Visualization of multi-dimensional data of bioactive chemicals using a hierarchical data visualization technique “HeiankyoView”

Takayuki ITOH* Fumiyoshi YAMASHITA**

*Department of Information Sciences, Ochanomizu University
**Graduate School of Pharmaceutical Sciences, Kyoto University

* itot@computer.org ** yama@pharm.kyoto-u.ac.jp

Abstract

In the age of combinatorial chemistry and high throughput screening, large-scale data of bioactive chemicals oriented to drug development are being accumulated. Due to the difficulties inherent in understanding such large quantities of data, information visualization techniques are increasingly attractive. Authors apply “HeiankyoView”, which is the technique for the representation of large-scale hierarchical data, for the visualization of multi-dimensional data of bioactive chemicals. In the present study, we investigated applicability of the visualization technique to the structure-activity relationship (SAR) analyses. The study first classifies chemicals according to similarity in their biological actions through self-organizing map analysis. It then applies a recursive partitioning method to find the relationship between biologically based categories and chemical structure, and finally it stores the drugs as hierarchical data. HeiankyoView is suitable for the visualization of such hierarchical data. This paper first describes the algorithmic overview of HeiankyoView, and then provides some example of visualization of multi-dimensional data of bioactive chemicals..

Keywords: Visualization, Hierarchical data, Rectangle packing, Combinational chemistry.

1 Introduction

Drug discovery and development is costly, time consuming, and high risk activity. The process starts with the discovery of a chemical or class of chemicals with particular biological activity. Lead compounds must then be identified, optimized, and only then tested in preclinical animal studies for efficacy, toxicity, etc. Those bioactive chemicals still considered viable after such rigorous scrutiny are then brought to human subjects for clinical evaluation of a variety of aspects of the chemical, including safety, effectiveness, and dosage determination.

Recent evolution of combinational chemistry and high-throughput screening technologies has brought exhaustive research of exploration of drugs. Large populations of chemical compounds are synthesized combinatorially, using sets of chemical "building blocks", and then subjected to automated biological assay systems. In vitro techniques of evaluating the metabolism and toxicity of drug candidates, which are major reasons of withdrawal of preclinical and clinical drug development, is also developing. These innovations have enabled collection of multi-dimensional and large-scale chemical compound data for drug discovery. Information technologies are potential to assist the selection and optimization of lead compounds and the selection of drug candidates; however, it is a very complicated problem to optimize the multi-dimensional large-scale data and design high-quality compound libraries. Information visualization techniques should be useful to discover characteristics from such multi-dimensional and large-scale data.

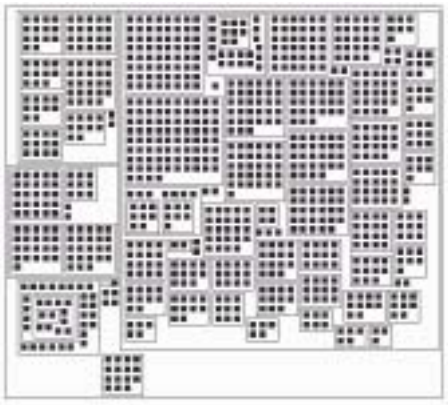

Figure 1. Example of hierarchical data visualization by HeiankyoView.

This paper presents the visualization of multi-dimensional data of bioactive chemicals by applying the hierarchical data visualization technique “HeiankyoView”. As shown in Figure 1, HeiankyoView visualizes the hierarchical data by mapping leaf-nodes as painted square icons, and non-leaf-nodes as rectangular borders. The technique targets to represent all leaf-nodes of large-scale hierarchical data in one display space without any focus-and-context techniques. One of authors has already applied the technique to visualize various data, including access trend of Web sites [Yam02], jobs in distributed computing environments [Yam03], distribution of network intrusion detection data [Ito05], and so on. These applications proof that HeiankyoView is useful not only for the overview of large-scale hierarchical data, but also discovery of interesting but minor local characteristics, and exploration of detailed local information.

