Using Treemaps to Visualize Gene Ontologies

Ketan Babaria
Human Computer Interaction Lab and Institute for Systems Research
University of Maryland, College Park, MD USA
12/04/2001

1. Introduction

Treemaps are a space-filling visualization for hierarchical structures that show attributes of leaf nodes by size and color-coding. Treemaps enable users to compare sizes of nodes and of sub-trees, and are especially helpful in revealing patterns.

The Gene Ontology Consortium maintains Gene Ontology for molecular functions, biological processes and cellular components of gene products. These ontologies can be represented as a hierarchy.

This paper describes how using the gene ontology hierarchy coupled with the treemap visualization technique can be used to effectively visualize genome data.

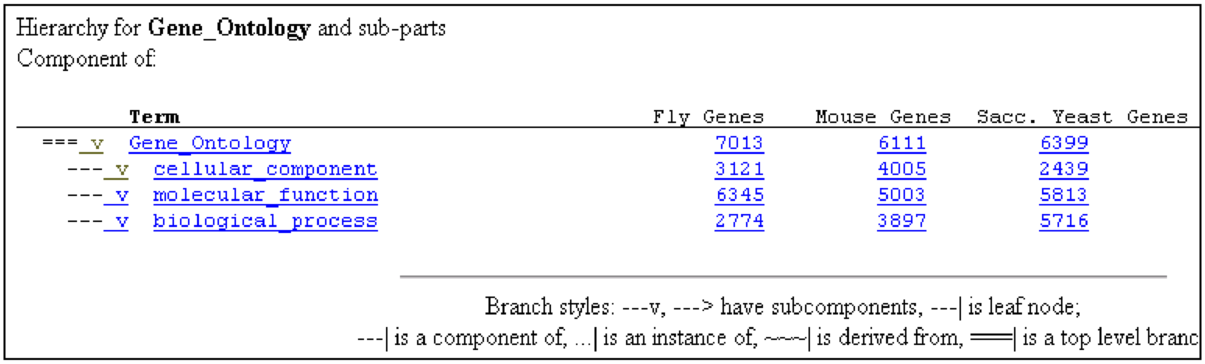
2. Gene Ontology

2.1 Objective

The objective of Gene Ontology (GO) is to provide controlled vocabularies for the description of the molecular function, biological process and cellular component of gene products. The controlled vocabularies of terms are structured to allow both attribution and querying to be at different levels of granularity. The three organizing principles of GO are molecular function, biological process and cellular component [1]. Each of these are further divided into subparts. For example cellular component is further divided into cell, external protective structure, extra-cellular, obsolete, unlocalized etc. Each of these nodes is further divided into sub categories.

2.2 Visualizing GO

For visualizing the GO hierarchy, the web site uses Windows Explorer style information seeking (Figure 1). The figures on the right show number of genes of a particular type (e.g. fly genes) in each hierarchy. If the users want to see the nodes inside cellular component, they have to click on cellular component and the web page will show the nodes inside it (Figure 2.)

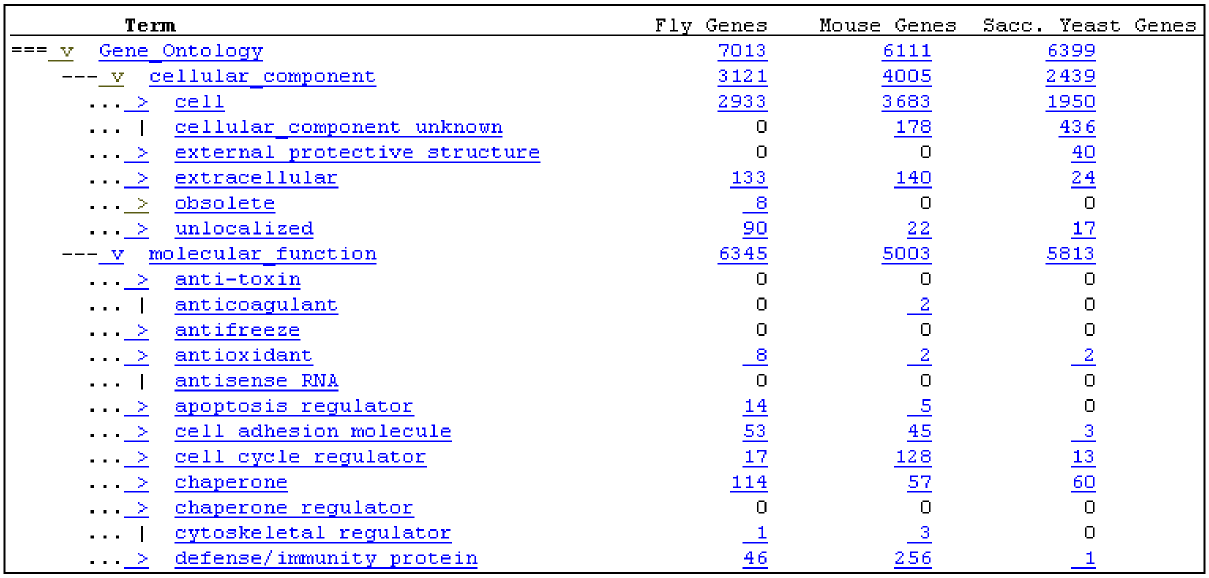


Hierarchy for **Gene_Ontology** and sub-parts
Component of:

Term	Fly Genes	Mouse Genes	Sacc. Yeast Genes
===_v Gene_Ontology	7013	6111	6399
---_v cellular_component	3121	4005	2439
---_v molecular_function	6345	5003	5813
---_v biological_process	2774	3897	5716

Branch styles: ---v, ---> have subcomponents, ---| is leaf node;
---| is a component of, ...| is an instance of, ~~~| is derived from, ===| is a top level branc

Figure 1. Classic Windows Explorer style view for exploring Gene Ontology



Term	Fly Genes	Mouse Genes	Sacc. Yeast Genes
=== v Gene_Ontology	7013	6111	6399
---v cellular_component	3121	4005	2439
...> cell	2933	3683	1950
... | cellular_component unknown	0	178	436
...> external protective structure	0	0	40
...> extracellular	133	140	24
...> obsolete	8	0	0
...> unlocalized	90	22	17
---v molecular_function	6345	5003	5813
...> anti-toxin	0	0	0
... | anticoagulant	0	2	0
...> antifreeze	0	0	0
...> antioxidant	8	2	2
... | antisense RNA	0	0	0
...> apoptosis regulator	14	5	0
...> cell adhesion molecule	53	45	3
...> cell cycle regulator	17	128	13
...> chaperone	114	57	60
...> chaperone regulator	0	0	0
... | cytoskeletal regulator	1	3	0
...> defense/immunity protein	46	256	1

Figure 2. Figure showing the 2nd level hierarchy for GO

Such representation provides detailed content information and it's especially helpful when the users want to scan through all of the listings. However the use of such a representation makes it difficult to extract information as the navigation of the structure is a burden and the content information is often hidden within individual nodes. The problem becomes more acute when there is more than one attribute for leaf nodes.

3. Treemaps

Treemaps graphically represent hierarchical information via a two-dimensional rectangular map, providing compact visual representations of complex data spaces through both area and color [2]. The treemap can represent both hierarchical structure and each element's quantitative information simultaneously utilizing 100% of the designated screen area. Application arenas for treemaps have included computer directory browsing, stock market portfolio visualizations, an NBA player statistical browser and a budget viewer. For example, Figure 3 shows a treemap for visualizing stock data [3]. The size of boxes show the market capitalization and the color show whether the stock prices went up (green) or down (red). We see that the market capitalization of Exxon Mobile is highest in Energy sector and on this day the stock went down.

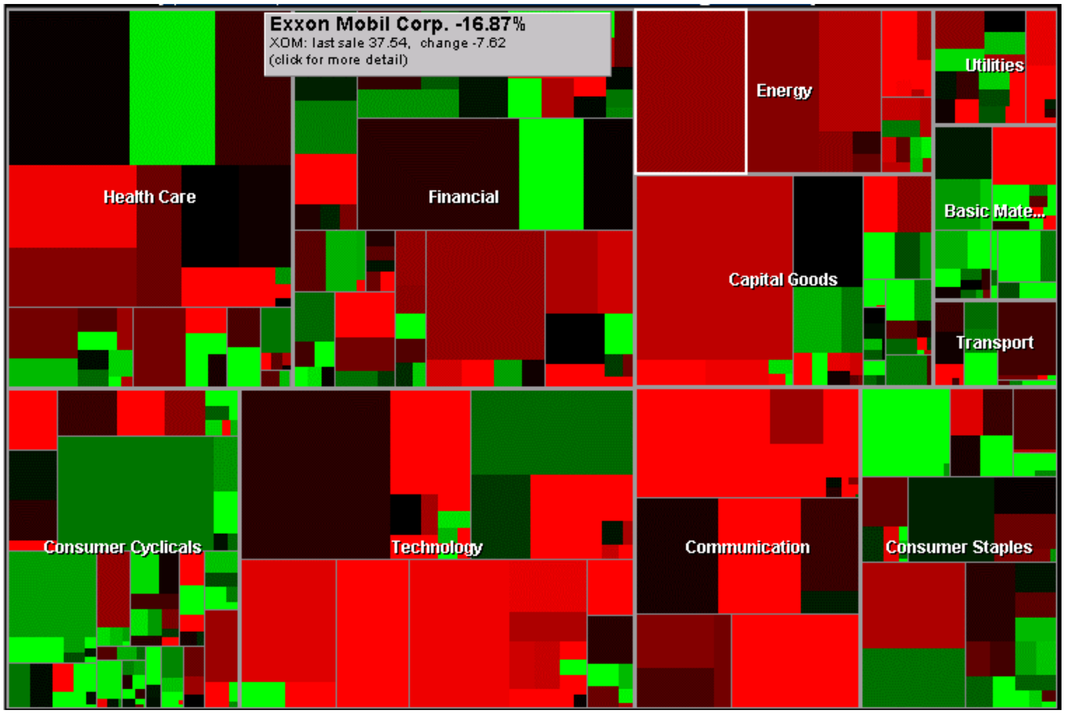


Figure 3. Treemap showing the performance of stock market.

