Extending Taxonomic Visualisation to Incorporate Synonymy and Structural Markers

Martin Graham
School of Computing
Napier University
10 Colinton Road
Edinburgh
EH10 5DT
UK
Tel: +44 131 455 2749
Email: m.graham@napier.ac.uk

Jessie Kennedy
School of Computing
Napier University
10 Colinton Road
Edinburgh
EH10 5DT
UK
Tel: +44 131 455 2772
Email: j.kennedy@napier.ac.uk

Running title: Extending Taxonomic Visualisations

Research Support : SEEK (NSF Grant award 0225676)

Abstract

The visualisation of taxonomic hierarchies has evolved from indented lists of names to techniques that can display thousands of nodes and onto hundreds of thousands of nodes over multiple taxonomies. However, challenges remain within multiple hierarchy visualisation, and for taxonomic hierarchy visualisation in particular. Firstly, at present, there is no support for handling specific taxonomic information such as synonymy, with current visualisations matching solely on names. Synonymy is extremely important as it reflects expert opinion on the compatibility of data held in separate taxonomies, and is needed to produce an accurate picture of taxonomic overlap. Also, current techniques for exploring large hierarchies find it difficult to convey internal re-organisations between hierarchies, with most systems showing only addition, removal or wide-ranging fragmentation of information between taxonomies. Finding the source of changes that have occurred within an existing structure is currently only achievable through exhaustive drill-down exploration. This paper describes work that tackles these problems, incorporating synonymy information into a model for multiple hierarchy visualisation of large taxonomies, and also detailing techniques that aid navigation for discovering structural re-organisations between hierarchies and for revealing information about nodes that lie below the effective display resolution of the hierarchy layout. Two examples on real taxonomic data sets are annotated to show the effectiveness of these techniques in operation.

Keywords

Tree layout, multiple tree visualisation, focus & context, drill-down

Introduction

Taxonomic data sets are natural candidates for exploration through Information Visualisation (IV) techniques given their hierarchical tree-like nature. Recognising this point, recent IV research [*1-5*] has focused on communicating relationships between multiple, related taxonomies. However to be useful to taxonomists working with real-world data, taxonomy visualisations must extend themselves beyond matching solely on names and include relationships that represent expert taxonomic opinion. Large taxonomies in the order of hundreds of thousands of nodes present their own problems too, in terms of scale and information discovery within such large data sets.

This paper introduces a basic description of taxonomy, and then discusses recent work in visualising taxonomic data. Issues with current techniques are raised which are then addressed by our improved techniques for visualising multiple, large, overlapping taxonomies. Examples are then given demonstrating how such techniques can help alleviate difficulties caused by manipulating and interrogating hierarchies of 100,000's of nodes and also help correctly interpret overlap between taxonomies with associated synonymy information.

Taxonomy

Taxonomy in general is the science of classification; recursively grouping objects on the basis of their properties into broader or narrower categories, depending on whether the process is applied bottom-up or top-down. Biological, or *Linnaean*, taxonomy [*6*] is this process practiced on organisms, wherein various specimens are grouped together according to perceived similarities. Taxonomies can thus be considered as sets of sets (see Figure 1), where each set (*taxon*) is a group composed of further constituent *taxa* or organisms, allowing basic comparison of taxa in alternative taxonomies by their set membership.

The taxonomic process and associated nomenclature issues can result in the creation of many alternative taxonomies constructed over the same data set. To accurately represent this multiplicity of taxonomic hierarchies, extensive research has been undertaken in the development of data models and database systems [*7-9*]. However, understanding the full complexity of the relationships contained in these databases through a textual medium is difficult, as it requires constant cross-referencing, checking, and backtracking across the individual hierarchies that form the overall data set. Our previous work [*1-3*] established that IV techniques can be used to explore multiple biological taxonomies of the order of thousands of nodes as found in these database systems, which is representative of the size and context of individual taxonomists' work. However, there are now several large international biodiversity initiatives such as SEEK, GBIF and Species 2000 [*10-12*] which require the integration of the taxonomic data found in multiple disparate database systems to produce taxonomies of the scale of hundreds of thousands of nodes or greater; their eventual goal being to provide access to information on the known species of the world.

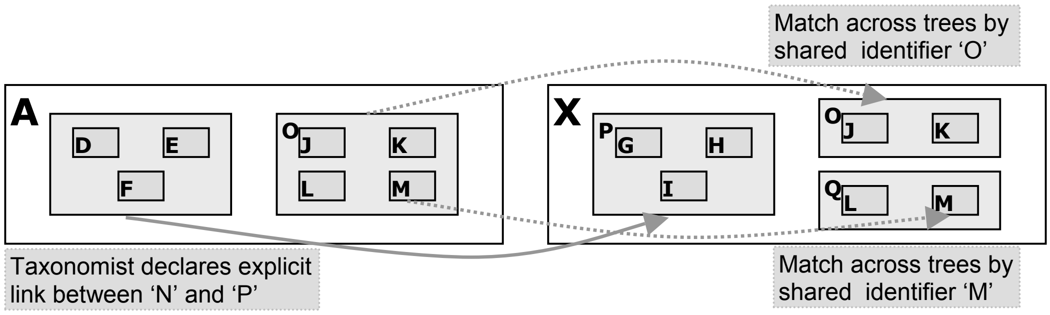

Match across trees by shared identifier 'O'
A
D
E
F
O
J
K
L
M
X
P
G
H
I
O
J
K
Q
L
M
Taxonomist declares explicit link between 'N' and 'P'
Match across trees by shared identifier 'M'

Figure 1. Matching by name and by expert-defined synonym relationship.

During revision of a taxonomy, taxonomists may also assert explicit equivalence relationships known as *synonyms* [*13*, p. 84-87] between the taxa they define and taxa described in existing taxonomies, or even between taxa in these other taxonomies. These relationships are based on opinion rather than simple set comparison. For example, in Figure 1, the sets *N* and *P* have been declared to be explicitly equivalent, even though they share no nodes with equivalent identifiers, and thus *N* is said to be synonymous with *P*. These relationships cannot be deduced simply from the data stored in taxonomic databases, but are founded on a deep, specialised understanding of the taxonomic group in question and are therefore extremely valuable. Thus, it is important these relationships are conserved and recognised in a visualisation to allow an accurate comparison of taxa.

Linnaean taxonomies are organised into strict layers known as *ranks* that differentiate them from other hierarchical structures such as file systems. Ranks such as family, genus, species etc form a discrete, ordered system that apply to all biological taxonomies, so that any two taxa sharing the same rank are said to reside at the same level in the Linnean taxonomic system, regardless of their placement within any individual taxonomy. The use of certain ranks is compulsory if they fall within the scope of a taxonomy under construction; however most ranks are optional, and taxonomists are at liberty to choose which of these are included in their classifications. Therefore mechanisms for comparing or displaying biological taxonomies cannot rely on equivalent sets of ranks being used consistently across different taxonomies, or even within the same taxonomy, though when comparing taxonomies it is essential to compare taxa by these ranks and not simply by depth from the root in any particular taxonomy.

Figure 2 supplies a simplified example, where Taxonomy A and B each cover a different range and selection of ranks. Taxonomy A spans the family to sub-genus ranks, but omits the tribe rank that Taxonomy B populates. Taxonomy B itself is inconsistent about its use of this rank, and does not go as deep as the sub-genus rank. The diagram also demonstrates that is it ranks rather than the distance of a taxon from its particular root that gives the measurement of depth in the Linnean system. Though the leaf nodes in Taxonomy B are either two or three links down from their root, they are all embedded within the genus rank, as are the immediate child nodes of the root of Taxonomy A. This system of rigid layers also contrasts markedly with phylogenetic hierarchies, in which depth is measured as a continuous rather than a discrete function, computed from similarity measures between related nodes. Phylogenetic trees also tend to be strictly binary; it is rare that an internal node splits into more than two sub-branches, whereas branching in Linnean taxonomy is *n*-ary in nature.

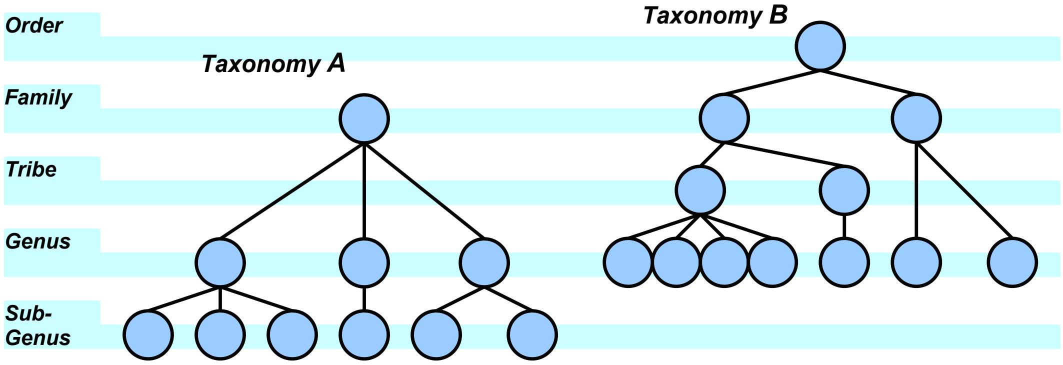


Figure 2. Taxonomies may not always use ranks consistently, even internally.

Individual taxonomists typically classify around 1,000 specimens when creating a taxonomic hierarchy, resulting in a hierarchy of up to seven ranks with up to 1,500 nodes (taxa). The taxonomists may then want to compare this classification with several other similarly sized taxonomies of the same data, seeking out differences of opinion regarding the membership or placement of taxa in the alternative taxonomies. By contrast, the large biodiversity projects are attempting to integrate existing taxonomies (several thousand), which vary from an individual taxonomist's hierarchy to large structures of 100,000+ nodes [*10-12*] generated by the previous integration of smaller taxonomies, into one large encompassing hierarchy. In this context, when integrating data, taxonomists are interested in finding taxa (sub-trees) that are similar. The similarity may be measured in terms of their composition by set comparison of taxa with the same name, or through synonymy. These similarities must be detected in taxonomies of varying scales spanning different ranks of taxonomic groups.

In conclusion, taxonomy is an intensive field of study, whose myriad alternative classifications and inclusion of expert knowledge can lead to a veritable spider's web of relations. In the next section, related work in IV on hierarchy visualisation is described, with specific attention paid to techniques that can manipulate and display multiple hierarchies to alleviate some of the complexity of understanding large, related taxonomic structures.

Related Work

Tree visualisation has been a cornerstone of information visualisation research since the influential Cone Tree [*14*] and TreeMap [*15*] visualisations of the early 1990s. Subsequently, tree visualisation development has concentrated on both data-based problems such as improving scalability [*16*], and user-centred perceptual and navigation issues such as focus + context effects [*17*] and animation of roving viewpoints [*18*]. However, few applications either cope with trees composed of 100,000's of nodes, or allow display of and detailed comparison between multiple trees.

Of those that do allow comparison of trees, some are limited in scale by the specific technique used to display the trees, such as Amenta and Klingner's node-link visualisation of tree sets [*19*], or by attempting to always display trees in their entirety such as our previous multiple classification prototype [*1*]. In these cases, trees of above a few thousand nodes become illegible. Another approach, Furnas and Zacks' Multitree visualisation [*20*], presented a browsing focus within an aggregated graph visualisation of overlapping hierarchies. However, even though graph approaches are more space efficient, visually fusing multiple hierarchies has been shown to be more confusing to users, both conceptually and visually, than displaying the hierarchies separately [*3, 21, 22*], due to effects such as visual edge crossings and disentangling the separate hierarchies to follow intra-hierarchy paths. It was for this reason our previous work concentrated on a visualisation that presented taxonomies as separate but linked entities.

