VISUALIZING PHYLOGENETIC TREES: ALGORITHMS AND VISUAL COMPARISON TECHNIQUES

by

WAN MOHD NAZMEE WAN ZAINON

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

December 2011

ACKNOWLEDGEMENTS

In the name of Allah, the most gracious and the most merciful. Alhamdulillah, highest praise to Allah for his wills that gave me the strength and patience to complete this thesis

I am extremely grateful for the opportunity to have had Assoc. Prof. Abdullah Zawawi as my supervisor and Assoc. Prof. Bahari Belaton as my co-supervisor. Both of them have been an inspiration to me and what I value most about these past years was the opportunity to absorb their insights into the specifics of my work and their fundamental approach to research. Both of them have given me endless support in all my writing.

Thanks to many current and former people at the School of Computer Sciences, USM for their friendship over the years. My study was mainly supported by the Universiti Sains Malaysia Academic Staff Training Program.

I acknowledge, appreciate, and return the love and support of my family, without whom I would be lost. My wife Hasnah and my 3 beautiful children, Nasuha, Hadi and Delisha who have been my emotional anchors through not only during my study years, but my entire life. Both my parents, siblings and their family have also become an important part in my world. I am forever indebted to all my family members for their affection, support, and constant encouragement.

TABLE OF CONTENTS

Acknowledgements .... **ii**

List of Figures .... **vii**

List of Tables .... **xii**

Abstrak .... **xiii**

Abstract .... **xv**

CHAPTER 1: INTRODUCTION **1**

1.1 Introduction to Visualization 1

1.2 Bioinformatics Visualization 3

1.3 Phylogenetic Tree Visualization 6

1.4 Problem Statement 7

1.5 Research Objectives 9

1.6 Research Approach 9

1.7 Scope and Limitations 10

1.8 Contributions 11

1.9 Thesis Structure 13

CHAPTER 2: LITERATURE REVIEW **15**

2.1 Tree-structured Visualization Techniques and Applications 15

2.1.1 Treemaps 16

2.1.2 Arc Trees 21

2.1.3 Cone Trees 22

2.1.4 Hyperbolic Tree 24

2.1.5 SpaceTree 26

2.1.5 Discussion on Tree Structure Visualization 28

2.2 Phylogenetic Analysis 29

2.2.1 Terminology 30

2.2.2 Phylogenetic Trees 31

2.2.3 Cladogram, Phylogram and Dendogram 32
2.2.4 Rooted and Unrooted Trees 34
2.2.5 Ancestor, Common Ancestor and Most Recent Common Ancestor ... 35
2.2.6 Creating a Phylogenetic Tree 36
2.2.6.1 Neighbor Joining (NJ) 37
2.2.6.2 Parsimony 37
2.2.6.3 Maximum Likelihood (ML) 37
2.3 Tree Comparison 38
2.3.1 Consensus Trees 39
2.3.2 Agreement Subtrees 41
2.3.3 Triplets 42
2.3.4 Nearest Neighbor Interchange (NNI) 43
2.5.7 Discussion on Current Tree Comparison Methods 44
2.4 Phylogenetic Tree Alignment Algorithm 45
2.5 Phylogenetic Tree Applications 49
2.5.1 PHYLIP 50
2.5.2 PAUP 51
2.5.3 COMPONENT 53
2.5.4 TreeView 54
2.5.5 Mesquite 55
2.5.6 TreeJuxtaposer 58
2.5.7 Discussion on Current Phylogenetic Tree Application 60
2.6 Summary 61

CHAPTER 3: RESEARCH METHODOLOGY & PROPOSED FRAMEWORK FOR TREE STRUCTURE VISUALIZATION 62

3.1 Research Methodology 62
3.2 Existing Frameworks 65
3.3 Proposed Framework 66
3.4 Summary 69

CHAPTER 4: THE PROPOSED ALGORITHMS AND VISUAL COMPARISON TECHNIQUES 70

4.1 The Alignment Algorithms 70

4.1.1 The Minimum Triplets Difference (MTD) Algorithm 71

4.1.2 The Maximum Branch Similarity (MBS) Algorithm 78

4.1.3 The All-But-*n* (AB1) Algorithm 81

4.1.4 Combining the Algorithms 102

4.2 Visual Comparison Approaches 104

4.2.1 Branch Offset 104

4.2.2 Common Branch Colouring 106

4.2.3 Node Pushing 108

4.3 Implementations 110

4.3.1 Implementations of Alignment Algorithm 110

4.3.2 Implementation of Visual Comparison Techniques 112

4.4 Summary 114

CHAPTER 5: EVALUATION 116

5.1 Performance Evaluation of Tree Alignment Algorithms 116

5.1.1 Evaluating the Effectiveness of the Algorithms 117

5.1.2 Evaluating the Efficiency of the Algorithms 119

5.1.2.1 Theoretical Evaluation 120

5.1.2.2 Empirical Evaluation 120

5.1.3 Discussion 122

5.2 Visualization Evaluation 123

5.2.1 Current Evaluation Practices in Information Visualization 123

5.2.2 The Evaluation Methods 125

5.2.3 Evaluation of Navigation Task 125

5.2.3.1 Experimental Setup 126

5.2.3.1 Result 133

5.2.4 Evaluation of the Visual Comparison Techniques 135

5.2.4.1 Experimental Setup 135

5.2.4.1 Result 138

5.2.5 Open Questions on Visual Comparison Techniques 145
5.2.6 Discussion 147
5.3 Summary 148

CHAPTER 6: CONCLUSIONS AND FUTURE WORK 150

6.1 Conclusions 150
6.2 Future Work 152
6.2.1 Colour 152
6.2.2 Tree Size 153
6.2.3 Minimum Triplet Difference Algorithm 153
6.2.4 Limitations 154
6.3 Concluding Remarks 155

Bibliography 156

Appendix A: The Example of NEXUS File Format 165

Appendix B: Evaluation of Navigation Task 168

Appendix C: Evaluation of the Visual Comparison Techniques 176

Appendix D: Open Questions on Visual Comparison Techniques 194

Appendix E: VCPT Development Environment and Graphical User Interface ... 196

List of Publications 206

LIST OF FIGURES

Figure 1-1 : Visualization maps the computer representation of reality into images or animation. 2

Figure 2-1: An example of tree-structure data and its treemap representation (Shneiderman, 1992) 17

Figure 2-2: Screen design for treemap representation of analytic hierarchy process (Asahi et al., 1994) 18

Figure 2-3: Screen shot of squarified cushion treemap (Bruls et al. 2000) 19

Figure 2-4: Screen capture of 3D treemaps (Bladh et al., 2004) 19

Figure 2-5:Example of Zoomable treemaps representation (Blanch & Lecolinet, 2007) 20

Figure 2-6: Example of arc trees (Neumann et al., 2005). 22

Figure 2-7: Example of a cone-tree (Robertson et al., 1991). 22

Figure 2-8: Example of reconfigurable disc tree (Jeong & Pang, 1998). 23

Figure 2-9: Example of hyperbolic tree (Lamping & Rao, 1995) 24

Figure 2-10: Example of walrus 3D hyperbolic display. 26

Figure 2-11: Example of a spaceTree (Plaisant et al., 2002). 27

Figure 2-12: Example of (fictitious) phylogenetic tree 31

Figure 2-13: Example of partially resolved and fully resolved phylogenetic tree. 32

