Hierarchical Visualization of High Dimensional Data

Interactive Exploration of ’Omics Type Data

Alexander Michael Macquisten

Supervisors: Dr Sara Johansson Fernstad
Prof. Nick Holliman

School of Computing Science
Newcastle University

This dissertation is submitted for the degree of
Doctor of Philosophy

December 2021

Abstract

Our ability to investigate biological entities has been improving over the years, thanks to next-generation sequencing technologies providing an ever-increasing efficiency of data collection. However, this superior data collection hasn't necessarily led to superior knowledge generation. Across the different fields of biological study, data is often high dimensional, with a single entity of interest correlating to a single dimension in the dataset. Datasets with more than a thousand dimensions are not uncommon, and visualising this without sacrificing some of the data is challenging. This thesis covers the creation of methods to support the visual exploration and analysis of high dimensional biological data, supporting users in applying their domain knowledge to discover patterns of interest.

The first contribution of this research is a study comparing five different hierarchical visualization methods for how well they represent the underlying dataset at different scales of hierarchical structure. This was used to inform decisions in developing a method consisting of two linked views to support the exploration and selection of subsets of data, a Sunburst Chart provides an overview, displaying the full dataset, this is complemented by a Treemap that displays the subset of the hierarchy, consisting of the currently selected node and its local structure of child nodes.

The second contribution was a pair of studies covering the evaluation of different statistical distributions for representing multivariate data features in aggregated data. The first was an initial usability study comparing two variations of a novel combination box plot-density distribution glyph called a density box plot, while the second study compared the glyph to box plots and density distribution plots independently. From this, the density box plot was added within each treemap node to show the aggregated distribution of child nodes.

The third contribution of the thesis is a software tool to demonstrate the methods discussed, implemented with JavaScript/D3, with a NodeJS server to handle data processing. Alongside the linked Sunburst-Treemap view and density box plot glyphs, nodes within both views are coloured based on the separation of their sample groups, using Silhouette clustering, with lighter nodes showing more similar sample group values, while darker nodes show more variance between sample groups. Additional support views provide more information on the currently selected node, an analysis view offers expanded information on the currently selected node, while a Parallel coordinates plot shows the counts for each sample for each entity. Additional overview of the full data is provided through multiple small sunbursts, allowing different sample combinations to be viewed side by side before delving into deeper exploration.

A final study was conducted to demonstrate the methods and the tool to end-users, using a series of heuristics the for them to evaluate the components of the tool. Overall, user response to the tool was positive, and the feedback they provided guided the future development of the tool.

It's been a long road, getting from there to here. It has certainly been a goal of mine to do a PhD for a number of years, the concept of getting to do pure research has been enticing ever since my first forays during a summer placement back in 2014. To say it has been an easy journey would be a heinous lie, but it nevertheless has been an interesting and enjoyable one. I have met so many incredible people I have had the privilege to call friends over the years, who have supported and been there for me along the way.

Firstly I'd like to thank my parents for all their support over the years in getting to this point.

To Dr Firat Batmaz, my undergraduate supervisor, who first got me interested in research.

To Dr Ryan MacDonald, thank you for being a wonderful friend, who kept me sane both through childhood and PhD.

To Chantel Maynard, a wonderful person and I am grateful for our friendship over the years.

To Dr Almajd Alhinai, you always had a wonderful story or three to tell, without your advice and guidance I don't know if I would have gotten this project in this first place.

To Alyssa Ridley, thank you for being there for me and for making this whole experience much more enjoyable.

To Robyn Robson, thank you for making my world that much brighter.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others. This dissertation contains fewer than 80,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Alexander Michael Macquisten
December 2021

Acknowledgements

Firstly, I would like to express my gratitude to Dr Sara Johansson Fernstad for being my supervisor and helping guide the research and writing, I am deeply grateful to all the support she has given me over these years. Next I would like to thank Professor Nick Holliman, my second supervisor for all his input and guidance.

I'd like to thank my Industrial Supervisor, Adrian Smith for all his support on the Unilever side, organising interviews, helping find participants, getting approval, along with all his general support and feedback over the years. Next I'd like to thanks Barry Murphy and all the people at Unilever who helped give ideas and tested my tools.

I'd like to thank Dr Christopher Stewart for providing the project with a detailed and useful dataset that greatly helped test things in the early project. Lastly I would like to thank some of my fellow researchers: Kenan Koc; Mike Adele; Hugh Garner, Alma Cantu, for their general assistance over the years, testing my experiments and generally making the whole research experience much more enjoyable.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Background and Motivation

Overall advancements in our ability to investigate biological entities over the last few decades, thanks to next-generation sequencing technologies, have greatly improved the quantity and quality of the data generated. However, this superior data collection hasn't directly led to superior knowledge generation. As the data has become increasingly high dimensional, the ability to process this data has become more complex. A single entity of interest correlates to a single dimension within the dataset. Datasets with thousands of dimensions are not uncommon, and methods designed to handle lower scales of dimensionality cannot handle this greater complexity without sacrificing some aspect of the data.

