


# Robust Cut for Hierarchical Clustering and Merge Trees

Divya Banesh , James Ahrens  and Roxana Bujack 

Los Alamos National Laboratory

## Abstract

*Hierarchical clustering arrange multi-dimensional data into a tree-like structure, organizing the data by increasing levels of similarity. A cut of the tree divides data into clusters, where cluster members share a likeness. Most common cutting techniques identify a single line, either by a metric or with user input, cutting horizontally through the tree, separating root from leaves. We present a new approach that algorithmically identifies cuts at **multiple levels** of the tree based on a metric we call **robustness**. We identify levels to maximize overall robustness by maximizing the height of the shortest branch of the hierarchical tree we must cut through. This technique minimizes the variation within clusters while maximizing the distance between clusters. We apply the same approach to merge trees from computational topology to find the most robust number of connected components. We apply the multi-level robust cut to two datasets to highlight the advantages compared to a traditional, single-level cut.*

## CCS Concepts

• *Mathematics of computing* → *Algebraic topology*; • *Information systems* → *Clustering and classification*;

## 1. Introduction

The field of data science is continuously evolving to better understand complex, high-dimensional data spaces. Hierarchical clustering methods establish a formal order in high-dimensional spaces by identifying a recursive similarity and defining an inherent relationship between clusters. However, clustered results can be many levels deep, and determining which levels are most significant for useful categorization of the data can be challenging. This paper introduces a novel algorithmic approach for identifying the *robust cut* of a hierarchically clustered dataset. We define a robust cut as the identification of a *set of multi-level breaks* of the hierarchical structure such that there is minimal variation *within* the clusters while maximizing the distance *between* clusters. The robustness of any cut is defined as the *height of the shortest branch* of the hierarchical tree we cut through. Our method differs from related work as we do not require the cut to be at the same height across the tree (hence multi-level). This method is applicable to both classical hierarchical tree visualizations (e.g. dendrograms, icicle plots, or sunburst charts) [ZS21], and merge trees, a fundamental concept in computational topology. More specifically, our method is geared towards two objectives: the formation of *flat clusters* from dendrograms to categorize data into distinct groups, and the determination of the *number of connected components* ( $0^{\text{th}}$  Betti number) in a robust manner for merge trees.

For hierarchical tree visualizations, our algorithm focuses on the identification of *robust flat clusters*. Specifically, we are interested in groupings of points that exhibit minimal internal variance while maintaining maximum separation from other groups. In addition, our method is deterministic so no user input is required.

Our algorithm is also applicable to merge trees, a topological representation of the connected components in a dataset. Our method can be used to robustly quantify the number of connected components ( $0^{\text{th}}$  Betti number) in the data by identifying the most robust cut of the merge tree. This metric is crucial in understanding the underlying topological structure, particularly in high-dimensional spaces where such structures are not readily apparent. Our approach enhances the robustness of this calculation, mitigating the influence of noise and outliers, and providing a more accurate representation of the data's fundamental topological features.

There is a significant overlap between the topological analysis of data using persistent homology, specifically the 1-skeleton of the alpha complex [EKS83], and hierarchical clustering [AA18]. In persistent homology, a set of points  $P$  can be connected into a single component by iteratively adding branches between two points when the distance between them is less than the filtration parameter,  $\epsilon \in \mathbb{R}^+$  [EH22]. This can be imagined as in Fig. 1 with  $\epsilon$  being the diameter of balls around the points and a merge occurring between two points as soon as their balls touch as  $\epsilon$  is continuously increased. The resulting merge tree is a hierarchical representation of  $P$ . This tree is identical to the dendrogram in hierarchical clustering if single-linkage clustering is used as the merging strategy. Any value of  $\epsilon$  defines a horizontal single-level cut of the merge tree into a set of connected components.