Figure 2-14: Example of cladogram, phylogram, and dendogram 33

Figure 2-15: Example of unrooted tree 34

Figure 2-16: Example of ancestor and common ancestor 35

Figure 2-17: Example of 2 fictitious phylogenetic trees 39

Figure 2-18: Strict consensus tree for trees in Figure 2-17 40

Figure 2-19: Greatest agreement subtrees for trees in Figure 2-17 41

Figure 2-20: Triplets for tree 1 in Figure 2-17 42

Figure 2-21: Example of one-step NNI for tree 1 of Figure 2-17 44

Figure 2-22: Top view of a 2.5D visualisation of eight phylogenetic trees with unmodified leaf orderings. (Dwyer & Schreiber, 2004) 47

Figure 2-23: Another top view of a 2.5D visualisation from Figure 3-1 but with crossings minimised using their proposed algorithm. (Dwyer & Schreiber, 2004) 48

Figure 2-24: Example of PAUP screen shot (Swofford, 2000) 52

Figure 2-25: Example of COMPONENT application screen shot (Swofford, 2000)..... 53

Figure 2-26: Example of the result window when comparing trees using COMPONENT 54

Figure 2-27: Example of TreeView application screen shot.. 55

Figure 2-28:TreeSet Interface with two consensus trees for the two selected groups of points (Amenta & Klingner, 2002). 57

Figure 2-29: An example of TreeJuxtaposer screen shot (Munzner et al., 2003). 59

Figure 3-1: Research Methodology.. 63

Figure 3-2: Current Framework 66

Figure 3-3: Proposed Framework.. 67

Figure 3-4: Example of 2 simple tree... 67

Figure 3-4: Creating visually identical presentation of phylogenetic tree 68

Figure 4-1: 12 possible labelled triplet patterns 71

Figure 4-2: An example of 2 fictitious trees 72

Figure 4-3: Extracting triplet 124... 72

Figure 4-4: All possible variants for a tree with 3 nodes 73

Figure 4-6: Two trees in original display... 77

Figure 4-7: Rearrangement of the trees of Figure 4-7 with MTD.................................. 77

Figure 4-8: Example of two trees... 78

Figure 4-10:Two trees before the maximum branch similarity algorithm has been applied. 80

Figure 4-11: The same trees after maximum branch similarity has been applied. 80

Figure 4-12: Example of 2 trees, T_1 and T_2. 82

Figure 4-13: Greatest agreement subtree for T_1 and T_2 82

Figure 4-14: Two attachment points on the agreement subtree. 83

Figure 4-15: Partitioning the GAS 83

Figure 4-16: The swapping of the upper and lower tree 84

Figure 4-17: Final result for the algorithm........ 85

Figure 4-18: The two trees are now drawn separately 85

Figure 4-19: New example for T_1 and T_2........ 86

Figure 4-20: Greatest agreement subtree and attachment points 86

Figure 4-21: The partitioned greatest agreement subtree........ 87

Figure 4-22: The rearrangement of the lower trees........ 88

Figure 4-23: The result for the special cases........ 89

Figure 4-24: The result is drawn seperately........ 89

Figure 4-26: Tree 1, tree 2 and their agreement subtree 91

Figure 4-27: Example of agreement subtree and its attachment point........ 92

Figure 4-28: Rearranged GAS for Figure 4.26 92

Figure 4-29: The final result with the nodes reinserted 93

Figure 4-30: Example for case 2 93

Figure 4-31: Rearranged GAS for Figure 3-29........ 95

Figure 4-32: The final result with the nodes reinserted 95

Figure 4-33: New example for case 2 96

Figure 4-34: Rearranged GAS for Figure 4-32........ 97

Figure 4-35: The final result with the nodes reinserted 97

Figure 4-36: Example of the “crossover” arrangement........ 98

Figure 4-37: The final results with the nodes reinserted ... 98

Figure 4-38: Example of “grouped” attachment case ... 99

Figure 4-39: The final result with the nodes reinserted ... 99

Figure 4-40: Example of AB3 ... 100

Figure 4-41:Rearranged GAS for Figure 4-38 ... 101

Figure 4-42:The final result with the nodes reinserted ... 101

Figure 4-43:Results of Figure 4-7 after applying MTD algorithm and both MTD and MBS algorithms ... 103

Figure 4-44:Results of Figure 4-10 after applying MBS algorithm and both MBS and MTD algorithms ... 103

Figure 4-45: An example of two trees before gaps are inserted ... 105

Figure 4-46: The results after gaps are inserted ... 105

Figure 4-47: Example of nodes collapsing action ... 104

Figure 4-48: Example of two trees ... 107

Figure 4-49: Example of two trees with the similar nodes coloured. ... 108

Figure 4-50: Push up and push down of the common ancestor ... 109

Figure 4-51: Example of two trees ... 109

Figure 4-52: The tree after pushing node 4 ... 110

Figure 5-1: Example of 3 different trees with their number of crossings ... 127

Figure 5-2: Effectiveness of the algorithms based on the number of crossings ... 127

Figure 5-3: The algorithms running time ... 127

Figure 5-4: Two trees presented for task 1 ... 127

Figure 5-5: The final result for task 1 ... 127

Figure 5-6: Two trees presented for task 2 ... 128

Figure 5-7: The result after the MBS button is clicked. ... 129

Figure 5-8: New result of the exercise ... 129

Figure 5-9: The result after the MTD button is clicked 130

Figure 5-10: The final result for task 2 131

Figure 5-11: Two trees presented for task 3... 132

Figure 5-12: The final result for task 3 132

Figure 5-13: The results of the first step in task 3 133

Figure 5-14: Example of tree transformation process.. 137

Figure 5-15: Pair of small trees with total alignment.. 140

Figure 5-16: Pair of medium sized trees with partial alignment.................................. 141

Figure 5-17: Pair of trees with their structural similarity highlighted 142

Figure 5-18: Pair of trees with their common ancestors “pushed” 144

Figure 5-19: The result of combining node alignment, highlighting the common structures and “pushing” the common ancestor methods. 148

LIST OF TABLES

Table 1-1: List of the main areas in bioinformatics visualization (Lesk, 2002) 5

Table 2-1: Summary of Tree Structure Visualization and Applications......................... 28

Table 2-2: Summary of current tree comparison method 33

Table 2-3: List of some of the PHYLIP programs and their functions. 50

Table 2-4: Advantages and disadvantages of current phylogenetic tree application. 60

Table 4-1: Matching triplets for tree 1 and tree 2 73

Table 4-2: Number of nodes and their possible tree variants... 74

Table 4-3: Matrix of differences for two trees 75

Table 5-1: Four Current Thematic Areas of Evaluation (Plaisant, 2004) 123

Table 5-2: The time taken (in minutes) for participants to complete the usability testing. 134

Table 5-3: The time taken (in minutes) for participants to complete the visualization evaluation. 139

Table 5-4: Summary of the comments for the open question. 146

Pemvisualan Pepohon Filogenetik: Algoritma dan Teknik Perbandingan Visual

ABSTRAK

Tesis ini mengkaji pemvisualan data berstruktur pepohon. Secara khususnya, penekanan diberi kepada pemvisualan keserupaan dan perbezaan antara pasangan pepohon. Terdapat banyak bidang penyelidikan (seperti biologi, linguistik, kimia dan sains komputer) yang menggunakan pepohon sebagai struktur data asas. Dorongan untuk membuat kajian ini datang dari bidang bioinformatik yang melibatkan pembinaan pepohon filogenetik kompleks oleh ahli biologi untuk mewakili evolusi spesies atau gen. Sebaik sahaja pepohon filogenetik mendedahkan potensi saling hubungan antara spesies yang dikaji, langkah seterusnya adalah untuk mengesahkan maklumat data tersebut. Pada ketika ini, perbandingan antara pepohon yang berasal daripada pelbagai data percubaan diperlukan supaya model terbaik untuk satu set spesies yang dikaji diperoleh. Antara dua isu utama yang timbul apabila membandingkan data tersebut adalah untuk mengetahui cara perbandingan pepohon filogenetik yang cekap dan berkesan, serta cara untuk mempersembahkan hasil perbandingan secara visual.