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at *Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006)*, Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Cytochrome P450 (CYP) enzymes are a super-family of drug metabolizing enzymes that extensively affect the elimination of drugs from the body. Competitive interaction of simultaneously administered drugs with CYPs [Gue97] or genetic polymorphism of CYPs [Ing04] increase the drug level in the body unexpectedly, and sometimes cause lethal adverse reactions. Therefore, to characterize the interaction of investigational compounds with CYPs is an important issue in their safety evaluations. In the present study, authors experiment the visualization for the structure-activity relationship (SAR) analyses of CYP-related metabolism. Susceptibility of 161 drugs to major five CYP isoforms (i.e., 1A2, 2C9, 2C19, 2D6, and 3A4) was analyzed. Of more than 40 CYPs encoded in the human genome, only the five CYP isoforms account for 95% of hepatic drug metabolism [Ren97]. By understanding what molecular structural attributes relate to substrate specificity of each CYP isoform, design of molecules or their libraries is more effective. The drugs were classified into 6 groups according to their metabolic susceptibility to the CYP isozymes through self-organizing map analysis. Next, a recursive partitioning method was applied to find the relationship between metabolic susceptibility profile and chemical structure, and it finally stores the drugs as hierarchical data. HeiankyoView is useful for the visualization of such large-scale hierarchical data.

The reminder of this paper is as follows. Section 2 summarizes requirements for the visualization of multi-dimensional data of bioactive chemicals, and introduces existing hierarchical data visualization techniques. Section 3 introduces the algorithm of HeiiankyoView. Section 4 describes about experiments of visualization of combinational chemistry data and some results. Section 5 summarizes this study and discusses about the future works.

2 Required visualization techniques

This section first summarizes requirements for the visualization of multi-dimensional data of bioactive chemicals, and then introduces well-known information visualization techniques related to HeiankyoView.

2.1 Requirements

HeiankyoView is the hierarchical data visualization that satisfies the following requirements, which are definitely preferable for the applications described in Section 1 [Yam02][Yam03][Ito05], and also for the visualization of large-scale multi-dimensional data of bioactive chemicals.

[Requirement 1: No overlaps between leaf-nodes] Some visualization methods may cause overlap of leaf-nodes in defocused regions. HeiankyoView does not overlap them, so it provides a uniform overview of the data. It is also useful for detail-on-demand user interface, because it lets every leaf-node as clickable metaphor.

[Requirement 2: Efficient use of display spaces] It is often useful if visualization techniques pack all data items in a limited display space to provide a good overview. Traditional orthogonal tree-based systems, such as well-known file system viewers, have a bottleneck in that they may need a large display space if there are many nodes under one non-leaf node, or if there is a deep hierarchy.

[Requirement 3: Aspect ratio of subspaces] When visualization techniques subdivide a display space to represent the parts of the given data, squarish subspaces are usually preferable over thin subspaces so that users can visually recognize the parts. Therefore, aspect ratios of subspaces should be considered.

[Requirement 4: Flexible placement of arbitrarily shaped nodes] When data items are represented as rectangular icons, we often assume that the aspect ratios and sizes of all icons should be entirely unified. It is especially preferable if applications require representing all leaf-nodes equally. On the other hand, we also assume that the aspect ratios and sizes of all the icons should be specifiable by users, and they may even be varied. This makes it possible to visually emphasize important data items.

2.2 Related techniques

This section introduces well-know related information visualization techniques.

2.2.1 Space-filling hierarchical data visualization techniques

The technique subdivides display spaces to represent each portion of hierarchical data. TreeMaps [Joh91] recursively subdivides the display spaces into rectangular regions to form nested bar charts. Another technique subdivides the display space into sectors to form nested pie charts [Chu98]. HeiankyoView is somewhat analogous to TreeMaps because both techniques subdivide display spaces into rectangular area.

Variation improved TreeMaps have been recently proposed. Squarified Treemap [Bru00] subdivides display spaces into rectangles as much as square. Ordered Treemap [Shn01] subdivides display spaces and assigns the subregions in the predefined order of leaf-nodes. Quantum Treemap [Bed02] applies modified Squarified or Ordered Treemap so that it can represent leaf-nodes that are equally shaped and sized. Target of the Quantum Treemap is very similar to HeiankyoView, and actually Quantum Treemap satisfies requirements described in Section 2.1. Experiments described in [Ito04] discusses trade-offs between Quantum Treemap and HeiankyoView.

