialogs.<br />
* <br /><br />
* Some tunable attributes affecting dialog display behaviour:<br />
* - maxDialogHt<br />
* - maxMessageLengthToScroll - Beyond this and dialogs are ht limited
* and scroll.<br />
* Attributes referring to names of DOM objects which need accessed by
* the application:<br />
* <br />
* - groupsTab, thresholdField, groupNoField, canvasContainer, largestMergeHtDiv,
* leafNodesDiv.<br />
* The constructor: <br />
* - Calls this.bindTabEvents() to set up the group tab event.<br />
* - Calls the jQuery tabs() method to invoke the tabs effect behaviour on
* the page<br />
* - Calls the jQuery button() method to invoke the button styling and
* behaviour on the page<br />
* <br />
* @class Gui
* /

//Define functions for the methods
/**
* Notifies the user the groups have been formed via a dialog
* Activates the pulsating Groups tab
*/
* @method notifyGroupFormation
*
* @param message {string} the message to be shown in the dialog
* @param groupListIn {GroupList object} the group list to be output
*/
function Gui_notifyGroupFormation(message, groupListIn) {
  //pulsate effect for the groups tab
  $('pulser').addClass('highlight');
  $('pulser').text(groupListIn.getSize() + ' Groups');
  //Make it pulse 30 times. Each cycle is 1 second
  //There is an event set to stop it on tab click
  $('pulser').effect('pulsate', { times:30 }, 1000);

  //show the groups dialog
  this.showMessage(message, 'Groups formed');
}
/**
* Clears groups from the tab notifier
*/
* @method clearGroupNotifier
*/
function Gui_clearGroupNotifier() {
  //stop it pulsing
  $('pulser').stop(true, true);
  //reset the pulser tab text
  $('pulser').text('Groups');
  //reset the opacity
  $('pulser').css('opacity', 1);
}
/**
* Show the message dialog using the dialog groups div
* with an OK button
*/
* @method showMessage
*
function Gui_showMessage(message, title) {
    var dialogWidth = 300; // default width
    // show the message dialog using the dialog groups div
    $('"#dialog-groups"').html(message);
    // check on length of message and adjust width if it is long
    if (message.length > this.maxMessageLengthToScroll / 5) {
        dialogWidth = this.maxMessageLengthToScroll / 4;
    }
    // if the message is very long then limit the dialog ht so it scrolls
    // but dont do that if it is a mobile device
    if ((message.length / 2 > this.maxMessageLengthToScroll) && (!this.mobile)) {
        $('"#dialog-groups"').dialog({
            height: this.maxDialogHt,
            width: dialogWidth,
            modal: true,
            buttons: {
                Ok: function() {
                    $('this').dialog("close");
                }
            }
        });
    } else {
        $('"#dialog-groups"').dialog({
            width: dialogWidth,
            modal: true,
            buttons: {
                Ok: function() {
                    $('this').dialog("close");
                }
            }
        });
    }
    $('"#dialog-groups"').dialog('option', 'title', title);
}

function Gui_showMessageWithLink(message, title, urlString) {
    // assemble the hyperlink html, append the message to it
    // and call the show message method
    var hyperlinkString = '<a href="' + this.leafSearchURL + urlString + '" target="_blank" ' + 'title="Search using ' + this.leafSearchURL + urlString + '" >Search Link</a>';
    this.showMessage(hyperlinkString + "</br> " + message, title);
}

function Gui_showThreshold(valIn) {
    // format the number
    if (valIn.toPrecision) {
        // if browser supports toPrecision() method
        valIn = valIn.toPrecision(5);
    }
    $('this.thresholdField').val(valIn);
}
dendrogrammer.js

}/**
 * Show the given value in the gui element for displaying no of groups.
 * @method showNoOfGroups
 * @param valIn {number} the value to show
 */
function Gui_showNoOfGroups(valIn){
  $(this).groupNoField.val(valIn);
}
/**
 * Show the given values in the gui element for displaying those.
 * used for showing facts about the current loaded data set
 * @method showDataDisplay
 * @param h {number} the value to show in largestMergeHtDiv
 * @param n {number} the value to show in leafNodesDiv
 */
function Gui_showDataDisplay(h,n){
  $(this).largestMergeHtDiv.text(h);
  $(this).leafNodesDiv.text(n);
}
/**
 * Get the current value of the threshold
 * @method getThresholdVal
 * @return {number} the value of the threshold
 */
function Gui_getThresholdVal(){
  return $(this).thresholdField.val();
}
/**
 * Attach functions to Gui buttons
 * @method bindButtonFunctions
 * @param dendrogramIn {Dendrogram object} the dendrogram object with all its functions
 * @param guiIn {Gui object } the gui object passed is required in as the "this" keyword references the event rather during a click.
 */
function Gui_bindButtonFunctions(dendrogramIn,guiIn){
  //bind the click event to the button with its handler
  $(function() {
    $('#groupingButton').bind('click', function() {
      dendrogramIn.applyGroupingThreshold($("#amount").val());
    });
    $('#removeGroupsButton').bind('click', function() {
      dendrogramIn.removeGroups();
    });
    $('#truncateButton').bind('click', function() {
      dendrogramIn.drawSummary();
    });
    $('#restoreButton').bind('click', function() {
      dendrogramIn.restore();
    });
    $('#maxButton').bind('click', function() {
      guiIn.showThreshold(dendrogramIn.setThresholdBarToMax());
    });
    $('#zeroButton').bind('click', function() {
      guiIn.showThreshold(dendrogramIn.setThresholdBarToZero());
    });
    $('#lessButton').bind('click', function() {
      guiIn.showThreshold(dendrogramIn.setThresholdBarToLess(20));
    });
    $('#moreButton').bind('click', function() {
      guiIn.showThreshold(dendrogramIn.setThresholdBarToMore(20));
dendrogrammer.js

$("#minusButton").bind('click', function() {
    guiIn.showThreshold(dendrogramIn.setThresholdBarToLess(190));
});

$("#plusButton").bind('click', function() {
    guiIn.showThreshold(dendrogramIn.setThresholdBarToMore(200));
});

$("#useGroupNoButton").bind('click', function() {
    dendrogramIn.setThresholdForNoOfGroups($("#groupNo").val());
});

function Gui_bindTabEvents()
{
    //bind the click event on groups tab to handling code
    $(function() {
        $("#pulser").bind('click', function() {
            //stop it pulsing
            $("#pulser").stop(true, true);
            //reset the opacity
            $("#pulser").css('opacity', 1);
        });
    });
}

//Constructor function
/**
 * @constructor Gui
 * @param clusterTableIn {ClusterTable object} the clusterTable
 */
function Gui(clusterTableIn){
    //initialise attributes
    this.clusterTable=clusterTableIn;
    this.maxDialogHt=$(window).height()*0.75;//dialog is max ht 75% of window
    this.maxMessageLengthToScroll=2000;//beyond this and dialogs are ht limited
    //set default leafSearchURL
    this.leafSearchURL="http://www.google.co.uk/search?q=";
    //set leafSearchURL using input data
    if (this.clusterTable.getSearchLink()!=null){
        this.leafSearchURL=this.clusterTable.getSearchLink();
    }
    //detect mobile device
    // Looks for the orientation property.
    // In a mobile device such as ipad/android it is defined
    if (typeof orientation !='undefined'){
        this.mobile = true;
    }else{
        this.mobile = false;
    }
    $( "#traceDump" ).append("gui.mobile="+this.mobile+"</br>");

    //names of DOM objects which need accessed by the application
    this.groupsTab=#groupsOutput;
    this.thresholdField=#amount;
    this.groupNoField=#groupNo;
    //used in a getElementById call, so no #
    this.canvasContainer='canvas_container';
    this.largestMergeHtDiv="#largestMergeHt";
    this.leafNodesDiv="#leafNodes";

    //associate methods
    this.notifyGroupFormation = Gui_notifyGroupFormation;
    this.showMessage = Gui_showMessage;
    this.showMessageWithLink = Gui_showMessageWithLink;
dendrogrammer.js
this.showThreshold = Gui_showThreshold;
this.bindButtonFunctions = Gui_bindButtonFunctions;
this.bindTabEvents = Gui_bindTabEvents;
this.showNoOfGroups = Gui_showNoOfGroups;
this.clearGroupNotifer = Gui_clearGroupNotifer;
this.getThresholdVal = Gui_getThresholdVal;
this.showDataDisplay = Gui_showDataDisplay;

//processing to be done by Constructor
this.bindTabEvents(); //set up the group tab event

//jQuery UI elements
//jQuery tabs Widget
$(function() {
  $("#tabs").tabs();
});
//END OF jQuery tabs Widget

//jQuery buttons
//apply button styling and interaction to the Toolbar buttons.
//function is bound at global level after creation of the dendrogram

$(function() {
  $("#zeroButton").button({
    text: false,
    icons: {
      primary: "ui-icon-seek-first"
    }
  });
  $("#lessButton").button({
    text: false,
    icons: {
      primary: "ui-icon-seek-prev"
    }
  });
  $("#minusButton").button({
    text: false,
    icons: {
      primary: "ui-icon-minus"
    }
  });
  $("#plusButton").button({
    text: false,
    icons: {
      primary: "ui-icon-plus"
    }
  });
  $("#moreButton").button({
    text: false,
    icons: {
      primary: "ui-icon-seek-next"
    }
  });
  $("#maxButton").button({
    text: false,
    icons: {
      primary: "ui-icon-seek-end"
    }
  });
  $("#groupingButton").button();
  $("#removeGroupsButton").button();
  $("#truncateButton").button();
});
dendrogrammer.js

```javascript
$( "#restoreButton" ).button();
$( "#useGroupNoButton" ).button();

//END OF jQuery buttons
//####################################################################

};// End of Gui class
//####################################################################
```
<?php
// The Dendrogrammer - production version
$version="1.0.1"; //Set the version number
//created by David Allan Robb March to August 2011

// PROCESSING STARTS HERE
// Get script location

$version
//     The Dendrogrammer - production version

// Define a path variable for use in a php redirection if needed
$webhost= $_SERVER['HTTP_HOST']; //the host server on MACS this is "www.macs.hw.ac.uk"

// Set the version number
$version;

//was placed in the top level of my www dir on MACS

// then it would be "/~darl4/show_server_vars.php"

// END OF Get script location

$redir = "http://".$webhost.$this_file;