Tesis ini meneliti teknik pemvisualan dan perbandingan pepohon yang ada pada masa ini dan mencadangkan algoritma yang memaparkan pepohon perduaan yang diselesaikan sepenuhnya dengan cara yang memudahkan perbandingan visual mereka. Pendekatan utama adalah untuk menghasilkan satu kerangka baru untuk teknik pemvisualan struktur pepohon yang akan memaparkan pasangan pepohon "muka ke muka" dengan nod daunnya terjajar. Secara umumnya, nod daun pepohon yang berbeza tidak mungkin dapat dijajarkan sepenuhnya. Tesis ini membincangkan beberapa algoritma yang menyusun dan menjajar nod dalam pelbagai cara: algoritma perbezaan triplet minimum, algoritma keserupaan cabang maksimum, dan algoritma semua kecuali n. Selain itu, pelbagai teknik perbandingan berasaskan pemaparan dicadangkan untuk memvisualkan keserupaan dan perbezaan antara pasangan pepohon. Akhirnya alatan interaktif prototaip yang dinamakan VCPT (*Visual Comparison of Phylogenetic Trees*) dibangunkan untuk meninjau dan menilai konsep yang dicadangkan dan isu-isu berkaitan dengan perbandingan dan manipulasi visual pepohon filogenetik.

Hasil kajian menunjukkan bahawa gabungan penyusunan semula automatik dan manual seringkalinya berkesan dalam menghasilkan susunan yang memudahkan perbandingan pepohon dengan cepat, walaupun untuk pepohon yang agak besar. Hasil kajian ini juga mengesahkan kerangka teknik pemvisualan struktur pepohon yang dicadangkan. Algoritma dan teknik berasaskan pemaparan yang dicadangkan dalam tesis ini akan membantu pengguna untuk memahami pasangan pepohon dengan memvisualkan keserupaan dan perbezaan antara pasangan berkenaan.

Visualizing Phylogenetic Trees: Algorithms and Visual Comparison Techniques

ABSTRACT

This thesis is about visualizing tree structured data. In particular, the emphasis is on visualizing the similarities and differences between pairs of trees. There are many research areas (such as biology, linguistics, chemistry and computer science) that use the tree as a basic data structure. The impetus for the work comes from the field of bioinformatics, where biologists construct complex phylogenetic trees to represent the evolution of species or genes. Once phylogenetic trees reveal potential interrelationships between examined species, the next step would be to validate the derived data. At this point, the comparison between the trees derived from various experimental data is necessary in order to find the best model for a given set of species. The two main issues that arise when comparing these data is to know how to efficiently and effectively compare phylogenetic trees, and how to visually present the results of the comparison

This thesis examines current tree visualization and comparison techniques and proposes algorithms that display fully resolved binary trees in a way that facilitates their visual comparison. The primary approach is to present a new framework for tree structure visualization techniques that will display pairs of trees “face to face” with leaf nodes aligned. In general, it will not be possible to fully align leaf nodes of different trees. This thesis presents several algorithms that arrange and align the nodes in various ways: the minimum triplet difference algorithm, the maximum branch similarity algorithm, and all-but-n algorithm. In addition, a variety of visual comparison techniques are proposed to visualize the similarities and differences between pairs of trees. Finally a prototype interactive tool named VCPT (Visual Comparison of Phylogenetic trees) is developed to explore and evaluate the proposed concepts and issues in regard to visual comparison and manipulation of phylogenetic trees.

The results show that a combination of automatic and manual rearrangement is often effective in rapidly generating an arrangement that facilitates tree comparison, even for quite large trees. The results also validate the proposed framework for tree structure visualization techniques. The algorithms and visual comparison techniques proposed in this thesis will help users to understand pairs of trees by visualizing the similarities and differences between them.

"Like good writing, good graphical displays of data communicate ideas with clarity, precision, and efficiency."

"Like poor writing, bad graphical displays distort or obscure the data, making it harder to understand or compare."

(Michael Friendly, York University)

CHAPTER 1

INTRODUCTION

1.1 Introduction to Visualization

Graphics has long been recognised as an effective and economical means of presenting and analysing large and complex data sets. Potential benefits arise both from the exploitation of the human visual system with its remarkable ability in organising and detecting objects, and from the strength of graphics in encoding intrinsic characteristics of a dataset into features that can be readily processed by our visual system.

Visualization is an old term, which has received a large amount of interest in the computer science community. Generally, visualization is the transformation of data and information into pictures. One definition of visualization is to form a mental vision, image, or picture of (something not visible or present to sight, or of an abstraction); to make visible to the mind or imagination [The Oxford English Dictionary, Third Edition, 2010].

Haber and McNabb (Haber & McNabb, 1990) defined visualization as use of computer imaging technology as a tool for comprehending data obtained by simulation or physical measurement. In their understanding, visualization technology is based on the integration of older technologies, including computer graphics, image processing, computer vision, computer-aided design, geometric modelling, approximation theory, perceptual psychology, and user interface studies. Figure 2-1 shows where visualization maps the computer representation into images or animation.

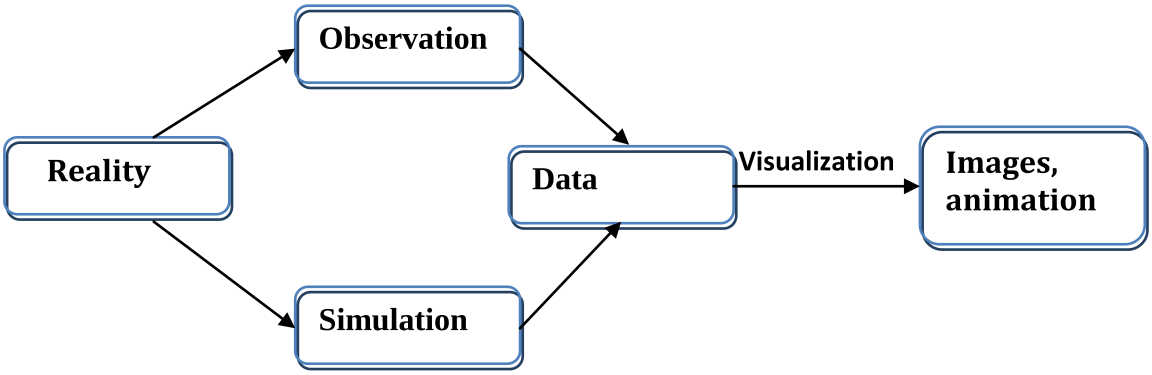


Figure 1-1: Visualization maps the computer representation of reality into images or animation (Belaton, 1995).

In the present day, visualization is used in many scientific areas. Each of these scientific areas has difference data types that need to be visualized. Ben Shneiderman (Shneiderman, 2002) has described seven data types by task taxonomy of information visualizations: 1-D linear, 2-D map, 3-D world, temporal, multi-dimensional, tree, and network.