2.2.2 Other hierarchical data visualization techniques

Tree visualization techniques are other well-known hierarchical data visualization techniques. Hyperbolic Tree [Lam96], Cone Tree [Car95], and Fractal Views [Koi95] provides navigation and exploration capabilities for large-scale hierarchical data. Space-filling techniques have advantages against tree visualization techniques in the view of requirements 1 and 2.

HeiankyoView represents hierarchical data as two-dimensional nested metaphor. Three-dimensional nested metaphor has been applied by Information Cube [Rek93] and H-BLOB [Spr00]. These techniques require

capability and skill of 3D graphics, and adjustment of semi-transparency may be difficult for large-scale data visualization.

3 Hierarchical data visualization technique "HeiankyoView"

This section describes the hierarchical data visualization technique "HeiankyoView".

3.1 Overview

The technique represents leaf-nodes of hierarchical data as square icons, and non-leaf-nodes as nested rectangular borders. Figure 2 denotes the processing order of display layout of the nodes. The technique first places the leaf-nodes in the lowest-level of the hierarchy onto a display space, and represents a non-leaf-node by enclosing the leaf-nodes. The technique then places leaf-nodes or non-leaf-nodes in a higher level, and again encloses them by another rectangle. The technique represents whole the hierarchical data by repeating the similar process until it arrives at the highest level of the hierarchy.

Assuming the square icons represents leaf-nodes as rectangles, the visual items of the hierarchical data can be treated as a collection of rectangles. In other words, the technique solves the rectangle packing problem satisfying the following conditions, starting the lowest level, toward the top of the hierarchy:

[Condition 1] Rectangles never overlap each other.
[Condition 2] Area of rectangular region enclosing the placed rectangles is to be minimized.
[Condition 3] Aspect ratio of rectangular region enclosing the placed rectangles is to be optimized.
[Condition 4] If ideal positions of the rectangles are given, the rectangles are to be placed where enough close to the ideal positions. (This condition is ignored in this paper, but often the condition is quite important.)

The technique places the rectangles onto the display space one-by-one, selecting each position of the rectangles from multiple candidate positions. Original rectangle placement algorithm [Ito04] calculates the candidates by referring Delaunay triangular mesh connecting the centers of previously placed rectangles, but the further improved rectangle placement algorithm has been briefly reported [Ito03]. Authors measured the rectangle layout results and computation time of the proposed algorithm [Ito03] and proofed that it improved the results from the previous algorithm [Ito04]. This section describes the detail of the improved rectangle placement algorithm.

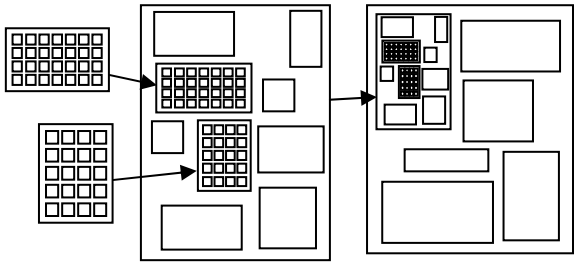

Figure 2. Processing flow of HeiankyoView.

3.2 Rectangle Placement algorithm

The visualization technique places rectangles in one level of the hierarchical data by the following processing order:

1. Select a rectangle R.
2. Calculate multiple candidate positions for R.
3. Repeat the following process for each candidate position, by the trial placement of R there:
   (a) If the candidate does not satisfy [condition 1], then select another candidate and repeat the similar process.
   (b) Otherwise, calculate the penalty value for other conditions, and record the candidate if the penalty value is smaller than any of penalty values of previously processed candidates.
4. Place R at the recorded candidate, and update the data structure for the placement of rectangles.

3.2.1 Definition of rectangles

The technique assumes that rectangles are placed onto a 2D display space represented as x-y orthogonal coordinates, and all edges of rectangles are parallel to x- or y- axis. This section describes parameters of a rectangle as follows;

w: width of a rectangle
h: height of a rectangle
x: x coordinate value of the center of a rectangle
y: y coordinate value of the center of a rectangle
u: x coordinate value of the ideal position of a rectangle
v: y coordinate value of the ideal position of a rectangle

Here the input and output of the rectangle placement algorithm is defined as follows:

Input: *w* and *h.*
Optional input: *u* and *v*.
Output: *x* and *y*.