4. Treemap representation of GO

Since the Gene Ontology can be represented by hierarchical trees, these trees translate directly to the treemap visualization method. Figure 4 is an example of a treemap generated with Treemap 3.0 software for the same GO hierarchy that was shown in Figure 3. A base rectangle represents the root of the GO hierarchy and is divided into small rectangular areas proportional to their relative importance. Users can size by any numerical attribute of the leaf node. The leaf nodes in this case are the genes. So the user can size by one of its attributes such as average fold change.

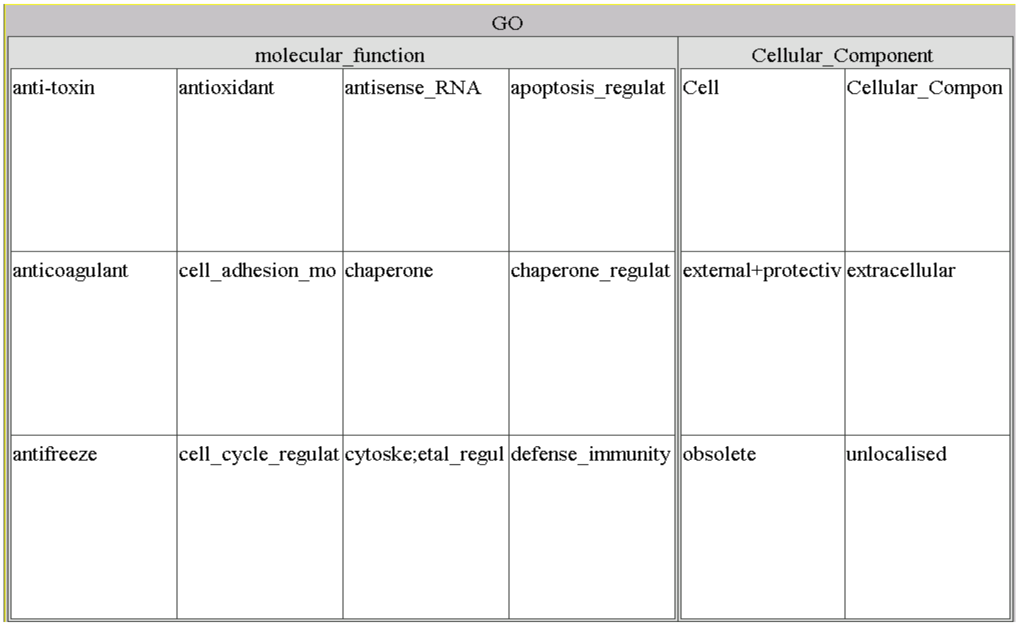


Figure 4 Treemap representation figure 2

While the Windows Explorer representation of GO data is useful, quantitative attributes of genes may be better displayed and queried using treemaps. For example if the genome is analyzed for genes that are transcribed during cell death, 3087 genes are expressed at different levels. The changes in gene level (fold change) and the representation of this information, and how genes are represented in the larger context of GO demonstrate the utility of treemap for this application.

In this dataset each of 3087 genes has a number of attributes (average fold change, base pairs, pH value, etc). For approximately half of the genes in the dataset, their position in the GO hierarchy is known and the rest are labeled as unknown. The user may want to know the distribution of this data set in the GO hierarchy. Figure 5 shows the treemap for this data. Each square represents a gene. Using the treemap a user can recognize that almost 50% of the genes fall in the unknown category and approximately 25% of the genes fall under the enzymes category and so on.

