Comparison of HyperTree and Phylogenetic Tree.

HyperTree requires data in JSON format.

Phylogenetic Tree requires data in Newick or PhyloXML formats.

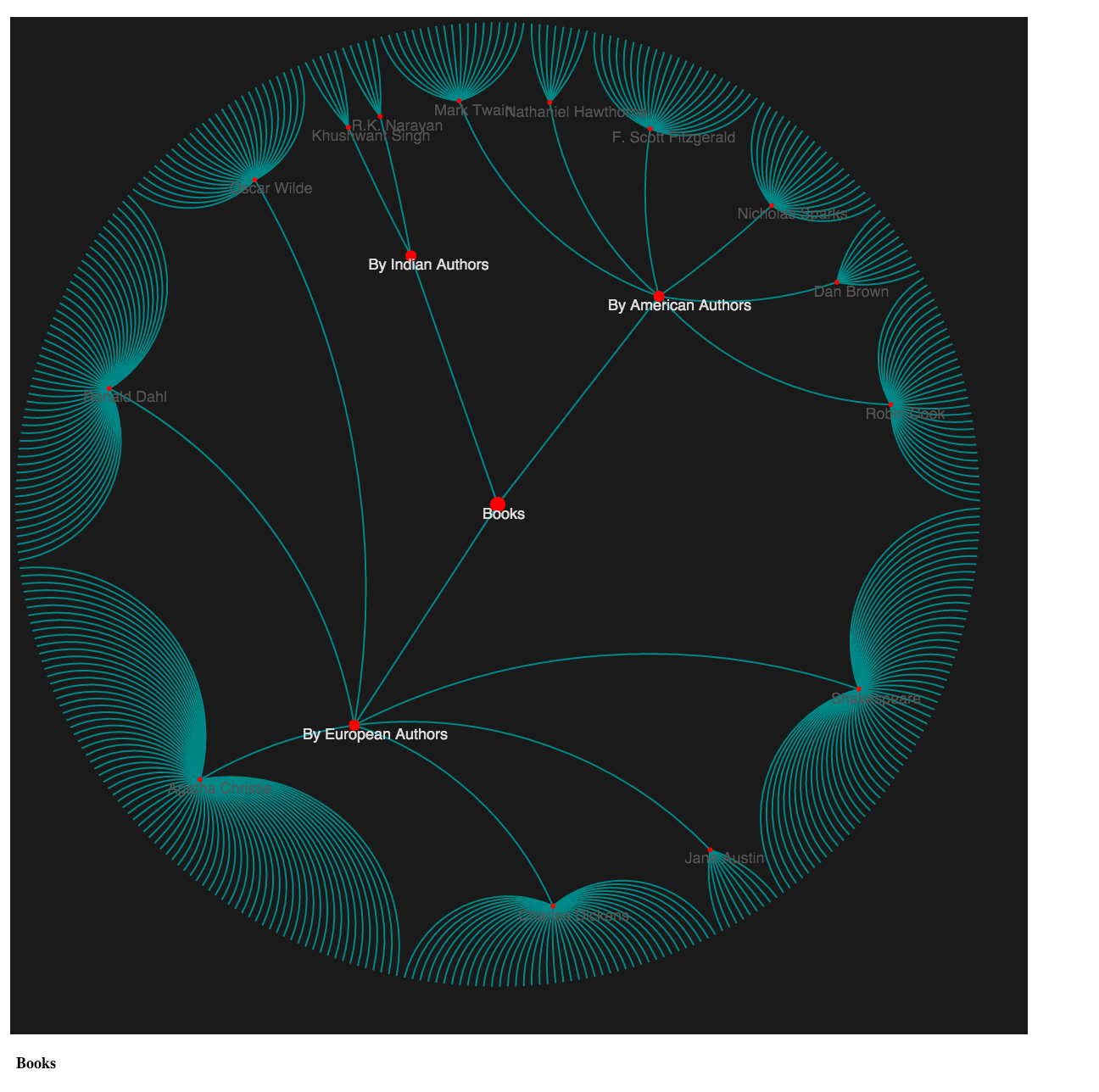
Observed that:

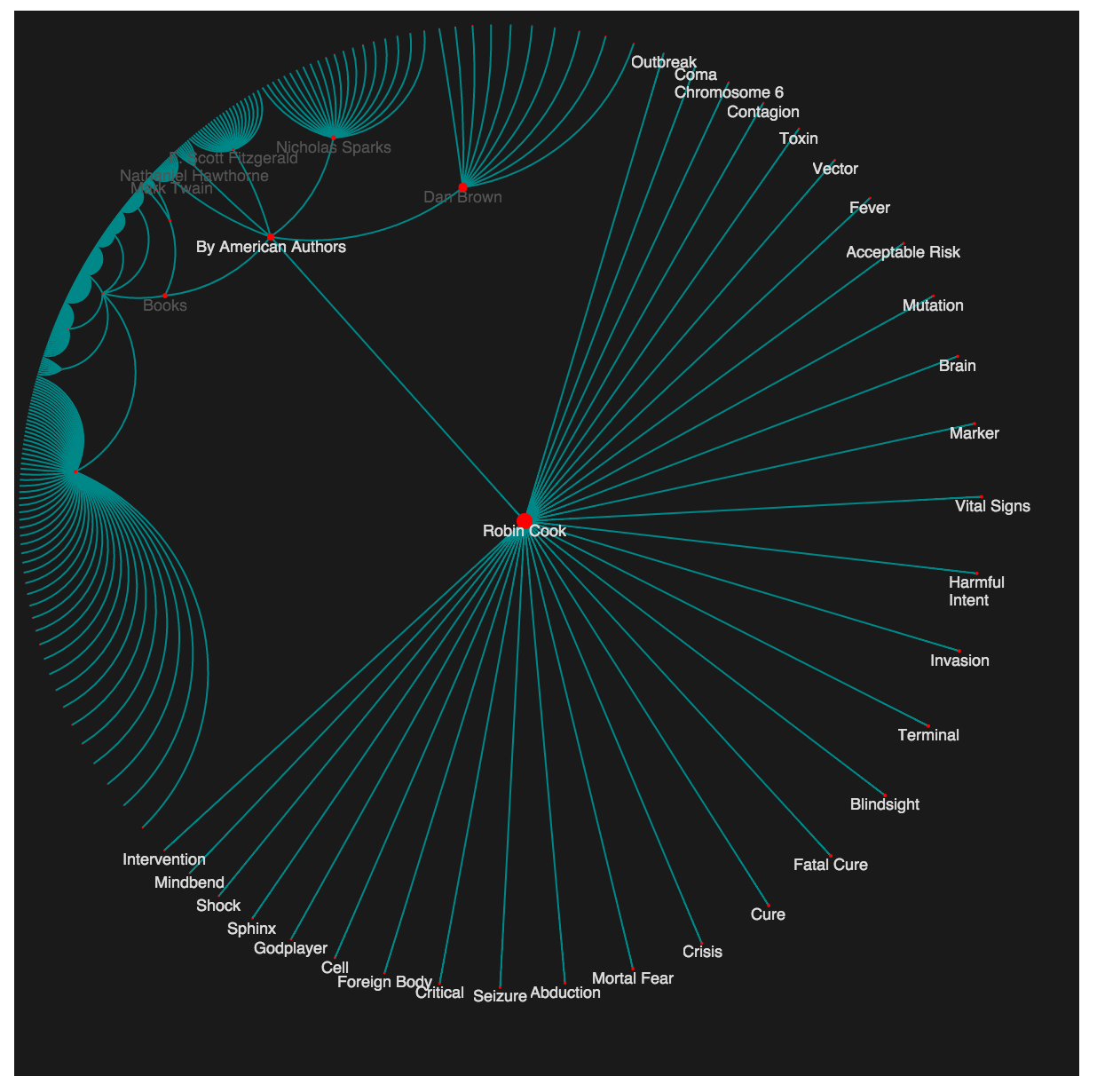
Both HyperTree and Phylogenetic Trees can display unbalanced (in terms of number of nodes) trees well.

HyperTree can handle large number of nodes well as long as the tree structure is not very deep. If the tree