// END OF declarations for jQuery UI

<!DOCTYPE html>
<html>
<meta http-equiv="X-UA-Compatible" content="IE=9">
<meta charset="UTF-8">
<!-- declarations for jQuery UI-->
<link rel="stylesheet" href="../libraries/ui/jquery.ui.all.css">
<!-- the application style sheet-->
<link href="dendrogrammer_styles.css" type="text/css" rel="stylesheet">
<!-- declare the library files
- the jQuery cross-browser API for JavaScript
- the Raphael cross-browser JavaScript graphics library
- the jQuery UI user JavaScript user interface API
-->
<script src="../libraries/jquery-latest.min.js"></script>
<script src="../libraries/raphael.js"></script>
<!-- The Application code-->
<script type="text/javascript" src="dendrogrammer.js"></script>
<!-- declarations for jQuery UI-->
<script src="../libraries/ui/jquery.ui.core.js"></script>
<!-- for widgets and effects -->
<script src="../libraries/ui/jquery.ui.widget.js"></script>
<!-- for widgets and effects -->
<script src="../libraries/ui/jquery.ui.tabs.js"></script>
<!-- tabs structure-->
<!-- for pulsate effect on tab -->
<script src="../libraries/ui/jquery.effects.core.js"></script>
<!-- for pulsate effect on tab-->
<script src="../libraries/ui/jquery.effects.pulsate.js"></script>
<!-- for draggable effect in groups and for dialog-->
<script src="../libraries/ui/jquery.ui.draggable.js"></script>
<!-- for dialog-->
<script src="../libraries/ui/jquery.ui.dialog.js"></script>
<!-- for dialog-->
<script src="../libraries/ui/jquery.ui.button.js"></script>
<!-- for buttons-->
Dendrogrammer.php

C:\Documents and Settings\david\workspace\dendrogrammer_proto01\release 1.0.1\dendrogrammer\version1_0_1\ Last modification: 16/08/2011 15:41:36

PSPad editor 4.5.0 (2183)  www.pspad.com  17/08/2011 10:54:21 david

C:\Documents and Settings\david\workspace\dendrogrammer_proto01\release 1.0.1\dendrogrammer\version1_0_1\ Last modification: 16/08/2011 15:41:36

dendrogrammer.php Page:2/10

<head>
</head>

<body>

<div id="wrapperForCentring">
</div>

<div id="centre">
</div>

<!-- centre div Page Content goes HERE -->

<span class="theHeading" >The Dendrogrammer</span>

<div id="tabs">

<!-- Tabs Div contains the tab list followed by all the tab divs -->

<!-- Tabs List -->

<ul>
    <li><a href="#tabs-1" title="View a dendrogram">Dendrogram</a></li>
    <li><a href="#tabs-2" title="Load a new data set">Data</a></li>
    <li><a href="#tabs-3" id="pulser" title="Access groups detail">Groups</a></li>
    <li><a href="#tabs-4" >Program Variables (Tracer)</a></li>
    <li><a href="#tabs-5" >About</a></li>
    <li><a href="#tabs-6" >Help</a></li>
    <li><a href="#tabs-7" >Zoom Help</a></li>
</ul>

<!-- END OF Tabs List -->

</div>

<!-- Start of the individual tab divs -->

<div id="tabs-1">

<span class="theHeading" title="A dendrogram is a tree diagram with all the leaves at one end.">Dendrogram</span>
<br/>
Look under the "Help" tab above for the Quick-start guide.
<br/>

<span id="adjusterToolbar" title="Threshold Toolbar">
    <button id="zeroButton" >zero</button>
    <button id="lessButton" >less</button>
    <button id="minusButton" >minus</button>
    <input type="text" id="amount" size="7" title="Threshold value field" style="border:0; color:#f6931f; font-weight:bold; text-align: center"/>
    <button id="plusButton" >plus</button>
    <button id="moreButton" >more</button>
    <button id="maxButton" >max</button>
    <input type="button" id="groupingButton" title="Apply new threshold" value="Apply threshold" />
    <input type="button" id="removeGroupsButton" title="Remove groups" value="Remove" />
    <input type="button" id="truncateButton" title="Truncate below threshold" value="Truncate" />
    <input type="button" id="restoreButton" title="Restore leaves" value="Restore" />
</span>&nbsp;&nbsp;

<span id="groupsNoSetToolbar" title="Set no. of groups Toolbar">
    <input type="text" id="groupNo" size="2" title="No. of groups field" style="border:0; color:#f6931f; font-weight:bold; text-align: center"/>
    <input type="button" id="useGroupNoButton" title="Apply no. of groups" value="Apply no. of groups" />
</span>

</div>

<!-- End of tab 1 div -->

</div>

<!-- Graph div contains the dendrogram div -->

<div id="canvas_container" ></div>
</div>
</div>
</div>
</body>
</html>
<?php

// File reading script starts here

//get the directory listing
$dirlist = scandir('./data');

//if no data file was found then output a message about this
if (!isset($_GET['read'])) {
    $filename=$_GET['read'];
    $filename=$first_found_dat;
}$filename="default.dat";

//http://www.macs.hw.ac.uk/cgi-bin/ips/echo

//check the GET data to see if the user has chosen a file
if(isset($_GET['read'])) {
    $filename=$_GET['read'];
} else {
    $filename=$first_found_dat;
    $filename="default.dat";
}

//if not then load the data for the first displayed csv file and offer that
if(!$found_one_dat){
    echo 'No .dat file in the data directory'</h2>';
    echo 'Regretably there is no loadable data in the data directory. ';
    echo 'A .dat file containing the names of the node and leaf data files needs to ';
    echo 'The node data needs to be a 3-column csv file containing clustering data';
    echo ' and the feaf data should be a 2-column csv file containing leaf ';
    echo 'description data. ';
}

//create a table for displaying links for changing selected file
echo "<table >";

//loop through the directory list
foreach ($dirlist as $key => $filename) {
    //start a new row
    echo "<tr>
        // if the filename contains the ".dat" then display it
        if (strpos($filename, ".dat")){
            //store the first found csv name
            if (!$found_one_dat){
                $first_found_dat=$filename;
                }
            $found_one_dat=true;
            //start the new cell
            echo "<td>
                //display the file loading link building a url querystring like this:
                //applicationfile.php?read=somefile.csv&w=1280
                //store the first found csv name
                if (!$found_one_dat){
                    $first_found_dat=$filename;
                    }
                $found_one_dat=true;
                echo "<a href='".$redir."?read=".$filename."'>".$filename."</a><br>
            
            // close the cell
            echo "</td>
        }
    //END if (strpos($filename, ".dat"))
    }
//end foreach ($dirlist as $key => $value)

// finish the table
echo "</table>";

//if no data file was found then output a message about this
if(!$found_one_dat){
    echo '<h2>No .dat file in the data directory</h2>';
    echo 'Regretably there is no loadable data in the data directory. ';
    echo 'A .dat file containing the names of the node and leaf data files needs to ';
    echo 'The node data needs to be a 3-column csv file containing clustering data';
    echo ' and the feaf data should be a 2-column csv file containing leaf ';
    echo 'description data. ';
}

//get the directory listing
$dirlist = scandir('./data');

//if no data file was found then output a message about this
if (!isset($_GET['read'])) {
    $filename=$_GET['read'];
    $filename=$first_found_dat;
}$filename="default.dat";

//check the GET data to see if the user has chosen a file
if(isset($_GET['read'])) {
    $filename=$_GET['read'];
} else {
    $filename=$first_found_dat;
    $filename="default.dat";
}

//http://www.macs.hw.ac.uk/cgi-bin/ips/echo
The largest merge height (dissimilarity value) is
Below are details of the currently loaded data set:
There are leaf nodes.
<!-- start the data form -->
<!-- the form has no submit button. It is purely for the purpose of containing-->
<!-- data within input elements which are harvested by the JavaScript code. -->
<form name="myForm" id="dataForm"
action="no-action.php"
method="POST">
<!-- the form has no submit button. It is purely for the purpose -->
</php
//if no data was found output a messege else read data and build the form.
if (!$found_one_dat){
echo 'ch2:Hard coded example data loaded</h2>';
echo 'Some hard coded example data has been loaded for the '
echo 'time being, for demonstration purposes.';
?}
<!-- dummy data -->
<!-- dummy file name data -->
<input type="hidden" name="fileName" value="example.dat"/>
<input name="dat1c101" type="hidden" value="example.node.csv">
<input name="dat2c102" type="hidden" value="example.leaf.csv">
<!-- leaf description data -->
<input name="node1c102" type="hidden" value="1">
<input name="node1c101" type="hidden" value="2">
<input name="node2c101" type="hidden" value="3">
<input name="node2c102" type="hidden" value="4">
<input name="node3c101" type="hidden" value="5">
<input name="node3c102" type="hidden" value="6">
<input name="node4c101" type="hidden" value="7">
<input name="node4c102" type="hidden" value="8">
<input name="node5c101" type="hidden" value="9">
<input name="node5c102" type="hidden" value="10">
<input name="leaf1c102" type="hidden" value="11">
<input name="leaf1c101" type="hidden" value="12">
<input name="leaf2c101" type="hidden" value="13">
<input name="leaf2c102" type="hidden" value="14">
<input name="leaf3c101" type="hidden" value="15">
<input name="leaf3c102" type="hidden" value="16">
<input name="leaf4c102" type="hidden" value="17">
<input name="leaf4c101" type="hidden" value="18">
<input name="leaf5c101" type="hidden" value="19">
<input name="leaf5c102" type="hidden" value="20">
<!-- lead description data -->
<input name="leaf1c100" type="hidden" value=""/>
<input name="leaf2c100" type="hidden" value=""/>
<input name="leaf3c100" type="hidden" value=""/>
<input name="leaf4c100" type="hidden" value=""/>
<input name="leaf5c100" type="hidden" value=""/>
<!-- END OF the dummy data -->
</php
}else{
echo '<input type="hidden" name="fileName" value="'.$filename.'"/>';
# PHP csv file read
# based on code from PHP.net


# Ready an array to hold two filenames. Initialise with two empty strings

```
$filenames = array('', '');
```

# First the dat file read

```
$row = 1;
if (($handle = fopen("data/".$filename, "r")) !== FALSE) {
    echo "<h3>Dat file name=".$filename</h3>";
    echo "It contains the following data file names and description text:"
    while (($data = fgetcsv($handle, 1000, ",\")) !== FALSE) {
        echo "<\tr>";
        $num = count($data); // not needed but keep for now
        // make an input element like this:
        // <input name="datrow$1no" type="hidden" value="1">
        echo '<\td>'. $row . '.<\td>'; // end of that data line
        $row++;
    }
    echo "</\table>";
    fclose($handle);
}
```