Two visualisations developed to handle tree comparison and display on the scale of 100,000's of nodes are Munzner et al's TreeJuxtaposer [*4*] and Spenke's InfoZoom [*5*] applications. The InfoZoom approach is based on a powerful spreadsheet model in which ranks map to the rows of the spreadsheet, and nodes map to the appropriately positioned cells. Higher nodes occupy a number of multiple, neighbouring cells in proportion to the size of their descendent set, and the spreadsheet subsequently merges these repeated cells into one continuous area to allow display of name labels. InfoZoom allows various operations on the data to be performed, but by taking the approach of a spreadsheet it is restricted to the standard interaction techniques of such an application. For example, it lacks the ability to brush entire sub-trees for direct visual comparison of group distributions, and its zooming mechanism is filter-based rather than a focus+context technique, which meant that visual context was lost when nodes were selected. In taxonomy, it is often advantageous to view and contrast many taxonomies simultaneously in context, e.g., in order to comprehend a process of continual re-organisation or change, as was the case with the smaller data sets we handled previously [*1*]. The TreeJuxtaposer application is able to handle up to four trees simultaneously, and the 'guaranteed visibility' technique it uses allows all selected areas of a hierarchy to be displayed or at least represented at all times. It does this by using a focus+context technique in which selected nodes formed the expanded, visible focus and the rest of the tree forming the necessarily much-reduced context. However, this technique also leads to overcrowding and occlusion in the display, with labels being particularly difficult to display if they are associated with internal nodes.

Both Munzner and Spenke's visualisations allow basic similarities and differences to be found between hierarchies i.e. addition or deletion of nodes, or the agglomeration or break-up of selected node groups across hierarchies. However, neither can reveal much in the way of internally rearranged data within selections that span multiple hierarchies, and neither supports the use of synonymy within their data models. Indeed, no tree comparison visualisation yet allows synonymous relationships or aids in the visual discovery of internal structural rearrangements.

Thus, we find that some existing multiple hierarchy visualisations scale to the necessary size and allow comparison, but don't include the explicitly declared relationships necessary for synonymy or aid discovery of structural re-organisations beyond addition or deletion of nodes. In the following sections, we describe our multiple hierarchy visualisation that addresses these challenges, allowing the inclusion of synonymy data, detailed interaction with the data sets and permits changes in structure between existing nodes to be visualised and explored.

Design

Our multiple hierarchy visualisation is based on previous work [*3*], a sample of which is shown in Figure 3. Trees were vertically stacked one above the other and a space-efficient approach used to draw the individual trees. Taxa were represented as groups of nodes displayed beneath their parent nodes, which in turn are laid out underneath their parent nodes in a bottom-up process until the root of the tree is reached. This 'abutment' method of indicating node relationships removes the need to explicitly display numerous link representations for the parent-child relationships, but still allows depth information to be conveyed clearly, which is often a problem in nested tree representations such as Treemaps [*15*]. Similar abutment layout methods can be seen in Stasko and Zhang's radial space-filling tree visualisation [*23*] and Sifer's filter for multiple classifications [*24*].

A user makes a selection by mouse clicking on a leaf node or sub-tree in one of the displayed classifications. All the nodes in this group are then coloured, and this colouring applied to wherever the nodes occurred across the other classifications, thus giving an intuitive mechanism for viewing distributions across classifications. Multiple selections can be made, each individual selection being marked with a different hue.

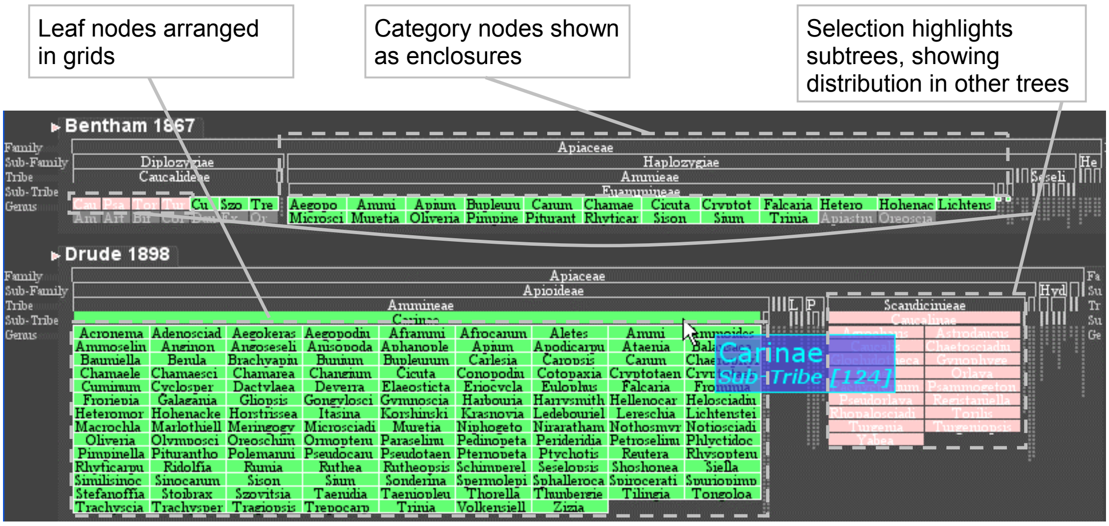


Figure 3. Our previous visualisation, which handled taxonomies of a few thousand nodes.

Our previous work had experimented mainly with two data sets, specifically the *Apiaceae* and *Globba* taxonomic data sets [*25*]. The *Apiaceae* data consists of eight different classifications of the family *Apiaceae*. Each classification consists of several hundred genera, where each classification is a, sometimes radical, revision and restructuring of previous classifications. The *Globba* data set consisted of four snapshots of a specimen level classification being organised from scratch by a single taxonomist, which consisted of several thousand plant specimens. The previous visualisation technique had been developed to cope with data sets of this size, however the *ITIS* (Integrated Taxonomic Information System) and *Moss* data sets now under consideration are of a much larger scale and contain detailed synonymy information.

The *Moss* data [*26*] is analogous to the *Apiaceae* data set, in that it has relatively few taxa but covers fifteen classifications and there are distinct revisions and re-categorisations of the data between the different classifications. It is also heavily populated with synonymy data, where a taxonomist has stated that they consider taxa in their classification to have specific relationships with taxa in other taxonomies. Classifications of this size are typical of taxonomists whose tasks include finding overlap and correlations between their classification and other related work. In detailed taxonomies matching by name alone is often rendered useless by simple changes to spelling of taxon names, or by the reclassification of species/specimens between genera. In the latter case, reclassification leads to a binomial name changing to reflect the genus it resides in, and so synonymy data is the only way to reliably link the taxa.

The *ITIS* data set [*27*] is formed from seven annual snapshots of the *ITIS* database between 1998-2004 inclusively, with each revision holding between 180,000 and 250,000 taxa. The challenge here is one of scale, though there are distinct re-organisations of taxa between revisions. There is also some synonymy information, though it forms a much smaller proportion of the overall data set than in the *Moss* data set, and has a subtly different meaning. Here, synonyms signify the substitution of one name with another, e.g. 'Name B in use from 2000 and onwards is the same as Name A in all previous revisions'. Taxonomic data providers such as data aggregators explore this type of data set to find when restructuring took place for particular taxa, to find taxa missing from a particular revision, or discover rates of growth of particular branches of the data set.

To encompass these larger tree sizes, the synonymy data, and to enable the discovery and exploration of non-trivial structural reorganisations between trees, modifications were necessary to both the underlying data model and to the visualisation techniques. The *ITIS* and *Moss* data sets are henceforth used to explore the effectiveness of the visualisation.

Synonymy Data Model

The data model describing the taxonomies is based on the visualisation data model described in Raguenaud *et al* [*28*]. In this model, one 'name' object is assigned per unique taxon name, representing all the non-classification specific information that can be associated with a particular name. Within a name object, sub-objects (classification objects) are generated on a per-classification basis, to hold information describing how different taxonomists use that name in their classification. Effectively, a classification object maps to one taxonomist's use of a particular name, equivalent to the taxonomic construct of a *concept*. Mapping one entire classification will thus produce a collection of classification objects across an equal number of name objects. By defining child and parent pointers between classification objects that belong to the same taxonomy we can build up the classifications structures themselves. In this way, multiple classifications are described separately and are only connected via the name objects, which act as containers that collect together the appropriate group of classification objects. Perhaps a useful analogy is to think of the classifications as different underground or metro lines. The name nodes act as "transfer stations" that allow traversal between one "line" (classification) and onto another.

To extend the model to hold synonyms, each classification object is assigned a further list of pointers to other classification objects that belong to different classifications, to which taxonomists have decided that there exists a type of direct relationship. These relationships can be seen to 'short-circuit' the paths described by the classification and name relationships. Figure 4 shows a simple example of three classifications straddling the same set of name nodes. It can be seen that the intra-classification links (shown as the grey-scaled lines) enable movement between nodes in the same classification, and traversing the inter-classification links (the short black lines) allows movement between classifications at the same name node. A synonymy relationship however allows a direct link to be engineered between a pair of classification objects representing both different classifications and names.

In Figure 4, there is a synonymous relationship indicated by the dashed line. Here the green classification is stating that one of the names it uses is synonymous with a different name in the orange classification. A new analogy of these different types of movement within the multiple classification graph is to consider intra-classification traversal as 'vertical' movement, and inter-classification movement as 'horizontal' movement. In this sense, a synonym relationship is a 'diagonal' movement, linking one particular name and classification combination with a different name and different classification.

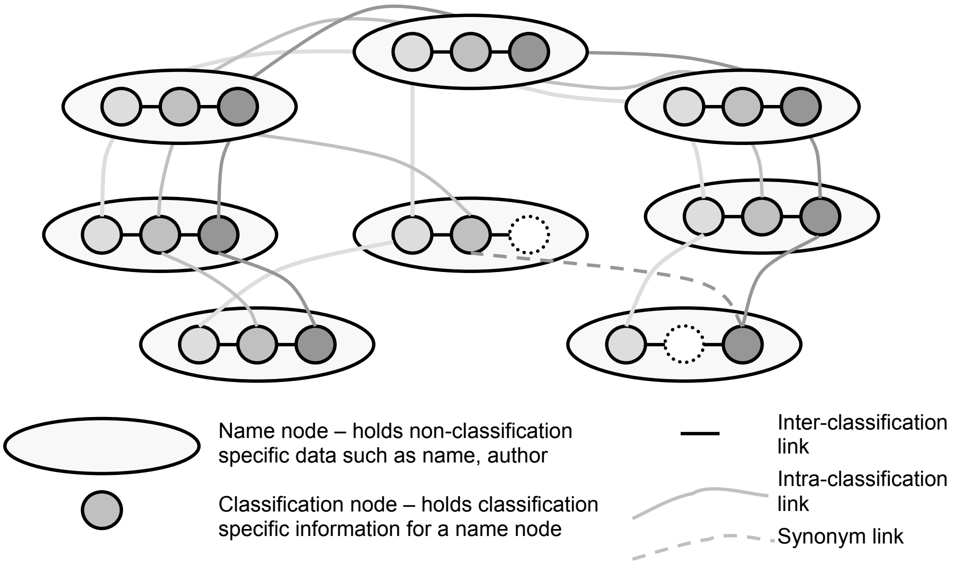

Name node – holds non-classification specific data such as name, author
Classification node – holds classification specific information for a name node
Inter-classification link
Intra-classification link
Synonym link

Figure 4. The different types of relationships possible in a multiple hierarchy classification.

