Visualizing Change over Time Using Dynamic Hierarchies: TreeVersity2 and the StemView

John Alexis Guerra-Gómez, Michael L. Pack, Catherine Plaisant, and Ben Shneiderman

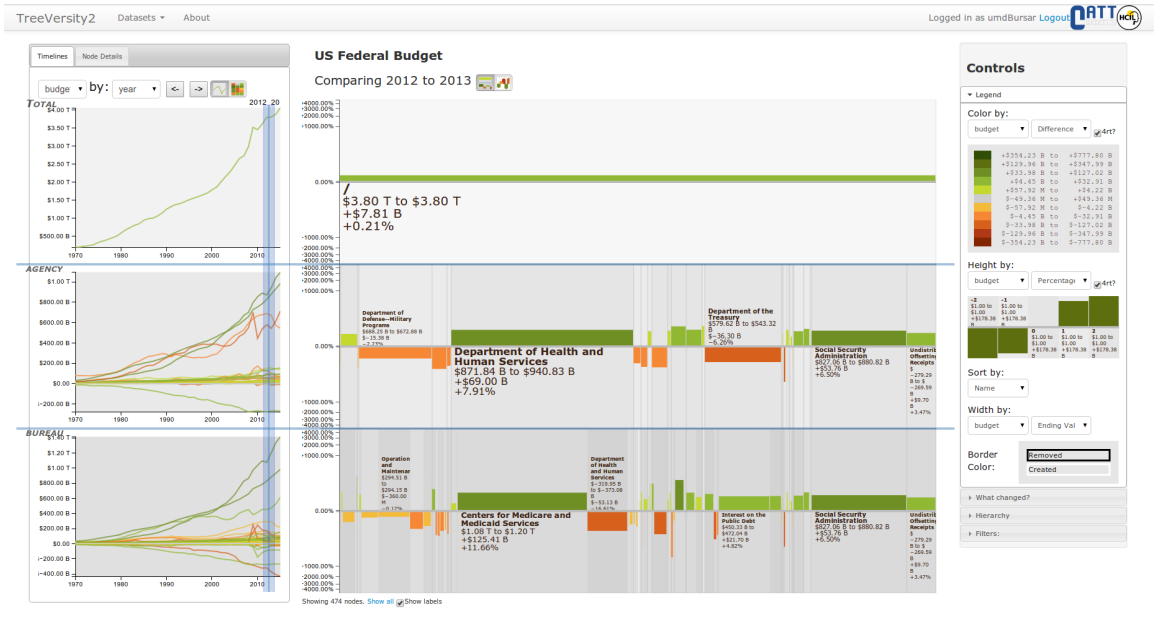

Fig. 1. Exploring change in the US Federal Budget since 1970. The left panel shows the timelines of the actual budgets for each element in the tree: overall at the top, by Agency in the middle and by Bureau at the bottom. The StemView (center panel) compares 2013 and 2012. Each box in the StemView represents an element in the Budget. The green box on the top tells us that overall the Budget increased by US$7.81 Billion. The middle row shows the changes by Agency. Color represent the change in dollars (green for an increase)while the height of the boxes show the percentage of change. The width shows the budget in 2013. Defense, Health, Treasury and Social Security are the main players, and all are increasing.

Abstract—To analyze data such as the US Federal Budget or characteristics of the student population of a University it is common to look for changes over time. This task can be made easier and more fruitful if the analysis is performed by grouping by attributes, such as by Agencies, Bureaus and Accounts for the Budget, or Ethnicity, Gender and Major in a University. We present TreeVersity2, a web based interactive data visualization tool that allows users to analyze change in datasets by creating dynamic hierarchies based on the data attributes. TreeVersity2 introduces a novel space filling visualization (StemView) to represent change in trees at multiple levels - not just at the leaf level. With this visualization users can explore absolute and relative changes, created and removed nodes, and each node's actual values, while maintaining the context of the tree. In addition, TreeVersity2 provides overviews of change over the entire time period, and a reporting tool that lists outliers in textual form, which helps users identify the major changes in the data without having to manually setup filters. We validated TreeVersity2 with 12 case studies with organizations as diverse as the National Cancer Institute, Federal Drug Administration, Department of Transportation, Office of the Bursar of the University of Maryland, or eBay. Our case studies demonstrated that TreeVersity2 is flexible enough to be used in different domains and provide useful insights for the data owners. A TreeVersity2 demo can be found at https://treeversity.cattlab.umd.edu

Index Terms—Information visualization, Tree comparison

- *John Alexis Guerra-Gómez was with HCIL & CATT, Department of Computer Science, University of Maryland, now with PARC, a Xerox Company. E-mail: john.guerra@gmail.com.*
- *Michael L. Pack is with CATT Lab, University of Maryland. E-mail: PackML@umd.edu.*
- *Catherine Plaisant is with HCIL, University of Maryland. E-mail: plaisant@cs.umd.edu.*
- *Ben Shneiderman is with HCIL & Department of Computer Science, University of Maryland. E-mail: ben@cs.umd.edu.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1 Introduction

Analyzing changes to numerical datasets over time is one of the most common and useful techniques of data exploration. However, for datasets that can be represented as trees, like the US Federal Budget, analyzing temporal changes is challenging. For example, if one wants to explore what has changed in the U.S. Federal Budget over the past 20 years, one can organize the Budget as a tree— grouping funding amounts by Agencies (like the Department of Health and Human Services) and their Bureaus (like Medicaid, Medicare, Child and Family Services, etc.). Each node can be labeled by the organizational name (e.g. Dept. of Health and Human Services), the amount of dollars spent during a fiscal year, and other attributes (e.g. Discre-

1077-2626/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

tionary/Mandatory/Net Interest).

With such a tree, users could ask questions like: which nodes (e.g. funded programs) increased or decreased the most compared to the previous year (both in relative and absolute values). These questions suggest that a visual analytics tool to explore these changes should illustrate the **direction of change** (increase or decrease), the **actual amount of change** (dollar amounts in the budget example) and the **relative change** (the percentage of change compared to the previous year).

One simple solution for this visualization would be to build a table that shows all the actual and percentages of change for each program in the budget, like the one shown in Fig. 2(a). However, the table will be insufficient if users want to maintain the **context of the hierarchy**, and look at **inner node's values**, such as finding Bureaus that change significantly but are part of an Agency that don't change at all. In addition users might want to find nodes that were **created** or **removed** in the tree, like finding all the Bureaus that were created in 2013.

Using a classic node-link based tree visualization with node glyphs that show change (e.g. the Bullet visualization [1, 2, 3] shown in Fig. 2(b) will allow the exploration of tasks that require the context of the hierarchy, while still providing insight about absolute and relative changes. Node link representations get too crowded even with a tree of only a hundred nodes, and do not show the actual **starting** and **ending** values of the nodes (e.g. when comparing the 2012 and 2013 budgets, the starting values are the actual dollar values for 2012 and the ending values those for 2013) which are required to answer questions such as "which Agencies lost the most funding throughout multiple years when compared against all other agencies?".

As shown in Fig. 2(c) a treemap where color represents the percent change and the area of each box represents the total dollar value could be used for this task [4, 5, 6], however treemaps can only show one change variable at a time (actual or relative change), cannot represent negative values of the size attribute, and hide the values of the inner nodes (which is a significant problem when the values do not aggregate up the tree).

We present TreeVersity2 (Fig. 1), an interactive data visualization tool that allows the exploration of change in hierarchically organized numerical data and tackles direction of change, actual and relative change, starting and ending values, created and removed nodes, and inner nodes' values - while keeping the hierarchy context. TreeVersity2 allows the detailed exploration of change between two time points (e.g. two years) with the StemView, coordinated with additional overviews to explore a larger time range. Finally TreeVersity2 includes a reporting tool, that guide users through the most significant differences in the tree, according to outlier detection algorithms.

We evaluated TreeVersity2 using 12 case studies, developed with partners from organizations as diverse as the National Cancer Institute, Federal Drug Administration, Department of Transportation, Office of the Bursar of the University of Maryland, or eBay. The diversity of the characteristics of the datasets of these case studies showcase the flexibility of TreeVersity2 and suggest that it is a useful tool for analyzing change in complex datasets.