# Next csv data files themselves

# read the node file

```
$filenames[$row-1] = $data[0];
```

# Now read the first csv data file. This should be
# the 3 column node dat file else errors will occur

```
$row = 1;
if (($handle = fopen("data/".$filenames[0], "r")) !== FALSE) {
    echo "<h3>Node file name=".$filenames[0]."</h3>";
    echo "Line No\tcol1\tcol2"
    while (($data = fgetcsv($handle, 1000, ",\")) !== FALSE) {
        echo "<\tr>";
        $num = count($data); // not needed but keep for now
        // make an input element like this:
        // <input name="rownno" type="hidden" value="1">
        echo '<\td>'. $row . '.<\td>'; // end of that data line
        $row++;
    }
    echo "</\table>";
    fclose($handle);
} else {
    echo "<h3>File name=".$filenames[0] . " not found</h3>";
}
```

# read the leaf file

```
# Now read the second csv data file. This should be
# the two column leaf data file else errors will occur
$row = 1;
if (($handle = fopen("data/".$filenames[1], "r")) !== FALSE) {
    echo "<h3>Leaf file name=".$filenames[1]."</h3>";
    echo "Line No\tcol1\tcol2"
    while (($data = fgetcsv($handle, 1000, ",\")) !== FALSE) {
        echo "<\tr>";
        $num = count($data); // not needed but keep for now
        // make an input element like this:
```
// <input name="row1no" type="hidden" value="1">
for ($c=0; $c < $num; $c++) {
    echo '<td>',$row['].$data[$c].','<input name="leaf'.$row.'col'.$c.'" type="hidden" value="',$data[$c],'"></td>
}  
$row++; 
} else {
    echo '"<h3>File name=$filenames[1] not found</h3>";
    
    //END if (!$found_one_dat)\n
    //************************************************************
    //     END of file reading script
    //************************************************************\n
    <?php echo $version; ?>

    <ul>The Dendrogrammer was created as one component in the analysis of multivariate data. In particular it is intended to allow already pre-processed data to be classified into groups and then either </li>
    <li>used in further analysis, or</li>
    <li>used in some form of presentation about the data.</li>
</ul>

Current Features

- Dendrogram construction
- File read
- File read now includes a manifest file which cues a clustering data file (node data) and a leaf description data file (leaf data).
2011 15:41:36

416: loads if no data files are found.</li>
417:  &lt;li&gt;Window width optimisation&lt;/li&gt;
418:  &lt;li&gt;Group formation by adjustable threshold&lt;/li&gt;
419:  &lt;li&gt;Cluster click/tap reports into dialog&lt;/li&gt;
420:  &lt;li&gt;Group formation reports with dialog, reports into separate tab, 
421:      then click the "Apply threshold" button to form groups.&lt;/li&gt;
422:  &lt;li&gt;The threshold is a draggable element on the graph&lt;/li&gt;
423:  &lt;li&gt;The threshold can be adjusted also using toolbar buttons&lt;/li&gt;
424:  &lt;li&gt;The desired number of groups can be entered and it will set 
425:      the threshold to fit&lt;/li&gt;
426:  &lt;li&gt;Leaf labels on the vertical axis. These are interactively styled in response 
427:      to events.&lt;/li&gt;
428:  &lt;li&gt;Dialogs now guaranteed to be under 75% of window ht, scrolling if needed 
429:      and using a default or wider width as appropriate on PC platform.
430:  &lt;li&gt;Scrolling dialogs are not deployed on tablet platform as these are 
431:      not suitable for tablets.  &lt;/li&gt;
432:  &lt;li&gt;The group banding (and groups) can be removed with the "Remove" button.&lt;/li&gt;
433:  &lt;li&gt;Summary dendrogram: For use in dendrograms with a very large number of 
434:      leaves. Effectively truncating the dendrogram up from 0 ht to a level at 
435:      which the structure can be discerned and then the clusters interrogated, 
436:      rather than having hundreds of leaves along the bottom.&lt;/li&gt;
437:  &lt;li&gt;The Descendant Nodes Dialog contains a search link. The URL for this is 
438:      specified in the .dat file which loads the data. The search link passes 
439:      the cluster's leaf terms to a search engine.&lt;/li&gt;
440:  &lt;/ul&gt;
441:  &lt;/ul&gt;
442:  &lt;ul&gt; Documentation
443:  &lt;li&gt;&lt;a href="../doc/index.html">JavaScript code documentation</a&gt;&lt;/li&gt;
444:  &lt;li&gt;&lt;a href="../Readme.txt">Readme file</a&gt;&lt;/li&gt;
445:  &lt;/ul&gt;
446:  &lt;/ul&gt;
447:  &lt;/ul&gt;
448:  &lt;/ul&gt;
449:  &lt;ul&gt; The technologies deployed in the application include
450:  &lt;li&gt;JavaScript - The major part of the application is written in JavaScript&lt;/li&gt;
451:  &lt;li&gt;&lt;a href="http://raphaeljs.com/">Raphael</a&gt; Javascript library  
452:      (by &lt;a href="http://dmitry.baranovskiy.com/">Dmitry Baranovskiy</a&gt;)&lt;/li&gt;
453:  &lt;li&gt; - Used to make the dendrogram graphic elements. Raphael is truly amazing.&lt;/li&gt;
454:  &lt;li&gt;(Indirectly) SVG and VML - Raphael uses SVG or VML to do the drawing.
455:  &lt;li&gt; Raphael switches seamlessly depending on the browser &lt;/li&gt;
456:  &lt;li&gt;&lt;a href="http://jqueryui.com/">jQueryUI</a> - For some of the cross browser Dynamic 
457:      HTML&lt;/li&gt;
458:  &lt;li&gt;&lt;a href="http://jqueryui.com/">jQuery</a>&lt;/li&gt; - For some of the user 
459:      interface supporting the application&lt;/li&gt;
460:  &lt;li&gt;HTML5 - Although there is nothing especially HTML5 about it really.  &lt;/li&gt;
461:  &lt;li&gt;&lt;a href="http://php.net/">PHP</a>&lt;/li&gt; - the data files are loaded using PHP.&lt;/li&gt;
462: &lt;/ul&gt;
463: &lt;/ul&gt; The Dendroammer was created
464: &lt;li&gt;by David Allan Robb&lt;/li&gt;
465: &lt;li&gt;for his MSc project at Heriot-Watt University&lt;/li&gt;
466: &lt;li&gt;in summer 2011&lt;/li&gt;
467: &lt;/ul&gt;
468: &lt;/div&gt;&lt;!-- END of tab 5 div --&gt;
469: &lt;div id="tab-6"><!-- Start of tab 6 div --&gt;
470: &lt;span class="theHeading" &gt;Quick-start Guide&lt;/span&gt;
471: &lt;/ul&gt;
472: &lt;li&gt;In a nutshell:&lt;/li&gt;
473: &lt;/ul&gt;
474: &lt;li&gt;Drag the green bar (Threshold bar) 
475:      to adjust the dissimilarity threshold for group formation.
476: Then click the "Apply threshold" button to form groups.&lt;/li&gt;
477: &lt;li&gt;"Remove" will get rid of the groups.&lt;/li&gt;
478: &lt;li&gt;Or enter the desired no. of groups in the "no. of groups field" and 
479: click the "Apply no. of groups" button.  &lt;/li&gt;
480: &lt;li&gt;To create a summary dendrogram, drag the threshold, then click 
481: "Truncate".  &lt;/li&gt;
482: &lt;li&gt;"Restore" redraws the original dendrogram.&lt;/li&gt;

PSPad editor 4.5.0 (2183) www.pspad.com 17/08/2011 10:54:21 david
Another way to adjust the threshold is to enter a number in the "Threshold value field". Click or tap a cluster (element on the graph). Then read about that cluster in the dialog that appears. That dialog also contains a search link. The URL for this is specified in the .dat file which loads the data. The search link passes the cluster's leaf terms to a search engine. It should open in a browser pop-up window or browser tab. Adjust the green threshold bar and click the button to form groups of leaves (data points). See more detail about the groups you form under the "Groups" tab. The threshold can be adjusted by click-and-drag on the green bar, or using the threshold toolbar buttons, or entering a number in the "Threshold value field". If you are using a tablet computer then drag on the green bar with your finger. If you find you can't drag the bar accurately then use the toolbar buttons to move the threshold. You might find that after moving the threshold bar away from the max with a button then you can get a better feel for how it drags using your finger. If using the toolbar fine adjust buttons (plus and minus), note that the amount of movement with minus is set slightly less that with plus. This allows even more fine adjustment if you alternate from plus to minus around the point you are looking for. To specify a given number of groups, enter the desired number in the field next to the "Apply no.of groups" button and click the button. The threshold will be chosen automatically for you. To view a summary dendrogram (or truncated dendrogram), drag the threshold to some point below which you wish to discard the leaf detail. Then click the "Truncate" button. The resulting summary dendrogram will have everything below the threshold removed and be reshaped to fit the previous space. The origin is at the dissimilarity ht. you chose with the threshold bar. Tapping/clicking a cluster will still report all the leaf detail as before, despite the leaves being invisible. Groups can be formed in the same manner as before. The "Restore" button will restore the original full dendrogram and the leaves as it was when you first viewed the current data. Loading new data or the same data again will have the same effect, as will reloading the page with your browser.

Use Control and + or - to zoom in and out ( or pinch on a tablet ). Look in your browser's View menu for more zoom controls.

What equipment to use: The dendrogrammer was designed to be cross platform (works on PCs with mouse and tablets with tap)

What browser to use: The dendrogrammer was designed to be cross browser. It is written in JavaScript (with a little bit of PHP to load the data). So it works in pretty much any browser that has Javascript enabled. However it is my experience that if one considers the latest browsers, the JavaScript interpreters in Firefox (Gecko with Tracemoney), Safari (Webkit with Squirrelfish), and Chrome (Webkit with V8) are significantly faster (e.g. by a factor of 2 and more) than the "JScript" engine in Internet Explorer 8 and below (However, IE9 32-bit is good, but IE9 64-bit is reputed to be slow). So if one is seeking faster smoother performance with this particular application then one might wish to use Firefox, Chrome, or Safari (as at July 2011).
Data tab - Load different data or find out about the current data used to make the dendrogram.

A table of links is shown representing all the available data sets.

Clicking a link loads that data set and makes a dendrogram from it.

One table shows the contents of the .dat (or manifest) file. This contains meta data about the data set: the names of the two data files and a description of the data set.

The next table shows the contents of the clustering data file (or node file). This should be a csv file generated by MATLAB. It should have 3 columns. The first two columns are cluster or leaf designation numbers. The third column contains the merge height or similarity measure for that cluster.

The last table shows the contents of the leaf description data file (or leaf file). This is a csv file. The first row describes leaf number one. The second row describes leaf number two and so on. It should contain two columns. Column one should hold the short leaf descriptions or labels. Column 2 should contain a longer description for each leaf (this can be a repeat of the short label).