structure is deep, the visualization looks shabby.

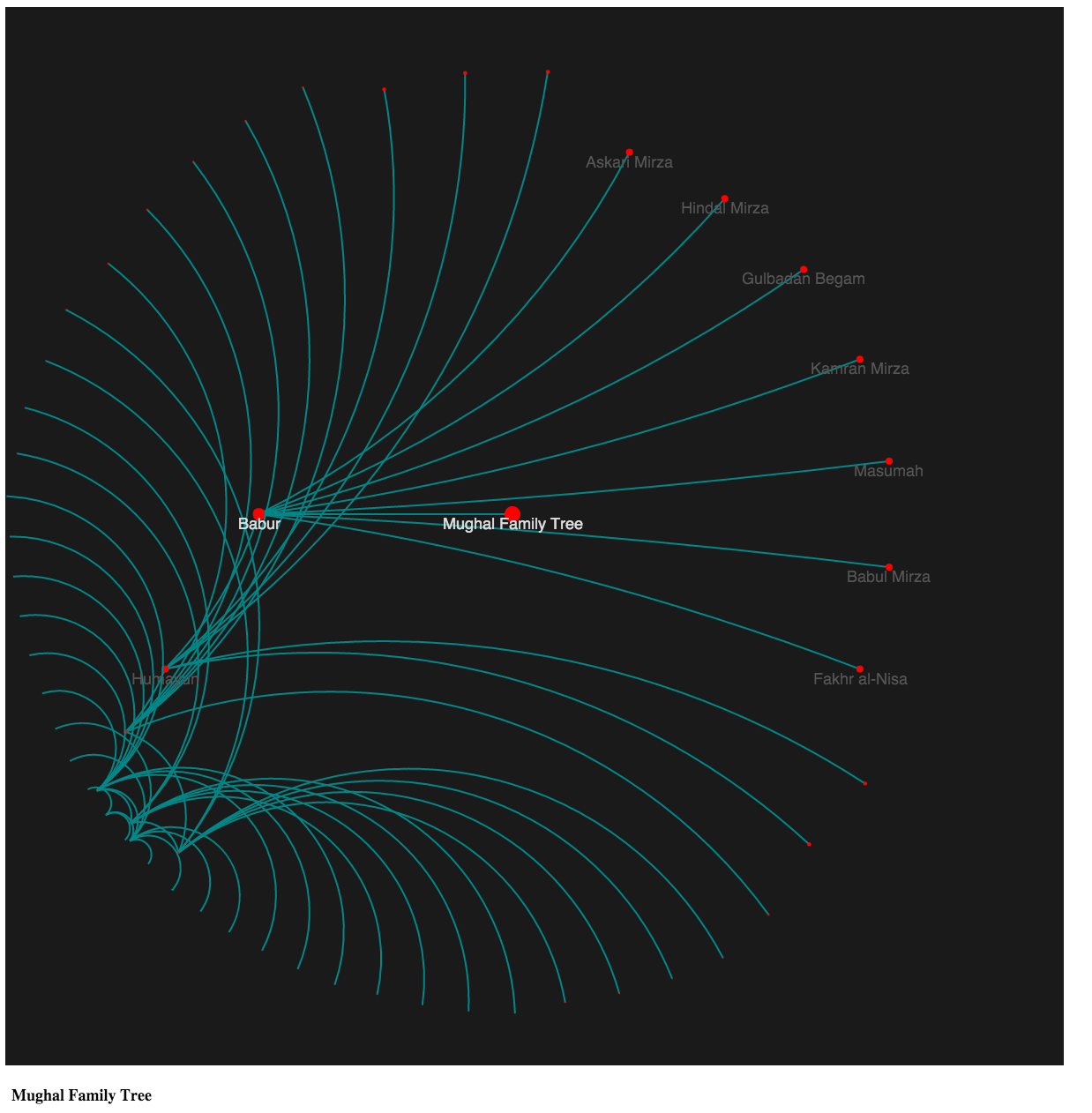
Phylogenetic Tree can handle large number of nodes well even when the tree structure is deep. But it displays only the leaf nodes and their paths from the root.