In this paper we start with definitions and a description of the type of comparison TreeVersity allows users to perform, then describe the related work. Next we introduce StemView, a novel space filling visualization artifact representing a wide variety of changes in trees, and describe TreeVersity2's Reporting tool. Finally we summarize a subset of our case studies.

1.1 Definitions

In this paper a tree is treated as the traditional data structure defined in computer science books, composed of nodes and links that express parent-to-child relationships, but where each node, regardless of being leaf or inner node, follows these rules: 1) It is uniquely labeled in the tree, 2) contains one or more numeric variables, with values over time, and 3) contains one or more categorical attributes that might have more than one value.

Much work has been done on visualizing [7, 8, 9, 10] and exploring [11, 12, 13] single tree structures; however, the problem of comparing trees is significantly harder. We have identified and classified the following five types of tree comparison (Fig. 3):

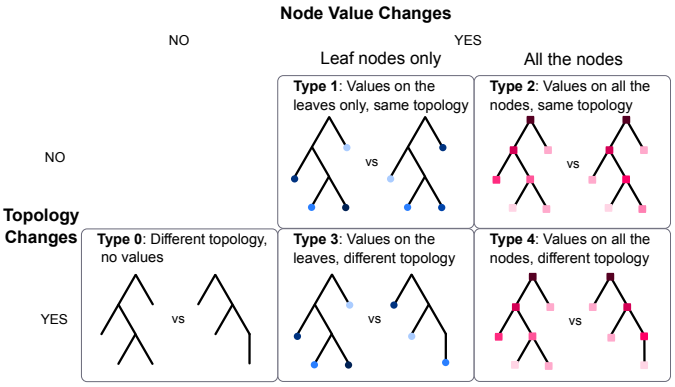

Fig. 3. Types of tree comparison problems. Current literature has addressed Types 0 and 1, with only one attempt at Type 3 [4]. TreeVersity2 supports all five cases, with emphasis on Types 1-4, the ones that include node value changes.

Type 0: Topological differences between two trees where the nodes contain only a label. Example: Finding differences between two phylogenetic trees, or trees of species, where biologists want to identify which species are in the same position on the tree, which are moved, appeared or disappeared.

Type 1: Positive and negative changes in leaf node values with aggregated values in the interior nodes (i.e. trees that can be visualized with a treemap [7]) and no changes in topology. Example: Comparing the stock market's closing prices between today and yesterday across a hierarchy of market sectors, assuming no stocks are created or deleted.

Type 2: Positive and negative changes in leaves and interior node values with no changes in topology. Example: Comparing the salaries in an organizational chart between two years, when no reorganization has occurred (note: salaries do not aggregate up the interior nodes)

Type 3: Positive and negative changes in leaf node values with aggregated values in the interior nodes and with changes in topology. Example: Finding changes in the U.S. Federal Budget, given that agencies or bureaus have been created or terminated.

Type 4: Positive and negative changes in leaves and interior node values, with changes in topology. Example: Comparing the number of page visits in a website between two months using the file hierarchy as a natural organization of the pages. Some pages might be created or removed, and each page in the hierarchy has an independent number of visits.

1.2 Characteristics of node changes

According to related work and our own experience, analysts want to be able to find and understand the following dimensions of change in the tree:

Direction of change: positive, negative or neutral (no change).

Absolute change: the actual amount of change, e.g. the Department of Defense budget decreased by 15.99 billion dollars between 2012 and 2013.

Percentage change: the absolute change with respect to the original value, e.g the change for Department of Defense represents a 2.32% decrease compared to the 2012 budget.

Relative change: how does the change for one node compares to the changes for other nodes in the tree, e.g. The cut in the Department of Defense ($-15.99 Billion, -2.32%) is considerably smaller than the decrease in the Department of Labor's budget ($-52.66 Billion, -29.84%).

Created and Removed: Which nodes were created, removed, or moved. e.g. The Bureau of Engraving and Printing ($140 million dollars) was to be removed from the Department of Treasury on 2013.

As described in Section 2, researchers have proposed a significant number of solutions for comparing trees on topology (Type 0)

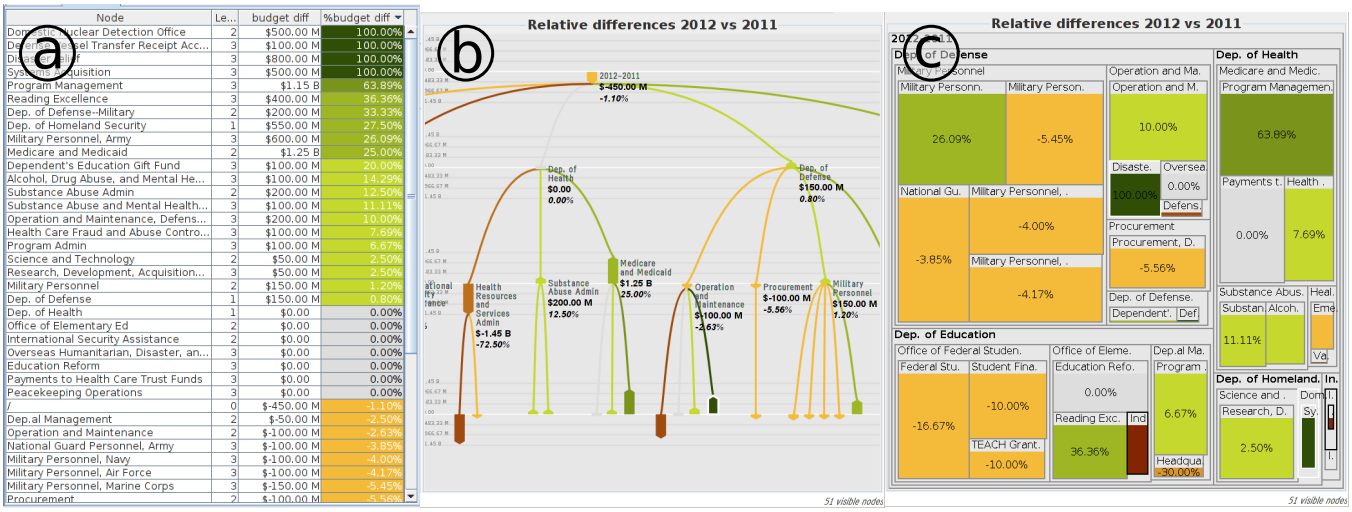

Fig. 2. Different ways of showing changes between trees: (a) table representation, (b) bullet visualization, (c) treemap representation.

[14, 11, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31] or for visualizing changes in node values with aggregated values in the interior nodes (Type 1). To the best of our knowledge, only one project [4] has attempted combining both types of differences at the same time (Type 3). TreeVersity2 is a tree comparison tool that tackles a richer set of problems by combining a novel visualization technique, interface design with coordinated views, interaction techniques and a comparison report algorithm to address all five types of tree comparisons, with the constrain of created and removed nodes on the topological comparisons.

2 Related Work

2.1 Tree Comparison

Most tree comparison work has focused on comparing topological changes between tree structures. This may have been influenced by the well-known problem of comparing taxonomies of species. TreeJuxtaposer by Munzner et al. [23] is one of the better known examples, presenting an efficient algorithm for comparing hierarchies. It uses a node-link representation with side-by-side comparison and a focus+context technique with guaranteed visibility. TreeJuxtaposer scales well with the number of nodes. MultiTrees by Holten & van Wijk [32] also compares two tree structures using side-by-side Icicle-like [8] representations, mirroring one of them and drawing connections between the tree's nodes using Hierarchical Edge Bundling [33] to reduce clutter. MultiTrees connections can get very busy, but are very useful to represent splits and joins between subtrees.