Loading a new data set or the same data set again will result in a page refresh and will clear any displayed groups or summary dendrogram which you may be displayed on the Dendrogram tab.

Adding further data sets can only be done by someone with write-access to the web space where the application resides (See 'Documentation' in the About tab.)

Groups tab - View the detail of the groups formed using the threshold adjuster and toolbar. The detail can be copied and pasted into other software for further analysis or presentation.

About tab - Read about the application.

Help tab - This page you are reading now is the Help.

Once you have zoomed, refreshing the page will adjust the dendrogram to fit the width you have chosen.

On a PC or laptop, once you have zoomed, refreshing the page will adjust the dendrogram to fit the width you have chosen.

On most browsers "<strong><em>Control</em></strong>" and "<strong><em>Command</em></strong>" will zoom in.

"<strong><em>Control</em></strong> + <em></em> will zoom out.

"<strong><em>Command</em></strong>" and "<strong><em>Command</em></strong> - <em></em> will zoom out.

"<strong><em>Command</em></strong>" works too on most browsers.

Look in your browser's View menu for more zoom commands. E.g. Some browsers offer a separate zoom for text only. If using that, bear in mind that your browser sees the entire dendrogram including leaf labels as graphics.

Once you have zoomed, refreshing the page will adjust the dendrogram to fit the width you have chosen.

On a tablet device

Use the open or close pinch gesture to zoom.

Once you have zoomed in, as you would expect, refreshing the page does not affect the dendrogram size. It remains zoomed until you zoom back.
Summary Dendrogram feature

By setting the threshold bar and clicking "Truncate" you can reduce the amount of leaf detail displayed and reduce the height of the dendrogram.

Clicking a cluster on a summary dendrogram still reveals all of its descendant leaf detail in the dialogs. Group formation works as normal.

The "Restore" button restores the normal dendrogram.
1: /* CSS Document */
2: /*Interactive Dendrogrammer production version 1.0.0*/
3: /*David Allan Robb March-August 2011 */
4: /* Based on Cascading Style Sheet by David A Robb */
5: /* For TT280 ECA December 2010 */
6: 
7: // Containers
8: 
9: body {
10: margin-left: 0px;
11: padding: 0px;
12: border: 0px;
13: margin-top: 0px;
14: margin-right: 0px;
15: text-align: center;
16: color: rgb(153,0,51);
17: font-size: 11pt;
18: font-family: Verdana, Arial, Helvetica, sans-serif
19: }
20: 
21: /* This id style is for the purpose of centering all of the content horizontally*/
22: /* I have decided to remove this centering from the app to allow the edge of the page */
23: /* to be featureless. This helps in zooming the page.*/
24: #wrapperForCentring {
25: /*so the div to have no effect asside from text alignment and colour
26: */
27: background-color:rgb(255,255,255);
28: text-align: left;
29: }
30: 
31: /*main div for content */
32: #centre {
33: position:static;
34: margin: 5px;
35: }
36: 
37: /*data div sits inside content and holds the cluster div and program data div */
38: #data {
39: position:static;
40: height: 6em;/* allows its two contained divs to float left and right of eachother*/
41: /* while keeping the following div below*/
42: 
43: /*This is purely a container div so have nothing outside/inside the actual size*/
44: margin: 0px;
45: padding: 0px;
46: border: 0px;
47: }
48: 
49: .block {
50: /*This is purely a container div class
51: //so set defaults for these to have nothing outside/inside the actual size*/
52: position:static;
53: width: 49%;/*These take part in forming a 2-column div layout*/
54: margin: 0px;
55: padding: 0px;
56: border: 0px;
57: text-align: center;
58: font-size: 90%;
59: }
60: 
61: 
62: /*clusterData div for table. sits inside data div */
63: #clusterData {
64: position:static;
65: float: left;/* makes it in the left hand column of the containing div*/
66: margin: 0px;
67: padding: 0px;
68: border: 0px;
69: }
/*programData div for table. sits inside data div */
#programData {
  position: static;
  float: right; /* makes it in the right hand column of the containing div*/
  margin: 0px;
  padding: 0px;
  border: 0px;
}

/*output div for output from cluster double events and such like */
.output {
  border: 1px solid black;
  border-color: rgb(153,0,51);
}

/*output div for output from cluster double-click events */
.outputSummary {
  border: 1px solid black;
  border-color: rgb(153,0,51);
}

/*graph div for table. sits inside data div */
#graph {
  position: static;
  margin: 0px;
  padding: 0px;
  border: 0px;
}

/* specific container for diagrammer */
#canvas_container {
  position: static;
  margin: 0px;
  padding: 0px;
  border: 0px solid #aaa;
}

/* style the input adjuster items */
.adjusterInputs {
  float: left;
}

#adjusterToolbar {
  padding: 10px 4px;
  font-size: 75%;
}

#groupsNoSetToolbar {
  padding: 10px 4px;
  font-size: 75%;
}

/* jQuery UI specific styling */
.ui-progressbar-value { background-image: url(images/pbar-ani.gif); }

/* might not be using this now*/

/* Styling for draggable objects */
#draggable { cursor: move; }

/* Additional style to allow tabs to notify a change */
.highlight {
color: red;

/* Text, Headings, Lists */

/* Body text */

p {
  margin-left: 5px;
  background-color: inherit;
}

/* Headings */

.theHeading {
  color: rgb(80, 80, 80);
  font-weight: bold; font-size: 14pt;
}

h1 {
  color: rgb(80, 80, 80);
  font-weight: bold; font-size: 18pt;
}

h2 {
  color: rgb(80, 80, 80);
  font-weight: bold; font-size: 14pt;
}

h3 {
  color: rgb(80, 80, 80);
  font-weight: bold; font-size: 12pt;
}

/* Bulleted list style */

ul {
  list-style-type: square;
  padding: 10px;
  margin: 5px;
}

/* Classes for text */

.dimension {
  font-weight: bold; font-size: 120%;
}

/* Tables */

table {
  border: 1px #ff9900;
  border-spacing: 0;
  font-size: 75%;
}

td {
  padding: 1px 1px;
  border: 1px solid #ff9900;
  margin-left: 5px;
}

th {
  padding: 4px 6px;
  border: 1px solid black;
  margin-left: 5px;
}

/* Classes for tables */

table.invisible {
  border: 0px;
  border-spacing: 0;
211: }
212:
213: td.invisible{
214:   padding: 2px;
215:   border: 0px;
216: }
217:
Appendix 7
Testing

- Records of testing including records of testing in different browsers.
- Example of how large simulation data was constructed in spreadsheets.
Testing the ClusterTree from the 5-row basic example data

The data

<table>
<thead>
<tr>
<th>row</th>
<th>First Data point/cluster in pair</th>
<th>Second Data point/cluster in pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 6 0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4 7 0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2 5 0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 8 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9 10 2</td>
<td></td>
</tr>
</tbody>
</table>

The tests

<table>
<thead>
<tr>
<th>Test command</th>
<th>Reason</th>
<th>Expected output</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>All are myTestvalueGlobal=</td>
<td>Checking root ID</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>theClusterTree.getRoot().idNo;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.idNo;</td>
<td>Checking children of root Child A</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.idNo;</td>
<td>Child B</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.idNo;</td>
<td>Checking children of child A: A</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childB.idNo;</td>
<td>B</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childA.idNo;</td>
<td>Checking children of child B A</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.idNo;</td>
<td>B</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.childA.idNo;</td>
<td>Checking other generations First a known leaf</td>
<td>Error Leaf</td>
<td>Error</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.childB.idNo;</td>
<td>As above</td>
<td>Error Leaf</td>
<td>Error</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childA.idNo;</td>
<td>Now a known cluster (8) A</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.idNo;</td>
<td>B</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childA.idNo;</td>
<td>And finally 7 A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childB.idNo;</td>
<td>B</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Conclusion: The tree is being built correctly.
Now testing leaf order property

<table>
<thead>
<tr>
<th>Test command</th>
<th>Reason</th>
<th>Expected output</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>All are myTestvalueGlobal=</td>
<td>To get leaf node N and show its LeafSproutCount property</td>
<td></td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.leafSproutNo</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childB.leafSproutNo</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childA.leafSproutNo</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childA.leafSproutNo</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childA.leafSproutNo</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childB.leafSproutNo;</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

**Conclusion:**
The tree is correctly recording the leaf sprout order in each leaf node’s leafSproutNo property.
Testing: ClusterTree method setClusterY()

Now testing yPos property produced by this new method

First test

ClusterTree method setClusterY(this.root) called in the constructor of ClusterTree

<table>
<thead>
<tr>
<th>Test command</th>
<th>Reason</th>
<th>Expected output</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>All are myTestvalueGlobal=</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().yPos;</td>
<td>Checking root yPos</td>
<td>2.6875</td>
<td>0.75</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.yPos;</td>
<td>9</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.yPos;</td>
<td>10</td>
<td>3.875</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.yPos;</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childB.yPos;</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childA.yPos;</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.yPos;</td>
<td>8</td>
<td>4.75</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childA.yPos</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.yPos</td>
<td>7</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.chil</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.chil</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Conclusion: There is a problem
2nd Test
Running the test again with call to set cluster y commented out. Checking yPos of leaves
Clusters should be null.