The core issue with any analysis with high dimensional data comes down to the scale of the data. Most visualization methods are only designed to handle a few dimensions; there are only so many axes you can add to a graph before it becomes too complex. And while there are visualization methods that can handle higher dimensionality, such as Parallel Coordinates (Inselberg, 1985) and Scatter plot Matrices (Becker and Cleveland, 1987), even these have limits from human understanding and screen size.

For analysis methods, the issue arises that as dimensionality increases, the data becomes increasingly sparse as the volume of the data space increases. This is known as the curse of dimensionality (Wright and Bellman, 1962); as fewer and fewer samples become attributed to individual dimensions, the distances between different objects become similar and it becomes difficult to discriminate between them.

Often users are limited to exploring and analysing high dimensional data with only a representation of the full dataset. Either only looking at a subset of the data or processed with methods such as dimensionality reduction techniques to map the dataset to a more reasonable level of complexity.

Many of these methods are often automated, making decisions of what parts of the data to keep based on their parameters and without being able to take advantage of the user's domain knowledge to know what parts may be interesting or not. While such methods have their applications when the analysis question is well defined, they become less useful for exploratory tasks when accessing the full dataset.

The research presented in this thesis was focused on providing explorative visualization support for the analysis of ’omics data. ‘Omic techniques allow for high-throughput and extensive characterisation of biological samples. They provide data based on DNA (e.g. Genomics), RNA (Transcriptomics), protein (Proteomics), and metabolites (Metabolomics). This provides high dimensional data spanning transcription and translation, and the eventual proteins synthesised or the functional small molecules produced by cells. These techniques can be used to understand microbial presence and function, for instance metagenomics for bacterial relative abundance or meta-transcriptomics to understand microbial gene expression.

The datasets used throughout the project were consistently structured around two tables. A data table containing the raw abundance counts, with entities as columns and samples as rows, along with a metadata table with samples as rows, and the sample groups as columns. A single entity correlates to a single dimension in the dataset, with a single sample being made up of all the entities with at least a count of 1. Attached to each sample in the data table is its full hierarchical path.

In the case of microbiology, an individual entity is an Operational Taxonomic Unit (OTU), these are a close approximation to a bacterial species detected within each sample, with each OTU having its detected counts. The hierarchical structure of an OTU will be formed from its associated taxonomy of species, genus, family, order, class, phylum, kingdom and domain.

1.2. Research questions and Summary of Contributions

1) To address high dimensionality challenges by investigating solutions using hierarchical visualization methods:

The issue of scalability is the main gap in the literature; often many existing visualization methods for high dimensionality cater for problems scaling in the tens or hundreds of dimensions (Anand et al., 2012) but are less effective at dealing with dimensionality of thousands or more. Few data analysis techniques support knowledge generation and investigation of the larger and more diverse datasets. Ultimately this is always going to be an issue, as methods for generating more complex datasets will be developed. Looking into ways to better handle this increased scalability and complexity is vital for data analysis to keep pace with data generation.

By taking advantage of included hierarchical metadata, such as the taxonomical classification of species in a metagenomics dataset, the project explored combinations of hierarchical visualization methods to display the data, evaluating the usability of the methods independently and in tandem with one another, to identify the most suitable combination for exploration of high dimensional hierarchies. The focus would be on microbiomic data, but with consideration to handling other kinds of ’omics data and hierarchical data if it was feasible.

2) To enable efficient exploration of extremely high dimensional data by utilising approaches of visual guidance:

‘Omics analysis tasks are explorative in nature, as the user looks for what they consider to be interesting patterns in the data, comparing different samples and datasets. Reduction needs to be controlled by the user and guided by the system, rather than automated reduction, which risks removing or ignoring relevant structures that may be useful to the user, yet entirely unguided exploration would be time-consuming.

Sometimes it will be better for the user to use an automated methods over an explorative method, such as when the task is well defined and the user knows what they are looking for (Munzner, 2018). But there are already options for these tasks, by presenting users with explorative methods as well, they can make the correct decision to use the right tools for the task at hand, be it using automated reduction for well defined questions, or explorative methods for when instead the task if to look for interesting structures within the dataset.

The project explored ways of highlighting patterns of potential interest within the data, such as the colouring of nodes to show measures of difference between samples or variation within a data-subset, and alternative views to show different aspects of the data.