Other good examples of side-by-side comparison are Graham & Kennedy's [16] Icicle-like [8] representation and Bremm et al. [31] node-link visualization. These two solutions scale to tens of trees by dividing the screen space into small interconnected views of the compared trees, but are limited by the screen size. In later work [29] Graham & Kennedy addressed this by switching from the small multiples to an aggregated representation using directed acyclic graphs (DAG). Others have used the concept of mixing DAGs and trees such as Furnas et al. [14]; CandidTree [28] used the concept with a node-link representation that uses color, shapes and dotted lines to represent uncertainty. Amenta and Klingner's TreeSet [19] takes a different approach to comparing a large number of taxonomies by calculating a bi-dimensional metric representing each tree and plotting them in a scatter plot. TimeTree [27] explored the concept of time changing hierarchies, combining Degree of Interest Trees (DOITrees) [34, 35] with time sliders to analyze hierarchies that evolve with time.

The InfoVis2003 contest [36] promoted the development of projects on topological tree comparison. Some of the winning submissions presented innovative solutions for the problem, such as TreeJuxtaposer [23], already described. Others include Zoomology [20] which used radial representations combined with zooming interfaces, InfoZoom [18] which used condensed side-by-side tables, EVAT [21] with radial side-by-side comparisons, and TaxoNote [22] with a condensed Microsoft Windows Explorer-like representation. However, many of these promising projects did not published anything else beyond the competition's two page submission requirement.

Finally other approaches use zooming interfaces such as Moire-Trees [26], which allows navigation of multi hierarchies (different trees that categorize a shared group of leaf nodes) using zooming and radial displays, and DoubleTree [24], that uses two connected, side-by-side SpaceTrees [11] to highlight topological differences between taxonomies.

Despite the substantial work on topological differences between trees, to the best of our knowledge, none of these solutions addresses the problem of comparing changes in node values. TreeVersity2 takes the task of comparing tree structures changing over time one step further, by looking also at created and removed nodes. However more complex topological comparison features already supported by these projects, like finding moved nodes and subtrees, have not yet been addressed in the TreeVersity2 design. More specifically, TreeVersity2 performs topological comparison of two trees, by identifying created and removed nodes and revealing changes in the node values, tackling a richer set of problems than those that are restricted to topological differences only.

2.2 Node Values Comparison

The work on comparing node values is more limited, usually employing treemaps. The original treemap tool [7, 5] allowed the display of one change value on the hierarchy but it was never expanded to allow further comparisons. For example SmartMoney's Map of the Market [6] represents stock market price changes using a treemap with nodes colored green for increases or red for decreases [1]. This approach has proven to be popular, however it only presents relative differences in the leaf nodes without topological changes, i.e. what we called problem Type 1 in the introduction. Animated TreeMaps [37] represent changes in the nodes' attribute values using animation, by stabilizing the layout. Both projects rely on user's memory to keep track of the amount of change and the location of the nodes which can be taxing and confusing. TreeVersity2 in contrast allow users to navigate differences in a more explicit way.

Contrast Treemap [4] is to the best of our knowledge, the only project that compares two trees using aggregated node value changes

[1] http://www.smartmoney.com/map-of-the-market/

and topology differences (tree comparison problem Type 3). It modified the traditional treemap technique by splitting each of the nodes' rectangular shapes into two complementary color triangles. The color shade and hue, and the areas of the triangles are used both to represent node value changes and topology differences. We believe that Contrast Treemaps sets of colors can be improved using palettes that are more commonly associated with increases and decreases, but the combination of node values and topology differences in one feature (the color) might lead to information overload. The use of treemaps facilitates the comparison of the biggest nodes in the tree, and hides the smaller one. However, Contrast Treemaps, are limited to aggregated Trees (problems Types 1 and 3) and they do not represent created and removed nodes. In contrast TreeVersity2 shows changes in multiple levels of the tree (opposed to only the leafs), works with non-aggregated trees (problems Type 2 and 4), and highlights the created and removed nodes.

The Multiple Skylines Graphs by Caemmerer is a visualization designed to show changes in datasets. It uses the concepts of variable width bar charts that are similar to the ideas used on the StemView, however it was not designed for tree structures and therefore does not support them. The Skylines were featured in an on-line article in the SAP Design Guild [38] and not formally published anywhere else. On the other hand, Brodbeck et al. work [39]on visualizing survey results use a area filling hierarchical visualization base with overlying line graphs. This is a similar technique to the one used for the StemView, but was not designed for showing change, and uses a different representation.

LifeFlow[40, 41] a temporal categorical data exploration tool, included an option for using non temporal attributes to compare different trees side by side. LifeFlow was the inspirational work of TreeVersity, however it only allows users to compare datasets by using side by side inspection and does not support complex comparison tasks.

3 Description of TreeVersity2

TreeVersity2 is a web-based interactive data visualization tool that allows the exploration of change over time in datasets using hierarchies. Users see multiple overviews of the entire time range, and can select two time points for detailed analysis while still maintaining contextual awareness.. In one of our case studies (explained in more detail in Section 4.1), TreeVersity2 allowed analysts from the Federal Drug Administration to compare changes in drug adverse effects reports between any two years between 2008 and 2012, while keeping the overall context of the tendencies for the whole five year period.

The time based visualizations are displayed on the left side of the main interface (Fig. 1) and show the entire time period. Users can switch between traditional timelines to compare actual values, or a custom visualization called the TimeBlocks for comparing differential values. The TimeBlocks use color boxes to represent differential change between sequential time points. An example is shown in Fig. 4, where decreases in the National Cancer Institute's (NCI) lung cancer death index are depicted with green boxes (green being good), and increases are red. Each horizontal line in the TimeBlocks represents an attribute's value, with a corresponding node in the tree. For example in the Fig., there are three lines marked with letters (d) and (e), that represent the three races contained in the dataset (i.e. *White*, *Black* and *Other*). To facilitate the mental mapping of the TimeBlocks and the the tree nodes, mouse interactions are provided that highlight corresponding nodes or TimeBlocks when users move their cursor over them.

User can select two points of time on the overview and explore the detailed changes between them in the StemView (shown in the center of the screen, and explained in more detail in Section 3.1).The possible time points can be set up by the user depending on the data, and can range from seconds (e.g. comparing number of tweets in periods of five seconds), to decades (e.g. compare the number of publications in a research field by decades).

Finally, the Control Panel on the right side of the interface enables users to see the legend and change the variable mappings, see what changed the most (reporting tool), modify the hierarchy or set filters. Since the StemView represents change, there are five modifiers for each data value: the actual difference, the relative difference, the starting value, the ending value, or the maximum of the starting and ending values. Users can assign those variables to color, height, width or sorting order. Different combinations of mappings allow for richer explorations. For example, in Fig. 4 NCI analysts choose to explore the actual and relative changes in lung cancer death rates (represented with the color and height of the StemView boxes respectively), while still being able to gauge the size of the populations compared (depicted by the width of the StemView boxes).

The control panel also includes a novel textual reporting tool ("What changed" tab) that helps users directly find the major differences by reviewing a textual list of outliers calculated for each pair of compared time points. For instance Fig. 5 shows how the analysts at the Office of Management and Budget (OMB) could identify all the accounts decreasing the most in value in the US Federal Budget (more than $14 million dollars between 2012 and 2013), all while keeping the context of the entire budget hierarchy. The reporting tool is described in more detail in Section 3.2. Lastly, users can apply specific range filters for each characteristics of change. For example users can filter to show only the accounts in the US Federal Budget that have a budget larger than $10 million dollars, or all the accounts increasing or decreasing more than $1 million dollars. Smooth animations and transitions allow users to remain oriented in the tree. When filtering, the nodes that do not match the criteria are removed first then the remaining nodes are animated to occupy the available space.