The input or raw material of visualization can be observation data from microscopes, scanners or satellites; experimental data from a measuring device; or simulation data from computation. The output or end product of visualization is an **image**, or more generally a multimedia object, which is a combination of images, animation, text and sound.

Information visualization, sometimes called InfoVis, is a special kind of visualization. In information visualization, the graphical models may represent abstract concepts and relationships that do not necessarily have a counterpart in the physical world, e.g., information describing user accesses to pages of an Internet portal or records describing selected properties of different car brands and models. Typically, each data unity describes multiple related attributes (usually more than four) that are not of a spatial or temporal nature. Although spatial and temporal attributes may occur, the data exist in an abstract (conceptual) data space. (Ferreira & Levkowitz, 2003).

1.2 Bioinformatics Visualization

Bioinformatics - sometimes called computational biology - is an emerging area in modern science that brings together computer science, mathematics and biology (Bryant, 1997). It is often referred to as the new molecular biology, because it adds expertise in computerised databases and specialised data-analysis tools to the traditional science of exploring the fundamental processes of life at the molecular level. Bioinformatics is being used heavily in the field of human genome research by the Human Genome Project, which focuses on determining and understanding the sequence of the entire human genome (about 3 billion base pairs), and is essential in using genomic information to understand diseases. It is also used extensively for the identification of new molecular targets for drug discovery.

It is difficult to pinpoint the exact beginnings of bioinformatics, but it is easy to see that the field is currently undergoing rapid, exciting growth. This growth has been fuelled by a revolution in DNA sequencing and mapping technology, which has been accompanied by rapid growth in many related areas of biology and biotechnology. All this new DNA and protein sequence data brings with it the challenge of how to turn the raw sequences into information that will lead to new drugs, new advances in health care, and a better overall understanding of how living organisms function.

Visualization is one of the important parts of the field of bioinformatics. It has roles not only in analysis, but also in building more user-friendly interfaces, implementing methods to navigate large information spaces intuitively, and developing powerful techniques to browse and query data interactively via visualization (Robinson & Flored, 1997). Bioinformatics visualization is about how to use the power of computation to visualize and transform the biological data into understandable graphics, which will be able to assist biologists to understand more about the data. Once the data are stored in an accessible, flexible format, the next step is to extract what is important to the biologist and visualize it.

Biologists have been dealing with the problem of information management since the 18th century. Taxonomy was the first informatics problem in biology. In the 1730s, Carolus Linnaeus catalogued 18,000 plant species and over 4,000 species of animals, and established the basis for the modern taxonomic naming system of kingdoms, classes, genera, and species. By the end of 19th century, Baron Cuvier had listed over 50,000 species of plants (Hall, 2007). Now, biologists have reached a point of information overload by collecting and cataloguing information about individual genes.

The evolution of computers over the last half-century has roughly paralleled the development in the physical sciences that allow us to see biological systems in increasingly fine detail. The Human Genome Project is fundamentally about information, and computing contributed not only the raw capacity for processing and storage of data, but also the mathematically sophisticated methods required to achieve the results.

Arthur M Lesk (Lesk, 2008) identified 10 areas of visualization that arise when dealing with the bioinformatics domain. Table 1-1 lists the areas and gives a brief description of each.

Table1-1: List of the main areas in bioinformatics visualization (Lesk 2007)

	Area	Description
1	**Sequence alignment and sequence searching**	Multiple sequence alignment methods assemble pair wise sequence alignment for many related sequences into a picture of sequence homology among all members of a gene family. It helps in visual identification of sites in a DNA sequence or protein that may be functionally important.
2	**Gene prediction**	Gene prediction is only one of the methods for attempting to detect meaningful signals in uncharacterised DNA sequences. It helps molecular biologist make sense out of this unmapped DNA.
3	**Protein Sequence analysis**	The amino acid content of a protein sequence can be used as the basis for many analyses, from computing the isoelectric point, to predicting secondary structure features and post-translational modification sites.
4	**Phylogenetic analysis**	Phylogenetic analysis attempts to describe the evolutionary relatedness of a group of sequences. A phylogenetic tree or cladogram groups species into a diagram that represents their relative evolutionary divergence.
5	**Biochemical simulation**	Biochemical simulation uses the tools of dynamical systems modelling to simulate the chemical reactions involved in metabolism.
6	**DNA microarray analysis**	DNA microarray analysis is a method that expands on classic probe hybridisation methods to provide access to thousands of genes at once.
7	**Whole genome analysis**	As more and more genomes are sequenced completely, the analysis of raw genome data has become a more important task. Users can start from a high-level map and navigate to the location of a specific gene sequence.
8	**Protein structure prediction**	The methods for predicting protein structure from protein sequence. Methods such as secondary structure prediction and threading can help determine how a protein might fold, classifying it with other proteins that have similar topology.
9	**Protein structure property analysis**	Protein structures have many measurable properties. Protein structure validation tools are used to measure how well a structure model conforms to structural rules extracted from existing structures or chemical models compounds.

10	**Protein structure alignment and comparison**	Even when the two gene sequences aren’t apparently homologous, the structures of the proteins they encode can be similar. New tools for computing structural similarity are making it possible to detect distant homologies by comparing structures.

This thesis is concerned with phylogenetic analysis (No. 4 in Lesk’s taxonomy). In particular, it focuses on visualizing tree-structured data to help biologists by providing a better understanding of phylogenetic trees.

1.3 Phylogenetic Tree Visualization

Many real-world domains can be represented as node-link graphs or tree-structured data. These types of data are simple, powerful, and elegant abstractions that have broad applicability in computer science and many other fields. Tree-structured data occurs in many domains: file systems, parse trees, organisational hierarchies, and classification schemes of many kinds.

This thesis is about visualizing tree structured data. The impetus for the work described in this thesis is in the domain of phylogenetic classification, which is used by biologists to describe possible evolutionary relationships between species or individuals based on their DNA or protein sequences. Although the techniques described here were developed specifically for this domain, it could also be applied to other domains that use similar tree structures.

Phylogenetics is a field with a growing impact on a variety of science areas and can benefit greatly from the use of visualization techniques. It presents a number of visualization challenges. Biologists and geneticists use phylogenetic trees to represent the evolutionary interrelationships between collections of related species or genes. The discovery and analysis of those relationships may help in many practical applications such as drug discovery, forensics, disease control, and ecological modelling. As Hall (Hall, 2007) plainly stated ‘Evolution is important because not only does it provide a scientific answer to the question of human existence but it also forms a framework for understanding the biological diversity we observe around us’.

Biologists construct phylogenetic trees by examining the phenotypes or genotypes of a collection of organisms and attempting to infer the evolutionary process by which the organisms came to be. For example, a geneticist might obtain DNA sequence data from a range of species or from individuals within a population. Then, by comparing the sequences, he or she could infer how the sampled organisms might have evolved via a series of mutations, each caused by a change in the DNA sequence. This hypothesised evolutionary history is then represented as a “tree of life” showing how possible ancestors could have led to the current organisms.