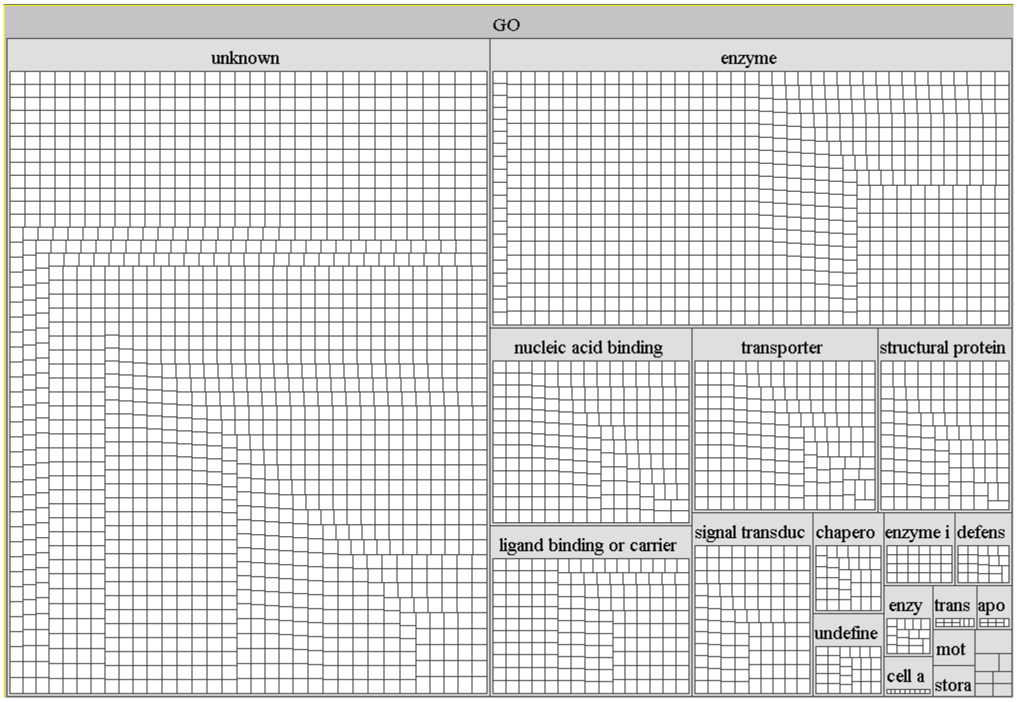


Figure 5 Gene distributions in each GO category using Treemap

Now if the user has information regarding the second level of the hierarchy for each gene, the treemap becomes more effective in visualizing the distribution. Figure 6 shows the distribution of genes in two levels of hierarchy. Such distribution can be extended to any level of the hierarchy.