<table>
<thead>
<tr>
<th>Test command</th>
<th>Reason</th>
<th>Expected output</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>All are myTestvalueGlobal=</td>
<td>Checking root yPos</td>
<td>null</td>
<td>.</td>
</tr>
<tr>
<td>theClusterTree.getRoot().yPos;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.yPos;</td>
<td>9</td>
<td>null</td>
<td>.</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.yPos;</td>
<td>10</td>
<td>null</td>
<td>.</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.yPos</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childB.yPos</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childA.yPos</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.yPos</td>
<td>8</td>
<td>null</td>
<td>.</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childA.yPos</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.yPos</td>
<td>7</td>
<td>null</td>
<td>.</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childA.yPos</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childB.yPos</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Conclusion: This is ok
Bug trace detail:

I looked at the code. I found I had an else in where I should not have

```java
if (node.childA.yPos==null){
    this.setClusterY(node.childA);
} else if(node.childB.yPos==null){
    this.setClusterY(node.childB);
}
```

Fixed it to be like this:

```java
//if the yPos if childA is null then set its yPos
//if the yPos of childB is null then set its ypos
if (node.childA.yPos==null){
    this.setClusterY(node.childA);
}
if (node.childB.yPos==null){
    this.setClusterY(node.childB);
}
//finally set the yPos based on yPoses of children
```
Testing: ClusterTree method setClusterY After Debug

Now testing yPos property produced by the new method

Third test

ClusterTree method setClusterY(this.root)

<table>
<thead>
<tr>
<th>Test command</th>
<th>Reason</th>
<th>Expected output</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>All are myTestvalueGlobal=</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().yPos;</td>
<td>Checking root yPos</td>
<td>2.6875</td>
<td>2.6875</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.yPos;</td>
<td>9</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.yPos;</td>
<td>10</td>
<td>3.875</td>
<td>3.875</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childA.yPos</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childA.childB.yPos</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childA.yPos</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.yPos</td>
<td>8</td>
<td>4.75</td>
<td>4.75</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childA.yPos</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.yPos</td>
<td>7</td>
<td>5.5</td>
<td>5.5</td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childA.yPos</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>theClusterTree.getRoot().childB.childB.childB.childB.yPos;</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Conclusion:
Code works fine now
Testing the Canvas class

The new style canvas was coming out too small
Traced a bug to an incorrectly named property.
The problem is in the ht, X. I
myTestValueGlobal=theCanvas.largestHt; produces 10
It should be 2, the ht of the root node
So it looks like ClusterTree has the value of node.xHt wrong. Tracing now.
The operative line in the ClusterNode constructor is:
```java
this.xHt=cTableIn.getHtByIdNo(idNoIn);
```

myTestValueGlobal=theClusterTable.getHtByIdNo(7);
reveals that all the leaf node hts are 0, correct. The cluster node ht are wrong
checking this ClusterTable method involved in returning ht
```java
this.isLeafByIdNo(idNoIn)
```

myTestValueGlobal=theClusterTable.isLeafByIdNo(1) TRUE
myTestValueGlobal=theClusterTable.isLeafByIdNo(7); FALSE
myTestValueGlobal=theClusterTable.isLeafByIdNo(11); FALSE

So that works ok.

Found it
```java
return this.getRowByIndex(i).itemB;
```
should be
```java
return this.getRowByIndex(i).mergeHt;
```

myTestValueGlobal=theClusterTable.getHtByIdNo(11);
now gives 2 which is correct.
myTestValueGlobal=theCanvas.xWidth; =2 Wrong
myTestValueGlobal=theCanvas.yHt; =138 Fine *
myTestValueGlobal=theCanvas.htScaleFactor; = 150 Fine
myTestValueGlobal=theCanvas.largestHt; = 2 Fine
myTestValueGlobal=theCanvas.margin; = 45 Fine

myTestValueGlobal=theCanvas.xWidth is now 390 which is fine
(not sure what was wrong there. Maybe I sampled the wrong value at “Wrong” above)

Compared the old paper set up values with the new canvas call setup and found a bug
in Canvas Constructor
\( \text{this.yHt} = ((\text{this.noOfLeafs} + 2) \times \text{this.noOfLeafs}) + (\text{this.margin} \times 2) \); should be

\( \text{this.yHt} = ((\text{this.noOfLeafs} + 2) \times \text{this.leafSpace}) + (\text{this.margin} \times 2) \);

* in the test further up 138 was too small for yHt

\( \text{myTestValueGlobal} = \text{theCanvas.yHt} \); now is 330 good

Biting the bullet now and commenting out the call to drawdendrogram(). Using IE error reporter which reports line nos. only Xaxis was appearing Found bug. Paper was being mis referecneced in GraphAxes Added proper local reference to paperIn and it works fine.

It draws the axes on the new Canvas.paper now 😊

Got it drawing the leaves now
Verifying that the groupList is being formed correctly contains all of the leaves grouped as expected. Checking that no leaves are being missed out.

Method: Compare the group list output from its reporting div with the cluster descendants reported from a cluster click.

Step 1)
Raw output pasted below copied from the output div. I have split up the groups by adding blank lines

Contents of Group List : Threshold : 3.02, Pointer : -1, The list is not empty. The Group List has size= 11,

The first group = Leaf ID=11, Leaf ID=39, Leaf ID=15, Leaf ID=41, Leaf ID=18, Leaf ID=76, Leaf ID=43, Leaf ID=48, Leaf ID=71, Leaf ID=26, Leaf ID=73, Leaf ID=48, Leaf ID=71, Leaf ID=26, Leaf ID=73, Leaf ID=82, Leaf ID=52, Leaf ID=30, Leaf ID=91, Leaf ID=27, Leaf ID=92!

Group 2 = Leaf ID=12, Leaf ID=10, Leaf ID=21, Leaf ID=51, Leaf ID=56, Leaf ID=25, Leaf ID=4, Leaf ID=22, Leaf ID=35!

Group 3 = Leaf ID=2, Leaf ID=16, Leaf ID=40, Leaf ID=53, Leaf ID=31, Leaf ID=58, Leaf ID=83, Leaf ID=37, Leaf ID=67, Leaf ID=74!

Group 4 = You clicked a leaf. ID= 20!

Group 5 = Leaf ID=17, Leaf ID=24, Leaf ID=47, Leaf ID=13, Leaf ID=28, Leaf ID=63, Leaf ID=7, Leaf ID=42, Leaf ID=81, Leaf ID=44, Leaf ID=55, Leaf ID=78, Leaf ID=79, Leaf ID=85, Leaf ID=86, Leaf ID=54, Leaf ID=80, Leaf ID=89, Leaf ID=93, Leaf ID=84, Leaf ID=99, Leaf ID=69, Leaf ID=98, Leaf ID=70, Leaf ID=7, Leaf ID=33, Leaf ID=90, Leaf ID=45, Leaf ID=87, Leaf ID=97, Leaf ID=61, Leaf ID=94, Leaf ID=72, Leaf ID=88, Leaf ID=8, Leaf ID=46, Leaf ID=64, Leaf ID=23, Leaf ID=100, Leaf ID=36, Leaf ID=95, Leaf ID=65, Leaf ID=66, Leaf ID=62, Leaf ID=75!

Group 6 = You clicked a leaf. ID= 3!

Group 7 = Leaf ID=1, Leaf ID=29!

Group 8 = Leaf ID=19, Leaf ID=50, Leaf ID=34, Leaf ID=38, Leaf ID=96!

Group 9 = Leaf ID=68, Leaf ID=9, Leaf ID=60!

Group 10 = Leaf ID=49, Leaf ID=57!

Group 11 = Leaf ID=6, Leaf ID=14, Leaf ID=32, Leaf ID=5, Leaf ID=59!

See table on next page for results analysis
Step 2) Do the comparison now. Expected output is from looking at the dendrogram and then clicking the clusters and examining the cluster report. Actual output is taken from the pasted group list above.

**Group lists generated by the Dendrogram_displayGroupList method**

<table>
<thead>
<tr>
<th>Test Data</th>
<th>Reason for test</th>
<th>Expected output (for group)</th>
<th>Actual output (for group)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold set to 3.02 and button clicked</td>
<td>To check that the method is correctly collating the group list</td>
<td>11 groups</td>
<td>Contents of Group List : Threshold : 3.02, Pointer : -1, The list is not empty. The Group List has size= 11,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 1: Around 17 leaves</td>
<td>As listed above 17 leaves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 2: Around 9 leaves</td>
<td>9 leaves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 3: Around 10 leaves</td>
<td>10 leaves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 4: a singleton leaf</td>
<td>You clicked a leaf. ID= 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 5: a large group 40 maybe? Top leaf should be #17 and bottom should be #75</td>
<td>45 leaves. Top is 17 and bottom is 75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 6: Singleton leaf #3</td>
<td>Single leaf #3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 7: 1 and 29</td>
<td>Pair = 1 and 29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 8: 5 leaves</td>
<td>5 leaves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 9: 3 leaves 68, 9, 60</td>
<td>As expected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 10: 2 leaves 49, and 57</td>
<td>As expected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group 11 : 5 leaves , top=6, bottom =59</td>
<td>As expected</td>
</tr>
<tr>
<td>There should be 100 leaves all tolled</td>
<td>100 leaves in total</td>
<td>17+9+10+1+45+1+2+5+3+2+5=100</td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion:**
The GroupList is being formed correctly. Nothing is being omitted.
Testing group display in iPad

Method:
1) Set a group threshold.
2) Tapped the form groups button.
3) Cleared the dialog (screen shot A).
4) Tapped the flashing tab.
5) Viewed the groups (screen shot B).
6) Copied the groups data by tapping and holding at the end of the last group list until the div selects in blue and a “copy” icon appears. (Screen shot C)
7) Tapped the copy icon
8) Switched to spreadsheet software
9) Opened a new document
10) Pasted from the clipboard
11) Viewed the resulting spreadsheet (screenshot D)

Screenshot A
Screenshot B

Prototype 2.0.0 dendrogrammer

Group 1:
Leaf-7-sd, Leaf-43-sd, Leaf-25-sd, Leaf-41-sd, Leaf-11-sd, Leaf-77-sd, Leaf-96-sd

Group 2:
Leaf-64-sd, Leaf-54-sd, Leaf-90-sd, Leaf-63-sd, Leaf-67-sd, Leaf-45-sd

Group 3:
Leaf-60-sd, Leaf-28-sd, Leaf-61-sd, Leaf-19-sd, Leaf-68-sd

Group 4:

Group 5:
Leaf-46-sd, Leaf-94-sd

Group 6:

Group 7:

Group 8:

Group 9:

Group 10:

Group 11:

Screenshot C

Prototype 2.0.0 dendrogrammer

Group 2:
Leaf-64-sd, Leaf-54-sd, Leaf-90-sd, Leaf-63-sd, Leaf-67-sd, Leaf-45-sd

Group 3:
Leaf-60-sd, Leaf-28-sd, Leaf-61-sd, Leaf-19-sd, Leaf-68-sd

Group 4:

Group 5:
Leaf-46-sd, Leaf-94-sd

Group 6:

Group 7:

Group 8:

Group 9:

Group 10:

Group 11:
### Conclusion:

The group lists as output can be copied and pasted successfully on iPad

Additionally about PC:
The same data displayed on a PC could be selected using the mouse, copied, pasted into a text file, saved as a CSV and opened in a spreadsheet.
**Aim:** to test that all leafs are displayed and that they all get allocated in the group analysis

The reason for this test is that I noticed that when I set the threshold to minimum and allocated groups it stated 99 groups were created. The data is supposed to be 100 data points = 100 leafs = should give 100 groups.