Here we assume that *w* and *h* values of leaf-nodes are given by users, but *w* and *h* values of non-leaf-nodes are calculated when child-level of the non-leaf-node is processed. *u* and *v* are not used in this paper.

3.2.2 Definition of grid-like subdivision of display spaces

As shown in Figure 3, the proposed algorithm subdivides a display space by using extensions of edges of previously placed rectangles. The technique manages of occupancy of display space using the grid-like subdivision. Painted area in Figure 3(right) has been already occupied by a rectangle, and white area has not been yet occupied by a rectangle.

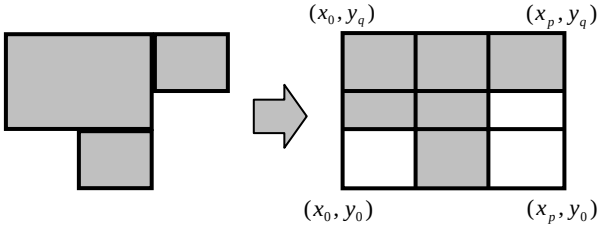

Figure 3. (Left) Already placed rectangles. (Right) Grid-like subdivision of display space by extension lines of edges of the placed rectangles.

Let us assume that the display space is divided into *p* along x-axis, and *q* along y-axis. Here the paper represents

the subdivision results as follows:

- x-coordinate values of *(p+1)* extension lines parallel to y-axis as x_0 to x_p.
- y-coordinate values of *(q+1)* extension lines parallel to x-axis as y_0 to y_q .

The paper also denotes a subdivided grid area, enclosed by extension lines $x=x_s$, $x=x_{s+1}$, $y=y_t$, and $y=y_{t+1}$, as *[s,t]*. Each grid area has a boolean value that denotes if it has been already occupied by a rectangle or not.

3.2.3 Calculation of candidate positions

Assume that *i-1* rectangles has been placed onto a display space, and the display space is divided into *p* by *q* grid subspaces. Here the technique attempts to place the *i*-th rectangle by calculating candidate positions inside the subspaces, and decide the best position to place the rectangle.

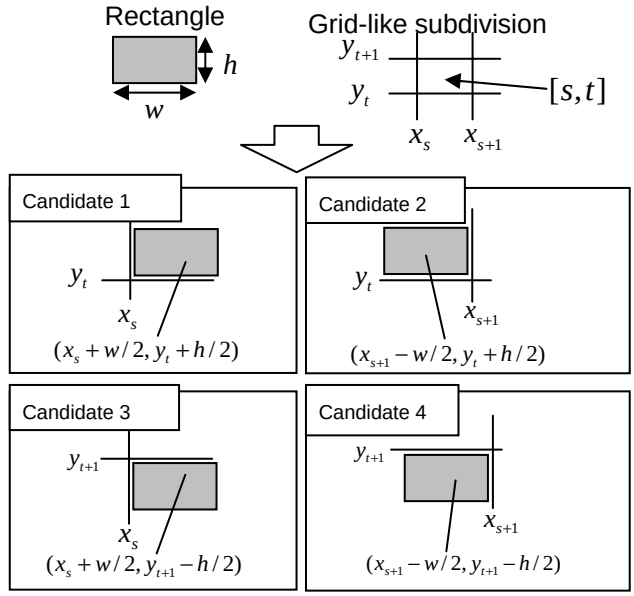

Figure 4. Four candidate positions for placing a rectangle.

As defined in Section 3.2.2, suppose x-coordinate values of the corners of the subspace *[s,t]* as x_s and x_{s+1}, and y-coordinate values of the corners of the subspace *[s,t]* as y_t and y_{t+1} . The technique first checks if the subspace *[s,t]* has been already occupied by a previously placed rectangle or not. If the subspace has not been occupied yet, the technique attempts to place the center of the *i*-th rectangle at the following four candidate positions:

Candidate 1: $((x_s + w/2), (y_t + h/2))$
Candidate 2: $((x_{s+1} - w/2), (y_t + h/2))$
Candidate 3: $((x_s + w/2), (y_{t+1} - h/2))$
Candidate 4: $((x_{s+1} - w/2), (y_{t+1} - h/2))$

As shown in Figure 4, the candidates are the positions that a corner of the *i*-th rectangle locates at a corner of the subspace.