## 2. Related Work

### 2.1. Hierarchical Clustering

Hierarchical clustering is a useful technique employed in various domains such as medical data analysis [RSK\*23,LLZZ23] and ma-

chine learning [BHA\*23, ZRK\*23]. In hierarchical clustering, flat clusters are identified using an approach known as tree or branch cutting. The most widely used method in this process, often referred to as the ‘static’ tree cut, identifies individual clusters as contiguous sets of branches below a predetermined cutoff height. Each of these branch sets, isolated at or below this fixed threshold, is considered a separate cluster [LZH08]. However, the use of a static cut for cluster determination may not accurately define the best categorization of the data. This limitation is due to the static cut’s reliance on a fixed threshold, which might not align perfectly with the natural divisions within the data, especially if the data is heterogeneous [VKA\*16, AA18].

While multi-level cut methods do exist, we are not aware of any publication that employs robustness as a metric for multi-level cuts. Some examples of multi-level methods include cuts by inconsistency (the deviation of a particle’s distance from the center over the average across the cluster), or by recursively splitting sub-clusters based on characteristic patterns [LZH08]. Obulkasim et al. point out that the fixed-height cuts do not provide good enough clusters in biomedical data [OMvdW15]. They suggest a semi-supervised piece-wise snipping that allows a flexible-height cut instead. Their algorithm incorporates external data, such as the life expectancy in DNA and mRNA datasets, to decide on the suggested cut. Vogogias et al. provide an interactive interface in which the user can select places to cut the tree taking data characteristics into account [VKA\*16]. They support the user with global (‘static’) and local automatic partition techniques. Alcaide and Aerts support interactive cuts with community finding algorithms from graph theory [AA18]. They aggregate nodes based on degree centrality and use the Infomap algorithm, which is based on random walks, for community detection [RB08].

The python package `scipy.cluster.hierarchy` [Sci23] allows the user to define linkage, and extract flat clustering based on inconsistency, number of clusters, static height, and hybrid methods, but the determination of flat clusters is only possible at a single height.

## 2.2. Persistent Homology

There is a noteworthy overlap between topological data analysis (TDA) and hierarchical clustering, explicitly pointed out by Alcaide et al. [AA18], but perceivable in many other works.

Delfinado and Edelsbrunner [DE95] describe how Betti numbers on triangulations up to three dimensions can be computed in almost linear time using two Union-Find algorithms [Tar75] to mark simplices that belong to the same cycle. Union-Find is also used in classical hierarchical clustering to compute the tree structure.

Related publications explore different complexes used to derive filtrations. Edelsbrunner et al. [ELZ00] suggest generating a filtration from scattered data using the alpha complex [EKS83]. Since it is homotopy equivalent to the distance field, it is an excellent tool for assessing the topological structure of a discrete point distribution. Ghrist [Ghr08] states that the Czech complex has the same homotopy type as the union of closed balls about the point set while the Rips complex, which can be computed faster, does not. However, he states that using the Rips complex and the barcode is justified because a Czech complex can be approximated by two Rips complexes through inclusion maps. When reduced to the

1-skeleton, i.e., to only points and edges without triangles, tetrahedra, or higher-dimensional simplices, these three complexes all produce the same filtration, which coincides with the ordering of adding smallest to largest edge used in hierarchical clustering with the single-linkage strategy. Siu et al. suggest a filtration function that takes the local density into account such that it becomes scale-invariant, i.e., the persistence diagrams of a pattern and its scaled version are identical [SSYY22]. They use an average instead of a minimum to be more robust against noise. In some ways, this is similar to the presented method because the robust cut is also scale invariant whereas most traditional single-level cut methods are not.

Barcodes, persistence plots, merge trees, and hierarchical tree drawings, like dendrograms, are related in that they all encode the births and deaths of features in data. The barcode [KMM04] is a visualization with a line for each feature stacked on top of each other such that the beginning of the line is its birth and the end is its death. The persistence plot is similar. Edelsbrunner et al. distinguish noise from feature through persistence, i.e., a feature’s life span in persistent homology [ELZ00]. A merge tree is a sub-graph of the contour tree [CSA03], which in turn coincides with the Reeb graph for simply connected domains [Ree46]. A merge tree is similar to the barcode of the  $0^{\text{th}}$  Betti number if the individual lines are connected based on the merges that caused their deaths. They are more commonly used for the visualization of the topology of scalar fields, where the filtration is generated from the scalar values at data points instead of distances [EH22]. A split tree is similar [CSA03].