**Test 1**

<table>
<thead>
<tr>
<th>Test data</th>
<th>Reason for test</th>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 leaf csv, set threshold to zero and apply threshold to form groups</td>
<td>Ensure groups properly formed and listed for threshold Ht zero</td>
<td>Appropriate no of groups. If any leaves merge as clusters at Ht zero then these should form one group and the group should show as a two-leaf group in the group list. This data has three clusters at Ht zero So expect I expect 3 groups. 1: leaves 1, 6 and 4 2: leaves 2 and 5 3: leaf 3</td>
<td>With threshold at zero Groups formed dialog: “3 groups were formed using the threshold set at 0.0000. See the 'Groups' tab for details.” In Groups tab (tab text stated “3 Groups”) Details text: “Threshold: 0.0000 Group-1: Leaf-5-sd Group-2: Leaf-3-sd Group-3: Leaf-4-sd, Leaf-6-sd” Note leaves 2 and 1 are not present in the group list</td>
</tr>
</tbody>
</table>

Note:

By clicking on the clusters I got the following Node Descendants dialog shown

Bottom cluster: Descendants of Cluster ID=8: Leaf-4-sd, Leaf-1-sd, Leaf-6-sd

Middle cluster: Descendants of Leaf-3-sd: Leaf-3-sd

Top cluster: Descendants of Cluster ID=9: Leaf-2-sd, Leaf-5-sd
### Test 2:

<table>
<thead>
<tr>
<th>Test data</th>
<th>Reason for test</th>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraser100D</td>
<td>Ensure groups properly formed and listed for threshold Ht zero</td>
<td>99 Groups as there is one leaf pair cluster at Ht zero (Leaves 14 and 30) So expect to see 99 groups all with a single leaf except for just that one pair of leaves in one group together.</td>
<td>Groups formed Dialog: “99 groups were formed using the threshold set at 0.0000. See the 'Groups' tab for details.” In Groups tab (tab text stated “99 Groups”) Details text: “Threshold: 0.0000 Group-1: Leaf-7-sd Group-2: Leaf-43-sd Group-3: …. Group-12: Leaf-63-sd Group-13: Leaf-67-sd Group-14: Leaf-44-sd Group-15: Leaf-45-sd Group-16: Leaf-30-sd Group-17: Leaf-35-sd Group-18: Leaf-84-sd …. Group-65-sd Group-98: Leaf-10-sd Group-99: Leaf-71-sd” NOTE: Leaf 14 is not present in the group list</td>
</tr>
</tbody>
</table>

Note: on the dendrogram I located the cluster of Leaf 14 and 30 designated correctly as cluster 101 i.e. the first cluster. Clicking on that cluster yielded the following “Node Descendants” dialog detail: “Descendants of Cluster ID=101: Leaf-14-sd, Leaf-30-sd”. So the cluster is being rendered correctly on the graph and is represented correctly in the data structure.

**Conclusion:**
1) The dendrogram is being drawn correctly.
2) The Cluster methods, getDescendants, appendDescendants, and displayDescendants are working fine.
3) There is a bug in the algorithm which forms the groups.
Bug trace

By loading the two sets of data which have clusters consisting of 2 leaves merging at ht zero, I have established that it is the leaf coming first in data order which is omitted from the group list when a group above ht zero or at ht zero is formed.

In default set
Leaves 1 and 6 share a cluster at ht 0
Leaf 1 is missing from the group which should contain 1 and 6

In fraser100Dset
Leaves 14 (created) and 30 (technologies) share a cluster at Ht 0
Leaf 14 (created is missing from the group with leaf 30(technologies)

However I have checked by making the node click event report to the tracer tab that the cluster node is formed and drawn. Because in fraser100Dset I can click on the cluster and it reports that it has descendants created, and technologies. Also both leaves are represented on the dendrograms visualised for both these data sets.

Now I have established that the bug is NOT related to Ht 0
It is affecting all groups. Each group is losing one member. => the bug is in the new Group code or the addToEnd method of the group List

The new Group code is very simple and simply assigns the leaflist and the ancestor node
The addTo end code is simple enough too

The one remaining method is the displayGroupList method of Dendrogram
Which uses the displayonegroup method.

Bug traced to the loop control in the displayonegroup method.

Test 3: Carried out on 13th July after bug fix
All leaves feature in the group lists when using both sets of data.

The particular highlighted clusters of two leafs are also properly listed.
Aim: To test the application output when no data files are present in the data directory

<table>
<thead>
<tr>
<th>Test data</th>
<th>Reason for test</th>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Removed all the data files from the data directory and ran the application</td>
<td>To ensure suitable output when no data files are present in the data directory</td>
<td>An example dendrogram based on hard coded data</td>
<td>Small example dendrogram was rendered and could be interacted with.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An appropriate message in the data tab about the lack of data and information about how this might be rectified.</td>
<td>Appropriate information was displayed in the data tab. See screenshots below.</td>
</tr>
</tbody>
</table>

Screen shots o output from hard coded data:

**Dendrogram**

![Dendrogram Image]

**Data Tab**

![Data Tab Image]
Aim: To test that group formation on the summary dendrogram. Are the groups properly formed and are any leaves missed out

Test A: Leaf Omission

Method
Step 1) truncated the graph, then set a threshold and applied the groups.
2) Copied the group list out of the groups tab and pasted into a spreadsheet (Actually I did this on the iPad as the spreadsheet software on the iPad accepts pasting text with commas as a csv and splits the paste over cells)
3) Using the spreadsheet I was able to count the leaf instances and add them up

<table>
<thead>
<tr>
<th>Test data</th>
<th>Reason for test</th>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraser100D</td>
<td>Check for leaf omission.</td>
<td>Groups lists</td>
<td>The groups totalled to 100 leafs</td>
</tr>
<tr>
<td>Has 100 leaves</td>
<td></td>
<td>Comprising 100 leafs</td>
<td>(in 7 groups)</td>
</tr>
<tr>
<td>Threshold for</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups set to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0881</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Max was 9.7272)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conclusion
No leaves are omitted when making groups on the summary dendrogram

Test B: Correct no. of groups for a given threshold

Method
Noting the threshold used in test A I set the same threshold on a refreshed page with the normal graph. Then I applied groups and checked that the same number of groups were formed.

<table>
<thead>
<tr>
<th>Test data</th>
<th>Reason for test</th>
<th>Expected output</th>
<th>Actual output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraser100D</td>
<td>Check for same no of groups as in test A</td>
<td>7 groups</td>
<td>7 groups</td>
</tr>
<tr>
<td>Has 100 leaves</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold for</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groups set to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.0872</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conclusion
The same groups are formed irrespective of whether or not in summary or normal mode (assuming same threshold is used)
Aim: To test how the application renders and behaves in different browser versions and on different platforms.

Method
The application was loaded and tested in various browser versions.
In the case of IE7 and 8, I used IE9 developer tools with IE9 running in debug mode. This allowed emulation of IE7. I used IE8 on my desktop PC and IE 9 on my laptop. All the rest were on my desktop unless otherwise stated.

<table>
<thead>
<tr>
<th>Page user interface</th>
<th>Raphael chart and drag-able threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Appearance</strong></td>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>IE9.0 – laptop.</td>
<td>Correct</td>
</tr>
<tr>
<td>IE8.0 – laptop (also Desktop See “Function”)</td>
<td>Identical to IE9</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>IE7.0 laptop (Emulated in IE9 developer tools)</td>
<td>Good. Slight height differences in the toolbars. See screenshot below</td>
</tr>
<tr>
<td>Firefox 5.0</td>
<td>Correct</td>
</tr>
<tr>
<td>Firefox 3.6 - laptop</td>
<td>Correct</td>
</tr>
<tr>
<td>Safari on iPad 1 (current version as on 2/8/11)</td>
<td>Correct</td>
</tr>
<tr>
<td>Opera v11.5</td>
<td>Correct</td>
</tr>
<tr>
<td>Chrome v12</td>
<td>Correct</td>
</tr>
</tbody>
</table>

Continued over page
Chrome v5.0 - laptop

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Correct</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safari v5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No noticeable delay on rendering chart

Conclusion:

Appearance
My test show it renders correctly in Firefox 3.6+, Safari 5.0+, Chrome 5.0+, Opera 11.5+ and Internet Explorer 7+

The jQuery toolbar buttons render acceptably in IE 7

The Raphael documentation claims that the graphics should render properly in Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+.

jQuery UI claims to render properly in “IE 6.0+, Firefox 3+, Safari 3.1+, Opera 9.6+ and Google Chrome”. There is the slight discrepancy here compared with my results for IE7

Function
My tests show that it functions well without noticeable processing delays in all the non Internet Explorer tested browsers and indeed also in IE9.

It seems slow in IE 7 and 8. It is particularly slow on less modern equipment.

In Addition:
iPad
It is notable that it functions well in Safari on iPad

Android Slate (Asus Transformer running Android Honeycomb v3.1)
Everything renders correctly. With the 10o leaf data there is a noticeable delay in responding to button taps. There is a noticeable lag in threshold drag response.
No of Leafs 300  
starting mergeHt 0.1  
largest mergeht 10  
merge dif 9.9  
merg dif/leafs 0.033

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Merge Ht</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0.133</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0.166</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.199</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>0.232</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.265</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.298</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0.331</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>0.364</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>0.397</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>0.43</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>0.463</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>0.496</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>0.529</td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td>0.562</td>
</tr>
<tr>
<td>14</td>
<td>17</td>
<td>0.595</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>0.628</td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>0.661</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>0.694</td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>0.727</td>
</tr>
</tbody>
</table>
No of Leafs 300
starting mergeHt 0.1
largest mergeht 10
merge dif \(=B3-B2\)
merge dif/leafs \(=B4/(B1*1)\)

<table>
<thead>
<tr>
<th>Item A</th>
<th>Item B</th>
<th>Merge Ht</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>=B2</td>
</tr>
<tr>
<td>=D8</td>
<td>=B8+1</td>
<td>=C8+$B$5</td>
</tr>
<tr>
<td>=A9+1</td>
<td>=B9+1</td>
<td>=C9+$B$5</td>
</tr>
<tr>
<td>=A10+1</td>
<td>=B10+1</td>
<td>=C10+$B$5</td>
</tr>
<tr>
<td>=A11+1</td>
<td>=B11+1</td>
<td>=C11+$B$5</td>
</tr>
<tr>
<td>=A12+1</td>
<td>=B12+1</td>
<td>=C12+$B$5</td>
</tr>
<tr>
<td>=A13+1</td>
<td>=B13+1</td>
<td>=C13+$B$5</td>
</tr>
<tr>
<td>=A14+1</td>
<td>=B14+1</td>
<td>=C14+$B$5</td>
</tr>
<tr>
<td>=A15+1</td>
<td>=B15+1</td>
<td>=C15+$B$5</td>
</tr>
<tr>
<td>=A16+1</td>
<td>=B16+1</td>
<td>=C16+$B$5</td>
</tr>
<tr>
<td>=A17+1</td>
<td>=B17+1</td>
<td>=C17+$B$5</td>
</tr>
<tr>
<td>=A18+1</td>
<td>=B18+1</td>
<td>=C18+$B$5</td>
</tr>
<tr>
<td>=A19+1</td>
<td>=B19+1</td>
<td>=C19+$B$5</td>
</tr>
<tr>
<td>=A20+1</td>
<td>=B20+1</td>
<td>=C20+$B$5</td>
</tr>
<tr>
<td>=A21+1</td>
<td>=B21+1</td>
<td>=C21+$B$5</td>
</tr>
<tr>
<td>=A22+1</td>
<td>=B22+1</td>
<td>=C22+$B$5</td>
</tr>
<tr>
<td>=A23+1</td>
<td>=B23+1</td>
<td>=C23+$B$5</td>
</tr>
<tr>
<td>=A24+1</td>
<td>=B24+1</td>
<td>=C24+$B$5</td>
</tr>
<tr>
<td>=A25+1</td>
<td>=B25+1</td>
<td>=C25+$B$5</td>
</tr>
<tr>
<td>=A26+1</td>
<td>=B26+1</td>
<td>=C26+$B$5</td>
</tr>
</tbody>
</table>
Appendix 8