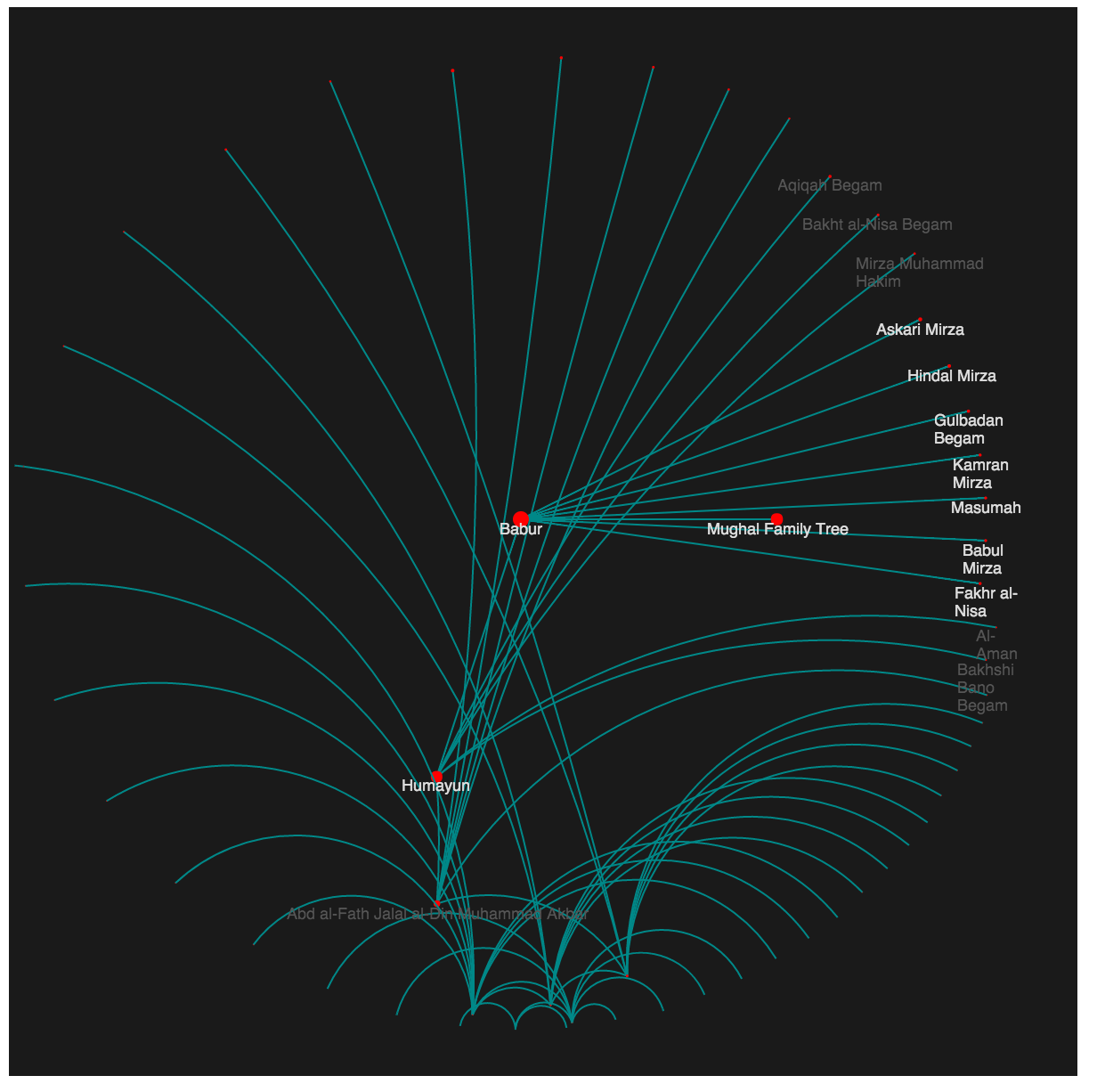
1. Hyperbolic tree visualization showing the popular books written by European, American and Indian authors. It’s an unbalanced tree with only 10 books written by Indian authors and around 385 books written by European and American authors. The Tree is not very deep and has just 3 levels.