This approach to modelling the taxonomic data gives access to ready-made hierarchies within a larger graph structure, eliminating the problem of extracting individual taxonomic classifications from the overall graph. Furthermore, as speed is an important factor for an interactive visualisation, having all the classifications connected together but easily distinguished makes inter-classification operations extremely efficient compared to the case for a general graph. Searching for matching names across classifications is unnecessary, as once a taxon node is known, the collection of classification objects effectively provide random access to the positions of that name across all classifications.

Layout

Previously, as shown in Figure 3, the layout technique was a bottom-up process limited to dealing with trees of a few thousand nodes in which groups of leaf nodes were laid out in 2D grids rather than in the traditional linear fashion. However, the size of the new data sets such as the multiple *ITIS* revisions meant that even intermediate-level nodes were now shrunk to sub-pixel size when we tried to display the data in full.

Our solution to this problem of scale, rather than attempt to draw the tree down to single-pixel dimensions as in Munzner *et* al [*4*], was to halt the layout algorithm at a predefined limit and subsequently provide a summary of interesting information that lies beneath this cut-off point. The first approach involved limiting the layout to displaying only a certain number of ranks, say five or seven, as this is generally the number of ranks that an individual taxonomist is interested in investigating. However, it was found that in deep, narrow subtrees, seven ranks would encompass only a couple of hundred nodes, whereas in flatter, broader subtrees, a smaller number of ranks would still encompass enough nodes to overwhelm the space-dividing layout algorithm. Also, inconsistent use of ranks meant that whilst one part of the display encompassing five ranks would yield a pleasing display, there might only be two of those five ranks present in another portion of the display.

This led to a decision to layout the hierarchies top-down from the current node of interest, and to stop drilling down into any particular branch when the size of the nodes dropped to below a threshold size. This prevents both the situation of trying to draw hundreds of indiscernible sub-pixel-sized nodes and also the event where the layout stops prematurely, when there is space left to draw in more nodes at a distinguishable level of detail.

Each individual node is depicted as a rectangle whose display area is used to convey information beyond simply an indication of presence in the hierarchy. Most importantly, to compensate for not drawing the trees in their entirety each node representation contains a summary of the user-selected nodes that lie beneath it. The top half of the node representation is coloured if it has been chosen by a previous selection operation, either through direct selection, or via a shared name or a synonymous relationship. For example, Figure 5 shows that *Animalia* has been selected as the top half of the bar is highlighted as opposed to the darker shade in the bottom right-hand section of the bar.

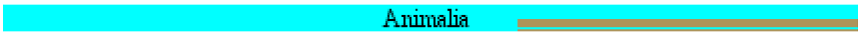


Figure 5. Information is conveyed within node representations. The horizontal proportions of shading in the lower half of the *Animalia* node reflects the ratio of descendents that are selected or not.

The bottom half of the node representation is used to draw a small bar chart that acts as a summary of the taxon's descendants' selection status. For instance, if any taxa in the sub-tree under *Animalia* had been selected then a coloured mark would be displayed within the bottom half of the *Animalia* node. When space permits, we can go beyond a simple marker of presence, and fill the bar in proportion to the amount of selected and unselected nodes in the subtree, e.g. Figure 5 shows that *Animalia* contains a proportion of about 50% selected taxa, as roughly half of the lower portion of the bar is shaded. When multiple selections have been made, the proportions of colours within the bar communicate the relative proportions of descendant nodes chosen by different selections within that sub-tree, as shown in Figure 10(a).

The node representation is also used to display as much of the associated name information as possible. Linnean taxonomy uses a binomial nomenclature below the genus rank, according to which a species epithet is composed of the genus name followed by the species-specific name. As displaying a hundred or so species within the same genus would lead to the replication of the genus name a hundred times, we follow the taxonomic convention of reducing the genus name to a single letter so that more of the species specific nomenclature is revealed. Below species level a trinomial nomenclature is used (genus-species-subspecies), and we compress the name in a similar manner.

The hierarchies are arranged on-screen as in Figure 6. Each individual hierarchy is composed of a tab indicating the node and rank of the current anchor node of that hierarchy, underneath which child nodes are laid out recursively, in a manner so that as far as possible nodes of equivalent rank within the same hierarchy are placed on the same vertical coordinate as each other. The final groups of nodes that can be displayed are laid out in a space-filling grid pattern. Icons for closing/opening individual hierarchies and navigating towards the root of each hierarchy are also provided in the upper left-hand corner of each tree.

Layout Proportions

Initially in the layout we apportioned screen space in ratio to subtree size, but this led to large subtrees dominating smaller subtrees to the point of visual exclusion. For example, at the top level of the most recent ITIS revision, there are four kingdoms: *Animalia, Fungi, Monera* and *Plantae*; *Animalia* includes over 215,000 taxa whilst *Monera* has just over 1,400, a ratio of over 150:1. If these proportions were calculated for all kingdoms and used to meter out screen space, *Monera* would receive roughly only a 5 pixel-wide display space for every 1000 pixels of width available. We found that allocating space according to the logarithm of the set size gave smaller groupings a more reasonable proportion of display space. For *Monera*, it would mean receiving a 180 pixel-wide segment of screen space in which it could be displayed. As an example, consider that two nodes representing 100,000 and 100 taxa respectively are to be displayed. Instead of dividing up the available space in the proportion of 100000:100, the space is divided up in the ratio of log100000: log100. Note that the logarithm base is irrelevant to this ratio, as dividing two logarithms of the same base results in a logarithm of the dividend number with the divisor number as the base i.e. $\log_A 100000$ / $\log_A 100$ = $\log_{100} 100000$ = 2.5. Thus, the taxa which ultimately holds 100,000 nodes is given two and a half times the screen space of the taxa that is an ancestor of 100 other nodes. Experience has shown that the larger groups are not overly compressed as taxonomists attempt to keep each individual taxon's number of children at a manageable size, so tend to differentiate these larger sets down through a greater number of ranks.

Navigation and Selection

Basic interaction involves brushing, navigating and selecting nodes within the hierarchies. Selecting a node with the mouse performs navigation or selection dependent on whether the left or right mouse button was pressed. Left-clicking the mouse will re-root a hierarchy at the selected node, whilst using the right-button will select the sub-tree rooted at the selected node and then iterate down the sub-tree, marking equivalent nodes in other hierarchies through their name and/or synonym relationships. Thus, the distribution of a sub-tree in one classification can be seen across the other classifications. A mouse brushing behaviour is also incorporated into the visualisation, such that brushing a node will temporarily highlight associated nodes in a similar manner to the selection process described above. A tool tip containing information on the node being probed by the mouse pointer is also displayed as seen in Figure 10(a).

We can often re-root the hierarchies upon a navigation or selection operation by simply re-anchoring all the hierarchies to start under the same name. However, if a name does not occur consistently across all the other hierarchies then we need to take a more involved approach. This consists of taking all the nodes in a selected sub-tree, matching them across to the other hierarchies, and then finding the deepest subtree that encompasses all of these nodes in any particular hierarchy, also known as the least common ancestor (LCA). The process is shown diagrammatically in Figure 7.

We calculate the LCA by building up a partial 'skeletal' tree for each hierarchy. To do this, we add every matched node and its ancestors to the skeletal tree, which results in a tree with the same root as the overall hierarchy, but with only selected nodes as leaves. Maintaining a hashtable of nodes in the skeletal tree during this process quickly culls unnecessary traversal operations, reducing the complexity of the operation from O(n log n) to O(n) for a given tree size. For instance, if consulting the hashtable tells us node X is already in the skeletal tree, we don't need to add it or any of its ancestors. Then, to find the LCA, we traverse down from the root, and the LCA is either the point at which a branch first occurs (where a node has more than one child), or where a selected node is first encountered.

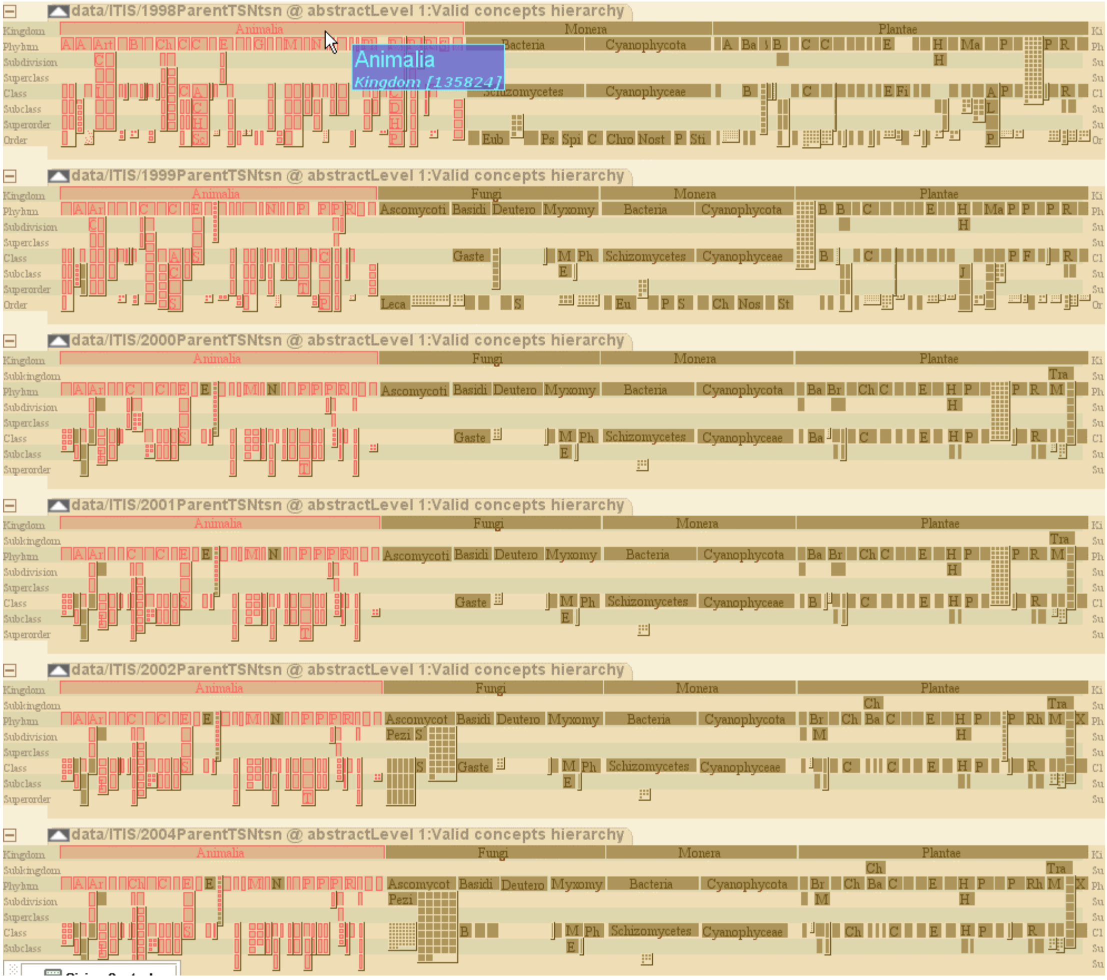
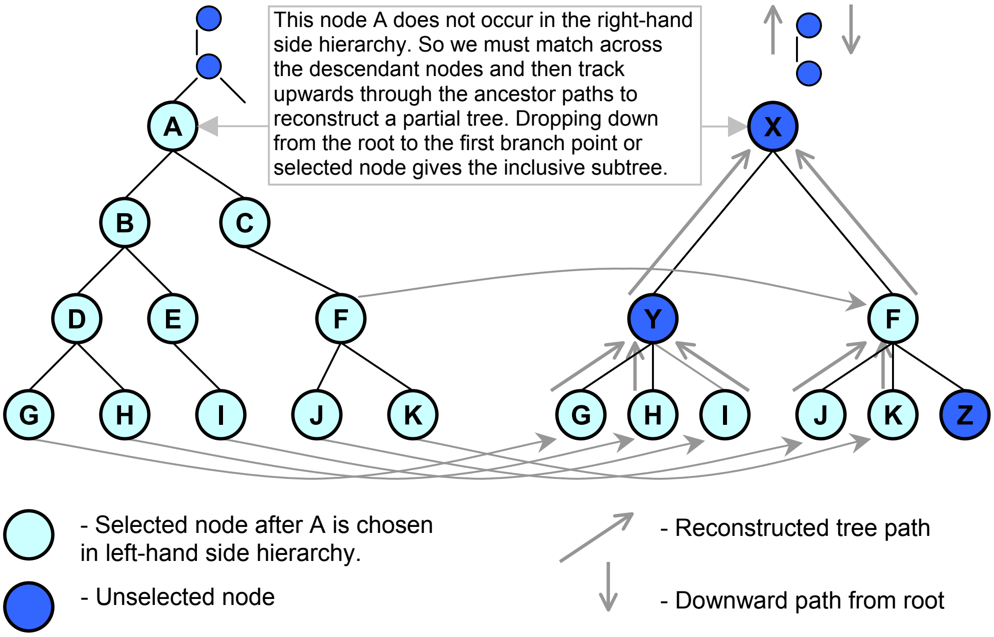