3) To design novel visualization to enable the representation of multivariate data features in aggregated data:

Due to the dimensionality and hierarchical structure, data is often aggregated rather than displayed in full. The features of these aggregated data may vary considerably, and a fast overview of these features could support the identification of interesting data patterns and considerably accelerate analysis.

The project investigated and compared glyph-based visualization to represent features of aggregated data, aiming to design a novel glyph visualization that can be used both as an enhancement to existing visualization methods and as a stand-alone visualization method.

Contributions

Through this Thesis, the project explores methods that contribute towards an overall solution for the exploratory analysis of high dimensional hierarchical data, presenting the following contributions:

1. A quantitative user study comparing five different hierarchical visualization methods, in their ability to represent underlying data features.
2. A pair of quantitative user studies comparing different glyphs for displaying multivariate data features in aggregated data.
3. The design of a novel visualization method for displaying aggregated multivariate data features.
4. A software tool, HieraViz, supports the exploration of high dimensional data through a combination of linked hierarchical views.
5. A qualitative user study which evaluated HieraViz through a series of heuristics and user feedback.

1.3. Thesis Structure

The Thesis is structured as follows:

Chapter 2 describes the state of the literature, covering how it relates to the project and the issues with existing methods.

Chapter 3 covers the main section of research conducted in this project, first by presenting the requirements and challenges generated through interviews with end-users.

Chapter 4 presents a user study comparing five different hierarchical visualization methods: Treemap, Icicle Plot, Sunburst Chart, Circular Treemap and Bubble Treemap, in their ability to represent underlying data structures. The study was presented in Macquisten et al. (2020) and expanded in Macquisten et al. (2022).

Chapter 5 covers two user studies, the first study covered the comparison of two variants of a novel statistical glyph, designed to show multivariate aggregated distributions, to determine how much complexity could be displayed within a single one of these glyphs. The second study followed on from the previous one, comparing the successful glyph from the previous experiment to Density Distribution plots and Box plots, again to determine which were best for representing multivariate aggregated distributions

Chapter 6 covers the application of the research discussed in the previous chapters into a hierarchical exploration and analysis tool called HieraViz. The section goes through each component view and functionality and how it supports the overall exploration process.

This is followed by the final user study in chapter 7, where five participants were given a demonstration and a chance to interact with the tool through an online session. Following this, the participants evaluated the tool both through feedback and a series of evaluation heuristics.

Finally, Chapter 8 provides a summary of the project and possible avenues for future work.

Chapter 2. Literature Review

This chapter will focus first on an overview of the challenges faced when dealing with high dimensional data, along with the current state of the art methods and solutions for visualizing and analysing high dimensional data. Following which, it will cover methods entailing hierarchical visualization, visualization of ‘omics data, statistical visualization and finally, glyph-based visualization.

The literature search started with a google scholar search for papers related to ’visualization’; ’high dimensionality’, and ’‘omics visualisation’ in order to get a general background on the fundamentals of the research, expanded on through following the references to other interesting papers mentioned with in, this backed up with setting up Scopus to provide alerts to new literature within the previously mentioned fields of interest. Of note, was the book "Visualization Analysis and Design" by Munzner (2018), which provided a good deal of understanding on good visualization practices, alongside the basis of the main methodology used within the project (Chapter 3).

Once enough general background was gathered, and an understanding of the problems being faced by the end users, the search was directed towards more specific topics of visualisation. Some of these searches went towards addressing the needs and issues presented through interactions with the end users, others went towards expanding knowledge on good visualization practices and current methods and tools.

2.1. Visualization background

For the exploration and analysis of large and complex datasets, visualization is a powerful tool, both for the presentation of the data in a way that is understandable to the user and as a way for them to directly interact and manipulate the data.

Automated systems with purely computation methods work well when there is a well-defined question, where large amounts of data can be popped into the system and come out with the answer (Munzner, 2018), be it using machine learning to trawl through data to improve search engine performance, or processing real-time financial data to make decisions on when to buy and sell stocks, having a human in a loop slows down the process in these cases.

However, when the analysis question is broader or where there is no set question at all, these methods become less powerful. While purely computational methods can efficiently process data, they are limited by what patterns and outliers the methods consider interesting to show. Visualization allows users to analyse data even when they don't know what questions they need to ask (Munzner, 2018), providing the data in a format that allows for the user to explore it themselves and apply their own domain knowledge of the data to find interesting patterns and structures.