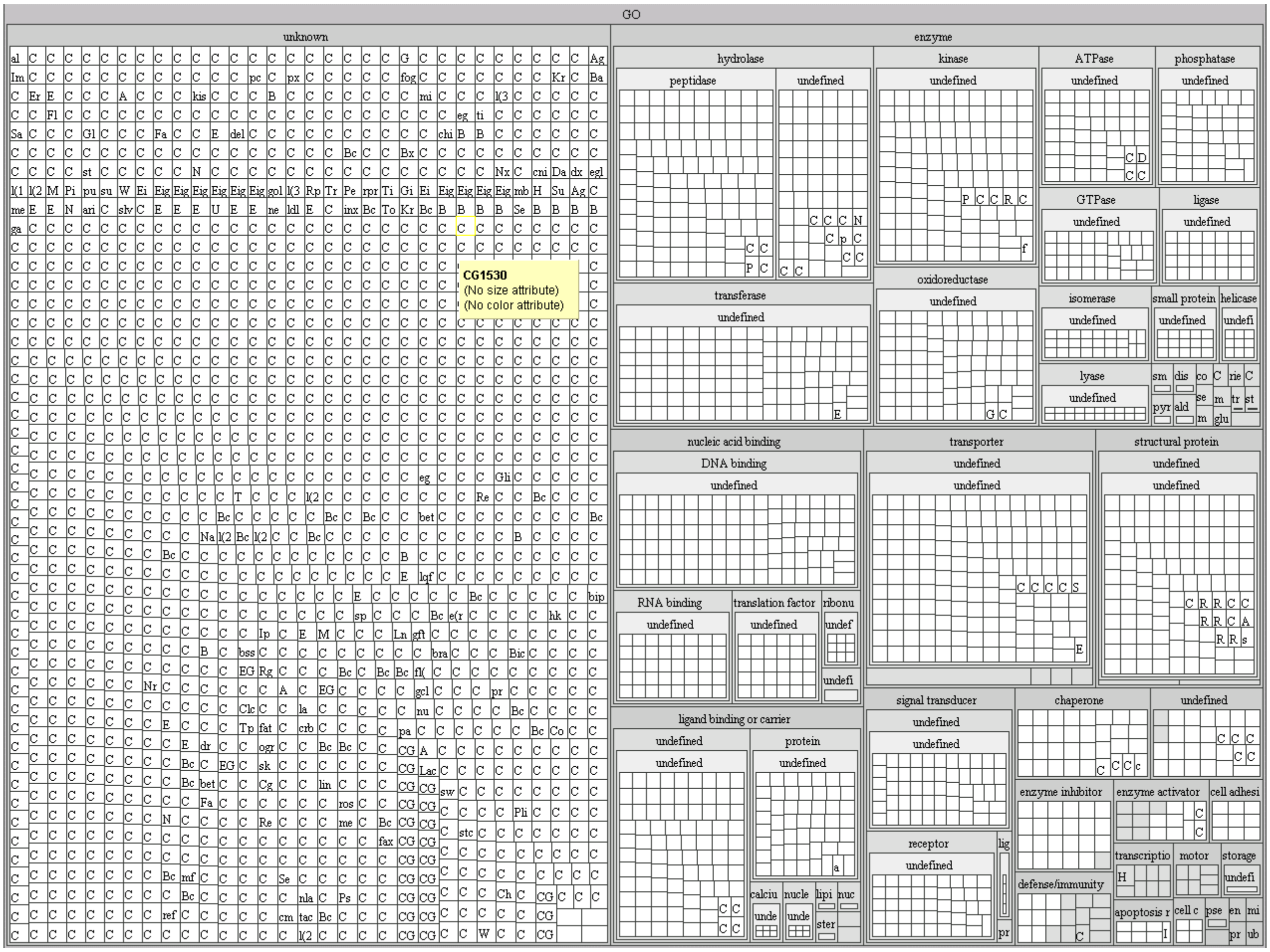


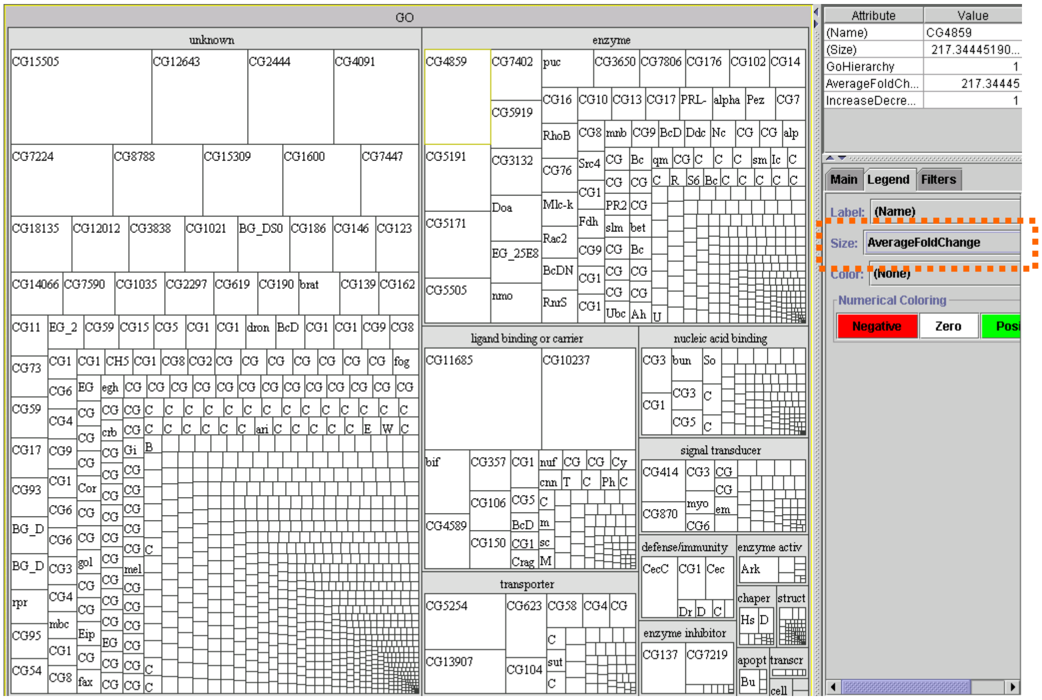
Figure 6 Gene distributions in 2nd level GO category using Treemap

In this example hydrolase has highest number of genes in enzyme group.

5. Data manipulation tools

5.1 Sizes and Color

In the GO hierarchy all of the leaf nodes (genes) have the same number of attributes. The attributes can be numerical (for example average fold change) and non numerical (for example acidic or basic) users can then modify the color and sizes of the nodes depending on a particular criterion. In the above example users can specify the size by criteria to be ‘Average Fold Change’. Figure 7 shows the resulting treemap.



Attribute	Value
(Name)	CG4859
(Size)	217.34445190...
GoHierarchy	1
AverageFoldCh...	217.34445
IncreaseDecre...	1

Main | Legend | Filters

Label: (Name)

Size: AverageFoldChange

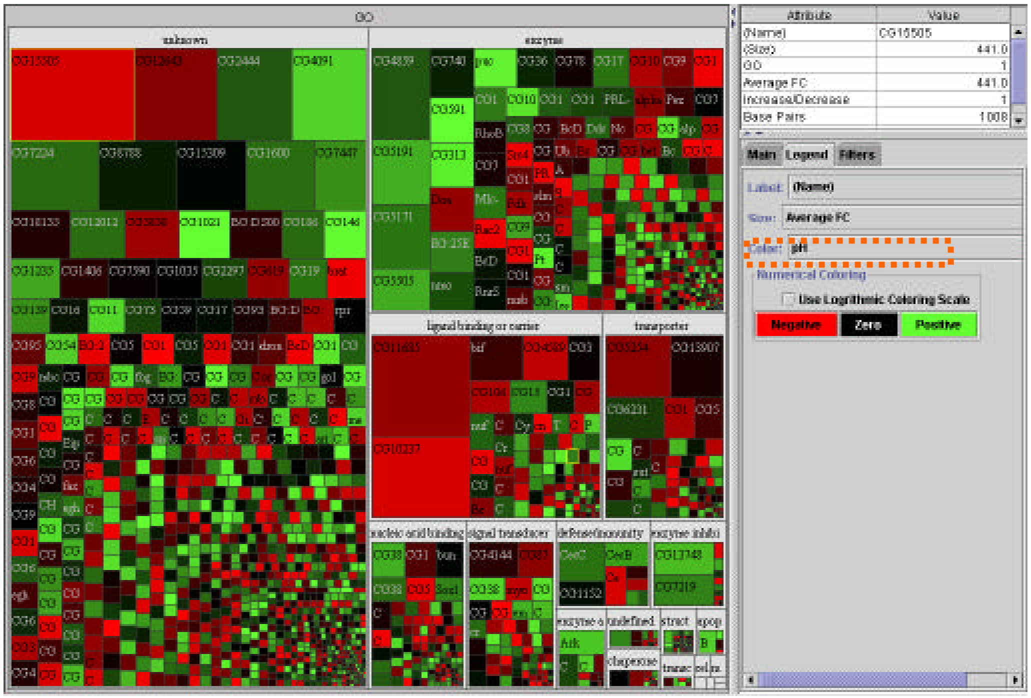
Color: (None)