TreeVersity2 allows the exploration of change over time in datasets using hierarchies. These hierarchies can be either **fixed**, when there is an inherent parent-to-child relationship (e.g. grouping the accounts in the U.S. Federal Budget by Agency and then by Bureaus, where grouping first by Bureaus and then by Agencies will not make sense, because each Bureau is part of only one Agency), **dynamic**, when the hierarchy is constructed by grouping rows by their attributes as defined in the original treemap paper [5] (e.g. Census population grouped by gender, race then age range), or **mixed,** where some levels of the hierarchy are fixed and some dynamic (e.g. grouping the U.S. Federal Budget by Discretionary/Mandatory Accounts, that is dynamic, and then by Agency/Bureau, that is fixed). For each of those hierarchy types the values can be **aggregated** if the values for the parent node are calculated as a function of the values of the children (e.g. adding up the values), or **non aggregated** if the values of the parent nodes are independent from the values of the children (e.g. The FDA's hierarchy of adverse effects of a drug presented in Fig. 4.1, where the values of the parent nodes are not calculated from the values of their children). Users can add or remove levels in the hierarchy, or swap the order of the levels of dynamic hierarchies.

TreeVersity2 was designed to allow rapid interactions with datasets on the order of hundred of thousands of records, which generate trees with thousands of nodes. This is achieved through client and server workload distribution. When the browser sends a request to the TreeVersity2 server, it is first processed with Python on a Django application server. It then accesses a PostgreSQL database that hosts the full extent of the dataset. The returned SQL query is a preprocessed data structure that is significantly smaller than the full database while still containing the information necessary to build the tree according to the parameters sent by the user (encoded in the URL). This data structure is then sent back to the browser, where a JavaScript application process it using the Crossfilter library, where it then draws all the visualizations using the D^3 [42] visualization library. TreeVersity2 also uses other libraries like Bootstrap, JQuery and JQuery UI, RequireJS and LESS. The use of this combination of technologies make TreeVersity2 flexible enough to support a wide range of datasets, as demonstrated by the 12 real world case studies developed with partners from government, industry and academia.

3.1 The StemView

The StemView is a novel visualization that represent changes using an area filling representation inspired by the icicle trees [8], where the levels of the hierarchy are distributed vertically in equally sized

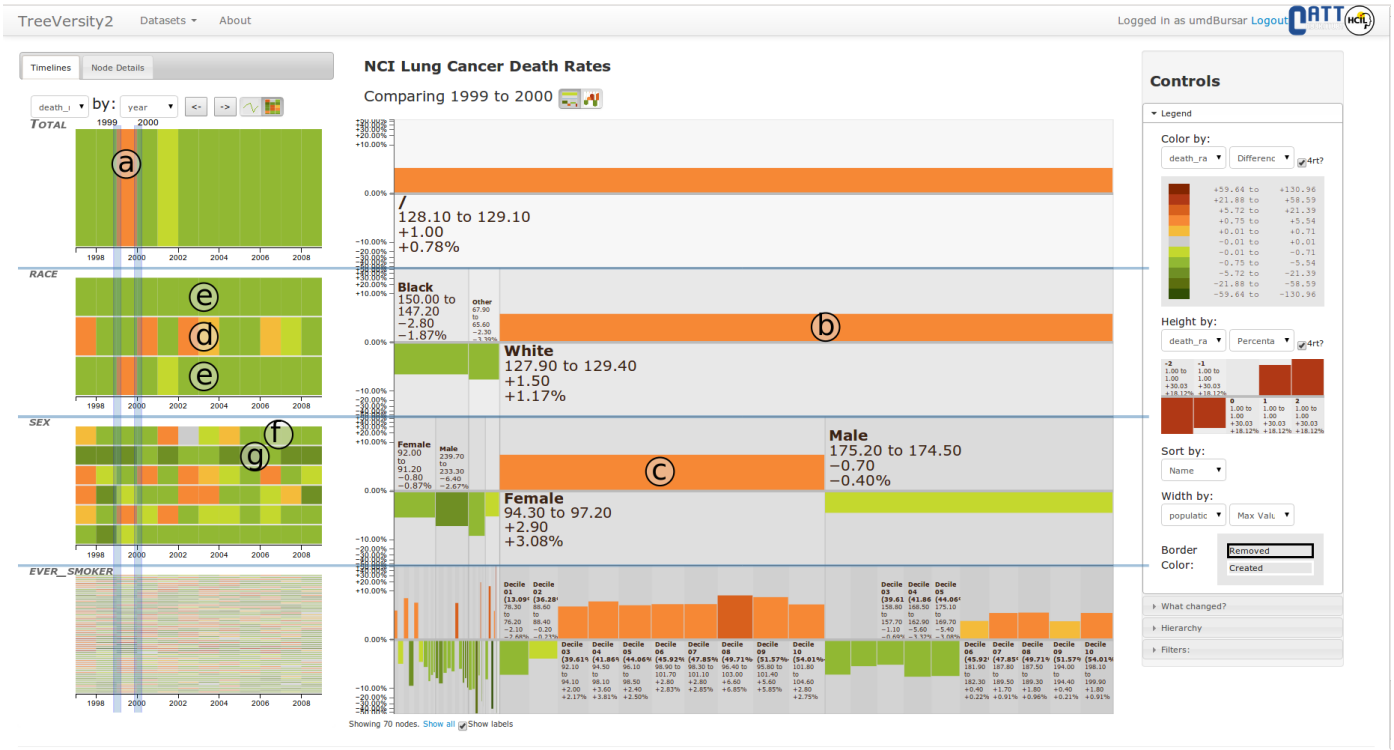

Fig. 4. National Cancer Institute Lung Cancer related death-rate change between 1999 and 2000 in the US. Color shows absolute change in the death-rate, while height represents the relative change (or percentage of change). The width encodes the population size for each group. The TimeBlocks show that the (a) overall rate increases only in 2000, however (b) the only race increasing is "White", that also happens to be more than 80% of the population. Among whites though, (c) women seem to be the ones contributing the most to the increase.

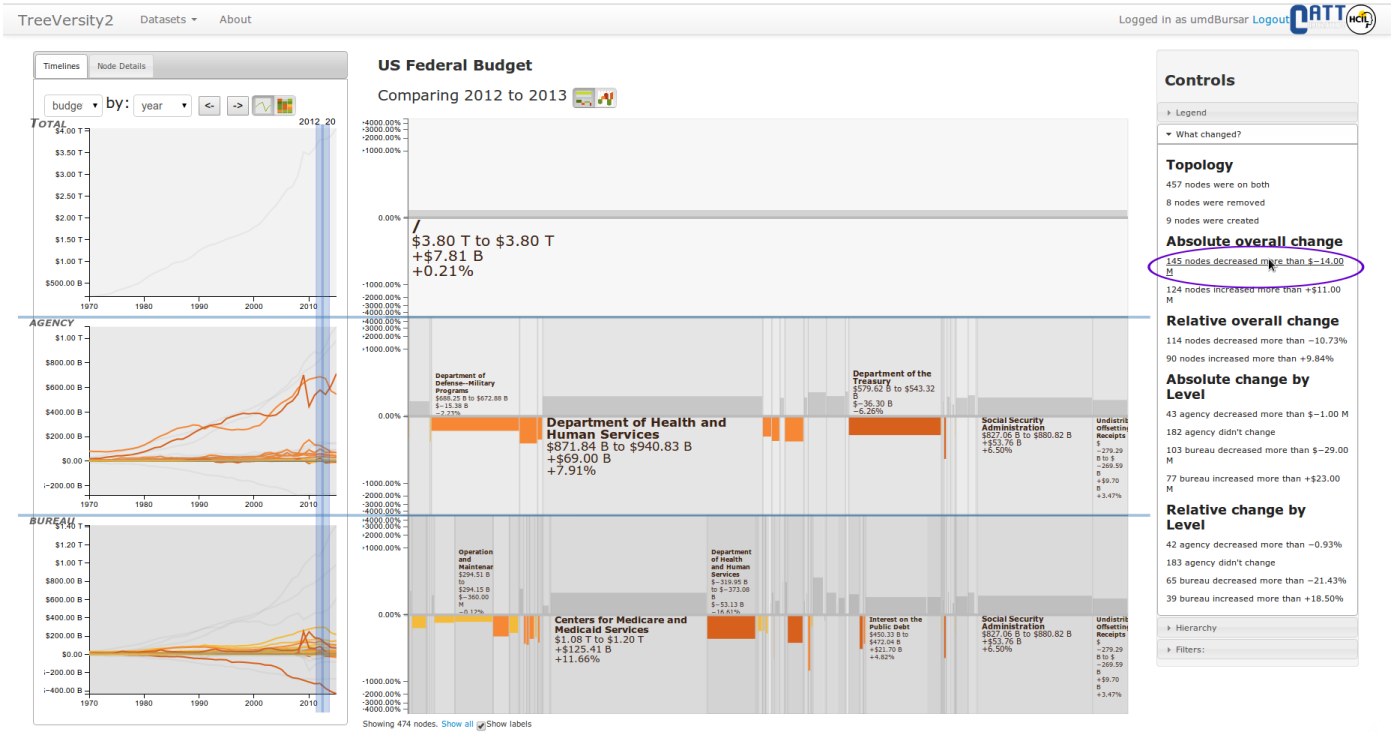

Fig. 5. The Reporting tool on the right side. One of the report entries was selected, highlighting all the agencies and bureaus in the US Federal Budget that decreased the most (here more than $14 million dollars). Users can filter the display to show only those accounts by clicking on the corresponding line of text in the reporting tool.