One of the core ideas presented often in visualization is the visual information seeking mantra (Shneiderman, 1996), a set of visual design guidelines that provides a framework for data visualization, Overview first, zoom and filter, then details-on-demand. The idea is that the visualization should present the user with the full dataset (Overview), allowing the user to see overarching patterns. From which they can select a subset to focus on (Zoom) and remove less relevant data (Filter). Then from this subset, individual data items or groups of items can be selected for more information (Details-on-demand). Within a fully interactive visualization system, the user can switch between these processes, go back to the overview to look for different patterns and see relationships between data items.

2.2. Visualization of High Dimensional Data

In this thesis, dimensions are referred to in the context as in features, variables or attributes that are in the dataset, as different measurements are taken from each sample. In the case of a Microbiomics dataset, the dimensions would relate to different microbial species with their values attributing to their counts within different samples.

The key interest in existing methods was in how they support the exploration process of high dimensional data visualisation and a number of criteria was considered to evaluate these methods. Table 7.4 compares HieraViz, the tool generated by the project, to the methods covered in the rest of the Visualization of High Dimensional Data section.

The first main aspect considered when exploring these methods was the scale of dimensionality the method could feasible handle. This will be limited to the dimensionality that the author mentions, which may be around the limit of what the method could handle, or it could be lower. The second aspect user engagement, how interactive is the method? how does the system keep the user involved in the exploration process? Can users make changes to parameters on the fly, or are they stuck having to restart everything whenever they need to make a change.

2.2.1. Challenges of High Dimensionality

One of the core problems in defining high dimensionality is what can be considered high dimensional is broad and often open to the interpretation of the writer, the terms ‘multi-dimensional’ and ‘high-dimensional’ are often used interchangeably. Bertini et al. (2011) define high dimensionality as when a dataset becomes challenging to extrapolate any meaningful relations among dimensions, specifically from ten dimensions in the case of that paper. A study by Halford et al. (2005) into how many variables humans can process the relationships for before they start having difficulty, found that people start having problems at even four variables.

As such, the baseline of what can be considered high dimensional isn’t actually that high, yet there isn’t a ceiling for what can be defined as high dimensional. Many methods for dealing with high dimensionality cater toward these ‘lower’ scales of dimensionality, often in the tens or hundreds of dimensions; they are not necessarily applicable to these larger datasets of thousands or more (Anand et al., 2012; Turkay et al., 2012).

A good example of this is the Fisher’s Iris dataset (Fischer and Fisher, 1936); it consists of five dimensions across 150 samples, it is commonly used as an example dataset for machine learning or to show off methods (Grinstein et al., 2001). By comparison, the Tara Oceans microbial reference catalogue (Sunagawa et al., 2015), which is made up of collected microbial data from oceans across the world, consists of a 35,650-dimensional OTU table across 139 samples. Both datasets can be defined as high dimensional, but the difference in scale is significant, and it isn’t viable to use the same methods for exploration and analysis. For the purposes of this project, high dimensional is defined as greater than 1000 dimensions, while this too is an arbitrary number that may eventually become deprecated through the advancement of methods, the majority of the datasets used by the project met this criteria or were at least close to it.

In addition to these problems, the increasing complexity in trying to analyse high dimensional data can be attributed to the curse of dimensionality (Wright and Bellman, 1962). As the dimensionality increases, the volume of the data space increases, the data becomes sparser as distances between different objects become similar, and it becomes difficult to discriminate between them. This limits traditional methods of automatically finding patterns in the data unaided, clustering methods become limited as there are simply too many features and combinations of features to consider (Tatu et al., 2012), methods of statistical analysis suffer from there potentially being significantly more dimensions than samples within the sparse dataset (Dunkler et al., 2011; Johnstone and Titterington, 2009). It is these issues of dimensionality and scarcity caused by dimensionality which are the key properties that impact our ability to interact with this kind of data.

2.2.2. *N-D Visualization Methods*

A high dimensional dataset can be represented as a 2-dimensional table where its columns represent individual dimensions while its rows show samples. But this is not human-usable given the sheer amount of data presented unless the user knows what they are looking for and can filter out data they don't need. Any sort of exploration task for high dimensional data needs more tools than just a table.

While most visualization methods are designed to handle one-to-three dimensions, such as bar charts and scatter plots, there are a number of methods that do scale beyond this to handle N-dimensions (Grinstein et al., 2001). Often the main limitation of these methods is scalability, as while they theoretically can handle any number of dimensions, there is a reasonable limit to the natural scalability of these methods before you run out of screen space and the visualization becomes too cluttered to reasonably understand (Bertini et al., 2011).