3.2.4 Penalty values for candidate positions

Placing the *i*-th rectangle at each of the candidate positions, the technique first checks if the rectangle overlaps with any of previously placed rectangles. Supposing a candidate position as *(x, y)*, the technique specifies the integer values *j,k,l,m*, As shown in Figure 5(left), these values satisfy the following inequation:

$$x_j \leqq x-w/2 \leqq x_{j+1}$$
$$x_k \leqq x+w/2 \leqq x_{k+1}$$
$$y_l \leqq y-h/2 \leqq y_{l+1}$$
$$y_m \leqq y+h/2 \leqq y_{m+1}$$

The technique then checks the boolean values of the subspaces *[j,l]* to *[k,m]*, to check if any of the subspaces has been already occupied by a rectangle or not. If at least one subspace has been occupied, the technique gives up the candidate and deals with the next candidate.

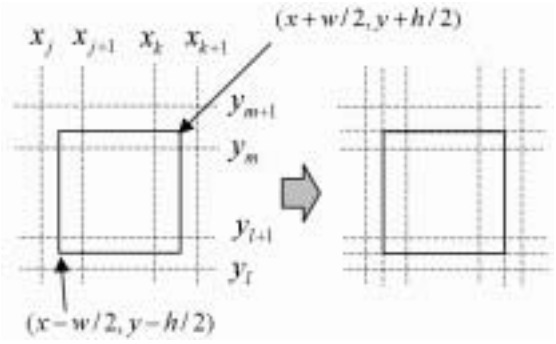

Figure 5. (Left) Integer values j,k,l,m and coordinates of vertices of a rectangle. (Right) Subdivision of grid.

The technique then calculates the penalty value for the candidate position using the equation (1), and decides to place the rectangle where the penalty value is minimum:

$$aD + bS + cA + dT \ldots(1)$$

where,

a,b,c,d: User-defined constant value,

D: Distance between the ideal position*(u,v)* of the rectangle and the position of the candidate *(x,y)*. *D* is ignored (treated as zero) in this paper.

S: Extension of area of the display space after the placement of the rectangle.

A: Aspect ratio of the display space.

T: Constant value lead by the below procedure.

T is calculated by the following procedure. Supposing the position of the corner of the rectangle lapping over a corner of a subspace as (x_s, y_t), the technique specifies the priority of the candidate position as follows:

Priority 1: If all of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 1. See Figure 6(upper-left).

Priority 2: If one of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 2. See Figure 6(upper-right).

Priority 3: If two of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 3. See Figure 6(mid-left).

Priority 4: If none of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 4. See Figure 6(mid-right).

The technique additionally calculates candidate positions outside the display space, if there is no candidate position adequate to place the rectangle. The outside

candidates are treated as lowest priority candidates as follows:

Priority 5: Outside candidates are categorized as priority 5. See Figure 6(lower).

The technique defines constant values $t_{k,}$ *(k=1..5)*, which satisfies the following inequation:

$$t_1 < t_2 < t_3 < t_4 < t_5$$

The above values are used as $T=t_k$ in the equation (1).