1.4 Problem Statement

Biologists have devised a range of algorithms, based on strategies such as Maximum Likelihood and Maximum Parsimony (Lesk, 2008), for computing such phylogenetic trees. However, there is no “gold standard”; current practice dictates that several different methods be applied to the sequence data (Farach & Thorup, 1994). Different theories and methods about the evolutionary relationship of the same set of species also result in different phylogenetic trees. A similar situation arises when several species have evolved in close association (co-evolution); the biologist might be interested in understanding how the phylogenetic tree of one species compares with that of the co-evolved species. A fundamental problem in computational biology is to determine how much the two theories have in common. To a certain extent, this problem can be solved by visually comparing these phylogenetic trees to get a more complete picture of the relationships involved.

While some numerical measures are currently being used as a basis for tree comparison, these tasks usually require extensive visual inspection. Numerous applications have been developed in this field to address these issues to varying degrees. However, while phylogenetic inference methods are comparatively well developed, tools in this domain are characterised by a lack of effective visualization techniques. It is not uncommon for biologist to "(fall) back on paper, tape and highlighter pens" due to current deficiencies in phylogenetic visualization programs (Munzner et al., 2003).

The comparison between the trees derived from various experimental data is necessary in order to find the best model for a given set of species. This is where computer science comes into the picture by providing the algorithms and applications that will give such results. Such application should be interactive; that is, it should be capable of various tree manipulations, in order to maximise the discovery of knowledge about the given data.

It should address two major issues that have risen in currently available applications:

- how to efficiently and effectively compare phylogenetic trees, and
- how to visually present the results of the comparison

1.5 Research Objectives

This research aims to study on tree-structured data visualization. In more detail, it seeks to fulfil the following research objectives:

- To propose a new framework for tree structure visualization techniques that incorporates visual aspect of the tree comparison process and result.

- To design several algorithms that can automatically increase the degree of alignment between pairs of trees that will facilitate visualization of tree structure data.

- To devise several visual comparison techniques that can help to visualize the similarities and differences between different but related trees.

- To develop an interactive prototype tool for visual comparison of phylogenetic tree that will be used to evaluate and validate the proposed framework and visual comparison techniques. This prototype tool will be able to help biologists in understanding their phylogenetic tree data.

1.6 Research Approach

In order to achieve the research objectives as stated in Section 1.5, this research firstly explores various visualization tools and techniques especially the ones that deal with tree structure data.

After understanding the current situation, this research proposes a framework that focuses on finding different ways to compare phylogenetic trees. Several algorithms and visual comparison techniques are then proposed as an alternative to current comparison methods. This experimental comparison technique will be implemented and tested using a prototype tool named VCPT (Visual Comparison of Phylogenetic Trees). The evaluation in then carried out by conducting an evaluation on the proposed algorithm and visualization. The overall research methodology for this research and the proposed framework for visualizing phylogenetic trees will be presented in Chapter 3.

1.7 Scope and Limitation

The scope of this research focuses mostly on comparison between two fully resolves binary tree that have the same number of leaf nodes. Having only two trees to "consider" simplifies the process and certainly the graphical presentation.

The phylogenetic trees that are compared usually contain conceptually the same information. The information may be derived from different laboratories, or created using different techniques. Having these different ways in which the information can be changed, finding out the differences between the trees is important to understand. Tree comparison techniques are trying to examine these differences and produce a solution that will (ideally) best represent the evolution of the given species. During the development it is assumed that the compared trees are more similar than different. This makes sense, because having totally different trees representing totally different information serves no purpose. Although, the trees may have high degree of similarity, there is still a challenge in understanding the differences, as the trees may be very large in size. That is why the aim is, once the comparison occurs, to visually enhance the derived results.

1.8 Contributions

This thesis makes several contributions to the fields of tree structure visualization. In particular, it describes methods (algorithms and techniques) for visualizing pairs of similar trees. The main aim is to develop a new framework as a way of presenting the information so that it highlights both the common structures of the trees and their points of difference. The primary strategy is to display the trees "face-to-face" with leaf nodes aligned. In general, it is not possible to fully align leaf nodes for different structured trees. Note that these alignment processes do not change the internal structure of the trees, which means that the interrelationship between the nodes (species) remain intact. However aligning the nodes makes it easier to see which leaf nodes match in the two trees and provides a good starting point for further examination and understanding of the trees.

This thesis proposes several algorithms for automatically increasing the degree of alignment of leaf nodes: the minimum triplet difference (MTD) algorithm, the maximum branch similarity (MBS) algorithm, the all-but-n (ABn) algorithm and hybrid algorithm. The comparison for the minimum triplet difference is based on triplet (three leaf nodes) analysis between the two trees. The idea is that the structure of each tree can be represented as a set of triplets, which can be used to determine the difference. The maximum branch similarity algorithm arranges one tree so that the branches of each internal node have the largest number of leaf nodes in common with the corresponding branches of the equivalent node in the other tree. The all-but-n algorithm can be used to arrange trees to maximise leaf nodes alignment in a face-to-face display where the greatest agreement subtree (GAS) of the two trees is almost as large as the trees themselves (in other words, where the trees differ with respect to just a few nodes). The hybrid algorithm which refers to the combination of MTD and MBS algorithms is also introduced in order to obtain better results.

In addition, several visual comparison techniques based on colour, node spacing, elision, and branch shaping have been proposed to visually highlight the tree's similarities and differences. Colour is used to highlight the common structure between the two trees. Gaps can be inserted in order to increase the chance of aligning the nodes. Collapsed nodes can be used to enhance visibility especially in case of large trees as it enables focusing on specific parts of the tree while ignoring other parts. Branch shaping helps in highlighting the differences between the two trees by "pushing" the common ancestor so that the nodes that are not in agreement between the two trees can be connected in different ways. All these techniques will help the users to visualize both the common structure of the trees and their points of difference.

An interactive prototype tool for visualizing pairs of phylogenetic trees has been implemented and used as a vehicle for developing and evaluating these ideas. The prototype application also provides the users with additional tree manipulation tools and other useful GUI elements. The application is implemented in Java using the Swing components.

1.9 Thesis Structure

This thesis is organised in the following way:

Chapter 2 reviews previous research and discusses preliminary knowledge related to this thesis. It focuses on presenting a broad literature review on existing techniques that are currently available to address tree visualization issues. This chapter also includes a brief description of well known applications that deal with phylogenetic trees.

Chapter 3 presents the overall research methodology and the proposed framework for phylogenetic tree structure visualization. It looks into the current framework and discussed the detail of the proposed framework that will help in visualizing the similarities and differences between pairs of trees.

Chapter 4 describes the proposed algorithms and visual comparison techniques. It presents several algorithms for automatically increasing the degree of alignment between pair of trees. This chapter then deals with a variety of different visual comparison techniques proposed to highlight the trees similarities and differences.

Chapter 5 looks into the evaluation aspect of this work. It discusses the basis of the evaluation process, the tree alignment algorithm evaluation and the visualization evaluation that has been conducted in order to validate the proposed framework and the research ideas. For each of the evaluation processes, an explanation of how the evaluation was done is presented and this is followed by an outline of expected outcomes. It finally provides a discussion on the results derived from the evaluation process.

Finally, **Chapter 6** presents some conclusions and discusses suggestions of possible improvements for future work associated with the research. This chapter summarises the key concepts and presents what has been achieved and learned from the derived results.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews some of the previous research related to the work in this thesis. It focuses on existing algorithms, techniques and applications that are currently available to address tree visualization issues. The chapter is divided into six main sections. Section 2.1 discuss the current tree-structure visualization techniques and application. Section 2.2 to Section 2.5 looks into phylogenetic analysis issues such as the phylogenetic tree comparison, phylogenetic tree alignment algorithm and current phlogenetic tree application. Section 2.6 presents the summary for this chapter.