One common N-D method is Scatter plot Matrices (Becker and Cleveland, 1987). These are made up of an N by N grid of scatter plots. Scatter plots themselves are some of the most common methods used in scientific visualization (Friendly and Denis, 2005) and combinations of scatter plots work well for showing the relationships and correlations for each pair of variables. As previously mentioned, the main issue is scalability; each matrix generates N2-N scatter plots, and even a 10-dimensional matrix will generate 90 scatter plots, a 100-dimensional matrix will generate 9900 scatter plots.

Another common method is Cluster Heatmaps. A 2D graphical display of a data matrix where node values are represented through a colour scale. Patterns displayed within the heatmap depend on the ordering of rows and columns, algorithms such as hierarchical clustering (Eisen et al., 1998) can be used to group and order nodes based on similarity, with the addition of a dendrogram to show the generated or included hierarchical structure.

Parallel Coordinates (Inselberg, 1985) make use of a series of parallel vertical axes to represent individual dimensions, with values scaling from lowest and the bottom to highest at the top for that specific dimension. Individual data items are displayed as polylines connecting across the axes at their specific values for that axis.

As an expansion to Parallel Coordinates, Hierarchical Parallel Coordinates (Ying-Huey Fua et al., 2003) make use of hierarchical clustering to structure the data and allow for the view to be abstracted at different levels, reducing the amount of clutter and allowing for trends in the data to be explored at the different levels without having to view the entire dataset. Aggregated Parallel Coordinates (Andrews et al., 2015) follow a similar concept, allowing for axes to be expanded to show child axes and collapsed.

The method is limited by axis arrangement, it is hard to compare axes that are not adjacent and different arrangements of axes may convey significantly different patterns. As such, often interactive methods allow for axes to be moved and individually filtered to see only polylines that pass through a certain range of an axis. There are also several different strategies to automatically order axes to show the most interesting correlations within the data. Zhou et al. (2018) use hierarchical clustering to order axes, while Blumenschein et al. (2020) orders axes based on their dissimilarity rather than similarity to better highlight patterns within high clutter datasets.

RadViz (Hoffman et al., 1997, 1999) is structured as a series of axes expanding out from the centre of the visualization to perimeter points equidistant from one another; data points are positioned based on the sum of their value for each dimensional axis, if a data point has a higher value for one dimension over another, it will be closer to that dimensions perimeter point whereas data points with lower values will appear closer to the centre.

As with Parallel coordinates, the method is limited by axis arrangement, a data point near the centre might have low values across multiple dimensions, or it may have high values in two dimensions opposite each other, resulting in interesting data points and outliers being potentially mixed with less interesting data points. One augmentation of the method, RadViz++ (de Carvalho Pagliosa and Telea, 2019) improves the clarity by making use of Pearson's correlation to rearrange axes based on their similarity.

By default, these methods only function well for small scales of dimensionality; however, the fact that these methods enable the visualization of multiple dimensions and the comparison of subsets of dimensions greater than two means they do feature greatly as components in overall solutions for high dimensional visualization. It allows the same method to be used to show the full dataset and the reduced dataset, allowing for overarching patterns in the dataset to be used as a start for exploration, while something like a 2D scatter plot is only useful after reduction has been performed, limiting exploration options.

2.2.3. Dimensionality Reduction

The amount of time it takes to understand a high dimensional dataset will vary, depending on if a user has a specific question they want to answer or if they intend to explore the dataset freely, looking for interesting patterns. The former will likely take less time than the later, since the user will know more specifically what they are looking for, but without some way to provide abstraction to the dataset, be it visual or programmatic, understanding the vast complexity of these high dimensional datasets unaided is not feasible for a person to accomplish. Presenting a user with a table with hundreds or thousands of rows and columns showing dimension and samples or something like a scatter-plot matrix with thousands of scatterplots would take too much effort to understand manually.

A common solution to mitigate high dimensionality problems is to make use of dimensionality reductions methods. They are used to represent high dimensional data in a lower-dimensional space while trying to preserve the structure and any interesting aspects of the dataset.

Multidimensional scaling (MDS) (Oppenheim and Torgerson, 1961) is a set of methods for representing data based on the similarity or dissimilarity of objects. The method performs reduction by calculating the pairwise distance between N objects in the dataset to produce N points in a 2D view.

One method of MDS, Principal Components Analysis (PCA) (Jolliffe, 2002) remains one of the most popular methods for dimensionality reduction, it looks for linear combinations of dimensions that have the most variance within the dataset. These linear combinations form the new features/dimensions of the reduced dataset. However, there is no interpretable connection between the original dimensions and the principal components, which limits usability if the user needs to know.

t-distributed stochastic neighbour embedding (t-SNE) (Van Der Maaten and Hinton, 2008) is an unsupervised, nonlinear reduction method that calculates the probability of similarity for points in the high-dimensional space and calculates the probability of similarity of points in the corresponding low-dimensional space. Then the method attempts to lower the difference between these two probabilities to map the high dimensional space to the low space.