rows. Fig. 6 shows an example StemView constructed for the US Federal Budget, between 2008 and 2009, aggregating the budget accounts by their pertinence to the budget (On or Off budget), and by their Budget Enforcement Act Category (BEA, that determines if they are Discretionary, Mandatory or Net Interest). The vertical space is shared equally among the levels, and then within each level the horizontal space is shared among the nodes, represented as boxes, here sized according to their ending budget value. Fig. 6(a) shows this first step, that is basically an icicle-tree showing the budgets of each node for 2009. The StemView expands on the icicle to also show additional dimension of change, here the actual and relative changes of each node. For this purpose it splits each level vertically so that then horizontal line represents zero change (see Fig. 6[b]]. Then from that zero line a sub-box is drawn with the same width of the node's containing box, but with a height proportional to the relative change of the node (e.g. +17.94% for the overall budget). The sub-boxes go upward from the horizontal line for increasing nodes, and downward for decreasing nodes (Fig. 6[c]). Fig. 6[d] shows the final step, where the sub-boxes are colored using the actual amount of change of each node (e.g. +$535.12 billion dollars) using two color scales. These scales are typically greens for increasing values and yellows-to-reds for decreasing values, but the color scheme can be customized for special purposes, as demonstrated in the Case Studies section. Finally, a thick white border is added around the sub-box of created nodes, and black borders are added to deleted ones. Each of the characteristics of the StemView: the height, width and color of the boxes, plus the sorting of the children under their parents can be assigned to any of the variables of the dataset and their modifiers (starting value, ending value, actual difference or relative difference).

3.2 The Reporting Tool

One of TreeVersity2 novel features is a change reporting tool that helps users locate significant changes in the tree. Every time the compared time points are moved, the reporting tool generates a new textual list of the major changes in the tree, grouped by type of change. Each item in the list describes a group of nodes and gives a node count and why they are interesting (e.g. *145 nodes decreased more than $-14 M*). Users can hover over an item in the report to highlight the corresponding nodes in the StemView and time visualizations, as shown in Fig. 5. To see them more clearly users can click on the report item to remove all the other nodes, leaving only the nodes referenced by the report item (and their grayed-out parents - for context), and explore from there by zooming, changing mappings or further filtering.

The current reporting tool find nodes based on topological changes (e.g. nodes created, removed), significant changes in the overall tree or by level only (e.g. find all Agencies that increase more than 20%). The current implementation classifies a node as significant when it is beyond 2.5 times the interquartile range, and additional outlier-finding algorithms are likely to be domain specific. An example was created for the analysts at the FDA in the case study detailed in Section 4.1 to detect adverse effects that start with a report index of less than 2.0 and then increase in more than 1.5.

While each report element can also be obtained by setting filtering manually, all of our case studies partners liked how the reporting tool facilitated the exploration process, reducing the number of steps and time needed to identify interesting changes. The information collected in our case studies also suggested that such reporting tool might help users get started with complex tools such as TreeVersity2.

4 Case Studies

To evaluate the potential of TreeVersity2 twelve case studies with partner organizations from government, industry and academia were developed. TreeVersity2 target audience are data analysts and data owners with deep knowledge about their data. Moreover TreeVersity2 requires a training process to obtain full benefit of its features. Because of this, the studies were developed using Multi-dimensional In-depth Long-term Case Studies as defined in [43]. Controlled experiments would have been inadequate because subjects will not have enough time or knowledge about the data to offer insightful feedback or performed exploration tasks. In a similar way, usability studies would have been more useful to provide feedback about specific components of TreeVersity2 rather than to evaluate it as a data exploration tool that requires training and knowledge of the data analyzed. The case studies were developed while Treeversity was still under development so we often used a "chauffeur-mode", were we sat with our partners to explore the data while we controlled the still hard-to-use interface. The studies were developed in periods of one to twelve months as the interface was refined, with periodic meetings to present new features, analyze newly obtained data and discuss findings.

Table 1 summarizes the studies, providing information on the size of the datasets, an size of an example tree generated with the data, the number of attributes and variables, the types of hierarchies created, and the type of tree comparison performed. Ten of the case studies were developed with partner organizations independent from the authors. Because of space limitations, this section describes only two of the case studies in detail. A more complete reference of the case studies can be found on [2] and for demos visit `http://treeversity.cattlab.umd.edu`. The demos for eBay product sales, the UMD Student Demographics and the Department of Transportation TRB publications are not publicly available because they contain sensitive data.

4.1 FDA Adverse Drug Effects

In this case study, TreeVersity2 was used to help analysts at the Federal Drug Administration identify changes in the number of adverse effects reports for an undisclosed drug. To characterize the significance of adverse effect reports for a certain drug, FDA's analysts use the Empiric Bayes Geometric Mean (**EBGM**) index. In lay-terms the EBGM gives an index of how many more than expected reports of an adverse effect have been received [44]. An EBGM value of 1.0 denotes an adverse effect with the expected number of reports, values bigger than 1.0 are "bad" and smaller than that are "good". The EBGM values are organized in a *fixed*, *non-aggregated* hierarchy defined by the Medical Dictionary for Regulatory Activities (MedDRA[2]), that consists of four levels (SOC->HLGT->HLT->PT), and groups effects by body systems.

Analysts at the FDA have been using a treemap based visualization called the Sector Maps [45] that shows the EBGM values for the adverse effects reported for a drug in a certain year. Analysts wanted to find changes in the EBGM values between years, and the only way of doing it was switching back and forth between the Sector Maps, or using side by side comparisons as in Fig. 7. A new treemap visualization could have been used where the color represented the change in the EBGM value, but doing so would hide the changes in the inner nodes of the hierarchy. This was undesired since analysts wanted to explore changes in the EBGM values in all the levels of the MedDRA hierarchy, while still watching the count of reports per adverse effect (and several other variables). Moreover, they wanted to highlight the adverse effects with non-overlapping confidence intervals and their current solutions were insufficient for addressing all these requirements.