2.1 Tree-structured Visualization Techniques and Applications

Tree-structured data is a specific kind of graph that is very important in many applications. Trees are simple, powerful, and elegant abstractions that have broad applicability in computer science and many other fields. For example, in the domain of the World-Wide Web, nodes represent web pages and links represent hyperlinks, and in biology nodes represent species, and links represent evolutionary descent. In the case of the internet, nodes could represent routers and links would imply direct network connectivity.

The size of the tree to view is usually considered as a key issue in tree visualization. Large trees sometimes pose several difficult problems. If the number of nodes is large, it can easily compromise performance or even reach the limits of the viewing platform. Many techniques have been proposed to show such tree structures more effectively. **Treemap**, **cone tree**, **hyperbolic tree**, and **spacetree** are examples of techniques and applications that are being developed to tackle this issue. These techniques are considered to be some of the major contributors to this area.

2.1.1 Treemaps

The treemap (Shneiderman, 1992) is a two-dimensional visualization technique for displaying large amounts of hierarchically structured information. According to its author, the original motivation for treemaps was to have a better representation of the utilisation of disk space where there are multiple directory levels and nested sub-directories and files. The goal was to display the entire set of files, hoping that this will allow users to quickly recognise large files as candidates for deletion when the disk is full.

A treemap is formed by using a rectangular display area and recursively subdividing it based on the tree structure, alternating between horizontal and vertical subdivision, and filling the terminal rectangular regions with a colour that can be used to represent different types of data. Figure 2-1 shows a simple tree structure with its corresponding treemap. In this case the amount of disk usage (indicated by the number beside each node) determines the size of the partition. The larger the disk usage, the greater the partition. Often, each partition is coloured based on file type.

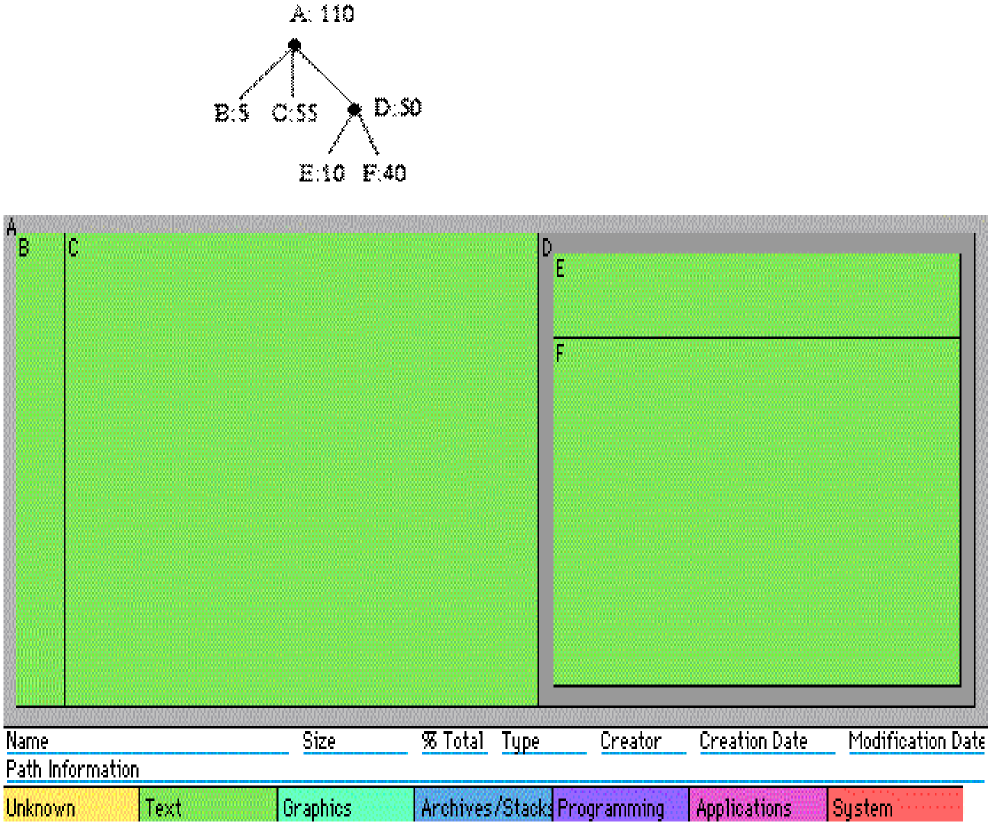
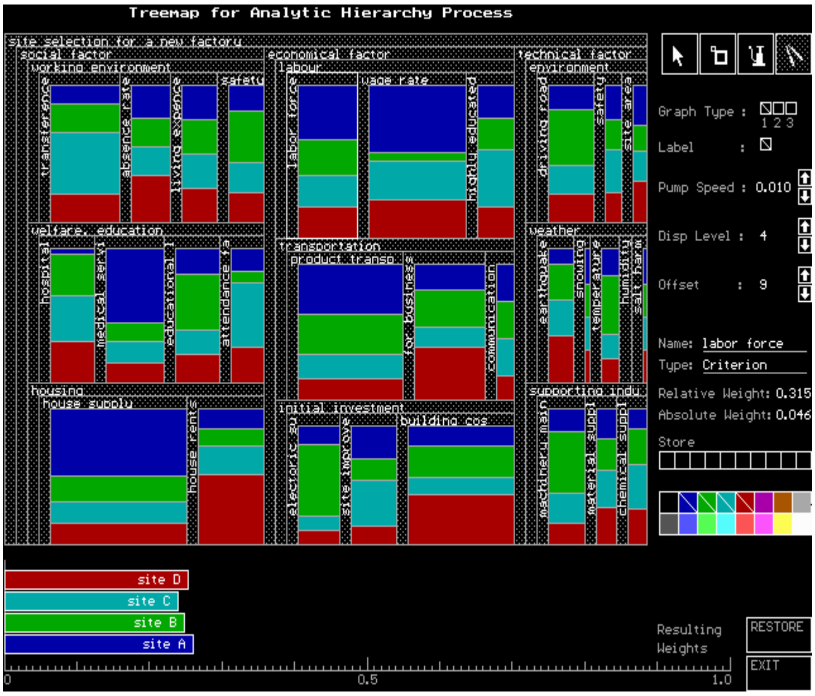


Figure 2-1 : An example of tree-structured data and its treemap representation (Shneiderman, 1992)

Treemap-based applications include many ways if filtering and sorting data. Later version of treemaps also included zooming, border variation, and dynamic query. For example clicking on a node might show the subtree of that node, or mousing over a node might show the details of the node in a pop-up window.

Treemaps have been further improved and reimplemented by others. For example, Asahi Toshiyuki, David Turo and Ben Shneiderman (Asahi et al., 1994) explored the use of treemaps to implement the Analytical Hierarchy Process in decision making (see Figure 2-2). The analytic hierarchy process, a decision-making method based upon division of problem spaces into hierarchies, is visualized through the use of treemaps, which packs large amounts of hierarchical information into small screen spaces. Apart from its traditional use for problem/information space visualization, the treemaps also serves as a potent visual tool for "what if" type analysis.



