

Visualizing large hierarchical clusters in hyperbolic space

Jonathan Bingham and Sucha Sudarsanam

SUGEN, 230 East Grand Avenue, South San Francisco, CA 94080, USA

Received on November 10, 1999; revised on January 5, 2000; accepted on January 8, 2000

Abstract

Summary: *HyperTree* is an application to visualize and navigate large trees in hyperbolic space. It includes color-coding, search mechanisms and navigational aids, as well as focus+context viewing, allowing enormous trees to fit within the fixed space of a computer screen or printed page.

Availability: A demo is available online at <http://www.kinase.com/tools/HyperTree.html>

Contact: sucha-sudarsanam@sugen.com

Introduction

With the rapid increase in availability of genomic, protein and expression data, it has become possible and even essential to analyse data sets orders of magnitude larger than previously available. Large data sets obviously pose challenges for clustering algorithms, but they also challenge our ability to visualize, navigate and ultimately understand the data. To cope with data sets of enormous size, specialists in computerized data visualization have introduced a variety of ‘focus+context’ techniques that zoom in on a particular subset of data while keeping the broader picture in view (Leung and Apperley, 1994). One ‘fish-eye’ variant (Sarkar and Brown, 1994) employs a hyperbolic, non-Euclidean geometry (Lamping *et al.*, 1995). Hyperbolic geometry allows the display of an arbitrarily large structure (infinite, in fact) within a bounded, finite space such as a computer screen: Figure 1 makes this clearer. The resulting view can be rotated, translated and otherwise manipulated using transformation matrices (Phillips and Gunn, 1991).

HyperTree displays large phylogenetic trees, phenetic trees and hierarchical clusters of gene expression data in hyperbolic (or Euclidean) space. It takes as input a standard Phylip format tree (Felsenstein, 1989), such as can be readily generated using a variety of phylogenetic and hierarchical clustering algorithms. It then generates a graphical view in one of three styles: a linear dendrogram view, a radial view or a hyperbolic view. The hyperbolic view is generated by applying an invertible transform to the radial plot. For each polar coordinate

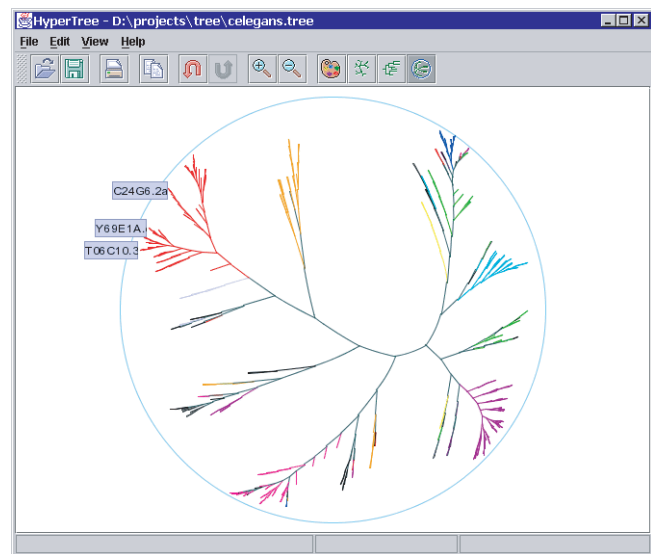


Fig. 1. *HyperTree* screenshot, showing *C.elegans* protein kinases

(r, θ) corresponding to the Cartesian coordinate (x, y) , we obtain the corresponding polar coordinate on the hyperbolic unit disk (r_h, θ_h) by transforming the radius so that $r_h = r/(r + 1)$. For display purposes, we then scale the hyperbolic unit disk to fill the screen or a printed page. For alternate implementations, see http://www.inxight.com/Demos/SLS_Demos/Site_Lens_Studio_Demos.html and <http://industry.ebi.ac.uk/~alan/BioWidget/>

Features

Navigation. In hyperbolic space, only a portion of the tree is ‘zoomed in’ at any given time. By translating the plot (holding the mouse button depressed while moving the mouse), users can bring different portions of the tree into focus. To rotate the image, users may hold down the shift key while dragging the mouse. To increase or decrease magnification, menu buttons, toolbar buttons and keyboard shortcuts are available.