We will use the topological concept of persistence, the life span of a feature, from TDA to mathematically motivate the robust cut to form flat clusters in hierarchical trees whether they originate from topology or from hierarchical clustering.

## 3. Robust Cut

Given a dataset  $D$  that comprises a set of elements  $E$ , each element is associated with a set of properties,  $a_i$ :

$$D = \{E \mid E = \{a_1, a_2, a_3, \dots, a_n\}\}.$$

From these elements and associated properties, we identify the relationship between points through a metric; here we use the Euclidean distance between  $a_i$ . From this measure, we generate the corresponding hierarchical tree.

From this hierarchical structure, a single-level flat cut identifies the height of the tree that results in a set of corresponding clusters, as all connected branches below a cut form a single cluster. However, finding the best height is not trivial. The most promising candidate is the height whose cut goes through as few *short* branches of the tree as possible. This is because cutting through a short branch indicates a configuration that is not stable. A small deviation could result in a different distribution of particles into different categories, making the cut invalid. However, there may not be a single value that generates a horizontal cut through the longest branches. A value that is good for one section of the tree may not be as meaningful in another.

We resolve this by allowing cuts that are not a single straight line. We only require that the cut goes through the whole tree, such that each leaf is separated from the root, but the height of the cut may vary. This allows us to minimize cuts through short branches and



(a) Data points (black points) and radii that cause merges (gray disks with luminance increasing with radius).

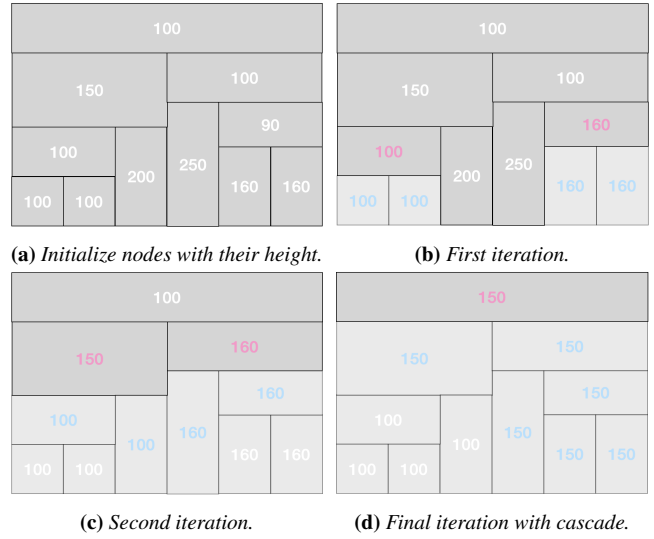


(b) Corresponding dendrogram and potential cuts for flat clustering.

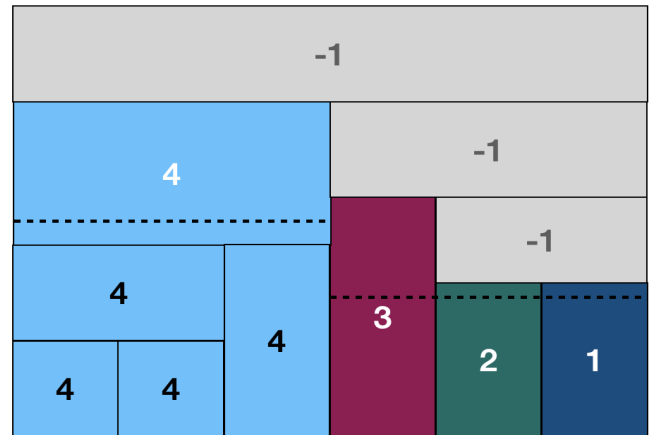
**Figure 1:** Toy 1D example data and corresponding dendrogram with potential cuts (dotted lines) to form flat clusters from the hierarchy. Regardless of the radius  $\epsilon$  or cut height, we pass through a node of height 100, resulting in the robustness of any cut being less than 100. The order of points left to right are the same in (a) & (b).

obtain the most robust one-to-one association of particles and connected components. Our algorithm computes this line without requiring optimization or initial guesses. We illustrate our algorithm on a 1D toy example with six data points at x-coordinates 0, 100, 300, 650, 900, and 1050, in Fig 1a.