data/ITIS/1998ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
Animalia
Kingdom [135824]
Monera
Plantae
Bacteria
Cyanophycota
Schizomycetes
Cyanophyceae
data/ITIS/1999ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
Fungi
Ascomycoti
Basidi
Deutero
Myxomy
data/ITIS/2000ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
data/ITIS/2001ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
data/ITIS/2002ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
data/ITIS/2004ParentTSNtsn @ abstractLevel 1:Valid concepts hierarchy
Kingdom
Subkingdom
Phylum
Subdivision
Superclass
Class
Subclass
Superorder
Order

Figure 6. Screenshot displaying six revisions of the ITIS data set.

Structural Comparison

One feature that multiple tree visualisations currently lack is the facility to find changes in structure between hierarchies. Most offer methods to discover new or deleted nodes, but none offer solutions for exploring and navigating any internal reorganisation of nodes that occurs between trees. Purely algorithmic tree comparators can perform this task at various levels of detail. String matching approaches work on string representations of trees [*29*] (e.g. (A(B(D(G,H),E(I)),C(F(J,K)))) is a depth-first string representation of the left-hand hierarchy in Figure 7), and can discover if trees match in linear time, but cannot show where differences occur unless every subtree is compared recursively. Other approaches calculate distance measurements between trees through subtree comparison functions [*30*] but location again requires recursive comparison. More computationally expensive techniques calculate detailed edit scripts - working out the exact pruning, addition and movement operations that are needed to convert one tree structure into another [*31*]. However they cannot process large trees efficiently, and as rapid feedback is crucial for interactive environments such as visualisations, they are unsuitable for our purpose of processing trees with 200,000+ nodes each (and for really small trees, an adequate visualisation will allow visual inspection to suffice in any case). To keep the speed of the visualisation as high as possible, we decided to use a simple comparison technique, but one that worked quickly enough in linear time to allow smooth interaction.

In short, our algorithm asks whether selected nodes keep the same parent nodes in the other hierarchies as they had in the originally selected classification. To do this, we take the skeletal, partial tree previously constructed for each hierarchy, and traverse it again to check whether the parent-child relationships are the same as in the original hierarchy. If a relationship is different within a particular classification, then the child node is flagged as having had a change in parent, and consequently all of its ancestors up to the root node are flagged as subtrees that contain structural change. This system captures changes in existing relationships, and also flags up additions and removals of internal nodes, as the affected child nodes will report the change in parentage. This method of storing the changes lends itself neatly to displaying within a tree visualisation, as rather than work out a final value or set of numbers to represent the structural similarities, we gain knowledge of specific nodes within classifications that are dissimilar in their parentage to the originally selected hierarchy – a collection of nodes termed the structural delta.


This node A does not occur in the right-hand side hierarchy. So we must match across the descendant nodes and then track upwards through the ancestor paths to reconstruct a partial tree. Dropping down from the root to the first branch point or selected node gives the inclusive subtree.

A
B
C
D
E
F
G
H
I
J
K
X
Y
F
G
H
I
J
K
Z
- Selected node after A is chosen in left-hand side hierarchy.
- Unselected node
- Reconstructed tree path
- Downward path from root

Figure 7. Finding the least common ancestor by backtracking up through a hierarchy.

In the visualisation, the flag values are read and then used to add textures to the top half of the appropriate node representations, either a strong hatching for the nodes whose parents have changed between trees, or a fainter striping texture applied to the nodes whose descendent set contains such an event. This second texture is useful if the change occurs deep within a tree, the fainter texturing acting as a navigational aid to help the user drill down until the node with the restructured relationship is reached. An example is shown in Figure 8, where a series of screenshots reveals that following any vertical stripe texturing will eventually reveal an incident of structural change. After drilling down, the species *Micropsitta pusio* in Figure 8(d) is marked with a hatched texturing, and is found to have switched genera between the first and second *ITIS* revisions. Conversely, in a tree with many changes, the most obvious pattern is that of the subtrees with no texturing, indicating sub-structures that have been preserved across hierarchies.

The only situation the technique will not cover is the addition or deletion of leaf nodes or entire sub-trees, but existing interaction and visualisation allow such situations to be discovered via brushing and selection comparison. Whilst the approach is algorithmically simple, its application allows the visualisation and user to operate rapidly to find the exact locations of structural reorganisation, something that existing visualisations do not support.

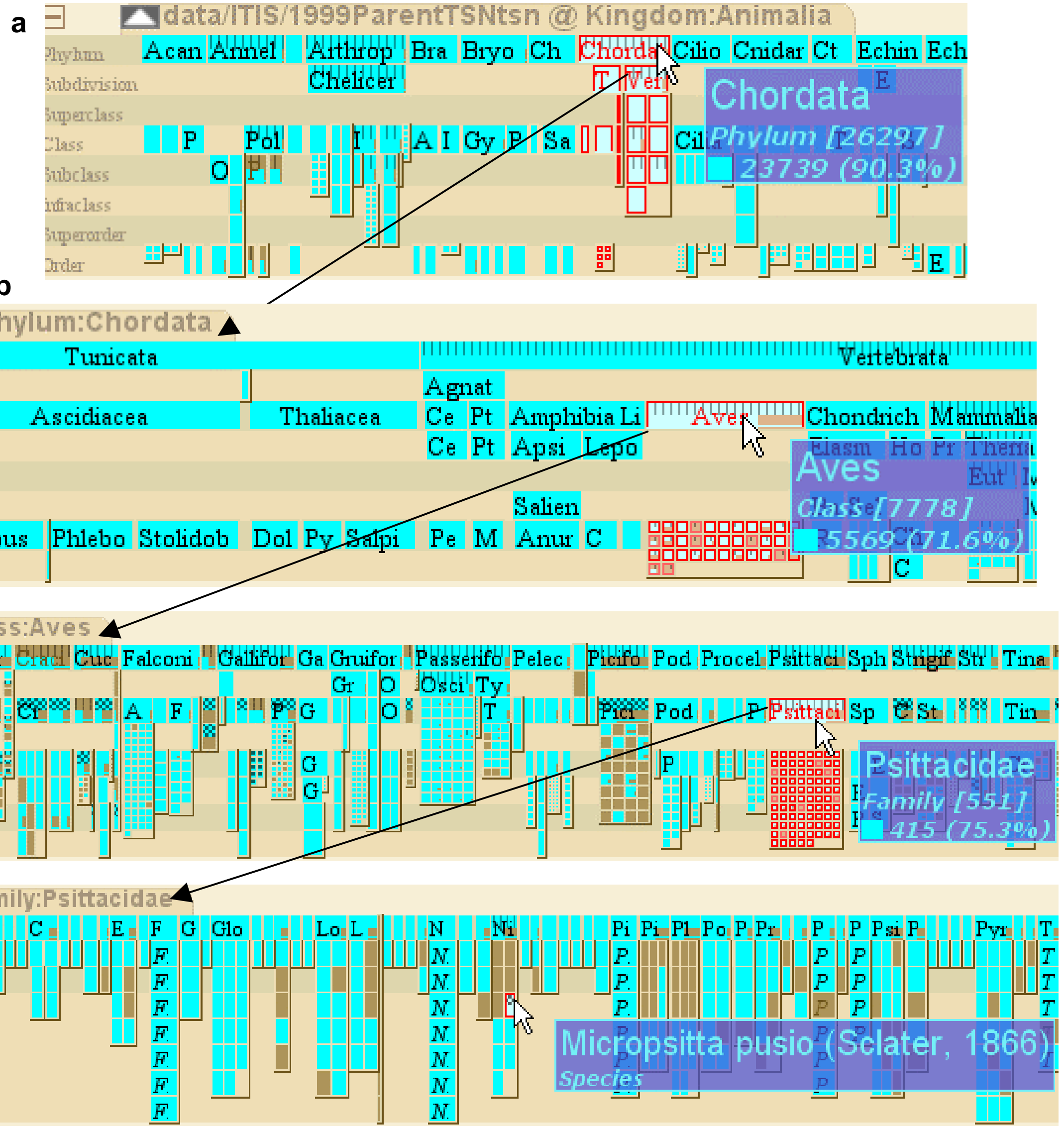

a
data/ITIS/1999ParentTSNtsn @ Kingdom:Animalia
Phylum
Subdivision
Superclass
Class
Subclass
Infraclass
Superorder
Order
Acan Annel Arthrop Bra Bryo Ch Chorda Cilio Cnidar Ct Echin Ech
Chelicer
Chordata
Phylum [26297]
23739 (90.3%)
b
Phylum:Chordata
Tunicata
Vertebrata
Agnat
Ascidiacea
Thaliacea
Amphibia
Aves
Chondrich
Mammalia
Aves
Class [7778]
5569 (71.6%)
Salien
Phlebo Stolidob Dol Py Salpi Pe M Anur C
c
Class:Aves
Cor Craci Cuc Falconi Gallifor Ga Gruifor Passerifo Pelec Picifo Pod Procel Psittaci Sph Strigif Str Tina
Osci Ty
Psittaci
Psittacidae
Family [551]
415 (75.3%)
d
Family:Psittacidae
C C E F G Glo Lo L N Ni Pi Pi Pl Po P Pr P P Psi P Pyr T
Micropsitta pusio (Sclater, 1866)
Species

Figure 8. Discovering internal restructuring by following texture markers. Figure parts (a) - (d) reveal the procedure of drilling-down through a hierarchy until the point of change is found.