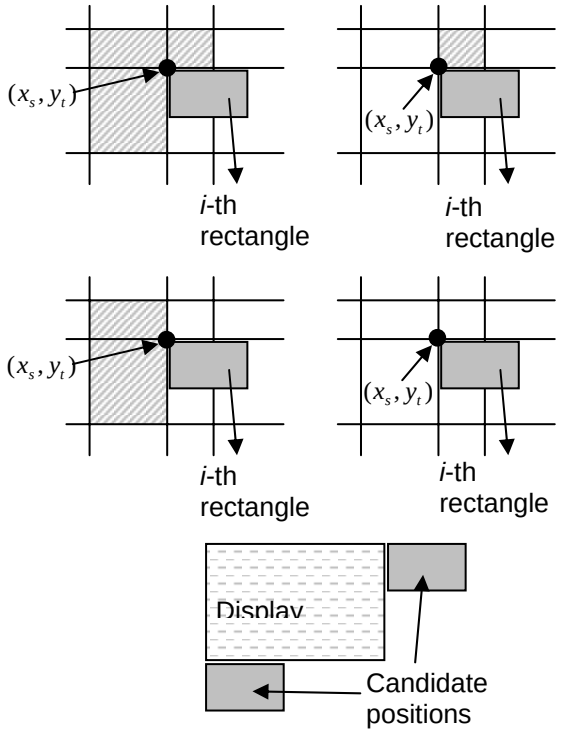

Figure 6. Five stages of priority of candidate

3.2.5 Decision of position of rectangles

The technique repeats the process described in Section 3.2.3 for all subspaces, and Section 3.2.4 for all candidate positions. Finally the technique decides the position of the *i*-th rectangle where the penalty value is minimum. As shown in Figure 5(right), the technique then divides the subspaces by the extension lines of the *i*-th rectangle, and repeats the similar process for *(i+1)*-th rectangle.

4 Experiments of visualization of multi-dimensional data of bioactive chemicals

This section described the experiments of visualization of multi-dimensional data of bioactive chemicals using HeiankyoView. In this experiments, authors used the metabolism data of 161 drugs against 5 CYPs (CYP1A2, CYP2C9, CYP2C19, CYP2D6, and CYP3A4).

4.1 Grouping of CYP isozymes according to their metabolic susceptibility

First stage of the experiment classifies the 161 drugs into 6 groups according to their metabolic susceptibility to the CYP isozymes. Here authors represented the characteristics of the drugs as five-dimensional vectors led from their metabolic susceptibility with each of 5 CYPs, and classified the drugs using self organizing map (SOM). Figure 7 shows an example of SOM, where the darkness of each node denotes the value of elements of the vectors. The SOM denotes that many drugs are metabolized by CYP3A4. It also denotes that substrate specificity of CYP1A2, CYP2C9, and CYP2C19 does not overlap with one another.

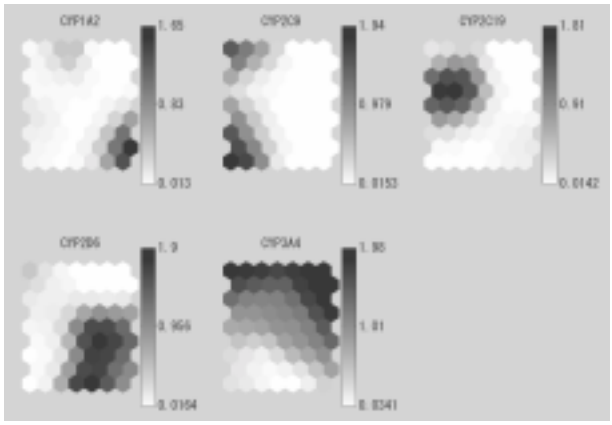

Figure 7. Example of SOM classified according to the metabolic susceptibility of 161 drugs.

Next authors divided the SOM nodes into 6 groups according to their Davies-Bouldin Index (DB) [Dav79]. They then divided the 161 drugs into 6 groups by mapping the drugs to their best-matched winning node. The group information was subjected to the following recursive partitioning analysis.

4.2 Recursive partitioning

Authors divided drugs into two subsets to maximally increase the internal homogeneity of groups within the subsets. Here, the internal homogeneity was defined as information entropy: *i.e.*, $\Sigma\{-P(s_i)\log_2 P(s_i)\}$ where $P(s_i)=n_i/N$; n_i and N are numbers of category i data and the total in (sub)data set, respectively. On the other hand, predictor variables for partitioning were molecular constitutional descriptors derived from chemical structure by using Dragon 5.2 (Talete srl, Italy) [Tod00]. Recursively repeating the division, they constructed a binary classification tree. The binary tree aimed at classification of the drugs varying in metabolic susceptibility based on their chemical structure.

4.3 Visualization by HeiankyoView

Authors visualized the hierarchical data of the 161 drugs using HeiankyoView. Figure 8 shows the example of visualization results, where icons denote the drugs, darkness of the icons denotes the metabolic susceptibility, and rectangular borders represent the group of drugs according to their chemical structures.