Numerical Coloring

Negative | Zero | Posi

Figure 7 Treemap with nodes sized by one of the attributes

In this figure the gene with highest fold change in the ligand binding branch has a higher fold change than the gene with the highest fold change in the enzyme category. Such global level knowledge discovery becomes easy in treemaps and is less tedious than going through a list of all of the leaf nodes and then comparing the value of their attributes. The users can further specify the color of the nodes depending on particular criteria. Figure 8 shows the treemap where the color of the nodes specifies if they are basic (green) or acidic (red).



Attribute	Value
(Name)	CG15505
(Size)	441.0
GO	1
Average FC	441.0
Increase/Decrease	1
Base Pairs	1008

Main | Legend | Filters

Label: (Name)

Size: Average FC

Color: pH

Numerical Coloring

□ Use Logarithmic Coloring Scale

Negative | Zero | Positive

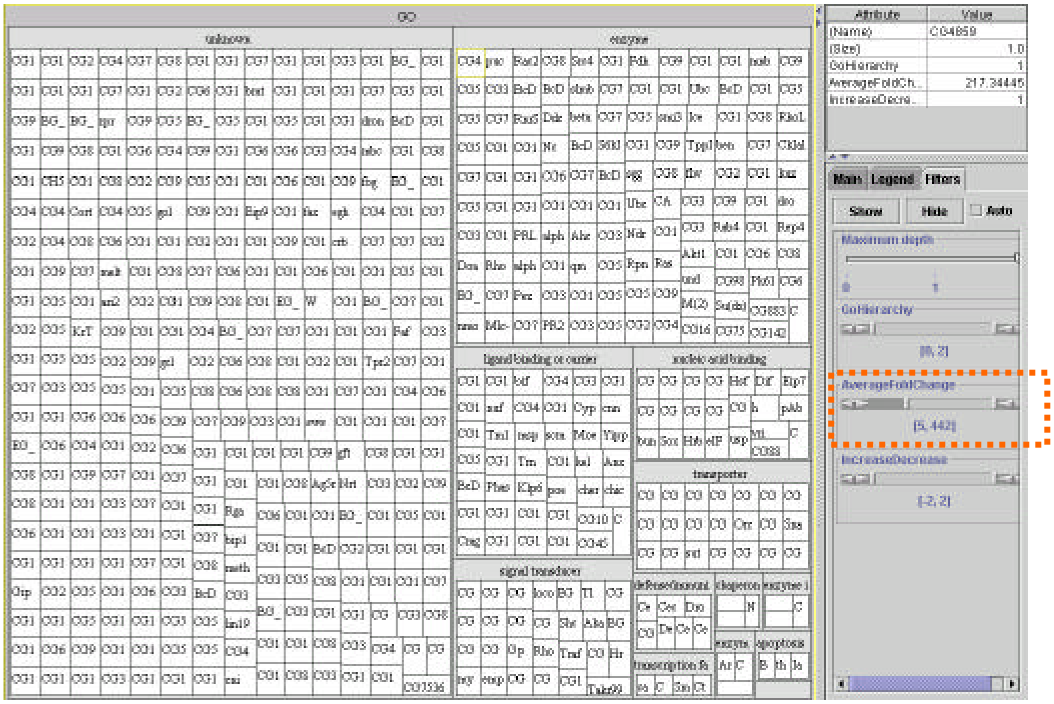
Figure 8 Treemap with nodes color by one of the attributes.

From this figure users can see that genes with highest fold change (size by attribute) in Unknown is also very acidic (Color by attribute).

5.2 Filtering and Dynamic queries

The control panel of the Treemap program provides the capability of dynamic queries. Dynamic queries are a visual alternative to Sequential Query Language (SQL) for querying a database. Dynamic queries are dramatically different from existing database query methods because they continuously update search results - within 100 milliseconds - as users adjust sliders or select buttons to ask simple questions, and find patterns or exceptions [4].

The control panel has sliders for each numerical attribute. Users can filter the nodes based on attributes of the leaf nodes. For example figure 9 shows the resulting treemap when all of the leaf nodes with less than 5 fold are change are filtered out.