Focus & Context

Further controls associated with the visualisation allow text size and the ratio of focus to context in the visualisation to be controlled dynamically. Altering the text size affects the number of levels or ranks in the hierarchies that can be displayed as the size of the node representations change to accommodate the text. Figure 9 shows this effect in action, where larger text sizes in the bottom screenshot produce a simpler presentation at the expense of the detail present in the top screenshot.

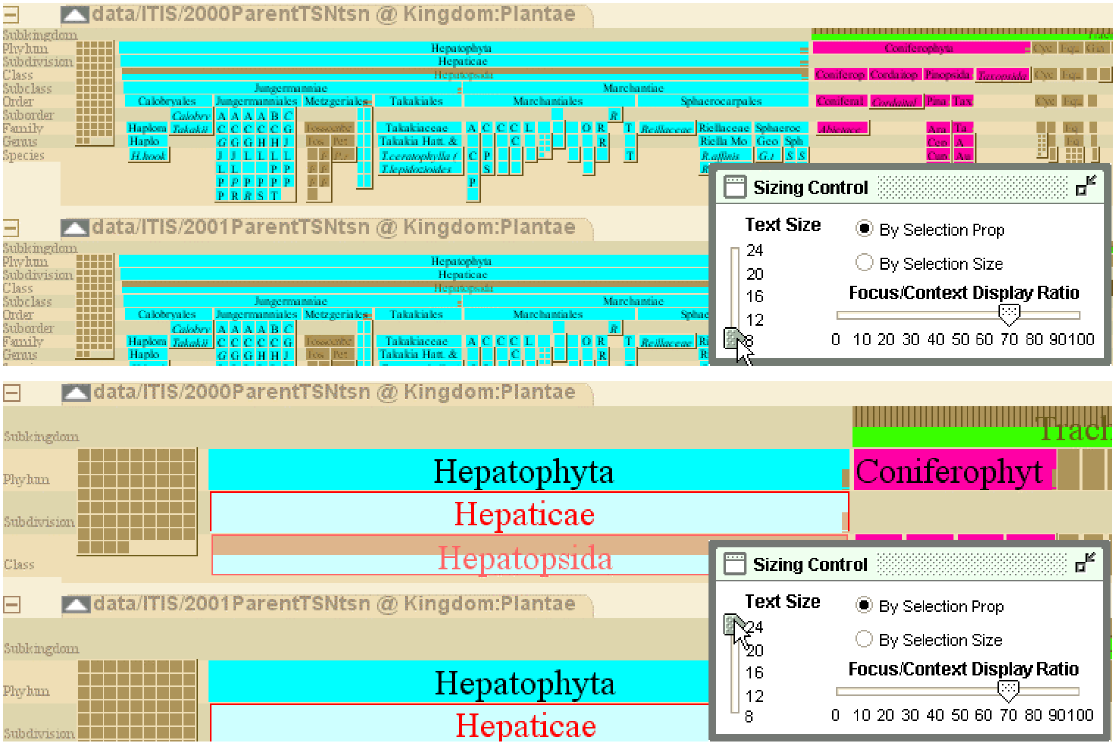
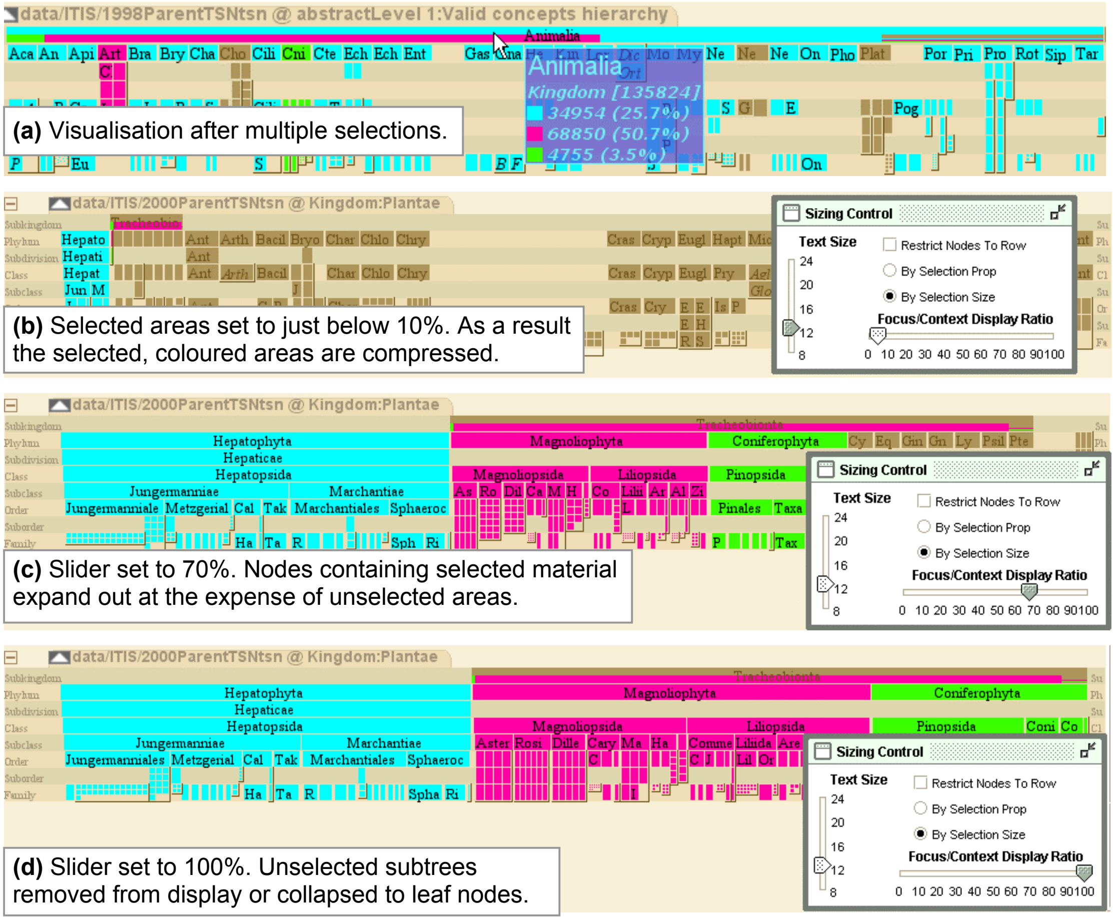


Figure 9. Changing the text size affects the number of ranks that can be displayed.



(a) Visualisation after multiple selections.

(b) Selected areas set to just below 10%. As a result the selected, coloured areas are compressed.

(c) Slider set to 70%. Nodes containing selected material expand out at the expense of unselected areas.

(d) Slider set to 100%. Unselected subtrees removed from display or collapsed to leaf nodes.

Figure 10. Multiple selections with the Focus & Context slider. (a) shows a standard result of a multiple selection. Changing the ratio of focus to context to extremes can either (b) compress subtrees with selected nodes, (c) give some space to all subtrees, or (d) hide all subtrees that have no selected nodes.

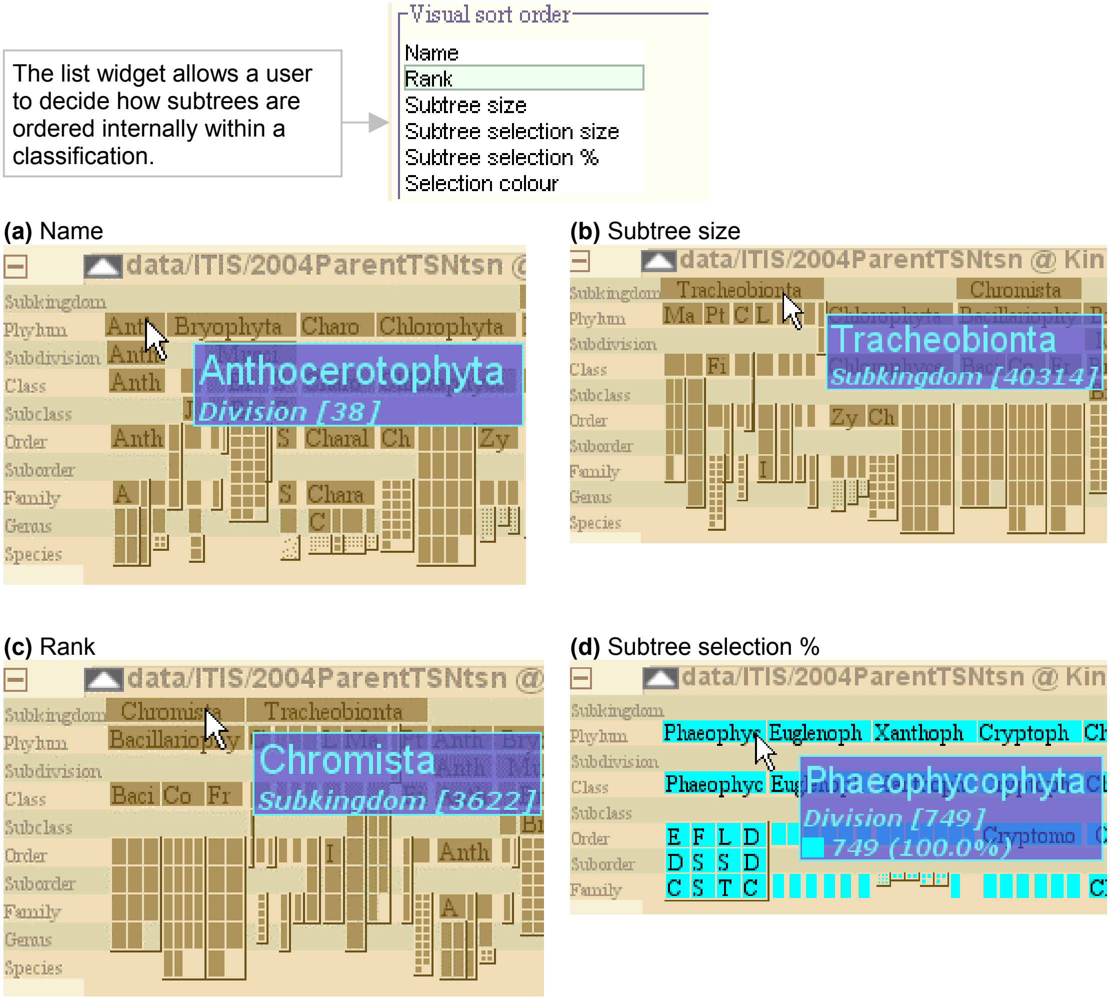
The focus and context control allows a user to change the proportion of display space that is allocated to groups containing selected nodes. This proportion can range from 0% (only groups or subtrees composed of unselected nodes are displayed) to 100% (only groups or subtrees containing selected nodes are presented). Figure 10(b)-(d) displays a series of screenshots that demonstrate the effect of altering the ratio via the slider in the pop-up control. A user has made a number of selections, indicated by the green, blue and magenta colourings. In Figure 10(b), the maximum space given to the selections has been set at below 10%; hence they appear compressed and under-represented. This type of manipulation though is useful when dealing with a subtree that has a majority of selected data; reducing the space allocated to focal areas allows unselected portions to gain prominence. In Figure 10(c), the slider has been moved up to 70%, and consequently the coloured areas have expanded to fill the majority of the available screen space. It can be seen that some unselected portions of the display to the right of the magenta and green blocks still have an appreciable amount of space. However, in Figure 10(d), the slider has been moved up to the maximum value of 100%, so only subtrees containing selected nodes are displayed. Subtrees composed of entirely unselected nodes have either been elided or reduced to leaf node status in the display.