Documentation

- Technical Guide. (For user guide see help tab in the application)
- Readme file
- The Code Documentation web pages can be viewed at
  http://www.macs.hw.ac.uk/~dar14/project/dendrogrammer/doc/index.html
  They are also available offline on the project CD. Open the file, index.html,
  located inside the docs folder.
The Dendrogrammer Technical Guide

Contents
- System Requirements
- Browser compatibility
- Compatibility list
- File organisation
- Data file format

System Requirements
The application files need to be placed in PHP enabled web space. The application is run from a web browser requesting the PHP file (dendrogrammer.php).

Browser compatibility
In theory the application can run on any modern Java enabled browser (see compatibility list below). In practice with larger data sets, The JScript engine in Internet Explorer 7 and 8 will struggle to cope with the processing demand unless run on high end PCs. All the other browsers should perform well even on modest equipment including tablet computers.

Compatibility list
Firefox 3.6+, Safari 5.0+ (including iPad), Chrome 5.0+, Opera 11.5+ and Internet Explorer 7+. Performance is slow on IE 7 and 8. Performance is good on IE 9+.

File organisation
The application files are organised as shown in the diagram below.

![Diagram of file organisation]
See the Readme.txt file in the top level of the release directory for specific file names. The Readme file is also available online via the “About” tab in the application.

**Data File Format**

The application requires that 3 files be present in the Data directory. 1default.dat, 1default.node.csv, and 1default.leaf.csv

(In fact it will run with no data directory at all. However it will only load demo data.)

The format of these files is specified below

1) The .dat file holds meta-data:
   - Line 1: the name of the node file to be loaded.
   - Line 2: the name of the leaf file to be loaded.
   - Line 3: a short text description of the origin of the data.
   - Line 4: a search URL to which any leaf data gets appended for to call a search engine.
     – Examine the supplied file, “1default.dat”, as an example

2) The other 2 files should be CSV files.

3) The node file holds the 3 column MATLAB clustering output.
   - The column headings would be Item-A, Item-B, and Merge Ht. (However, any column headings should be removed from the file).
     - Examine the supplied file, “1default.node.csv”, as an example

4) The leaf file holds the leaf descriptions (short and long descriptions) keyed to their MATLAB leaf cluster ID number. The three column headings would be Leaf ID No, Short Description, and Long Description. (However, any column headings should be removed from the file).
    - Examine the supplied file, “1default.node.csv”, as an example.
The Dendrogrammer v 1.0.1 Readme.txt

In this file
- Installation
- System requirements
- Browser compatibility and Compatibility list
- The application files
- Data file format
- Licence and copyright notices

Installation:

- Extract or copy the "dendrogrammer" directory into PHP enabled web space.

Running the application:

- Assuming you have installed the "dendrogrammer" directory in web space that can be addressed with the following URL www.mywebspace.com
then you can run the application by asking your browser to request this URL
www.mywebspace.com/dendrogrammer/version1_0_1/dendrogrammer.php

Loading data using the calling URL:

- If the name of the desired data file selection is known in advance, and that data file is present in the data directory, then the data to load can be specified as a query string in the calling URL.
e.g.
www.mywebspace.com/dendrogrammer/version1_0_1/dendrogrammer.php?mydatafile.dat

System Requirements:

The application files need to be placed in PHP enabled web space. The application is run from a web browser requesting the PHP file (dendrogrammer.php).

Browser compatibility:

In theory the application can run on any modern JavaScript enabled browser (see compatibility list below). In practice with larger data sets, the 'JScript' engine in Internet Explorer 7 and 8 will struggle to cope with the processing demand unless run on high end PCs. All the other browsers should perform well even on modest equipment including tablet computers.

Compatibility list:

Firefox 3.6+, Safari 5.0+ (including iPad), Chrome 5.0+, Opera 11.5+ and Internet Explorer 7+. Performance is slow on IE 7 and 8. Performance is good on IE 9+.

These are the compatible browsers proved in tests. It is a conservative list.

The Raphael documentation claims that the graphics should render properly in Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+.

jQuery UI claims to render properly in "IE 6.0+, Firefox 3+, Safari 3.1+, Opera 9.6+ and Google Chrome".

The Files:
The application directory should contain this file and 2 directories:
Readme.txt
/version1_0_0
and
/libraries

The version1_0_0 directory should contain 3 files and a data directory
dendrogrammer.js
dendrogrammer.php
dendrogrammer_styles.css
and
/data

The data directory should contain 3 files (See Data file format section below)
1default.dat
1default.node.csv
1default.leaf.csv

The libraries directory should contain 2 files and the ui directory:
jquery-latest.min.js
raphael.js
and
/ui

The ui directory should contain the following files and an image directory:
The jQuery UI js files required in the JQuery UI folder are:
jquery.effects.core.js
jquery.effects.pulsate.js
jquery.ui.button.js
jquery.ui.core.js
jquery.ui.dialog.js
jquery.ui.draggable.js
jquery.ui.mouse.js
jquery.ui.position.js
jquery.ui.tabs.js
jquery.ui.widget.js

Associated css style sheet files are:
jquery.ui.all.css
jquery.ui.base.css
jquery.ui.button.css
jquery.ui.core.css
jquery.ui.dialog.css
jquery.ui.resizable.css
jquery.ui.selectable.css
jquery.ui.tabs.css
jquery.ui.theme.css

There is a directory of images in the ui directory
/images

The images directory should contain the following files:
ui-bg_flat_0_aaaaaa_40x100.png
ui-bg_flat_75_ffffff_40x100.png
ui-bg_glass_55_fbf9ee_1x400.png
ui-bg_glass_65_ffffff_1x400.png
ui-bg_glass_75_dadada_1x400.png
ui-bg_glass_75_dadada_1x400.png
ui-bg_glass_75_e6e6e6_1x400.png
ui-bg_glass_95_fef1ec_1x400.png
ui-icons_2e83ff_256x240.png
ui-icons_222222_256x240.png
ui-icons_454545_256x240.png
Data File Format:

The application requires that 3 files be present in the Data directory. 1default.dat, 1default.node.csv, and 1default.leaf.csv.

(In fact it will run with no data directory at all. However it will only load demo data.)

Further sets of data files should be added to the data directory.

The format of these files is specified below

1) The .dat file holds meta-data:
   Line 1: the name of the node file to be loaded.
   Line 2: the name of the leaf file to be loaded.
   Line 3: a short text description of the origin of the data.
   Line 4: a search URL to which any leaf data gets appended to call a search engine.
   - Examine the supplied file, “1default.dat”, as an example
2) The other 2 files should be CSV files.
3) The node file holds the 3 column MATLAB clustering output.
   The column headings would be Item-A, Item-B, and Merge Ht. (However, any column headings should be removed from the file).
   - Examine the supplied file, “1default.node.csv”, as an example
4) The leaf file holds the leaf descriptions (short and long descriptions) keyed to their MATLAB leaf cluster ID number. The three column headings would be Leaf ID No, Short Description, and Long Description. (However, any column headings should be removed from the file).
   - Examine the supplied file, “1default.leaf.csv”, as an example.

Licence and copyright notices:

Copyright 2011 David Robb.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jQuery and Raphael

The application makes use of the the jQuery and Raphael JavaScript libraries

The following notices apply to those:
jQuery

Copyright (c) 2011 John Resig, http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Raphael

Copyright © 2008 Dmitry Baranovskiy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.
Appendix 9

Evaluation

- Text of the evaluation task
- Closed question response collation
- Open question response collation
The Dendrogrammer Evaluation Task

The Dendrogrammer was created as one component in the analysis of multivariate data. In particular it is intended to allow already pre-processed data to be
- classified into groups and then either
- used in further analysis, or
- used in some form of presentation about the data.

This evaluation should help to assess how well, the application does this.

Thank you for taking part. Please follow the steps below.

Launching the application (section A)
1) Using Firefox go to this URL
   http://www.macs.hw.ac.uk/~dar14/project/prototype_1_dendrogramer/evaluation_v3_1_0/prototype_3_1_0.php
2) You should see a page entitled “The Dendrogrammer” with a small tree diagram (dendrogram). This small dendrogram has 6 leaf nodes at the left-hand end with names such as “Thing Two”.

Loading a new data set (section B)
1) Click the “Data” tab
2) From the “Load different data” table click “the_evaluation_data.dat”
3) You should see a new dendrogram.
4) Verify that the correct data set has been loaded by clicking the “Data” tab again.
   • Under “The current data” check that it shows “Dat file name = the_evaluation_data.dat”.
   • In the 4-column table under “Node file name” check that Line No. 1 contains the numbers 14, 30, and 0.
5) Scroll down to the 3-column table under “Leaf file name”.
   • Find line numbers 14 and 30. Enter the corresponding words you find in the box below.

<table>
<thead>
<tr>
<th>Leaf 14 =</th>
<th>Leaf 30 =</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• In terms of this test data, this means that items 14 and 30 share a dissimilarity measure of zero. You should have found that 14 is “created” and 30 is “technologies”. If the data was text-mined from a set of web pages then these two words must have appeared together often and, for that reason, have been measured as highly similar, i.e. of zero dissimilarity.

6) Please give your opinion about the ease of use of the “Load different data” feature. Where offered a choice of answers please choose only one and indicate your choice by placing an “X” in the appropriate place.

| a) Would you say that, with the application, loading a new data set is… |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| very easy?      | easy?           | neither easy nor | difficult?      | very difficult? |
|                 |                 |                  |                 |                 |
b) If you were unable to load a new data set, briefly describe what went wrong in the box below and move on. Otherwise leave this box blank. (If you wish, you may simply state that the feature did not work).


c) Would you say that accessing the information in the data tab to verify that one has loaded the correct data is...