Attribute	Value
(Name)	CG4859
(Size)	1.0
GoHierarchy	1
AverageFoldCh...	217.34445
IncreaseDecre...	1

Maximum depth
0 1

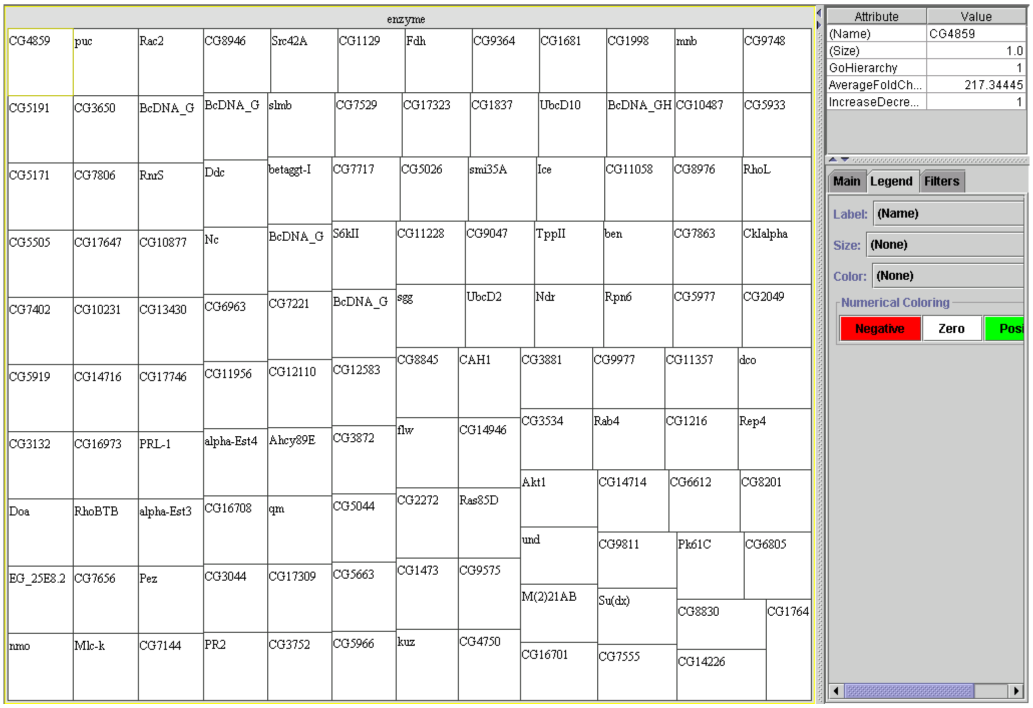
GoHierarchy
[0, 2]

AverageFoldChange
[5, 442]

IncreaseDecrease
[-2, 2]

Figure 9 Treemap with nodes less than 5 fold change filtered out.

Treemap also allows any of the nodes to be set as the current node by clicking the mouse. The attributes of the current node are displayed at the right side of the screen, in the control panel. Additionally, a zooming capability is provided to allow any node in the tree to become the root node, thus obtaining more screen area for that node's descendants. For example, figure 10 shows the treemap zoomed to only enzymes.



Attribute	Value
(Name)	CG4859
(Size)	1.0
GoHierarchy	1
AverageFoldCh...	217.34445
IncreaseDecre...	1

Label: (Name)

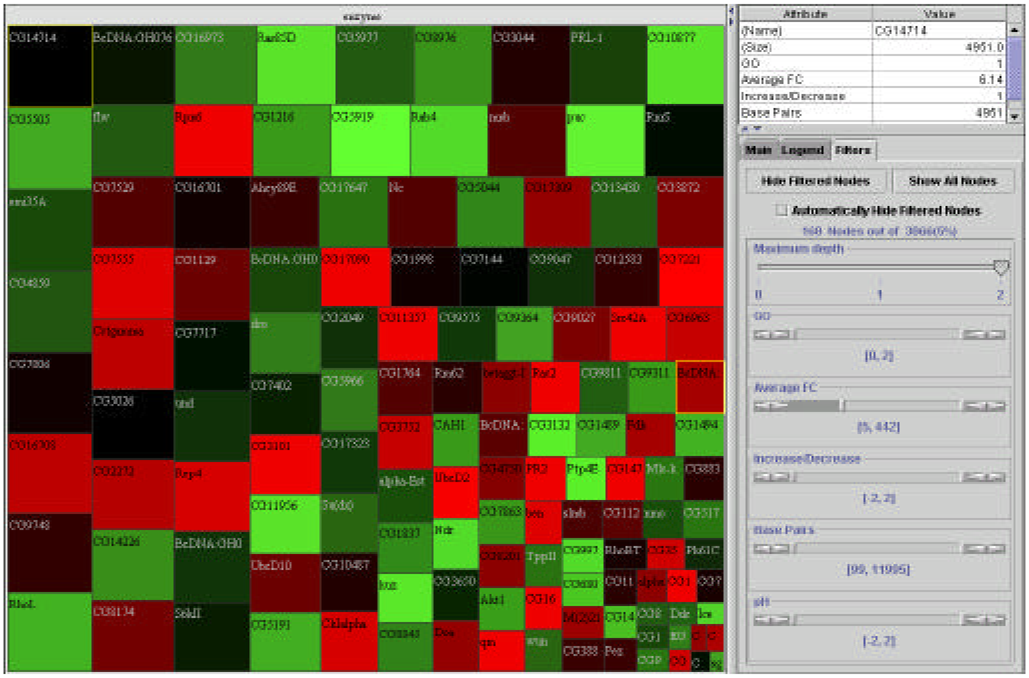
Size: (None)

Color: (None)

Figure 10 Treemap with root set to enzyme and with nodes whose fold change less than 5 filtered out.