Such a technique has general applicability for allowing users to control the balance between focus and context. In situations where a selection affects the majority of a data set, often the unaffected items themselves become the focus of interest. This control allows visual focus to move smoothly between selected and unselected data. In other circumstances, one or two items chosen in a large data set may appear over-represented and a user may wish to restore more emphasis to the context.

Using a related control, a user can then choose between the absolute number or proportion of selected nodes as the factor for allocating space to the focal parts of the visualisation. This distinction is made available because in some cases the groups of interest may be those that have all their component nodes selected, and we wish to prevent crowding out by larger groups that contain more selected nodes in total but hold less in terms of the proportion selected.

Users can also make internal comparisons within a classification by changing the ordering of nodes within sibling groups. As taxonomies are unordered trees we can reorder siblings without losing any important structural information, with siblings usually arranged initially by alphabetical order. By manipulating a small list widget they can also be arranged according to metrics such as rank, number of child taxa, selection status, and the number or proportion of selected child taxa. The list widget can be used to arrange these metrics such that if two nodes have equal value according to the first property in the list, they are then compared according to the second property and so on until a difference is found or all the comparison methods are exhausted.

Re-ordering the taxa in this manner can help determine size distributions, find the highest rank within a group of child nodes and discover which node in a particular group contains the highest proportion of selected taxa. Figure 11 shows the effect of re-ordering nodes by various metrics. In Figure 11(a), the nodes are arranged according to the default ordering, alphabetically by name. In Figure 11(b), the nodes have been arranged by subtree size, so that we can see *Tracheobionta*, with 40,314 descendants is the largest subtree below the current root. Within *Tracheobionta* itself, we can see it contains several sub-groupings at the Phylum rank, and these in turn are laid out in order of size. Within Figure 11(c) the nodes are ordered by rank, with *Chromista* and *Tracheobionta* displayed first as they occur at the Subkingdom rank, with taxa at the Phylum rank displayed next, and so on until the child nodes are all displayed. As *Chromista* is shown before *Tracheobionta* the next active metric must be alphabetical ordering by name, rather than subtree size. Finally, Figure 11(d) shows the effect of ordering taxa by the proportion of selected descendant taxa they contain. Thus, taxa that contain 100% selected descendants are displayed first, with the proportion descending from left to right.



The list widget allows a user to decide how subtrees are ordered internally within a classification.

(a) Name

(b) Subtree size

(c) Rank

(d) Subtree selection %

Figure 11. Ordering nodes within the visualisation can reveal patterns. Here siblings are ordered according to (a) name, (b) subtree size, (c) rank and (d) proportion of subtree selected.

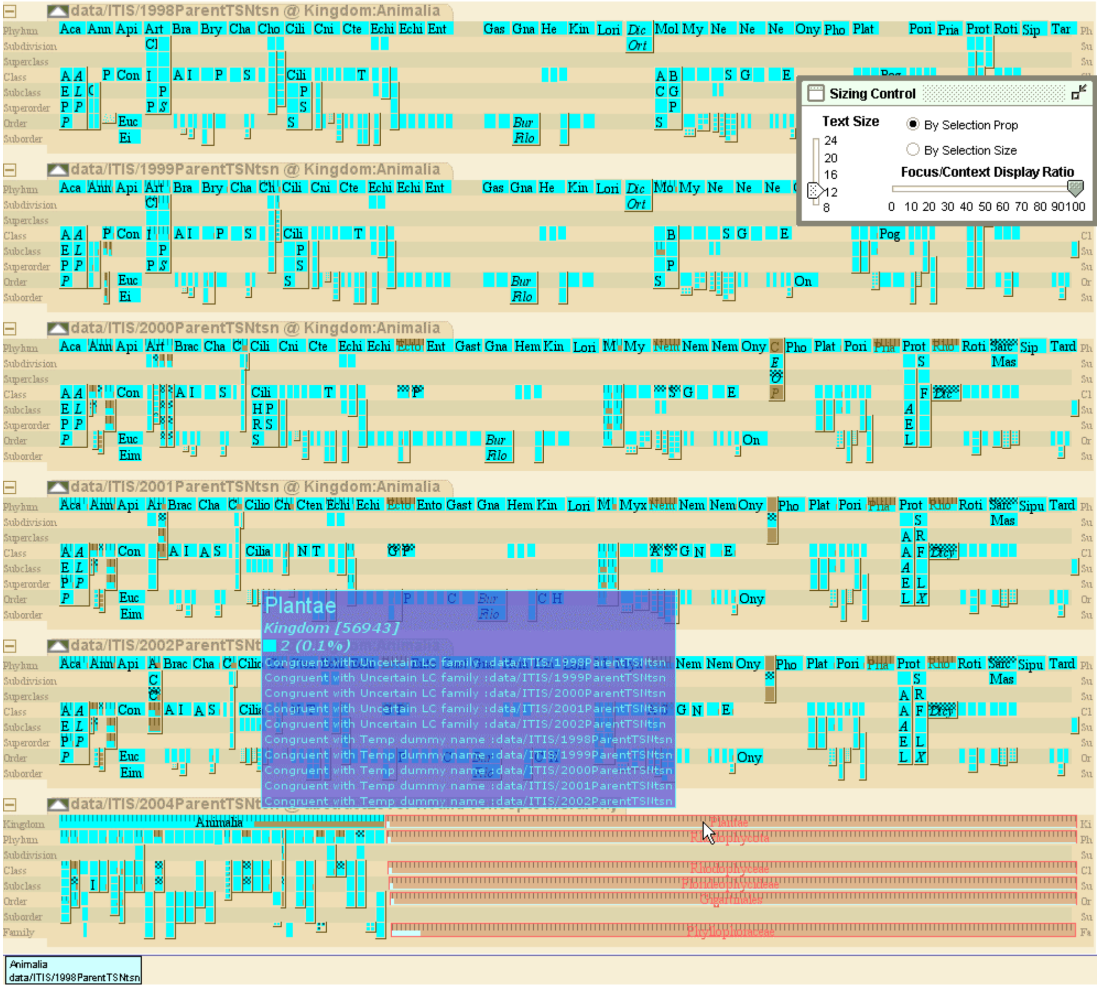
Examples of use

This section describes two annotated examples that demonstrate where the novel features of this visualisation would be of use to taxonomists. The first example covers the massive *ITIS* data sets and reports on the use of the coloured summary bars in node representations to find information that has been marked as interesting. The example then features the use of the structural change flags to aid navigation towards areas of reorganisation. The second example shows the benefits of having a synonymy capable visualisation, demonstrating a marked difference for the *Moss* data sets in correlation between hierarchies when relying on name matching and then synonym information.

Example 1 – The ITIS data

As a first example, we consider the *ITIS* data sets – formed from seven annual snapshots of the ITIS taxonomy from 1998 to 2004 inclusive and totalling over 1 million nodes, with each revision holding between 180,000 and 250,000 taxa. The ITIS data undergoes revision between snapshots as new branches are added to the tree, some are removed, and existing data is reorganised. In a case where a large data set such as ITIS has undergone a process of constant revision then data aggregators and database administrators wish to discover and track significant changes between snapshots.

Visualising the data set allows the context of particular pieces of information to be viewed. For instance, if we discover that a certain taxa has moved between revisions, we would like to know if any of its siblings performed the same manoeuvre, or whether the taxa it moved into was otherwise unchanged or, diametrically, newly formed from a collection of itinerant taxa. Similarly, by selecting entire classifications, and observing how ratios of selected to unselected information change across later revisions, we can not only see if growth occurs but where that growth is concentrated. We can perform such operations within the visualisation, whereas consulting the data in a textual form would need considerably more time and effort to establish the same facts. In the case of large hierarchies the scale of the data set is enough to overwhelm.


data/ITIS/1998ParentTSNtsn @ Kingdom:Animalia
data/ITIS/1999ParentTSNtsn @ Kingdom:Animalia
data/ITIS/2000ParentTSNtsn @ Kingdom:Animalia
data/ITIS/2001ParentTSNtsn @ Kingdom:Animalia
data/ITIS/2002ParentTSNt
data/ITIS/2004ParentTSNt
Phylum
Subdivision
Superclass
Class
Subclass
Superorder
Order
Suborder
Kingdom
Family
Animalia
Animalia
data/ITIS/1998ParentTSNtsn

Sizing Control

Text Size: 24, 20, 16, 12, 8

● By Selection Prop
○ By Selection Size

Focus/Context Display Ratio: 0 10 20 30 40 50 60 70 80 90 100

Plantae
Kingdom [56943]
2 (0.1%)
Congruent with Uncertain LC family :data/ITIS/1998ParentTSNtsn
Congruent with Uncertain LC family :data/ITIS/1999ParentTSNtsn
Congruent with Uncertain LC family :data/ITIS/2000ParentTSNtsn
Congruent with Uncertain LC family :data/ITIS/2001ParentTSNtsn
Congruent with Uncertain LC family :data/ITIS/2002ParentTSNtsn
Congruent with Temp dummy name :data/ITIS/1998ParentTSNtsn
Congruent with Temp dummy name :data/ITIS/1999ParentTSNtsn
Congruent with Temp dummy name :data/ITIS/2000ParentTSNtsn
Congruent with Temp dummy name :data/ITIS/2001ParentTSNtsn
Congruent with Temp dummy name :data/ITIS/2002ParentTSNtsn

Figure 12. Selection of a large sub-tree. Tracking the first instance of *Animalia* down the revisions reveals a split of the nodes between *Animalia* and *Plantae* in the newest revision.

We decide to select *Animalia* to view its changes over six revisions of the data set. Pressing the right mouse button marks *Animalia* and its descendents in this hierarchy, and also where they occur throughout the other classifications, as shown in Figure 12. The first conclusion to be made is that *Animalia* has been constantly growing in size since the first revision. By the sixth revision, brushing over *Animalia* reveals that it has reached over 215,000 nodes in size, of which only 130,000 were present in the first revision, approximately 60% of the total. Since the first revision contained 135,000 nodes we also know that roughly 5,000 have been either moved or dropped along the way.

One feature apparent in Figure 12 is that *Animalia* forms the root for all revisions, except for the final revision. Here, the tree is anchored at the root for all valid ITIS concepts, above the kingdom rank. The reason for this is that *Plantae* also holds selected taxa from the first ITIS revision of *Animalia*. Effectively, something has been moved from the animal to the plant kingdom in this revision. Drilling down into *Plantae*, and following the guide provided by the summary bars leads us to the troublesome taxa, *Schottera*, in Figure 13. If interested, we could then perform a synchronised navigation or selection on *Schottera* to find where it originated from within *Animalia*.