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

d) If you were unable to access the information in the data tab briefly describe what went wrong in the box below and move on.


**Examining the dendrogram (section C)**

1) Click the “Dendrogram” tab.

2) Move your mouse around over the dendrogram and see how the leaf labels down the left-hand side respond as you mouse-over the elements (clusters and leaves) of the dendrogram. Together, the clusters and leaves are referred to as nodes on a dendrogram.

3) Locate the two leaves, “created” and “technologies” (about 16 leaves down from the top).

4) Now move your mouse to the far right-hand side over the cluster that is furthest to the right at “Max ht=9.7272”. This is the root cluster. With your mouse over the root the whole dendrogram should be highlighted.

5) Click the root cluster. You will see a dialog containing a “Search link”, the cluster ID number and a list of descendant leaves.

6) Enter the Cluster ID number in the box below

   The root cluster ID =

7) Clear the dialog by clicking “OK” and try clicking other clusters to see the list of their descendant leaves.

8) Return your mouse over the root cluster and, bearing in mind where “created” and “technologies” are, follow the branches “down” the tree until you get to “Cluster ID 101, created, technologies”.

9) Trace back towards the root from “Cluster ID 101, created, technologies” until you reach its immediate ancestor, **Cluster ID129**.

10) Find out the descendants of Cluster ID129 and use Copy and Paste from the dialog to complete the box below to show the descendant leaves of that cluster.

   Descendants of Cluster ID129:
11) Find any other cluster on the dendrogram that has 4, 5 or 6 descendant leaves. Paste the cluster details, including ID number in the box below.

This cluster has between 4 and 6 descendant leaves:

12) Please give your opinion about the “interactive nodes” feature.

a) Would you say that accessing a node’s information is...

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) If you were unable to access a node’s information, briefly describe what went wrong in the box below.

Using the Threshold adjuster bar (section D)

1) Find the “Threshold Toolbar”. It is the longer toolbar immediately above the horizontal axis of the dendrogram. (It has several buttons and a 5-digit number field). Assuming the threshold has not been moved the orange number should read 9.7272. This is the “Threshold value field”.

2) Click several times on the “minus” button. You should see the orange number in the Threshold value field change. You should also see the green “Threshold adjuster bar” move.

3) With your mouse click and drag on the Threshold adjuster bar and slide it left and right along the Dissimilarity axis.

4) Locate Cluster 129 again. Its descendant leaves are “created, technologies, tablet-computers, and university”.

5) Drag the Threshold adjuster bar to where Cluster 129 branches between “created” and “technologies” on one side and “tablet-computers”, and “university” on the other. If you read off the Threshold value field the Threshold adjuster bar should be at roughly 2.54 (between, say, 2.51 – 2.58). These four leaves all share a dissimilarity of 2.54 . i.e. the descendants of Cluster 129 merge with a dissimilarity of around 2.54 units.

Forming groups of leaves using the “Threshold Toolbar” (section E)

1) Drag the Threshold adjuster bar to approximately 5.6

2) Click the “Apply threshold” button. Notice what occurs:
   - A dialog appears (entitled “Groups formed”) stating the number of groups and the threshold setting.
   - Coloured bands are overlaid on the dendrogram delineating the groups.
   - The “Groups” tab flashes and indicates the number of groups.

3) Clear the dialog by clicking Ok

4) Look in the Groups tab. Copy all the information about the groups and paste it into the box below.

   The box will expand to take the pasted text. Don’t worry about pagination or font-size. Just paste it in and move on.

5) Using copy and paste, complete the following boxes about the groups that were formed.

   a)
The threshold =

b) The details of the smallest group=

c) The details of the largest group=

6) Return to the Dendrogram tab and Click the “Remove” button to clear the groups from the dendrogram.

7) Now use the threshold bar and “Apply threshold” button to form the leaves into 4 groups. Paste all of the resulting group detail from the Groups tab (including the threshold detail) into the box below.

8) Now specify an exact threshold at which to form groups by selecting the Threshold value field (the 5-digit orange number), deleting the current number, entering the number 6.4 using your keyboard and clicking Apply threshold. Paste the detail of the groups that are formed into the box below.

Forming groups of leaves using the “Set no. of groups Toolbar” (section F)

1) Clear any previous groups away using the “Remove” button.

2) Locate the “Set no. of groups Toolbar”. (It has one button and a small number-field. This is the “No of groups field”. Assuming no groups have been set it should read “1”).

3) Select the “No of groups field” and change the number to 6.

4) Click the “Apply no of groups” button.

5) You should see that 6 groups have been formed.

6) Paste all of the resulting group detail about the 6 groups from the Groups tab (including the threshold detail) into the box below.

7) Use the “Set no. of groups Toolbar” to form 14 groups. Paste the groups detail into the box below

8) Please give your opinion about the “group formation” features.

a) Would you say that adjusting the threshold is…

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) If you were unable to adjust the threshold at all, briefly describe what went wrong in the box below.
c) Once a threshold has been set and applied, would you say that accessing the information about the groups in the Groups tab is…

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


d) If you were unable to access the information about the groups in the Groups tab, briefly describe what went wrong in the box below.


e) Would you say creating groups by specifying a particular number of groups is…

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


f) If you were unable to create groups by specifying a particular number of groups, briefly describe what went wrong in the box below.


Using the “Threshold Toolbar” to create a summary dendrogram (section G)

1) Clear any groups using the “Remove” button
2) Adjust the threshold bar to approximately 3.0
3) Click the “Truncate” button. Notice what occurs:
   - The leaf labels disappear
   - The dissimilarity figure at the origin is no longer zero
   - The dendrogram occupies less vertical space
4) Adjust the threshold again and truncate the dendrogram a little higher up the dissimilarity scale.
5) Click on one or two clusters and view the leaf data associated with those clusters.
6) Use the threshold to form some groups.
7) View the group list in the groups tab. All the leaf data is still available via the interactive nodes and when groups are formed. The summary dendrogram just offers an alternative way to visualise the data.
8) Click the “Restore” button to return to the original view of the dendrogram.
9) Try creating a summary dendrogram again, applying and removing groups, and restoring the dendrogram.
10) Please give your opinion about the “summary dendrogram” feature.
    a) Would you say that creating a summary dendrogram is…

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


b) If you were unable to create a summary dendrogram, briefly describe what went wrong in the box below.
c) Would you say that restoring the original dendrogram is…

<table>
<thead>
<tr>
<th>very easy?</th>
<th>easy?</th>
<th>neither easy nor difficult?</th>
<th>difficult?</th>
<th>very difficult?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

d) If you were unable to restore the original dendrogram, briefly describe what went wrong in the box below.

Final section (section H)

To finish there are some questions seeking your opinion about the application as a whole. Your comments can be a short or as long as you wish, but there is no need to feel you need to give extended responses.

Please comment giving your opinion of the following areas of the application

1) Navigation within the application

2) Responsiveness of the interactive dendrogram

3) Presentation of the commands used to manipulate the dendrogram

4) If you have any other comments please make them here

This is the end of the evaluation. Please save and let the evaluation administrator know you have finished.

Thank you again for taking part.
### Evaluation Response Collation

**Task questions**

<table>
<thead>
<tr>
<th>Serial</th>
<th>B5</th>
<th>C6</th>
<th>C10</th>
<th>C11</th>
<th>E4</th>
<th>E5a</th>
<th>E5b</th>
<th>E5c</th>
<th>E7</th>
<th>E8</th>
<th>F6</th>
<th>F7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.5054</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6013</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6235</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.5993</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6088</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6549</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6104</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.6016</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Threshold setting**

**Opinion Qs**

5 = very easy 1 = very difficult

<table>
<thead>
<tr>
<th>Serial</th>
<th>B6a</th>
<th>B6b</th>
<th>B6c</th>
<th>B6d</th>
<th>C12a</th>
<th>C12b</th>
<th>F8a</th>
<th>F8b</th>
<th>F8c</th>
<th>F8d</th>
<th>F8e</th>
<th>F8f</th>
<th>G10a</th>
<th>G10b</th>
<th>G10c</th>
<th>G10d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Collation of responses to the open questions:

Question H1 - Navigation within the application
- The navigation was simple and fluid.
- Working perfectly
- Navigation is quite intuitive and simple. I would put the data menu before the dendrogram as you first load the data and then see the dendrogram.
- Easy. But the leaf texts are in small fonts.
- All navigation controls are clear and with obvious functionality.
- Navigation around the dendrogram and the different sections is easy and having the group tab flash on change is useful.

Question H2 - Responsiveness of the interactive dendrogram
- Responsiveness was quick and easy to follow
- Very simple and easy to use.
- Dendrogram runs very smoothly, no problems at all.
- It’s a little bit slow
- Quick.
- The application responded very quickly to input and there appeared to be no lag.
- The dendrogram is very responsive and the interactive nature of it (being able to click on groups etc.) is very clear and useful.

Question H3 - Presentation of the commands used to manipulate the dendrogram
- Easy to understand especially when following the instructions
- Useable and understandable.
- Dragging the threshold works really well, I think you don't need the minus and plus signs (plus extra arrows) as users can simply type the number and they should be able to press enter. You can also add bigger labels to your buttons like "truncate tree", "remove grouping", "restore dendrogram" and separate them a bit more.
- Text size for nodes is too small
- Good.
- As an experienced user of dendrograms for visualising data, the commands were clear and exactly what I would have expected to see.
- Most of the commands are very self-explanatory. It was, however, somewhat unclear of the difference between 'Remove' and 'Restore' when starting use. Also, the error message when you try to apply too many groups is very long and quite unclear.

Question H4 - If you have any other comments please make them here
- Text is very small not sure if that can be helped.
- Really awesome.
- Labels' font is a bit too small to read in dendrogram leaves. I don't know what is the point of the pop-up if you open a node and click "search link". Loading data looks rushed and not finished.
- There is no need to have 4 decimal places on the threshold
- Since each item in one group has different length, I believe that it will be better to display group information in a grid. In this way, users can learn the sizes of the groups more easily.2. Sort the groups in ascending or descending order according to the size of them. In tab ‘Data’, the items, such as ‘Load different data’, ‘The current data’, could be displayed at the top. And add hyperlinks to them. I think it will be more convenient for users to access them. Additionally, there should be more obviously separator between any two of them.
- Professional looking application with conservative but helpful colour scheme
- I found the 'summary dendrogram' quite confusing and am unsure of how useful it will be. In addition, I found the group tab, while useful, difficult to read due to lack of formatting. Using colours like on the dendrogram overlay would alleviate this. Overall, however, it is very simple to use and has nice interactive features.