Graph Type : ☒☐☐ 1 2 3

Label : ☒

Pump Speed : 0.010

Disp Level : 4

Offset : 9

Name: labor force

Type: Criterion

Relative Weight: 0.315

Absolute Weight: 0.046

Store

Resulting Weights

Figure 2-2: Screen design for treemap representation of analytic hierarchy process (Asahi et al., 1994)

Van Wijk and Van de Wetering (Wijk & Wetering, 1999) developed **cushion treemaps** that use shading techniques on cushion-like 3-D mounds to make the tree structure more visible. Bruls, Van Vijk and Huizing (Bruls et al., 2000) also created a new layout strategy called “squarified treemaps” (Figure 2-3) that avoids high aspect ratio rectangles by using an alternative subdivision algorithm.

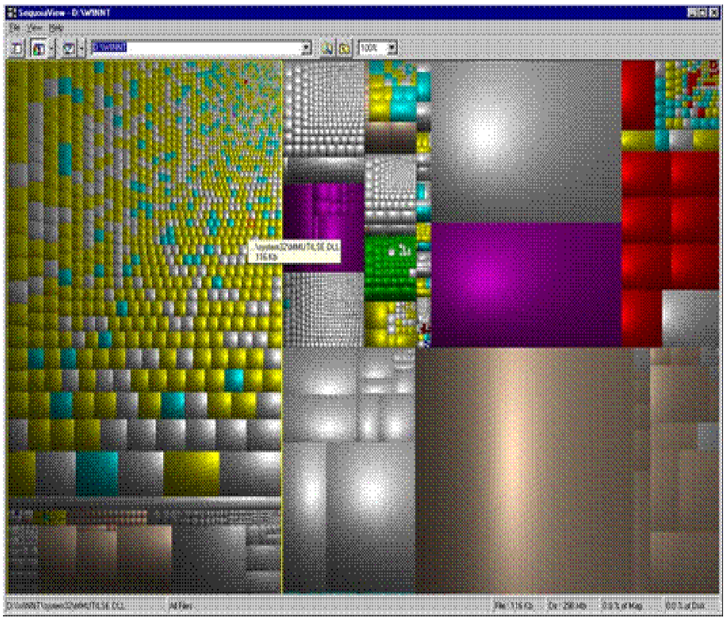
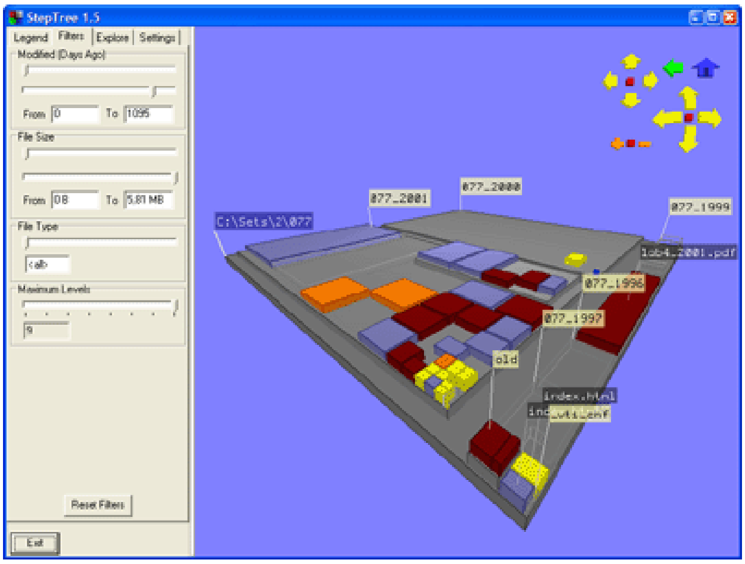


Figure 2-3: Screen shot of squarified cushion treemap (Bruls et al., 2000)

A group at Lulea University of Technology in Sweden developed a 3D treemap (Bladh et al., 2004) for file browsing that shows depth in the tree as the height of steps (see Figure 2-4). Their study with 20 participants have shown the benefits of 3D layout for a task by asking them to locate the deepest directory.



Modified (Days Ago) From 0 To 1095

File Size From 0B To 5.81 MB

File Type <all>

Maximum Levels 9

Figure 2-4: Screen capture of 3D treemaps (Bladh et al., 2004)

Renaud Blanch and Éric Lecolinet (Blanch & Lecolinet, 2007) developed another variation of treemaps known as Zoomable treemaps (ZTMs) to navigate the hierarchy with a multi-scale technique (Blanch & Lecolinet, 2007). ZTMs enhance classical treemaps by using the zoomable user interface (ZUI) paradigm to navigate efficiently in a hierarchical data space. Traditional ZUIs let users interact directly and continuously with the information space through panning and zooming (see Figure 2-5).

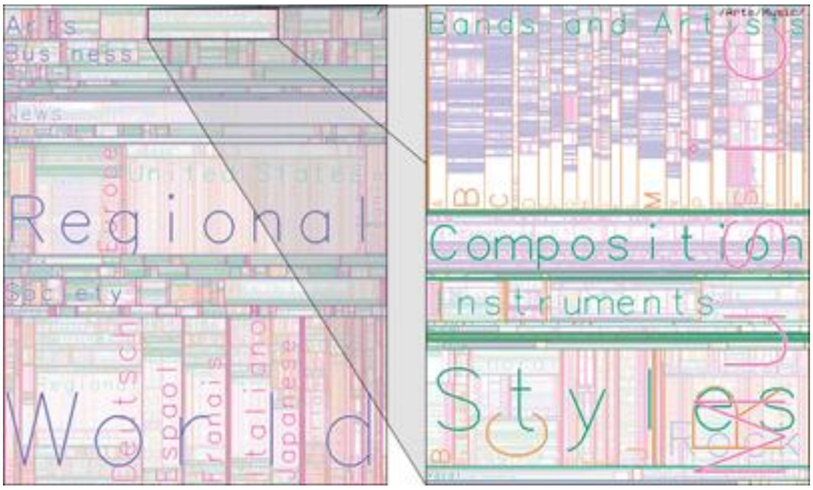


Figure 2-5: Example of Zoomable treemaps representation (Blanch & Lecolinet, 2007)

Apart from the few systems that are briefly discussed in this chapter, there is a lot more research that aims to improve treemaps. A detailed discussion of treemaps is beyond the scope of this thesis.

2.1.2 Arc Trees

Arc trees (Neumann et al., 2005) are a novel way of visualizing hierarchical and non-hierarchical relations within one interactive visualization. Arc trees utilise a concept similar to tree maps in presenting hierarchical relationships, in that they fold subtopics' rectangles into those of main topics (Figure 2-6). The difference is that tree maps use two- or three-dimensional space to draw and layout rectangles, while arc trees use one linear dimension. The other dimension is utilised to present additional, supplementary links and relations using arcs. Different coloured rectangles are employed to denote the parent-child relation of topics, and are usually coloured in shades of the main topic's colour. Arcs of different heights, colours, and thicknesses are then used to represent different link types, subtypes, or kinds of relations.

Logical minds find arc trees easy to understand because they are ideal to denote, read, and trace complex relations and subjects. However, they are not intuitive and are difficult to understand by most users. In addition, by using only one dimension, arc trees tend to grow lengthy and cumbersome when the numbers of nodes grow too large.