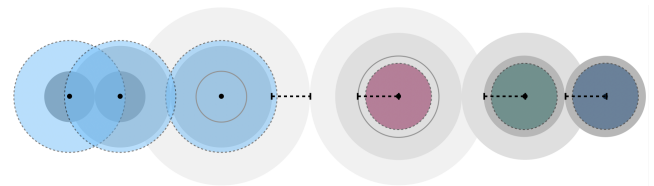
The algorithm to define a robust multi-level cut is as follows. If we cut through a branch of the tree, we must also cut through the branch of its sibling, and the combined robustness is the minimum of their individual robustness values. For a sub-tree, the overall robustness is the maximum of all robustness values down to the leaf. We therefore first initialize each node with its robustness (death length minus start length), Fig. 2a. We start by treating all leaf sibling pairs  $s_1, s_2$  that have only leaf siblings and 1) minimize the robustness among them  $m = \min(r(s_1), r(s_2))$ , Fig. 2b. 2) Going up, we assign the maximum of that minimum and the robustness of the parent  $\max(m, r(p))$  to each parent of only leaf children. 3) We then mark the treated leaves as removed, making the parent a leaf. 4) If the leaf siblings originally had descendants  $d_i$ , assign the minimum of  $m$  and each descendant's robustness  $\min(r(d_i), m)$  down the whole sub-graph, Fig. 2d. This process is applied iteratively until the tree has been reduced to the root. Now that we have assigned the cutting robustness for the branches leading to all nodes, we cut the most robust one first, Fig. 3. Upon a cut, we mark its descendants with the id of the node and its ancestors with a non-id, e.g., -1 to mark that they are treated. The process is repeated until all leaves have been assigned. In a tie, the lower node gets cut first.



**Figure 2:** Algorithm that assigns cutting robustness values. (a) Initialization. (b) to (d) iteratively for all unmarked leaf siblings: minimize across siblings (blue), maximize for parent (pink), and mark as removed (transparent).



(a) Dendrogram with the robust cut indicating 4 robust flat clusters.



(b) Data set colored by cluster ID. The dotted lines show the robustness of each cut. The hollow circles indicate the downward adjustment of the algorithm across siblings.

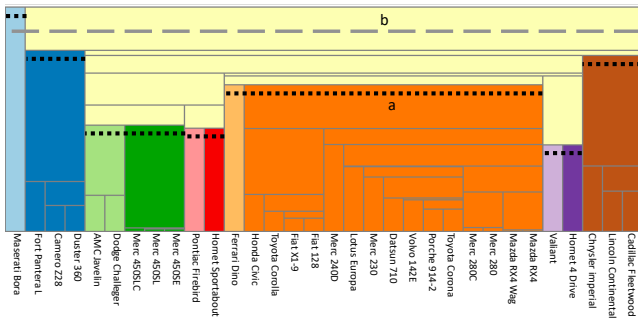
**Figure 3:** Toy example with robust cut forming flat clusters with a robustness of 150.

The necessity of step 4 in the algorithm becomes apparent if we imagine the robustness of the root had been 155. Then, the most robust cut would go right through it leaving everything in one cluster for a total robustness of 155. But without the cascading of the 150 down the right half sub-tree, the cutting part would have started at the 160 lower right value, falsely marking off the 155 value and resulting in a cut of robustness of only 150.

## 4. Case Studies

### 4.1. Motor Car

The *Motor Trend Car Roads Tests* data in this case study is from the *datasets* library included in the *R Programming Language* [RC19]. The data was first published in the 1974 *Motor Trend* US magazine and lists 11 numeric attributes of 32 different cars of the time. The 11 attributes are miles per gallon, number of cylinders, displacement in cubic inches, gross horsepower, rear axle ratio, weight in thousand pounds, time to travel  $\frac{1}{4}$  mile, engine type (V-shaped or straight), transmission type (automatic or manual), number of forward gears and number of carburetors. The Euclidean distance between the 11 attributes of each pair of cars determines their similarity. From the resulting icicle plot [ZS21], we identify the robust cut, the black dotted line in Figure 4. Our method simplifies the 32 automotive entries into 11 flat clusters, each identified by a unique color. Grey solid lines delineate the hierarchical structure of the data. Our multilevel robust cut has a robustness of 16.95, as defined by the height of box *a*.