Random projections accept a controlled amount of error in return for being simple and computationally efficient. The method looks to project each dimension into a lower-dimensional subspace, while dimensions and distribution of random projection matrices are controlled in order to approximately preserve the pairwise distances between any two samples of the dataset. Anand et al. (2012) use random projections alongside binning to reduce datasets of 16000 dimensions in order to generate interesting views to be further explored within a set of coordinated views.

As previously mentioned, these methods are often limited by the curse of dimensionality (Wright and Bellman, 1962). As the dimensionality increases, the amount of important dimensions increases, and the data becomes sparser as the volume of the dataset increases.

In a study by Espadoto et al. (2019) comparing a number of different dimensionality reduction methods in the context of machine learning, they found the most challenging aspect for a reduction method was the intrinsic dimensionality of the dataset, followed by the sparsity of the dataset. Intrinsic dimensionality is effectively the number of reduced features required to describe the dataset. The higher the intrinsic dimensionality of the dataset is, the harder it is the project the dataset to a lower dimensionality. Sparsity in the context of their study refers to the range of data values rather than there being more dimensions than samples, but describes a similar problem. The sparser the dataset, the more values the reduction method needs to consider.

Subspace clustering offers an alternative approach to reducing high dimensional datasets to more manageable representations. Rather than looking to remove, merge or select individual dimensions to keep, instead it looks for interesting subsets of dimensions. Interesting patterns may only be present in subspaces made of certain combinations of features, and dimensionality reduction may find the individual components of these subspaces to not be individually interesting and discard them

Tatu et al. (2012) present a method to visually explore potential subspaces of interest within the dataset, based on using the SURFING subspace algorithm (Baumgartner et al., 2004) to generate subspaces which are then processed with subspace grouping and filtering to understand the relationships of subspaces. These filtered subspaces are then displayed for the with selecting a subspace highlighting its dimensions across different clusters. The method is limited however to handling 10's of dimensions, due to a combination of the computational limitations of the subspace search and the space required to display its views.

Pattern trails (Jackle et al., 2018) order and compare patterns between subspaces in a series of linked views. Subspaces are generated through SURFING to form a hierarchical clustering structure, following which the subspace projections can be grouped together based on similarity measures set by the users, whilst showing the connections between data points in different subspaces. The method provides automated support finding interesting patterns based on user input. The method focuses on a dataset of only 8 dimensions, as the subspace set they were dealing with was large. No usability study was conducted to evaluate the method or to determine user acceptance.

Zhou et al. (2016) expand on subspace analysis, with a method generates subspaces, allows the user to explore the subspaces, and interact with them through methods including assigning weights and merging subspaces, after which dimensions can be reconstructed either manually or automatically, which can be fed back into the subspace analysis process. This method is able to handle high dimensional datasets than the previous subspace methods, with a 221 dimensional dataset being the largest mentioned, though this is still lower than the 1000+ dimensional datasets the project is aiming for, discussions with expert users also stated concerns of how scalable the method would be.

Hierarchical clustering (Eisen et al., 1998) is a method that seeks to build a hierarchical structure of clusters through merging dimensions. The first variant, Agglomerative clustering, looks for the two most similar dimensions based on a selected distance metric and merges them into a clustering, repeating the process until all clusters have been merged into one, providing a hierarchy based on how dimensions have been merged, normally represented through a dendrogram attached to the visualization. The second variant, Divisive clustering, starts all dimensions in one cluster and begins splitting ones that are not similar to form the hierarchy.

Fully automated methods can lead to an unintuitive relationship between the original and reduced datasets, due to the relevance of individual and sets of dimensions are decided by the system. These reduced datasets will be linear combinations of the original dimensions, if the original dimensions selected by the reduction method are relevant, then the reduced dataset will be intuitive, but alone, the methods cannot take advantage of the domain knowledge of the user to understand why certain dimensions are important and risk eliminating useful subspaces in favour of what could potentially be useless.

2.2.4. Interactive Dimensionality Reduction

A common approach to dealing with high dimensionality in visualization is to use quality metrics, which can be defined as calculated metrics that capture data properties that are useful for the extraction of meaningful information about data (Behrisch et al., 2018; Bertini et al., 2011). Quality metrics are commonly used for tasks such as projection, ordering, abstraction and view optimisation (Bertini et al., 2011) and can support the data analyst to concentrate on the most information, bearing part of the data. A variety of Quality Metrics have been suggested for the exploration of high dimensional data.