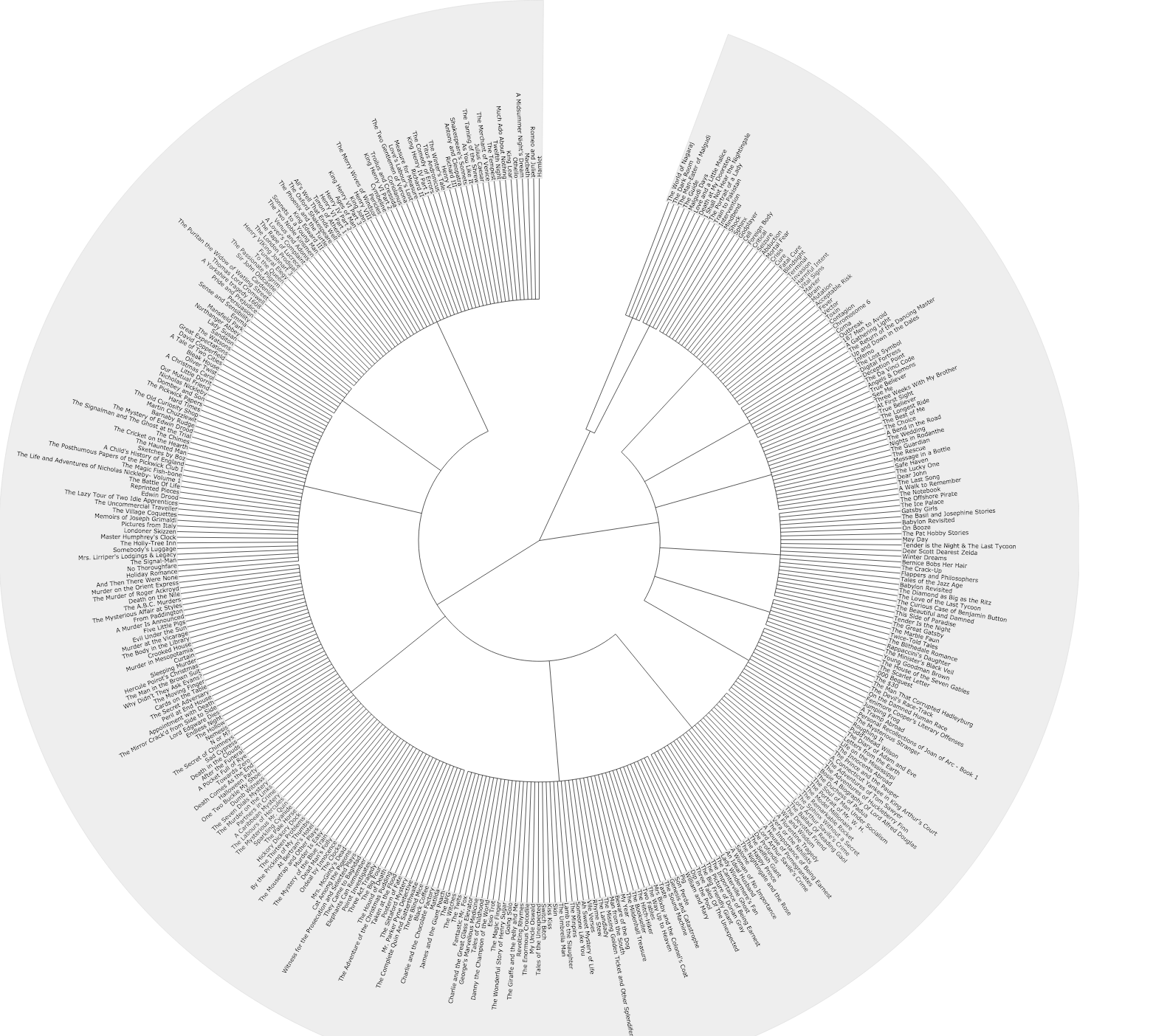


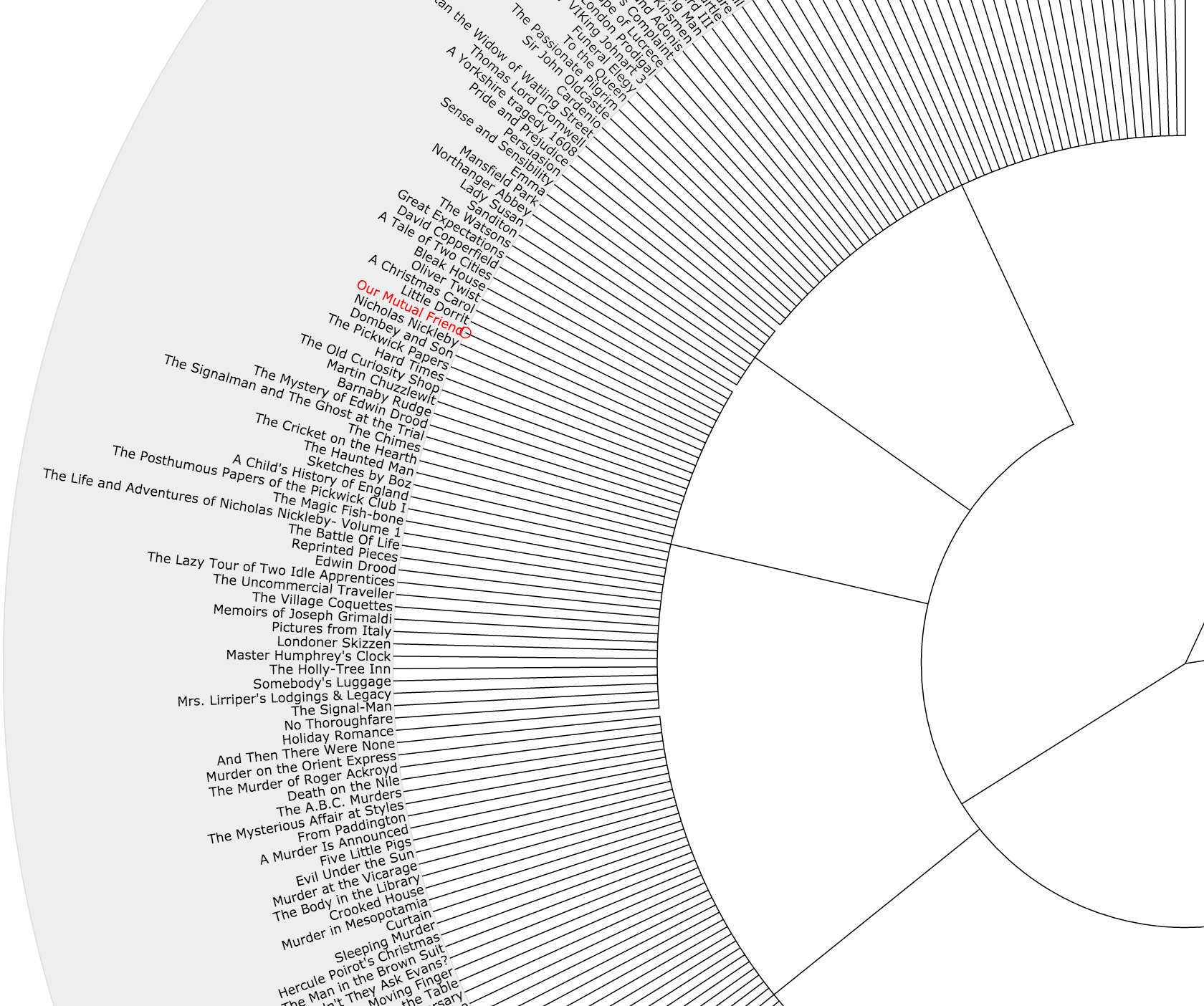
1. Hyperbolic tree visualization showing the Mughal dynasty family tree. It’s an unbalanced tree with around 70 nodes but 13 levels deep.