Labels. Labels can be edited, rendered in any native font, and hidden if they prove too cluttered or distracting. Even when hidden, labels appear on mouse-over.

Searching. Users can select from a list of node labels (and in a subsequent version, enter a search string). The corresponding labels will be highlighted in the tree.

Selection. Clicking on a node label selects that node. Users can then select a subtree from a pop-up menu (on right click) or from the main Edit menu.

Color coding. Any selected node or subtree can be color coded using a menu item or tool button. The results can be saved to file. As this process can be laborious for large trees, HyperTree also supports the importing of color coded files with a simple text format. These can be generated programmatically for automated high-throughput analysis. For example, here are a few valid color code specifications: ZC104.1, green; ZC104.2,0,0,255; ZC104.3, kinase.

Implementation

Hardware requirements

HyperTree is written in Java 2; it currently runs on Windows, Linux and various Unix platforms. All performance-critical aspects of the visualization tool scale linearly with respect to the number of sequences. A standard personal computer with typical RAM and clock speeds can support trees of the order of 1000 nodes (the largest size tested).

Component-based architecture

HyperTree is written using a component-based architecture, which will enable it to be easily integrated with other software components, such as a multiple sequence alignment viewer, a genome browser or a relational database (Bingham *et al.*, 2000). Integrated software tools will be increasingly important for the future of bioinformatics as researchers need to combine genomic sequence analysis, expression analysis, proteomics, genetic maps, functional genomics and biochemical pathways (Stein *et al.*, 1994; Robinson and Flores, 1997), so we have designed HyperTree to be readily extended and integrated with new tools.

Discussion

As a practical application, we have used HyperTree to help analyze the 492 protein kinases in *Canorhabditis elegans* (Bingham *et al.*, 2000; Plowman *et al.*, 1999). In order to help classify novel kinases, we created a color file with a distinct color for each family (casein kinases, cyclin-dependent kinases, PKC-related kinases, tyrosine kinases, etc). Upon importing the color file, we were quickly able to identify familiar families, unexpected outliers and novel members. HyperTree will play a similar role in our analysis of *Drosophila* and the human genome over the coming year.

Acknowledgement

We thank Greg Plowman for critically using HyperTree and providing useful suggestions for

References

- Bingham,J., Plowman,G.D. and Sudarsanam,S. (2000) Issues in large-scale sequence analysis: elucidating the protein kinases of *C.elegans*. (Submitted).
- Felsenstein,J. (1989) PHYLIP—Phylogeny inference package (Version 3.2). *Cladistics*, **5**, 164–166.
- Lamping,J., Rao,R. and Pirolli,P. (1995) A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *ACM Computer-Human Interaction 1995 Proceedings*.
- Leung,Y.K. and Apperley,M.D. (1994) A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Computer-Human Interaction*, **1**(2), 126–160.
- Phillips,M. and Gunn,C. (1991) Visualizing hyperbolic space: unusual uses of 4×4 matrices. *ACM*. Stanford.
- Plowman,G.D., Sudarsanam,S., Bingham,J., Whyte,D. and Hunter,T. (1999) The protein kinases of *C.elegans*: a model for signal transduction in multicellular organisms. *Proc. Natl Acad. Sci. USA*, **96**, 13603–13610.
- Robinson,A. and Flores,T. (1997) Novel techniques for visualizing biological information. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*.
- Sarkar,M. and Brown,M.H. (1994) Graphical fish-eye views. *Commun. ACM*, **37**(12), 73–84.
- Shinsato,H. (1999) Writing high-performance graphical Java components. *Dr. Dobbs's J.*, **303**, 50–54.
- Stein,L.D., Rozen,S. and Goodman,N. (1994) The case for componentry in genome information systems. In *Proceedings of the Meeting on the Interconnection of Molecular Biology Databases*.