Fig. 8 shows the changes between EBGM values for an undisclosed drug between 2010 and 2011 using TreeVersity2. Each box in the StemView represents an adverse effect, yellow-to-red colored sub-boxes denote adverse effects with non-overlapping confidence levels. Height encoded the relative change of the EBGM index, so sub-boxes going up represent adverse effects getting more reports (with a fourth root scale). Finally the width of the boxes shows the total number of reports by effect, so more significant effects have wider boxes. With these encoding, analysts were able to find that in 2011 the *Pulmonary Embolism* went from not having any reports in 2010 to having a EBGM score of 25.20 which is really bad. Analysts reported that "it was incredible that we could see that important effect this way" and "it was significant given the drug in question". They are interested in using TreeVersity2 in a regular basis and praised TreeVersity2's visualizations for encoding many of the variables they needed

[2] `http://www.meddramsso.com/`

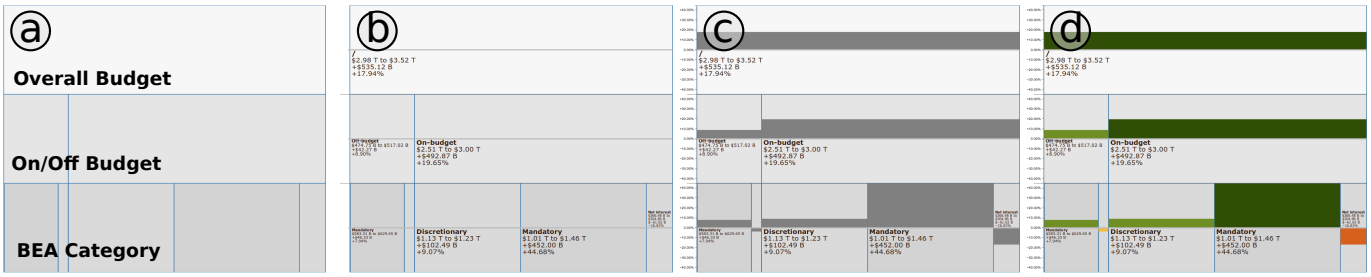

Fig. 6. Steps of the StemView construction using a commonly used mapping for budget data: (a) First an icicle tree for the ending values is drawn, (b) then inside each level a horizon line is added to represent the no change reference. (c) Sub-boxes with height corresponding to the relative % change are drawn inside each node. (d) Finally the nodes are colored using the actual dollar amount of change.

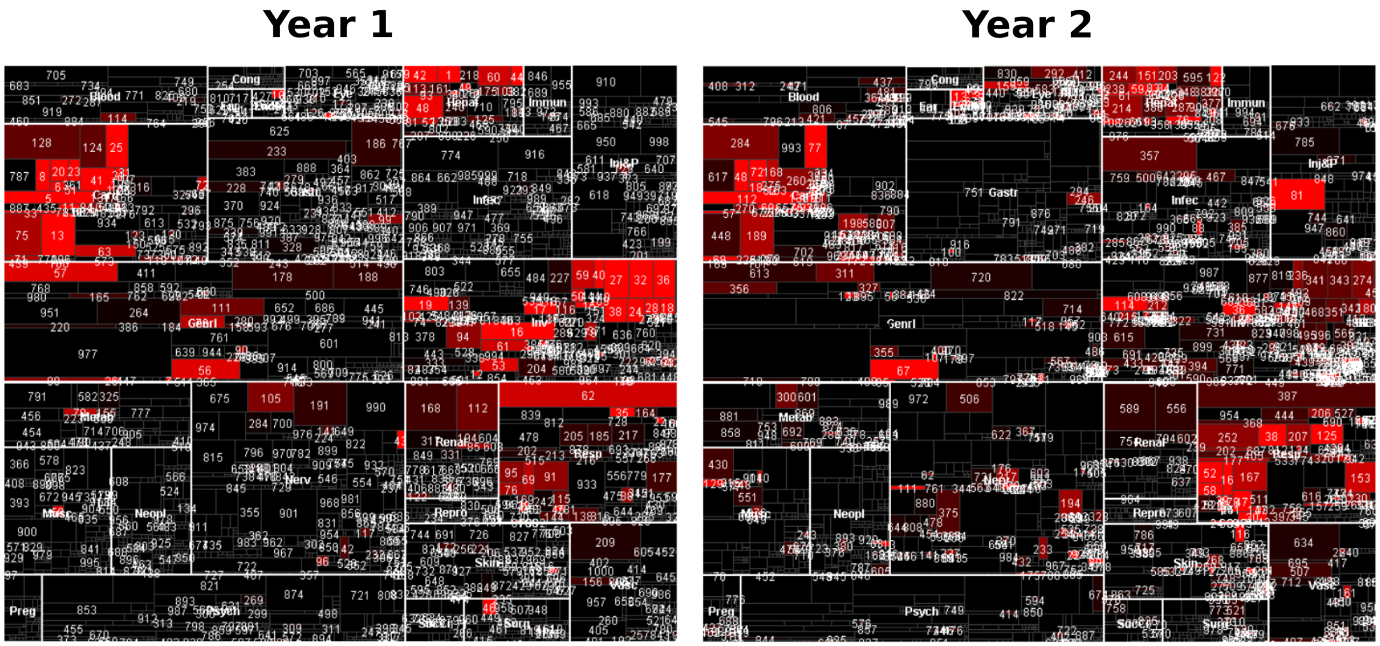

Fig. 7. Sector Maps for the EBGM values of a non-disclosed drug for 2011 and 2010. Each box represents a type of adverse effect, with red values encoding high EBGM, which is “bad”. EBGM Values of the inner nodes cannot be seen, and could be exposed only by redrawing the Sector Map at a different level. FDA Analysts rely on side by side comparisons like this to identify changes effects of interest, then have to review differences for individual effects one by one.

Table 1. TreeVersity2 Case Studies

Organization	Case Study	MILCS Stage	Driving Mode	Num. Of Records	Time Points	Example Tree Size	Number Attribs.	Number Vars.	Type of Tree
OMB	**US. Federal Budget**	Early	Chauffeur	4,845	56	1,393 (4 Levels)	7	1	Mixed
DOT	**TRB Publications**	Early	Chauffeur	52,135	8,012	674 (2 Levels)	20	1	Dynamic
DOT	**Nat. Trans. Library Publications**	Early	Chauffeur	38,351	374	294 (3 Levels)	10	1	Dynamic
DOT	**Passengers flying in the US**	Early	Chauffeur	65,534	162	4,194 (3 Levels)	4	1	Mixed
NCI	**National Cancer Institute**	Early	Chauffeur	1,716	13	101 (3 Levels)	3	3	Dynamic
FDA	**FDA Drug Adverse Effects**	Mature	Chauffeur	2,964	5	1,614 (4 Levels)	4	4	Fixed
UMD	**UMD Budget**	Early	Chauffeur	16,332	5	1,296 (3 levels)	6	1	Mixed
UMD Bursar	**UMD Students Information**	Mature	Chauffeur	227,158	5	715 (5 Levels)	219	3	Mixed
eBay	**eBay Product Sales Data**	Early	User-driven	63,098	4	5,443 (4 Levels)	6	2	Fixed
CATT Lab	**Transportation Bottleneck Data**	Early	User-driven	96,205	24	286 (3 Levels)	7	4	Mixed
IDB	**Imports and Exports in the Americas**	Early	User-driven	119,741	19	3,766 (4 Levels)	5	1	Dynamic
DUTO	**Blind Students in Colombia**	Mature	User-driven	33,802	4	1,098 (3 Levels)	21	1	Mixed