**Figure 4:** The hierarchical clustering (visualized as an icicle plot) and robust cut of the 1974 Motor Trend data of 32 different cars. Each color (except yellow) identifies a cluster as determined by the multi-level robust cut (black dotted line). Solid grey lines show the hierarchical tree. The gray dashed line represents the most robust single-level cut for comparison.

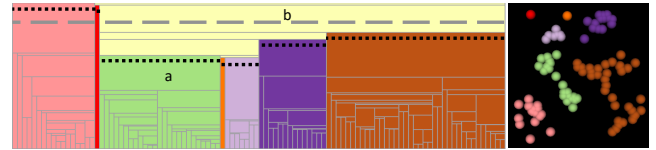
#### 4.1.1. Evaluation

The grey dashed line indicates the most robust single-level cut, i.e., the horizontal line where the smallest segment it intersects is maximized. The robustness of this cut, defined by the height of box *b*, is 16.76. Though the difference in robustness is not large, it's clear that the multi-level robust cut results in a much more informative clustering than the single-level cut. The single-level cut separates the Maserati Bora from the other cars, which is significant as the Maserati is the clear outlier in this group of cars. However, it categorizes everything else into one single group. The multi-level cut, however, does a much better job separating sports cars from higher-performing Mercedes and family sedans.

### 4.2. Birds

This *birds simulation* models flocking behavior that can be used to study and predict the real phenomenon, but can also be considered a swarm intelligence algorithmic optimizer. Similar to Boids [Rey87], the model encourages particles to move towards their neighbors, with a small random component added to their movement. Swarm behavior has previously been studied in persistent homology by Topaz et al. to evaluate the Betti numbers of the Vietoris Rips complex for simulations of birds and fish [TZH15].

In this example, we look at time 26 from a 100 step simulation. Particles are seeded randomly at time 0 and allowed to roam. The robust cut identifies seven clusters in this data, Fig. 5 left. The color of each cluster corresponds to the particle in Fig. 5 right. Our multi-level robust cut has a robustness of 0.047, as defined by box *a*.



**Figure 5:** The hierarchical clustering and robust cut (black dotted line) of a flocking simulation show the clustering of 100 particles about  $\frac{1}{4}$  of the way through the simulation. Each unique color (other than yellow) identifies a multi-level robust cluster in both the tree and the physical particle. Solid grey lines represent the underlying merge tree. The grey dashed line shows the most robust single-level cut for comparison (defined by the height of box *b*).

#### 4.2.1. Evaluation

The most robust single-level cut, at a robustness of 0.0365, would combine the five clusters on the right (green, orange, lavender, purple, and brown) into a single cluster. This would result in three clusters: the pink, the red, and everything else. Comparing these results to the locations of the particles in Fig. 5 right shows that this is a drastic oversimplification of the data. A visual analysis clearly shows the separation between particles that are being missed by the single-level cut. The multi-level cut, however, much more clearly represents the proximity of particles in this configuration.

## 5. Conclusions

We have presented an approach to improve the robustness of flat clusters in hierarchical clustering by softening the restrictions of the cut from a single line to a multi-line configuration. The resulting clusters are more robust and allow semantically more meaningful clustering of data. Additionally, we can use the same algorithm to better analyze the topology of point clouds, as it provides the most robust number of connected components ( $0^{th}$  Betti number) in computational topology. Moreover, our method computes the robust multi-level cut without the need for initial guesses or convergence issues.

However, it's worth noting that the robust cut has a notable limitation: it cannot be used in applications that need a constant cutting value for the physical validity of the results. Therefore, it's crucial to ensure that the results correspond to a semantic understanding of the data. We will explore this further in future work.