Users can now further size, color and filter the leaf nodes by any of its attributes. Figure 11 shows Treemap with root set to enzyme and with nodes whose fold change is less than 5 filtered out. The size-by attribute is number of base pairs and color-by attribute is pH value.

We see that among the nodes remaining after filtering, the node with highest number of base pairs (top left) is also neutral (red for acidic, green for basic and black for neutral).



Attribute	Value
(Name)	CG14714
(Size)	4951.0
GO	1
Average FC	6.14
Increase/Decrease	1
Base Pairs	4951

Hide Filtered Nodes | Show All Nodes

□ Automatically Hide Filtered Nodes

169 Nodes out of 3866(5%)

Figure 11 Treemap with root set to enzyme and with nodes whose fold change less than 5 filtered out. The treemap is sized by 'Base Pairs' and colored by 'pH' value.

6. Treemap strength and limitations

All static hierarchical presentations have limits as to the quantity of information they are capable of presenting on a finite display space. When these limits are reached, navigational techniques such as scrolling or panning must be used, creating the potential for loss of context [5]. Common character-based applications use a set number of lines to display the hierarchy. In graphical tree diagrams depending upon the drawing algorithm and the size of the display space, a hundred or so nodes can be adequately represented on screen without the need for panning or zooming. The number of nodes that can be displayed by a treemap can be an order of magnitude greater than traditional graphical tree diagrams. One of the treemap applications in HCIL displays 1 million nodes.

Treemap visualizations are limited to hierarchical data sets and hierarchical decompositions of categorical data [6]. As with any new visualization technique, users need some training to learn this technique.

The slice and dice algorithm, although it maintains fixed position of the nodes, usually results in thin high aspect ratio rectangles making node comparison difficult. Squarified algorithm [7] maintains low aspect ratio but results in nodes changing their position as sizes change making it difficult to find nodes consistently at one position.

7. Suggestions and future research

Since the success of Market Map [8] there has been renewed interest in Treemaps. Several algorithms were proposed in last few years to overcome limitations of the slice and dice algorithm (for example ordered, pivot my middle etc). A few algorithms were specialized to solve particular problems (for example Quantum treemaps). New algorithms may be developed specially for solving bio-informatics problems for example an algorithm that shows gene clustering with treemaps. Also Treemaps can be coupled with an existing visualization tool as an alternative view of data. As a web application, an applet for treemap may be created that will help users to navigate GO hierarchy.

With the present pace of developments in the field of genetics, proteomics, many more attributes of genes and proteins will be discovered and visualizing this data will be a challenging problem. New visualization techniques will be needed to organize and visualize this data. Treemap is one such powerful tool for visualizing this data. This paper discussed how treemaps could be applied to visualize gene data and its limitations.

8. Acknowledgements

I would like to thank Dr Ben Shneiderman who introduced me to the interesting and challenging field of information visualization. My education in the field of HCI would have been incomplete without him. I would also like to thank Dr Eric Baehrecke for helpful discussions and comments on this manuscript.

All the demonstrations in this paper used Treemap 3.0 [9] [10] software and I would like to thank members of the Treemap team (Dr Catherine Plaisant, Jim Blowitski, Steve Betten, Jesse Grosjean) who were always helpful and supportive. I would also like to thank Agilent Technologies for supporting us on this project.

References

[1] Gene Ontology Consortium- www.go.org
[2] Shneiderman, B. Tree Visualization with Tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics* 11, 1 (Jan. 1992), pp. 92-99.
[3] SmartMoney- www.smartmoney.com/marketmap
[4] Shneiderman, B., Dynamic queries for visual information seeking, *IEEE Software* 11, 6 (1994), 70-77.
[5] D. Beard and J. Walker II. Navigational techniques to improve the display of large two-dimensional spaces. *Behavior & Information Technology*, (1990), 9(6): pp. 451-466.
[6] Johnson, B.S. "Treemaps: Visualizing Hierarchical and Categorical Data" PhD thesis, Computer Science Department, University of Maryland. (1995), p. 148.
[7] Van Wijk, J., Wetering, H. "Cushion Treemaps: Visualization of Hierarchical Information" *IEEE Symposium on Information Visualization*, (1999), pp. 73-78
[8] 2001 National Magazine award for Best Interactive Design
[9] http://www.cs.umd.edu/hcil/treemap3/
[10] http://www.cs.umd.edu/hcil/treemaps/