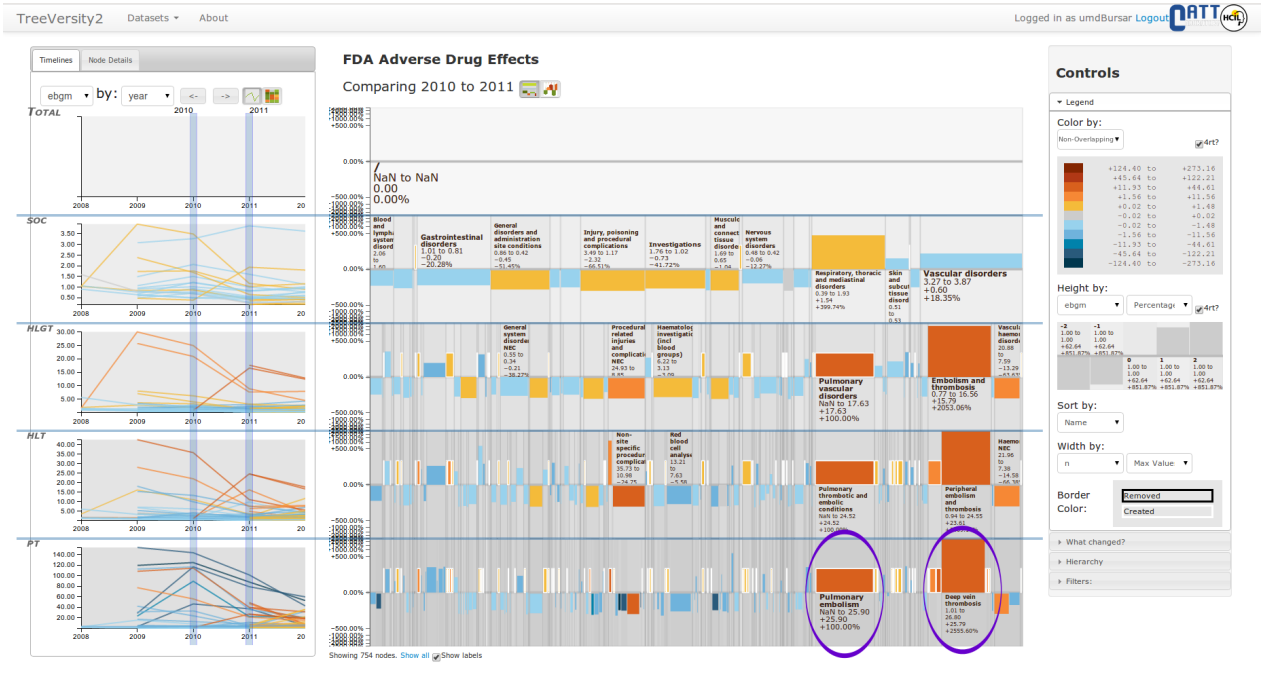

Fig. 8. Changes in the FDA's EBGM index of adverse effects (e.g. Pulmonary Embolism) for the same drug between 2011 and 2010. Analysts were able to rapidly identify two major growing adverse effects in the context of other changes. We circled *Deep Vein Thrombosis* and *Pulmonary Embolism* (which was not even reported in 2010 as indicated by the thick white border indicating a create node). The EBGM index is distributed in a fixed, non-aggregated tree and it is a measure of how many more reports than expected are received for a certain adverse effect. Boxes going up from the no-change line are showing an adverse effect getting worse (boxes going down the effects that are reducing). The width of the boxes in the StemView represents the total number of reports, so wide boxes are more important. The color scale was specially customized to draw attention to the adverse effects whose indexes's confidence intervals did not overlap (shown in yellow to red). Therefore, analysts started with the wide, red boxes going up.

for the comparison in one single view, as well as the possibility of exploring changes over a period of time. However, they were concerned with the color coding inconsistency with the Sector Map (The Sector Map shows high index values in red, while TreeVersity2 uses red for significant changes in the index value), and wanted to add even more variables to the interface, e.g. the seriousness of the effect, which may further increase complexity. A demo of TreeVersity2 with this case study as well as related videos are available at `http://treeversity.cattlab.umd.edu`.

4.2 National Cancer Institute

Analysts at the National Cancer Institute used TreeVersity2 to explore changes in lung cancer related death-rates in the US between 1997 and 2008. They calculated a normalized lung death-rate across the counties in the US, splitting them in ten comparable groups, i.e. by deciles, according to what percentage of the population have ever smoked. The dataset was also subsequently divided by ethnicity and gender; moreover, the population and death counts were also included with the data. For a first exploration of the dataset a dynamic, aggregated (using the average function) hierarchy was used, that grouped it by ethnicity, then by gender and finally by the counties deciles, as shown in Fig. 4. In the image, color represents the relative change of the death rate (decreasing values on green), and the height of the sub-boxes encode the actual change in the death rate. In order to highlight the group sizes, the value of the population counts (the max between the values of 1997 and 1998) of each node of the tree was selected for the width. Finally the TimeBlocks were used to compare the change of each group across time.

As illustrated in Fig. 4, analysts first noticed that the death-rate increased only in 2000 (a). They reflected on being able to see how this increase was due mainly to whites (b) in general, and to white female in specific (c). Other relevant findings show how the "other" race fluctuated between increases and decreases between years (d), when the remaining races decreased more consistently (e). They discussed that it might have been due to inconsistencies in the definition of the race "other" between years for the population count purposes. The initial exploration also suggested that African American men death rates (f) decreased more significantly than those of African American Females (g). From there the grouping order of the hierarchy was changed to Gender->Ethnicity->Counties-Deciles (not shown in the Figure) which confirmed the tendency. Analysts explained that this might have been due to smoking reduction campaigns being targeted mainly to men. Finally the hierarchy was changed again to put the grouping of the counties at the top, which revealed the expected correlation between the smoking and lung cancer death.

Analysts at the NCI were excited to see the changes in their datasets in a visual way, and liked the flexibility of TreeVersity2 to switch parameters. They discussed the idea that TreeVersity2 could be used to communicate their findings in an rich way to the general public, however they they were concerned with the learning curve required to understand the StemView.

5 Discussion and Conclusions

We presented TreeVersity2, an interactive data visualization tool that allows the exploration of changes in fixed or dynamic hierarchies, addressing direction of change, actual and relative change, starting and ending values, created and removed nodes, and inner nodes' values while keeping the hierarchy context. TreeVersity2 uses novel interactive data visualizations for exploring changes in numerical datasets between two time points (e.g. years), coordinated with an overview of the entire time period. TreeVersity2 includes a reporting tool to guide users through the major differences in the tree, which helps users get started with their analysis. Many challenges remain, several being classic problems found in many visualizations including:

- Currently TreeVersity2 interactions start to slow down with trees that have more than 7,000 nodes. This number is expected to improve with more modern web browsers that increase the speed of their Javascript engines. Also, improvements could be made on the drawing algorithm to reduce the number of svg shapes displayed on the screen, especially when the boxes are very narrow. Similar improvements could be made on the server side. Allowing users to exclude branches of deep trees for further exploration might also be useful.

- Displaying readable labels on the StemView and the timelines remains a challenge. Zooming is useful but users need to make heavy use of the tooltip to read labels. The current implementation labels boxes when possible, and resizes fonts dynamically to adjust to the available space, however smarter techniques could be implemented.

- When working with real datasets, it is common to find a small number of significant outliers, e.g. relative increases of 500,000% or more that overshadows the majority of the changes, e.g. in the ranges of 100%. TreeVersity2 allocates for this using a fourth root scale that emphasizes the smaller changes, but this solution makes it harder to compare values. Better customization controls could be included to provide more user control on this issue.

- The TreeVersity2 reporting tool only includes a small sample of metrics of interest. Future versions should allow users to save complex sets of filters, or add new custom algorithms.

- TreeVersity2 requires training. While interactive tutorials are helpful, an interface that would guide users through an orderly process of analysis might help novice users learn to use the tool more quickly.

In summary we designed, developed and evaluated TreeVersity2 in partnership with organizations such as the National Cancer Institute, Federal Drug Administration, Department of Transportation, Office of the Bursar of the University of Maryland, and eBay. While many challenges remain, the diversity of the characteristics of the datasets of these case studies illustrates the flexibility of TreeVersity2 and suggests that it is a useful tool for exploring what has changed over time.

Acknowledgments

We want to thank the Fulbright International Science and Technology Scholarship, the Center for Integrated Transportation Systems Management (a Tier 1 Transportation Center at the University of Maryland) and the Center for Advanced Transportation Technology Laboratory (CATT LAB) for partial support of this research; our case studies partners: David Rowe, Amanda Wilson, Pat Hu, Martin Akerman, Carol Kosary, Bradford Hesse, Anna Szarfman, Theresa Gil, Michelle S. Appel, Sharon A. La Voy, Andy Edmonds, Jeremy Harris and María Fernanda Zúñiga Zabala, for taking the time to work with us applying TreeVersity2 in real world problems; Mike Bostock and Jason Davies for their work on $D^3$and the crossfilter library; finally Audra Buck-Coleman for her thoughtful advice on design, that made previous versions of TreeVersity more beautiful and easier to understand, and that became later the aesthetic foundations of this work.

References

[1] J. A. Guerra Gómez, A. Buck-coleman, C. Plaisant, and B. Shneiderman, "TreeVersity: Visualizing Hierarchal Data for Value with Topology Changes," *Proceedings of the Digital Research Society 2012: Bangkok Vol 2*, vol. 2, no. July, pp. 640–653, 2012.