The visualization result denotes that the most of drugs metabolized by CYP1A2 or CYP2C19 are found in the right side of the display, but drugs metabolized by CYP2D6 or CYP3A4 are found in both left and right sides of the display. In the recursive partitioning analysis, the primary concern raised for classification was whether sum

of atomic Sanderson's electronegativity (*Se*) be less than 44.89 or not. Taking together, it was found that drugs metabolized by CYP1A2 and CYP2C19 mostly possess the *Se* less than 44.89. Such visualization result proofs that the proposed visualization technique can contribute to analyze the relationship between metabolic susceptibility profile and chemical structure.

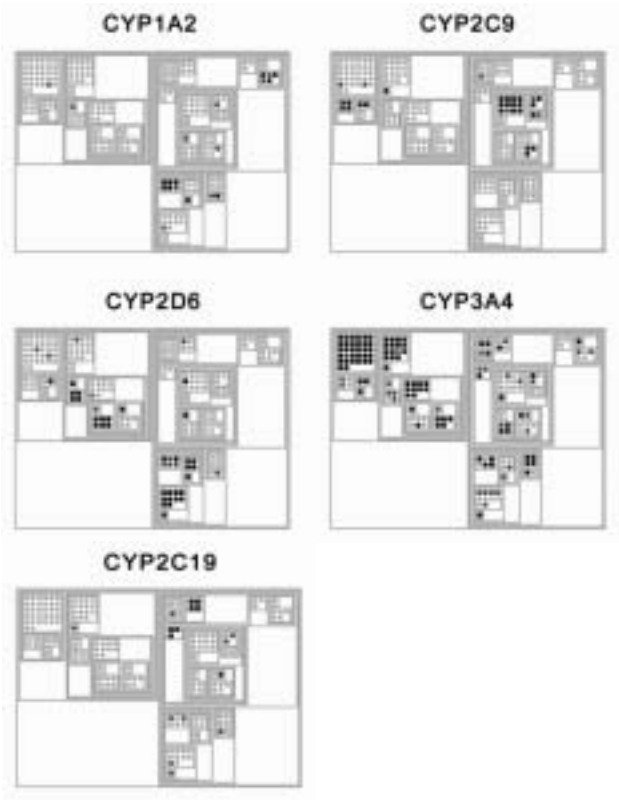

Figure 8. Visualization of hierarchical data of the 161 drugs using HeiankyoView. Dark leaf-nodes denote the drugs whose metabolic susceptibilities are high.

4.4 Discussion

This study was the first experiments for authors to deal with binary trees as hierarchical data. Visually comparing between Figures 1 and 8, HeiankyoView does not provide good display layout for binary trees as much as those of non-binary trees. Authors are currently studying some algorithmic improvement, especially for reducing waste spaces of layout results with binary trees.

Authors are implementing the metaphor of bioactive chemicals as clickable icons, so that users can obtain detailed information of the chemicals on demand. [Requirement 1] listed in the Section 2.1 is very important for developing clickable visualization techniques. HeiankyoView is advance in this point, as well as some of space-filling hierarchical data visualization technique such as Treemap. HeiankyoView is somewhat similar to Quantum Treemap [Bed02] because both techniques attempt to represent metaphors as equally- shaped and sized icons. Comparison between authors' previous technique and Quantum Treemap is provided in [Ito04], and authors are currently evaluating HeiankyoView and Quantum Treemap to provide the comparison.

5 Conclusion

This paper introduced an application of "HeiankyoView", a large-scale hierarchical data visualization technique, to the visualization of multi-dimensional data of bioactive chemicals. In the application authors constructed the hierarchical data of 161 drugs according to the chemical structure, and represented their metabolic susceptibilities as darkness of icons denoting the drugs. They proofed that the visualization technique can contribute to analyze the relationship between metabolic susceptibility profile and chemical structure.

Authors are improving the rectangle packing algorithm of HeiankyoView according to the issue discussed in the Section 4.4, and enhancing the implementation of the presented visualization technique according to the following issues:

- All-in-one representation of multi-dimensional values in one rectangular space.
- Filtering of non-interesting compounds.
- Sophisticated indication of detailed information.