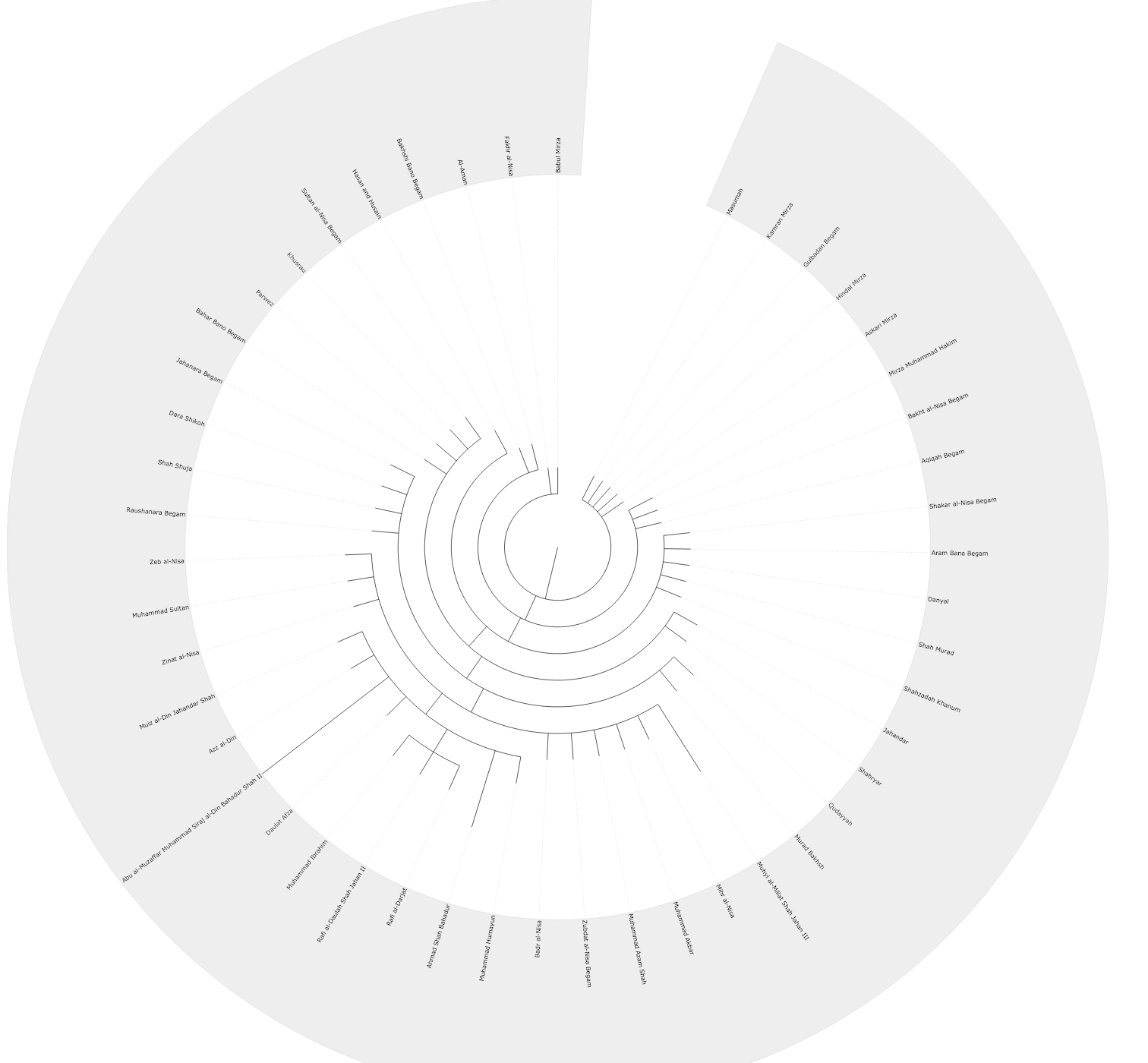


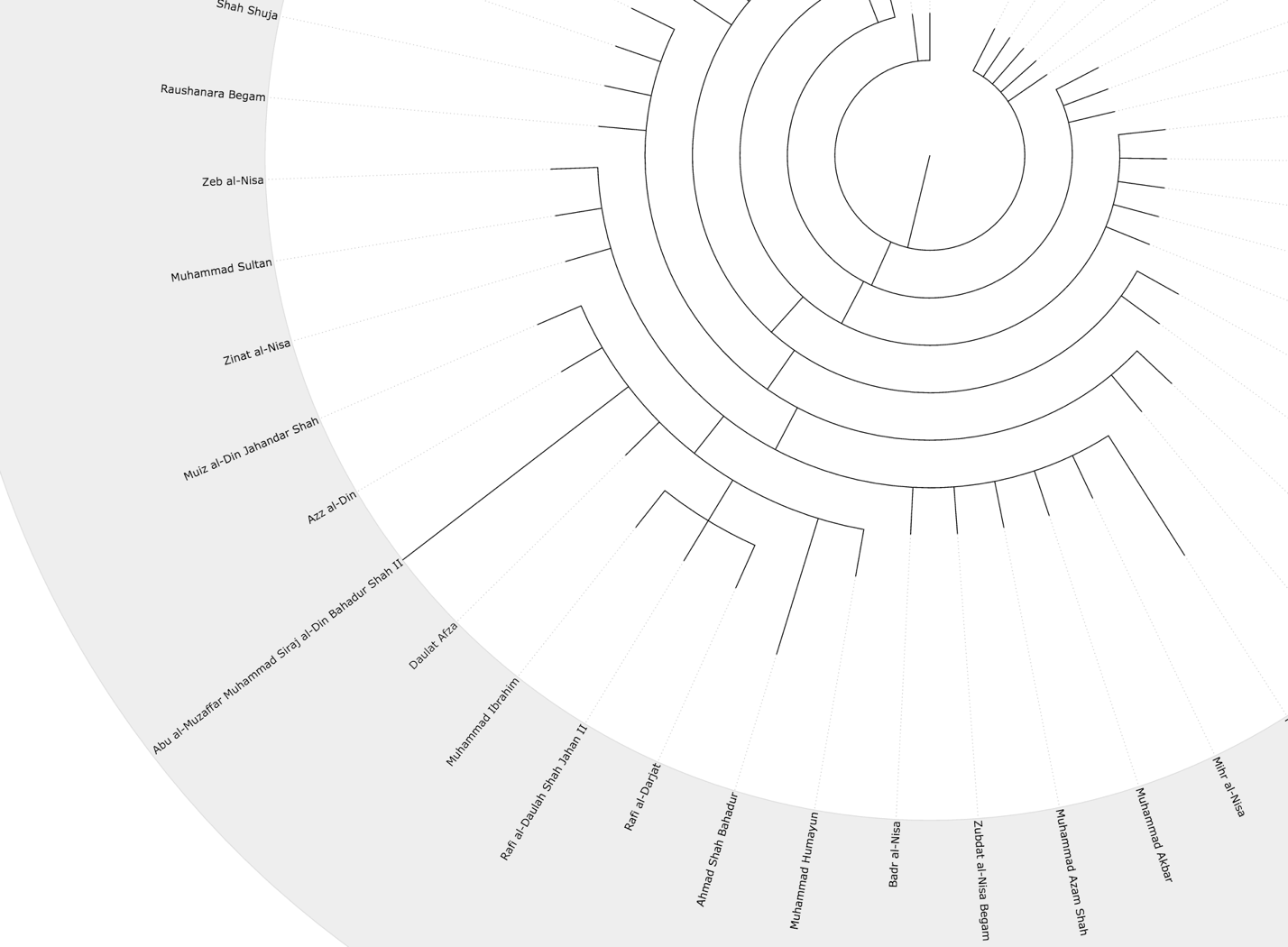
1. Phylogenetic tree visualization showing the popular books written by European, American and Indian authors. It’s an unbalanced tree with only 10 books written by Indian authors and around 385 books written by European and American authors. The Tree is not very deep and has just 3 levels.