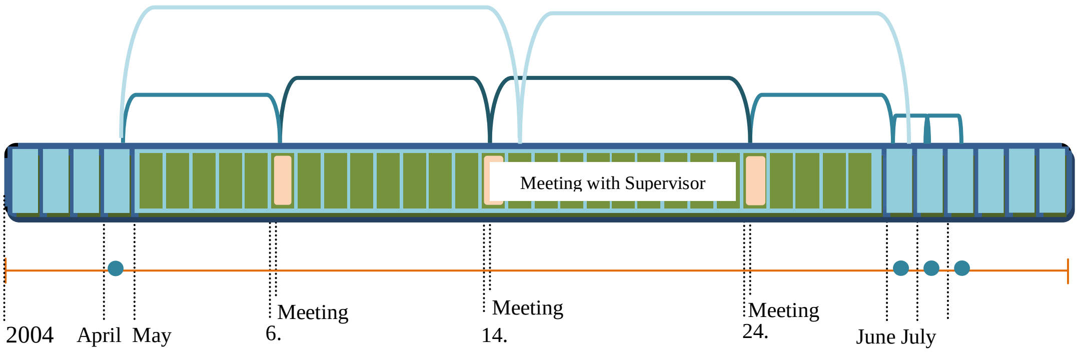


Figure 2-6: Example of an arc tree (Neumann et al. 2005)

2.1.3 Cone Trees

The cone tree, developed by Rebertson, Mackinlay, and Card (Robertson et al., 1991) is considered to be one of the best known 3D graph (in this case, tree) layout techniques in information visualization (Herman et al., 2000). The tree is presented in 3D to maximise effective use of available screen space and enable visualization of the whole structure (Robertson et al., 1991). The root of the tree is located at the apex of a cone and all its children are arranged around the circular base of the cone in 3D. Users interact with the cone trees by selecting and rotating certain nodes on the screen. The cones themselves are transparent. So users are able to see what is behind them.

Figure 2-7 shows a snapshot of a cone tree. The root of the hierarchy is placed at the top with its children placed evenly spaced along its base. A common application of the cone tree includes representation of directory structures, organizational charts and companies' operating plans.

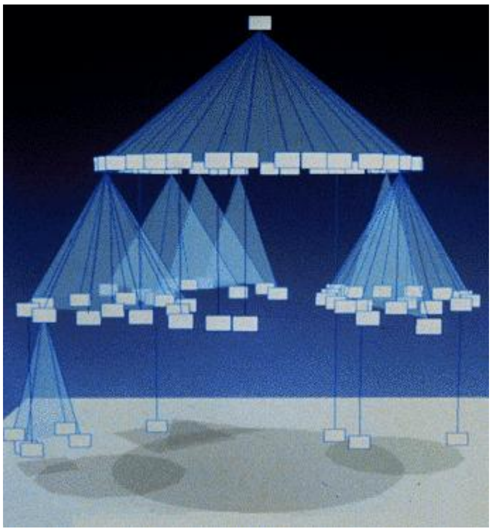


Figure 2-7: Example of a cone-tree (Robertson et al., 1991)

Compared to a 2D tree structure, more nodes can be displayed at once using a cone tree, while still allowing users to understand the tree structure. Any subtrees can be hidden or shown using the “prune” and “grow” commands. When a user selects a node, that node will be rotated so that it is displayed in the front. Because of the way nodes are placed next to each other, the names of the nodes are mostly hidden. Properties of a node can be shown by clicking on it.

Jeong and Pang (Jeong & Pang, 1998) presented the reconfigurable disc tree (Figure 2-8). Instead of using a cone, they use a disc to represent the nodes. Each child node is a disc itself, placed underneath its parent. Using the disc, they claim to reduce occlusion and increased the number of nodes that could be displayed effectively.

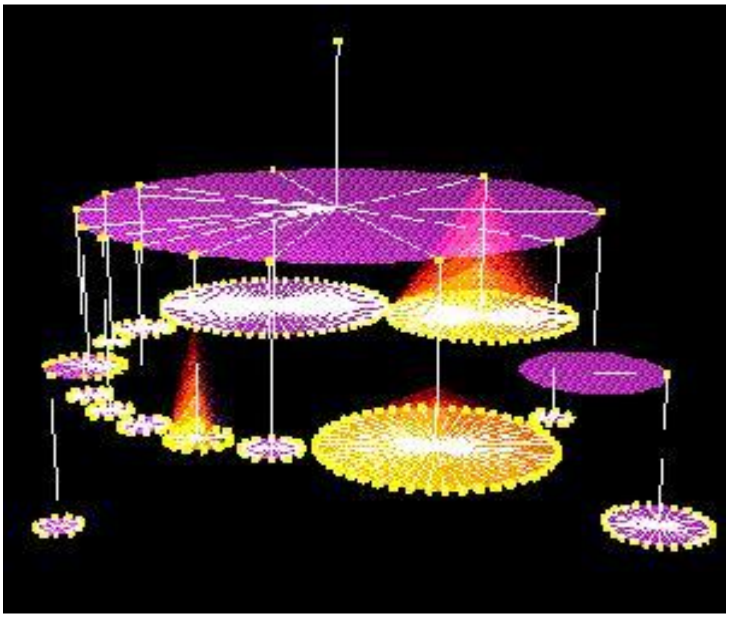


Figure 2-8: Example of reconfigurable disc tree (Jeong & Pang, 1998)

2.1.4 Hyperbolic Tree

The hyperbolic tree (Lamping & Rao, 1995), also known as star tree, is a tree that is laid out as a radial view in hyperbolic space. This view is then mapped to an Euclidean plane so that an arbitrarily large tree fits within an oval-shaped area on the screen (see Figure 2-9). The root is placed in the centre, and its children are fanned outward. Any part of the tree can be moved to the centre with a simple mouse-click or mouse-drag.

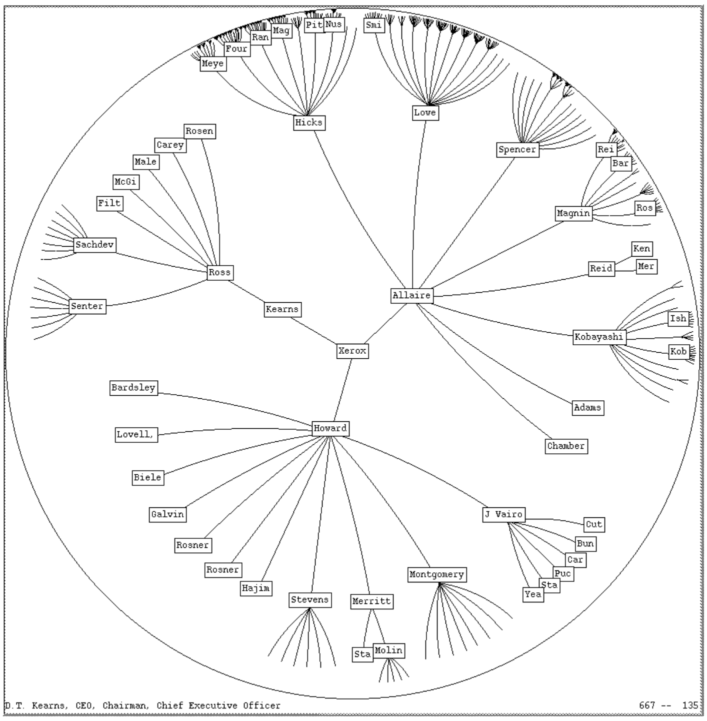


Figure 2-9: Example of hyperbolic tree (Lamping & Rao, 1995)

A hyperbolic tree follows the same basic principles as a common tree, with a link between a parent and a child. Users can easily grasp the hierarchical structure. However, since nodes are spread out evenly, users may find it hard in telling exactly how balanced or unbalanced a tree is.