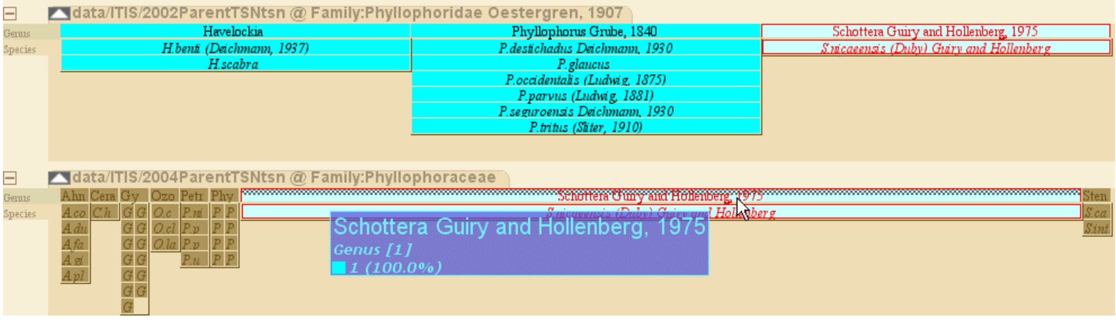

data/ITIS/2002ParentTSNtsn @ Family:Phyllophoridae Oestergren, 1907
Genus
Species
Havelockia
H.benti (Deichmann, 1937)
H.scabra
Phyllophorus Grube, 1840
P.destichadus Deichmann, 1930
P.glaucus
P.occidentalis (Ludwig, 1875)
P.parvus (Ludwig, 1881)
P.seguroensis Deichmann, 1930
P.tritus (Sliter, 1910)
Schottera Guiry and Hollenberg, 1975
S.nicaeensis (Duby) Guiry and Hollenberg
data/ITIS/2004ParentTSNtsn @ Family:Phyllophoraceae
Genus
Species
Ahn Cera Gy Ozo Petr Phy
A.co C.h G G O.c P.m P P
A.du O.cl P.p
A.fa O.la P.p
A.gi P.u
A.pl
Schottera Guiry and Hollenberg, 1975
S.nicaeensis (Duby) Guiry and Hollenberg
Sten
S.ca
S.int
Schottera Guiry and Hollenberg, 1975
Genus [1]
1 (100.0%)

Figure 13. The *Schottera* genus has been moved from *Animalia* to *Plantae* in this latest revision.

Whilst this demonstrates a taxon splitting across revisions, it is more problematic to discover a taxon that has undergone a purely internal revision. In this situation, all would appear well at a high level as the visualisation would communicate that over the course of the revisions no nodes had spilled out of the original taxa into other groups. The same taxon would be the root of all the revisions and the bar would indicate all the nodes to be present and correct. The structural comparison technique described earlier was developed to deal with this type of situation.

To demonstrate, we explore *Plantae* this time, and brush the *Hepatophyta* phylum in the first revision by steering the mouse over it. An intermediate class has been inserted in later revisions, as revealed by the presence of a non-highlighted node in the otherwise solidly highlighted sub-tree displayed in Figure 14. We then select *Hepatophyta* within the first revision and consequently reveal the distribution of its descendants throughout the other revisions.

After selection the screen shows six versions of *Hepatophyta*. When we brush the first two revisions of *Hepatophyta*, as shown in Figure 15, the mouse tooltip reveals the same number of total nodes and the same number of selected nodes for each revision. There is also no texturing of the nodes in the second revision that would indicate any internal structural changes to this selection.

However changes have occurred in the third revision as shown in the bottom half of Figure 16. A number of nodes now have texturing appearing within them, indicating internal structural changes have taken place, and the top level node is showing a small unselected presence within the value bar, indicating the presence of new nodes. The subclass level taxa in Figure 16 are obviously reporting changes as a new taxon at the Class rank has been inserted between them and their previous parent taxon. This new taxa, *Hepatopsida*, is the node that attracted our attention when we originally brushed over *Hepatophyta*.

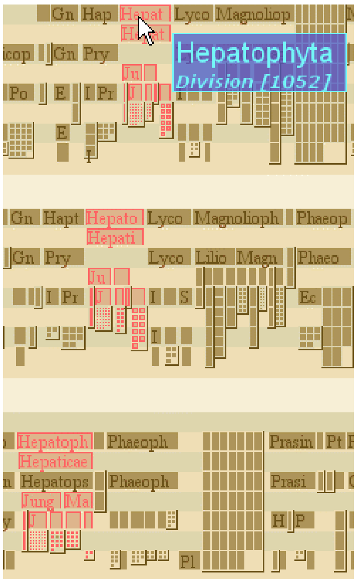

Gn Hap Hepat Lyco Magnoliop
Hepat
Hepatophyta
Division [1052]
Gn Pry
Jul
Po E I Pr
Gn Hapt Hepato Lyco Magnolioph Phaeop
Hepati
Gn Pry Lyco Lilio Magn Phaeo
Ju
I Pr J I S Ec
Hepatoph Phaeoph Prasin Pt
Hepaticae
Hepatops Phaeoph Prasi
Jung Ma
H P
Pl

Figure 14. Brushing the first revision of *Hepatophyta* within the *Plantae* kingdom reveals a new node inserted high up within the third revision.

Also noticeable is that *Metzgeriales* beneath *Jungermanniae* is reporting a change in its parent taxon. Highlighting *Metzgeriales* reveals that it has changed from being a direct child node of the *Hepaticae* subdivision to being included within *Jungermanniae*, and now has two taxa between it and its former parent. Without the explicit marking of the nodes, such a discovery would be either a matter of luck or exhaustive search, neither of which are satisfactory solutions. As such, using this tool, taxonomists and data aggregators can discover low-level and internal changes within a chosen structure, which is especially useful when comparing revisions of the same data set.

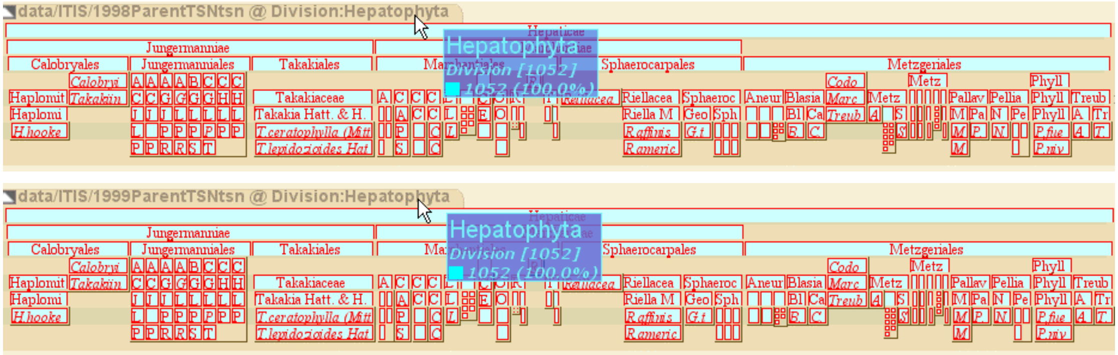


Figure 15. No change between the first two revisions of *Hepatophyta*.

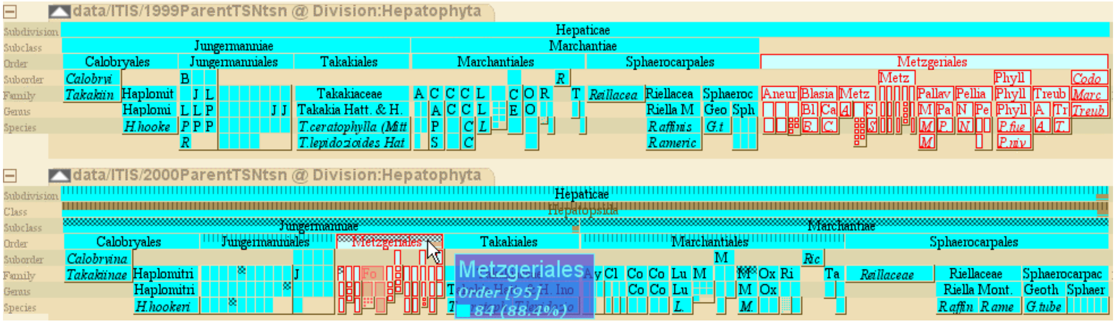


Figure 16. Second and third revisions of *Hepatophyta.* The hatching in the bottom-most *Metzgeriales* node represents a change in parent taxa. Brushing reveals it has been changed from a direct child of *Hepaticae* to being included in the *Jungermanniae* subclass.

Example 2 – The Moss data

The ability to view synonymy is critical when dealing with detailed taxonomies. Simple name matching often results in a misleading correlation being conveyed, as it can be nullified by events ranging from changes in spelling of taxa to reclassification of specimens within new genera. A taxonomist constructing a classification can view patterns through existing synonymy that may suggest where further synonyms could be declared.

To demonstrate the importance of synonymy when matching nodes across classifications, an example is given from the Moss data set. Synonyms in this data set have been described from one particular hierarchy - Koperski - to fourteen others, and reveal many differences between the patterns of their relationships and that produced by purely name-based matching.

With the visualisation set to match taxa by names only, we choose the genus *Tortula* within Koperski's classification to reveal the degree of overlap between it and the other taxonomies. The screenshot of Figure 17 reveals hardly any matches to the genus of the same name as compiled by Monkemeyer in particular. However, when we brush a few nodes in Monkemeyer's classification the tool tip reveals that most of them do have equivalent taxa in Koperski's classification, but they are related as synonyms - not by name.

So we reset the visualisation and change the selection mode to match taxa by synonymy only, and then repeat the selection of *Tortula*. Figure 18 now reveals that Monkmeyer's classification shows much more overlap, and that many of Koperski's *Tortula* taxa are also linked to a completely different genus, *Syntrichia*, in Monkemeyer's classification. Koperski has therefore stated through synonymy that they believe their *Tortula* genus is broadly equivalent to both *Tortula* and *Syntrichia* in Monkmeyer's classification. Brushing the *Tortula* taxon itself in Koperski reveals that it does have direct synonymy with both *Tortula* and *Syntrichia* in Monkemeyer.

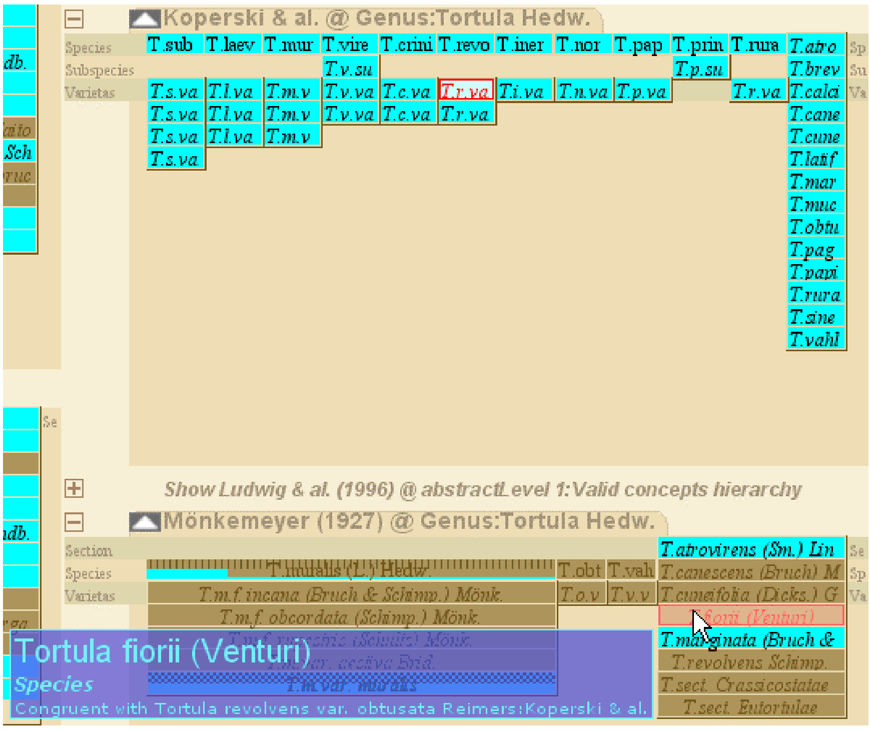


Figure 17. Matching by names in the Moss data set. It reveals a particularly sparse correlation between Koperski and Monkmeyer for *Tortula*, but brushing indicates there are synonymous relationships.