## References

- [AA18] ALCAIDE D., AERTS J.: Mclean: Multilevel clustering exploration as network. *PeerJ Computer Science* 4 (2018), e145. 1, 2
- [BHA\*23] BERTUCCI D., HAMID M. M., ANAND Y., RUANGROTSAKUN A., TABATABAI D., PEREZ M., KAHNG M.: Dendromap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2023), 320–330. doi:10.1109/TVCG.2022.3209425. 2
- [CSA03] CARR H., SNOEYINK J., AXEN U.: Computing contour trees in all dimensions. *Computational Geometry* 24, 2 (2003), 75–94. 2
- [DE95] DELFINADO C. J. A., EDELSBRUNNER H.: An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design* 12, 7 (1995), 771–784. 2
- [EH22] EDELSBRUNNER H., HARER J. L.: *Computational topology: an introduction*. American Mathematical Society, 2022. 1, 2
- [EKS83] EDELSBRUNNER H., KIRKPATRICK D., SEIDEL R.: On the shape of a set of points in the plane. *IEEE Transactions on information theory* 29, 4 (1983), 551–559. 1, 2
- [ELZ00] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science* (2000), IEEE, pp. 454–463. 2
- [Ghr08] GHRIST R.: Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society* 45, 1 (2008), 61–75. 2
- [KMM04] KACZYNSKI T., MISCHAIKOW K. M., MROZEK M.: *Computational homology*, vol. 3. Springer, 2004. 2
- [LLZZ23] LIU T., LU Y., ZHU B., ZHAO H.: Clustering high-dimensional data via feature selection. *Biometrics* 79, 2 (2023), 940–950. 1
- [LZH08] LANGFELDER P., ZHANG B., HORVATH S.: Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. *Bioinformatics* 24, 5 (2008), 719–720. 2
- [OMvdW15] OBULKASIM A., MEIJER G. A., VAN DE WIEL M. A.: Semi-supervised adaptive-height snipping of the hierarchical clustering tree. *BMC bioinformatics* 16, 1 (2015), 1–11. 2
- [R C19] R CORE TEAM: *Motor Trend Car Road Tests*, 2019. R package version 3.6.2. URL: <https://www.rdocumentation.org/packages/datasets/versions/3.6.2/topics/mtcars>. 4
- [RB08] ROSVALL M., BERGSTROM C. T.: Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences* 105, 4 (2008), 1118–1123. 2
- [Ree46] REEB G.: Sur les points singuliers d’une forme de pfaff complètement intégrable ou d’une fonction numérique [on the singular points of a completely integrable pfaff form or of a numerical function]. *Comptes Rendus Acad. Sciences Paris* 222 (1946), 847–849. 2
- [Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 25–34. 4
- [RSK\*23] RAGHAV S., SURI A., KUMAR D., AAKANSHA A., RATHORE M., ROY S.: A hierarchical clustering approach for identification of colorectal cancer molecular subtypes from gene expression data. *Intelligent Medicine* (2023). 1
- [Sci23] SCIPY DEVELOPERS: Hierarchical clustering: `scipy.cluster.hierarchy`. <https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html>, 2023. Accessed: 2024-02-05. 2
- [SSYY22] SIU C., SAMORODNITSKY G., YU C. L., YAO A.: Detection of small holes by the scale-invariant robust density-aware distance (rdad) filtration. *arXiv preprint arXiv:2204.07821* (2022). 2
- [Tar75] TARIAN R. E.: Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)* 22, 2 (1975), 215–225. 2
- [TZH15] TOPAZ C. M., ZIEGELMEIER L., HALVERSON T.: Topological data analysis of biological aggregation models. *PLoS one* 10, 5 (2015), e0126383. 4
- [VKA\*16] VOGOGLIAS A., KENNEDY J., ARCHAUMBAULT D., SMITH V. A., CURRANT H.: Mlcut: Exploring multi-level cuts in dendrograms for biological data. In *Computer graphics and visual computing conference (CGVC) 2016* (2016), Eurographics Association. 2
- [ZRK\*23] ZNALEZNIAK M., ROLA P., KASZUBA P., TABOR J., ŚMIEJA M.: Contrastive hierarchical clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2023), Springer, pp. 627–643. 2
- [ZS21] ZHENG B., SADLO F.: On the visualization of hierarchical multivariate data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)* (2021), pp. 136–145. doi:10.1109/PacificVis52677.2021.00026. 1, 4