Johansson and Johansson (2009) rank dimensions based on cluster, correlation and outlier metrics, to guide dimensionality reduction. The user is presented with the relationship between the number of variables retained and the amount of information they will lose as they change the values of various quality metrics before performing reduction using the selected parameters. The reduction process can be modified and repeated until the users reach a reduction they are happy with. The method lists working with datasets of 100 variables. Building upon that, Fernstad et al. (2011, 2013) used Parallel Coordinates as a visual representation of the dimensions' Quality Metric profiles to provide overview of patterns in the full dataset and to guide the selection of dimension subsets.

Turkay et al. (2011) presented a related approach where high dimensional data was represented in a separate item and dimension spaces, where dimensions were presented in the context of various statistical properties. Data points in each space can be selected through brushing, updating the subset of items or dimensions shown in the other view, allowing for iterative reduction and exploration. In its use case, the method works with a dataset consisting of 7129 gene dimensions.

Krause et al. (2017) present the dataset to the user in the form of one-dimensional frequency plots, which allows the user to build their own subspaces through manual selection and from suggestions of interesting dimensions presented by the system based on the currently selected dimensions through subspace clustering and correlation. These subspaces can then be explored and iteratively updated within a series of combined views. Doing so makes the generation of subspaces entirely transparent The paper stated the method could comfortably handle >100 dimensions, with a use-case using a 147 dimensional dataset.

Albuquerque et al. (2010) presented a slightly different approach extending existing quality metrics used by other visualization methods and adapting them for optimising the visual analysis process of Radviz, Pixel-Oriented Displays and Table Lens methods. Some interaction is presented through the selecting of regions, but nothing else was mentioned regarding user engagement. The highest dimensional dataset mentioned being used in the paper was at 73 dimensions.

Wang et al. (2019) provides subspace comparison through aggregating high dimensional data and allowing for incremental analysis. Subspaces are represented in the form of triangular matrices in order to better present the similarity/dissimilarity between subspaces. These are then clustered hierarchical based on their similarity. As with some of the earlier subspace approaches mentioned, the paper states scale of dimensionality that can be processed by the method is limited by the number of interspersing subspaces filtered out by the clustering algorithms, and therefore the method can mostly only handle around 20 dimensions comfortably. The method allows the user to interact with the views to select, add and remove subspaces, recomputing and making updates to the visualisation on the fly.

Turkay et al. (2012) ranks dimensions against statistics and then performs reduction locally on a group of similar dimensions to form a representative dimension to compare with the rest of the dataset. Doing so mitigates the issues of using dimensionality reduction on high dimensional data, as each representative factor is performed on a smaller number of dimensions. The reduction process is iterative, with users selecting groups of dimensions and existing representative factors to be formed into new representative factors, repeating the process across different structures in the dataset. The largest dataset mentioned using the tool was 357 dimensions with 83 samples.

Progressive computations (Turkay et al., 2017) is another method presented by Turkay. It mitigates the issues of waiting for the visualization to update when processing large amounts of data by computing as good as possible result within one second, then updating the visualization, and iteratively repeating this process until the computation is complete. The process allows users to see results evolve and make hypothesis decisions earlier without having to wait for the full computational process to finish, keeping them engaged with the process and able to make new decisions on what to do next even if the system hasn't finished updating. 77 Dimensions was the largest dataset mentioned.

While these are useful approaches for dealing with high dimensional data with hundreds or possibly a few thousand dimensions, they are unlikely to be able to deal with tens of thousands

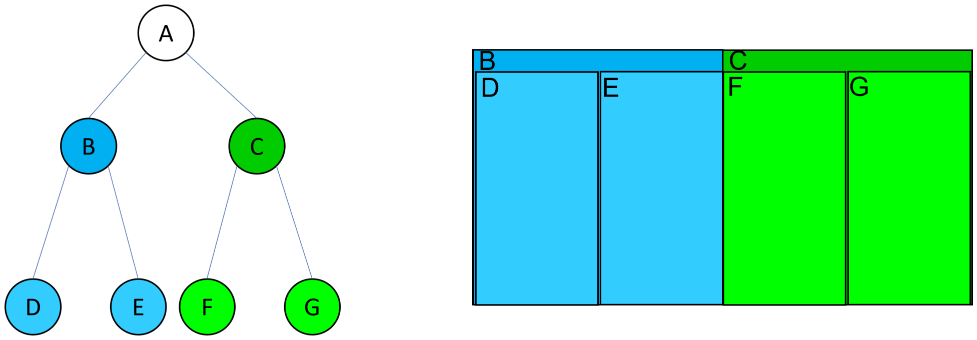


Figure 2.1 A Explicit Node link diagram on the left and an Implicit Treemap on the Right.

of dimensions efficiently. Furthermore, they are generally unable to provide a visual overview of patterns in the whole high dimensional dataset.