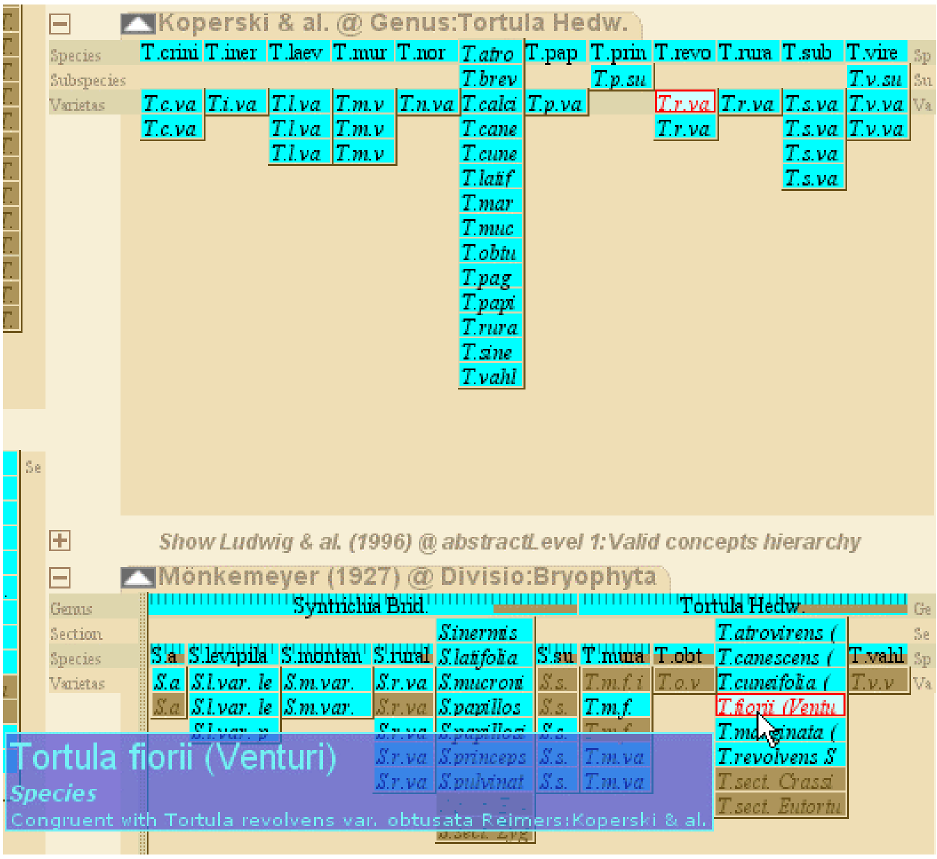

Koperski & al. @ Genus:Tortula Hedw.
Species
Subspecies
Varietas
Show Ludwig & al. (1996) @ abstractLevel 1:Valid concepts hierarchy
Mönkemeyer (1927) @ Divisio:Bryophyta
Genus
Section
Syntrichia Brid.
Tortula Hedw.
Tortula fiorii (Venturi)
Species
Congruent with Tortula revolvens var. obtusata Reimers:Koperski & al.

Figure 18. Matching by synonymy. This reveals a much greater overlap with Monkemeyer's classification, and reveals that some of Koperski's *Tortula* taxa are classified in a completely different genera, *Syntrichia*.

Conclusion

This paper describes a multiple hierarchy visualisation that allows detailed comparison of intersecting taxonomic hierarchies. As well as matching nodes across hierarchies by name, the underlying model allows synonymy relationships to be incorporated into the visualisation, allowing specific taxonomic knowledge to be acknowledged in the visualisation. This permits the visualisation to reflect real-world taxonomic data and comparisons currently possible only through textual taxonomic database interfaces.

The application also provides navigational aids for guiding drill-down interaction within trees. Firstly, we developed a node representation that summarised the proportion of selected and unselected node descendents, as well as the proportion of different selections to each other. These coloured bar representations act as navigational markers for finding selected nodes that have been scattered far and wide across multiple hierarchies. Correspondingly, they can also act to reveal new data that is added beneath selected nodes by communicating the presence of unselected nodes.

Finally, a mechanism that aids users in navigating to and finding structural changes between selected taxa was described. For the first time in a multiple hierarchy visualisation, groups of selected nodes that remain cohesive across hierarchies can be navigated and examined for examples of internal re-organisation. Previously, discovering changes in groups across hierarchies relied on being able to see the changes directly, or by framing the fragmentation of selected groups against a backdrop of unselected data. Despite being a simple metric of the consistency of a node's parents across hierarchies, it helps reveal many internal changes that could otherwise go unnoticed in a large tree.

Annotated examples were given, demonstrating situations where the techniques allowed discovery of relationships and changes that would have either required much painstaking exploration or, in all likelihood, remained undetected. In the case of the synonymy information, we revealed that names alone aren't enough to visualise the true overlap of multiple taxonomies.

Future work

Future work could include an ability to perform tree comparison at multiple resolutions, dependent on sub-tree size. Whilst exhaustive tree comparison functions are too slow for intensively interactive applications when applied to large sub-trees, smaller sets of nodes could benefit from more computationally expensive approaches to finding structural change without a noticeable drop in speed. Hence, a basic metric as we proposed could find changes underneath the high-level nodes, and once a user drilled down to the deeper nodes and smaller subtrees the more complex algorithms could be deployed to provide more detail on the nature of the reorganisation that they contain.

User testing is an obvious course of action for empirically validating the techniques presented here. We plan to perform such testing in the future with representative users such as taxonomic data providers, data aggregators, ecologists and taxonomists to guide future development of the application and its techniques.

Acknowledgements

This work was funded through the SEEK project (Science Environment for Ecological Knowledge) - NSF Grant award 0225676. We would also like to thank the reviewers for their detailed feedback on the original draft of this paper.

References

1 Graham M and Kennedy J. Combining linking & focusing techniques for a multiple hierarchy visualisation. *IEEE Conference on Information Visualization* 2001 (London, UK), IEEE Computer Society Press; 425-432.

2 Graham M, Watson MF, and Kennedy JB. Novel visualisation techniques for working with multiple, overlapping classification hierarchies*.* *Taxon* 2002; **51**(2): 351-358.

3 Graham M, Kennedy JB, and Hand C. A Comparison of Set-Based and Graph-Based Visualisations of Overlapping Classification Hierarchies. *ACM AVI* 2000 (Palermo, Italy), ACM Press; 41-50.

4 Munzner T, et al. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. *ACM Transactions on Graphics* 2003; **22**(3): 453-462.

5 Spenke M. Visualization and interactive analysis of blood parameters with InfoZoom. *Artificial Intelligence in Medicine* 2001; **22**(2): 159-172.

6 Stace CA, *Plant Taxonomy and Biosystematics*. 2nd ed. Hodder Arnold: London, UK, 1989; 272pp.

7 Ytow N, Morse DR, and Roberts DM. Nomencurator: a nomenclatural history model to handle multiple taxonomic views. *Biological Journal of the Linnean Society* 2001; **73**(1): 81-98.

8 Zhong Y, et al. HICLAS: a taxonomic database system for displaying and comparing biological classification and phylogenetic trees. *Bioinformatics* 1999; **15**(2): 149-156.

9 Pullan MR, et al. The Prometheus Taxonomic Model. *Taxon* 2000; **49**(1): 55-75.

10 SEEK. Science Environment for Ecological Knowledge [Web Site] http://seek.ecoinformatics.org (accessed January 15, 2005).

11 GBIF. Global Biodiversity Information Facility [Website] http://www.gbif.org/ (accessed June 14, 2004).

12 Species 2000. Species 2000 Home Page [Website] http://www.sp2000.org/ (accessed January 15, 2005).

13 Jeffrey C, *An introduction to plant taxonomy*. 2nd ed. Cambridge University Press: Cambridge, UK, 1982; 154pp.

14 Robertson GG, Mackinlay JD, and Card SK. Cone Trees: Animated 3D Visualizations of Hierarchical Information. *ACM CHI: Human Factors in Computing Systems* 1991 (New Orleans, Louisiana, USA), ACM Press; 189-194.

15 Johnson B and Shneiderman B. Treemaps: A Space-Filling approach to the visualization of hierarchical information structures. *IEEE Visualization* 1991 (San Diego, California, USA), IEEE Computer Society Press; 284-291.

16 Song H, Curran EP, and Sterritt R. Multiple foci visualisation of large hierarchies with FlexTree. *Information Visualization* 2004; **3**(1): 19-35.

17 Lamping J and Rao R. The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing* 1996; **7**(1): 33-55.

18 Heer J and Card SK. DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. *ACM AVI* 2004 (Gallipoli, Lecce, Italy), ACM Press; 421-424.

19 Amenta N and Klingner J. Case Study: Visualizing Sets of Evolutionary Trees. *IEEE InfoVis* 2002 (Boston, Massachusetts, USA), IEEE Computer Society Press; 71-74.

20 Furnas GW and Zacks J. Multitrees: Enriching and Reusing Hierarchical Structure. *ACM CHI* 1994 (Boston, Massachusetts, USA), ACM Press; 330-336.

21 Mukherjea S, Foley JD, and Hudson S. Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views. *ACM CHI* 1995 (Denver, Colorado, USA), ACM Press; 331-337.

22 Parunak HVD. Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning. *ACM Hypertext* 1991 (San Antonio, Texas, USA), ACM Press; 233-242.

23 Stasko J and Zhang E. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. *IEEE InfoVis* 2000 (Salt Lake City, Utah, USA), IEEE Computer Society Press; 57-65.

24 Sifer M. A Filter Co-ordination for Exploring Multi-Classification Sitemaps. *Coordinated and Multiple Views In Exploratory Visualization* 2003 (London, UK), IEEE Computer Society Press; 112-123.

25 Graham M. Multiple Hierarchy Applet [Webpage] http://www.dcs.napier.ac.uk/~marting/phd/my_applet/index.html (accessed 30th March, 2005).

26 Koperski M, et al., *Referenzliste der Moose Deutschlands*. Schriftenreihe für Vegetationskunde. Vol. 34. LV Druck im Landwirtschaftsverlag GmbH: Münster-Hiltrup, 2000; 519pp.

27 ITIS. Integrated Taxonomic Information System [Website] http://www.itis.usda.gov/index.html (accessed 30th March, 2005).

28 Raguenaud C, Graham M, and Kennedy J. Two approaches to representing multiple overlapping classifications: a comparison. *SSDBM* 2001 (Fairfax, Virginia, USA), IEEE Computer Society Press; 239-244.

29 Chi Y, Yang Y, and Muntz RR. HybridTreeMiner: An Efficient Algorithm for Mining Frequent Rooted Trees and Free Trees Using Canonical Forms. *Sixteenth International Conference on Scientific and Statistical Database Management (SSDBM)* 2004 (Santorini, Greece), IEEE Computer Society; 11-20.

30 Zhong Y, Meacham CA, and Pramanik S. A general method for tree-comparison based on subtree similarity and its use in a taxonomic database*. Biosystems* 1997; **42**: 1-8.

31 Chawathe SS and Garcia-Molina H. Meaningful Change Detection in Structured Data. *SIGMOD* 1997 (Tucson, Arizona, USA), ACM Press; 26-37.