[2] J. A. Guerra Gómez, A. Buck-Coleman, M. L. Pack, C. Plaisant, and B. Shneiderman, "TreeVersity: Interactive Visualizations for Comparing Hierarchical Data Sets," in *Proceedings of the 2013 Transportation Research Board Annual Meeting*, 2013.

[3] J. A. Guerra Gómez, *Exploring differences in multivariate datasets using hierarchies, An interactive information visualization Approach.* PhD thesis, University of Maryland at College Park, 2013.

[4] Y. Tu and H. Shen, "Visualizing changes of hierarchical data using treemaps," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1286–1293, 2007.

[5] B. Shneiderman, "Tree visualization with tree-maps: A 2-d space-filling approach," *ACM Transactions on Graphics*, vol. 11, p. 92–99, 1991.

[6] M. Wattenberg, "Visualizing the stock market," in *CHI'99 extended abstracts on Human factors in computing systems*, p. 188–189, 1999.

[7] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proc. IEEE Conference on Visualization (Vis)*, pp. 284 – 291, IEEE, 1991.

[8] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162 – 168, 1983.

[9] J. Lamping, "The hyperbolic browser: A Focus+Context technique for visualizing large hierarchies," *Journal of Visual Languages & Computing*, vol. 7, p. 33–55, Mar. 1996.

[10] G. Robertson, J. Mackinlay, and S. Card, "Cone trees: animated 3D visualizations of hierarchical information," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, p. 189–194, 1991.

[11] C. Plaisant, J. Grosjean, and B. B. Bederson, "SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation," in *Proceedings of the IEEE Symposium on Information Visualization*, p. 57–64, IEEE, 1998.

[12] J. Heer and S. Card, "DOITrees revisited: scalable, space-constrained visualization of hierarchical data," in *Proceedings of the working conference on Advanced visual interfaces*, p. 421–424, 2004.

[13] S. Card and D. Nation, "Degree-of-interest trees: A component of an attention-reactive user interface," in *Proc Working Conference on Advanced Visual Interfaces*, p. 231–245, 2002.

[14] G. W. Furnas and J. Zacks, "Multitrees: enriching and reusing hierarchical structure," in *Proc. SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, (New York, NY, USA), p. 330–336, ACM, 1994. ACM ID: 191778.

[15] M. Graham, J. B. Kennedy, and C. Hand, "A comparison of set-based and graph-based visualisations of overlapping classification hierarchies," in *Proceedings of the working conference on Advanced visual interfaces*, p. 41–50, 2000.

[16] M. Graham and J. Kennedy, "Combining linking and focusing techniques for a multiple hierarchy visualisation," in *Information Visualisation, 2001. Proc. 5th International Conference on*, p. 425–432, 2001.

[17] M. Graham, M. F. Watson, and J. B. Kennedy, "Novel visualisation techniques for working with multiple, overlapping classification hierarchies," *Taxon*, vol. 51, no. 2, p. 351–358, 2002.

[18] M. Spenke, "Visualization and interactive analysis of blood parameters with InfoZoom," *Artificial Intelligence in Medicine*, vol. 22, no. 2, pp. 159–172, 2001.

[19] N. Amenta and J. Klingner, "Case study: Visualizing sets of evolutionary trees," in *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, p. 71–74, 2002.

[20] J. Y. Hong, J. D'Andries, M. Richman, and M. Westfall, "Zoomology: comparing two large hierarchical trees," *Poster Compendium of IEEE Information Visualization*, 2003.

[21] D. Auber, M. Delest, J. P. Domenger, P. Ferraro, and R. Strandh, "EVAT: environment for vizualisation and analysis of trees," *IEEE InfoVis Poster Compendium*, p. 124–125, 2003.

[22] D. R. Morse, N. Ytow, D. M. Roberts, and A. Sato, "Comparison of multiple taxonomic hierarchies using TaxoNote," in *Compendium of Symposium on Information Visualization*, p. 126–127, 2003.

[23] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, "TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility," *ACM Transactions on Graphics*, vol. 22, no. 3, p. 453, 2003.

[24] C. S. Parr, B. Lee, D. Campbell, and B. B. Bederson, "Visualizations for taxonomic and phylogenetic trees," *Bioinformatics*, vol. 20, no. 17, p. 2997, 2004.

[25] M. Graham and J. Kennedy, "Extending taxonomic visualisation to incorporate synonymy and structural markers," *Information Visualization*, vol. 4, no. 3, p. 206–223, 2005.

[26] M. J. Mohammadi-Aragh and T. J. Jankun-Kelly, "MoireTrees: visualization and interaction for multi-hierarchical data," 2005.

[27] S. K. Card, B. Suh, B. A. Pendleton, J. Heer, and J. W. Bodnar, "Timetree: exploring time changing hierarchies," in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, vol. 7, p. 3–10, IEEE, 2006.

[28] B. Lee, G. G. Robertson, M. Czerwinski, and C. S. Parr, "CandidTree: visualizing structural uncertainty in similar hierarchies," *Information Visualization*, vol. 6, no. 3, p. 233–246, 2007.

[29] M. Graham and J. Kennedy, "Exploring multiple trees through DAG representations," *IEEE Transactions on Visualization and Computer Graphics*, p. 1294–1301, 2007.

[30] M. Graham and J. Kennedy, "Multiform views of multiple trees," *Proceedings of the 2008 12th International Conference Information Visualisation*, p. 252–257, 2008. ACM ID: 1440153.

[31] S. Bremm, T. von Landesberger, M. Hess, T. Schreck, P. Weil, and K. Hamacherk, "Interactive visual comparison of multiple trees," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 31–40, IEEE, Oct. 2011.

[32] D. Holten and J. J. van Wijk, "Visual comparison of hierarchically organized data," *Computer Graphics Forum*, vol. 27, pp. 759–766, May 2008.

[33] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 741–748, Oct. 2006.

[34] S. K. Card and D. Nation, "Degree-of-interest trees," in *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '02*, (New York, New York, USA), pp. 231–245, ACM Press, May 2002.

[35] J. Heer and S. K. Card, "DOITrees revisited: scalable, space-constrained visualization of hierarchical data," in *Proceedings of the working conference on Advanced visual interfaces - AVI '04*, (New York, New York, USA), p. 421, ACM Press, May 2004.

[36] HCIL, "Infovis benchmark - PairWise comparison of trees." http://www.cs.umd.edu/hcil/InfovisRepository/contest-2003/, 2011.

[37] M. Ghoniem and J. D. Fekete, "Animating treemaps," in *Proc. of 18th HCIL Symposium-Workshop on Treemap Implementations and Applications*, 2001.

[38] "SAP Design Guild – UI Design Blinks (2012)," 2013.

[39] D. Brodbeck and L. Girardin, "Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree," in *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*, pp. 197–201, IEEE.

[40] K. Wongsuphasawat, J. A. Gomez, C. Plaisant, T. D. Wang, B. Shneiderman, and M. Taieb-Maimon, "LifeFlow: Visualizing an Overview of Event Sequences," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, (New York, New York, USA), p. 1747, ACM Press, 2011.

[41] J. A. Guerra Gómez, K. Wongsuphasawat, T. D. Wang, M. L. Pack, and C. Plaisant, "Analyzing incident management event sequences with interactive visualization," in *Transportation Research Board 90th Annual Meeting Compendium of Papers*, 2011.

[42] M. Bostock, V. Ogievetsky, and J. Heer, "D³ Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2301–2309, Dec. 2011.

[43] B. Shneiderman and C. Plaisant, "Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies," in *Proc. 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, BELIV '06, (New York, NY, USA), p. 1–7, ACM, 2006.

[44] A. Szarfman, J. M. Tonning, J. G. Levine, and P. M. Doraiswamy, "Atypical antipsychotics and pituitary tumors: a pharmacovigilance study.," *Pharmacotherapy*, vol. 26, pp. 748–58, July 2006.

[45] S. A. Rivkees and A. Szarfman, "Dissimilar hepatotoxicity profiles of propylthiouracil and methimazole in children.," *The Journal of clinical endocrinology and metabolism*, vol. 95, pp. 3260–7, July 2010.