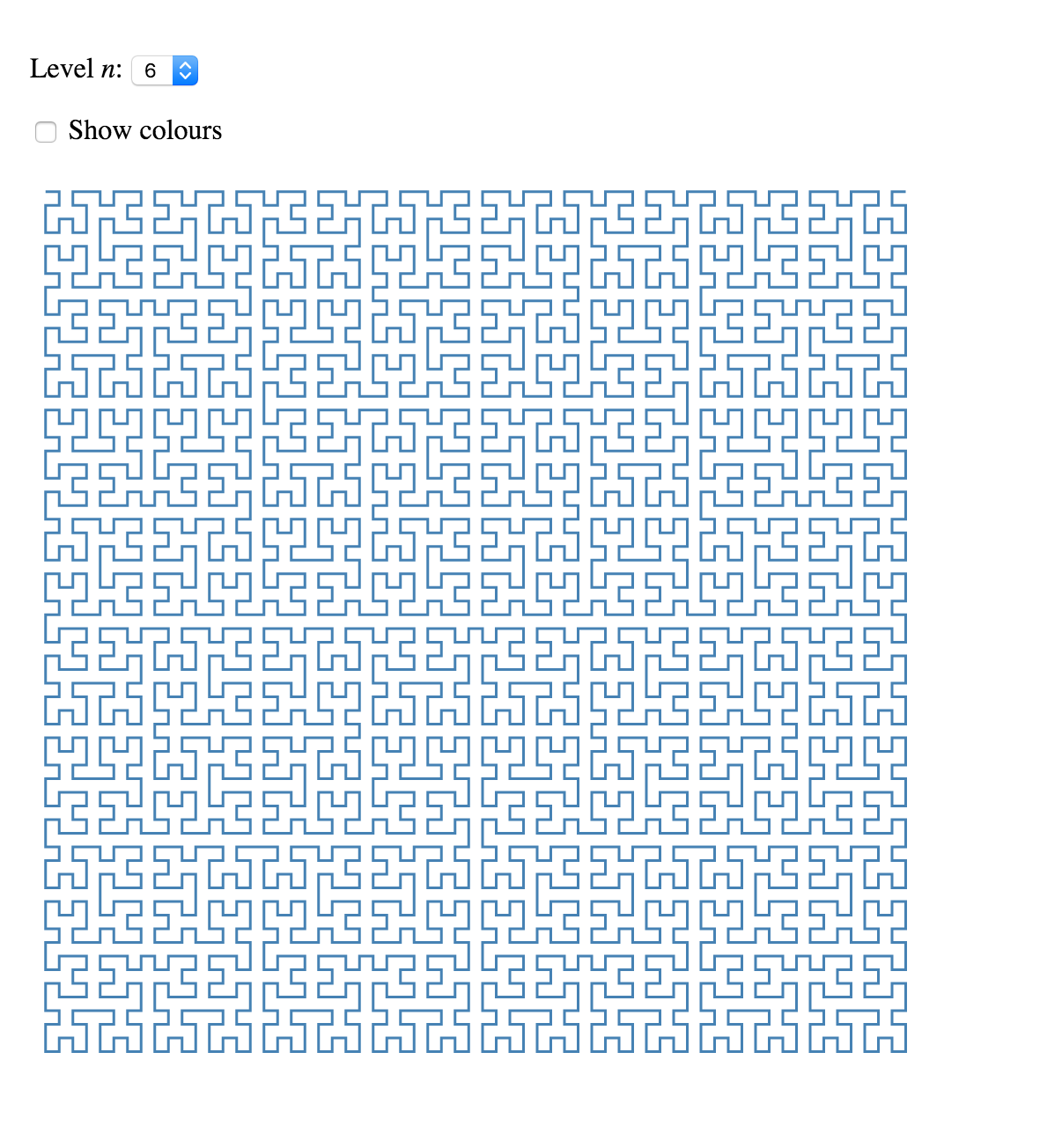


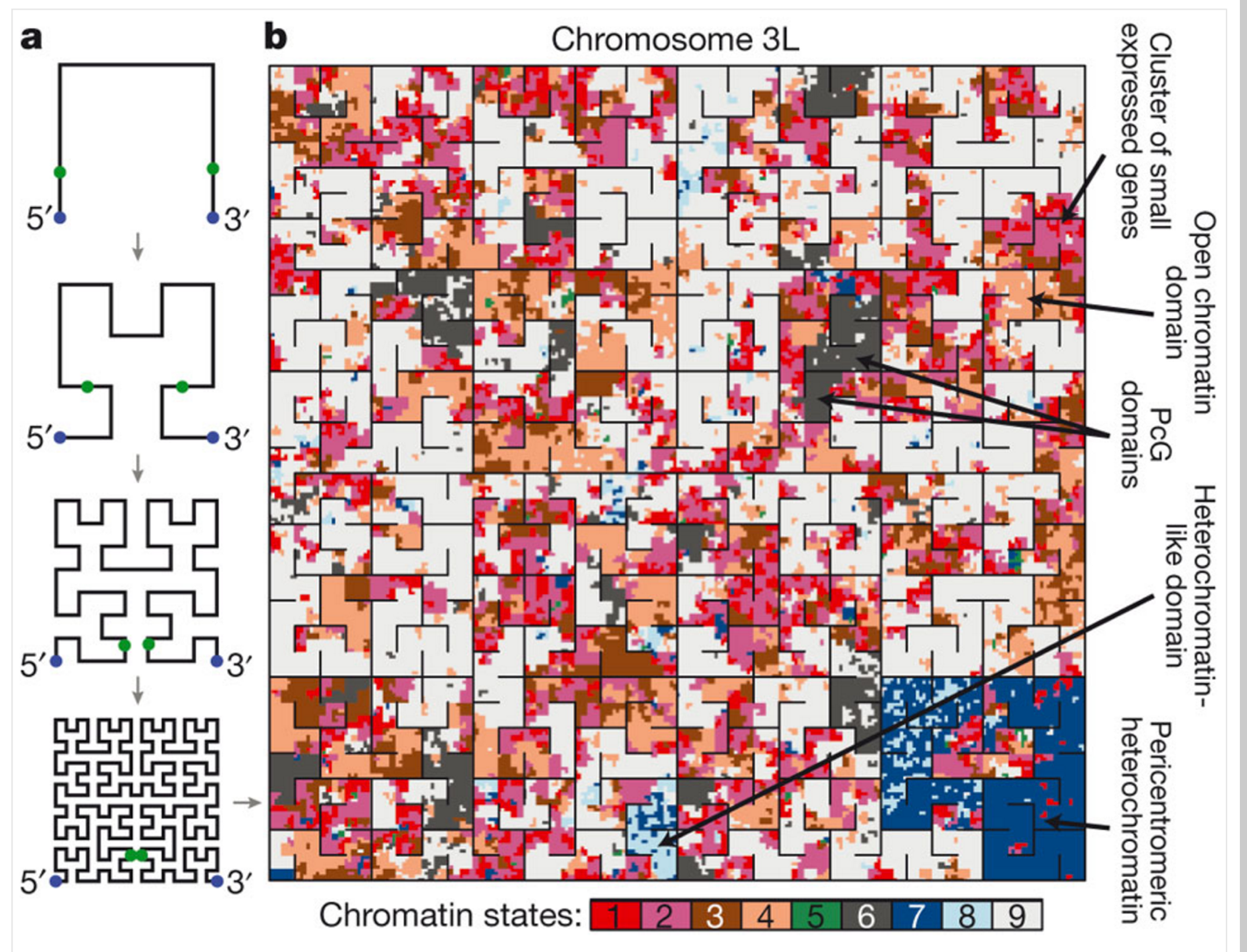
1. Hyperbolic tree visualization showing the Mughal dynasty family tree. It’s an unbalanced tree with around 70 nodes but 13 levels deep.





Hilbert Space Filling Curve





Common questions that the visualization should be able to answer:

1) Return no. of children of the given node.

2) Search for and navigate to a given node.

3) Given a node, highlight all its siblings

4) Find among the sibling nodes the node with greatest number of children

5) Find among the sibling nodes the node with the least number of children

6) Given any two nodes in the tree, find the closest common parent

7) Return the depth of the sub-tree of the given node

8) Find among the sibling nodes the node with the deepest sub-tree

9) Find among the sibling nodes the node with the least deep sub-tree

10) Highlight the shortest path from one given node to another given node

11) Given a node, return its ancestor at depth d

12) Given any two nodes, find the one with the greatest number of children

13) Given any two nodes, find the one with the deepest sub-tree

14) Highlight all nodes at depth d from the given node