In addition to improvements of visualization techniques themselves, it is also important to combine other computing techniques with the visualization techniques. For example, it should be useful if multi-dimensional data of virtual compounds are visualized with those of real compounds at the same time. The virtual compounds can be developed on a computer, and their multi-dimensional data can be predicted from the data of structurally similar real compounds. Such simulation and visualization can contribute to shorten the development cycle and expand the possibility of new compounds.

6 References

[Bed02] Bederson B., Schneiderman B.,Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, ACM Transactions on Graphics, Vol. 21, No. 4, pp. 833-854 (2002).

[Bon01] Bonnabry, P., Sievering, J., Leemann, T., Dayer, P. Quantitative drug interactions prediction system (Q-DIPS): a dynamic computer-based method to assist in the choice of clinically relevant in vivo studies. Clinical Pharmacokinetics, Vol. 40, pp. 631-640 (2001).

[Bru00] Bruls D.M., Huizing K., Wijk J. J., Squarified Treemaps, Proceedings of Data Visualization 2000, pp. 33-42 (2000).

[Car95] Carriere J., et al., Research Paper: Interacting with Huge Hierarchies beyond Cone Trees, IEEE Information Visualization 95, pp.74-81 (1995).

[Chu98] Chuah M., Dynamic Aggregation with Circular Visual Designs, IEEE Information Visualization '98, pp.35-43 (1998).

[Dav79] Davies D. L., and Bouldin D. W., A Cluster Separation Measure, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, pp. 22-227 (1979).

[Gue97] Guengerich F. P., Role of cytochrome P450 enzymes in drug-drug interactions, Advances in Pharmacology, Vol. 43, pp. 7-35 (1997).

[Ing04] Ingelman-Sundberg M., Pharmacogenetics of cytochrome P450 and its applications in drug therapy: the past, present and future. Trends in Pharmacological Sciences, Vol. 25, pp. 193-200 (2004).

[Ito03] Itoh T., Koyamada K., HeiankyoView: Orthogonal Representation of Large-scale Hierarchical Data, International Symposium on Towards Peta-Bit Ultra Networks (PBit 2003), pp. 125-130, 2003.

[Ito04] Itoh T., Yamaguchi Y., Ikehata Y., and Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 3, pp. 302-313 (2004).

[Ito05] Itoh T., Takakura H., Sawada A., Koyamada K., Interactive Poster: Visualization of Network Intrusion Detection Data Using a Hierarchical Data Visualization Technique, IEEE Information Visualization, 2005.

[Joh91] Johnson B., et al., Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, IEEE Visualization '91, pp. 275-282 (1991).

[Koi95] Koike H., Fractal Views: A Fractal-Based Method for Controlling Information Display, ACM Transactions on Information Systems, Vol. 13, No. 3, pp.305-323 (1995).

[Lam96] Lamping J., Rao R., The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, Journal of Visual Languages and Computing, Vol. 7, No. 1, pp. 33-55 (1996).

[Ren97] Rendic, S., Di Carlo, F. J., Human cytochrome P450 enzymes: a status report summarizing their reactions, substrates, inducers, and inhibitors. Drug Metabolism Reviews, Vol. 29, pp. 413-580 (1997).

[Rek93] Rekimoto J., The Information Cube: Using Transparency in 3D Information Visualization, Third Annual Workshop on Information Technologies & Systems, pp.125-132 (1993).

[Shn01] Shneiderman B., et al., Ordered Treemap Layouts, IEEE Information Visualization Symposium 2001, pp. 73-78 (2001).

[Spr00] Sprenger T. C., et al, H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces, IEEE Visualization 2000, pp. 61-68 (2000).

[Tod00] Todeschini R., and Consonni V., *Handbook of Molecular Descriptors*, Wiley-VCH, Weinheim and New York, 2000.

[Yam02] Yamaguchi Y., Itoh T., Ikehata Y., Kajinaga Y., Interactive Poster: Web Site Visualization Using a Hierarchical Rectangle Packing Technique, IEEE Information Visualization Symposium, 2002

[Yam03] Yamaguchi Y., Itoh T., Visualization of Distributed Processes Using "Data Jewelry Box" Algorithm, CG International 2003, pp. 162-169, 2003.