2.3. Hierarchical Visualization

For the exploration and analysis of high dimensional data, hierarchical visualization provides a useful abstraction. The complexity of having a high number of dimensions is mitigated by the hierarchical structure, which reveals meaningful relations between the dimensions. This fits well into the information-seeking mantra (Shneiderman, 1996); the top-level provides a full overview of the dataset, from which the user can zoom and filter through the hierarchy at different levels to find details that may be of interest.

These methods can take advantage of an included hierarchical structure within the dataset, such as a filesystem or taxonomical classification of species in biology. Alternatively, a hierarchical structure can be generated for the data using hierarchical clustering or subspace clustering (Tatu et al., 2012) methods, though these may be limited when dealing with higher scales of dimensionality.

Hierarchical visualization methods can be implicit or explicit (Schulz and Schumann, 2006). Explicit methods display their parent-child structure between nodes through lines. In contrast, Implicit methods are space filing and show their structure and relations through node arrangement, as seen in Figure 2.1.

Explicit methods are limited for high dimensionality problems unaided, as they take up a lot of screen space since the parent-child structure is made through external edges, as seen in Figure 2.2. For any large and complex dataset, this can be a problem as, without a method to interact with the visualization, it will quickly become exponentially larger than the screen allows for. Even for the simple case of just a simple binary tree, the maximum number of nodes at any given level N is 2N.

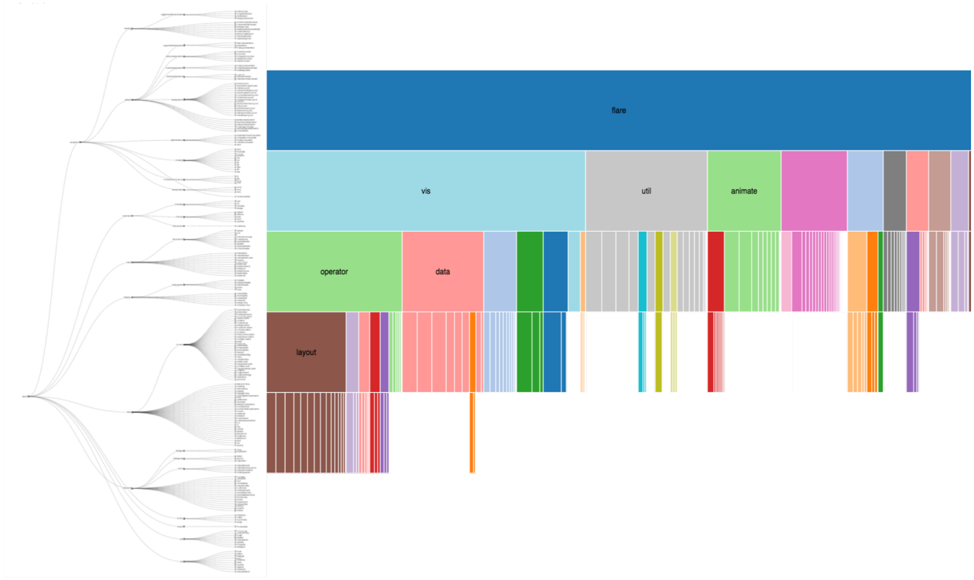


Figure 2.2 A Explicit Node link diagram on the left and an Implicit Icicle Plot on the Right. Both use the same dataset.

Implicit hierarchical methods, on the other hand, are less affected by this issue (Schulz et al., 2011) as they are often more space efficient since they do not rely on edges to show relations. They do still suffer from clutter as the data complexity increases. It becomes difficult to view smaller nodes and difficult to interact with without a method to zoom in or filter data.

There are a large number of different methods and variations of methods, the full scope of which can be explored on TreeViz.net (Schulz, 2011). The following sections will discuss some of the main hierarchical visualizations of most relevance for the research of this thesis.

2.3.1. Explicit methods

Explicit methods were not considered for use within the project. Their structure requires too much space to handle the scale of a high dimensional dataset without making compromises to how much of the dataset can be shown at once or requiring some abstraction of the data. Additionally, implicit methods are considered more interesting for the project due to their ability to show additional information within their nodes.

However, this isn't to say explicit methods cannot be used as part of a solution for displaying high dimensional datasets. Dinkla et al. (2011) use a node-link diagram as the main view to show the structure of the dataset but has it linked to separate views to show the details of selected nodes. Cuenca et al. (2018) combine streamgraphs with a hierarchical tree to allow for the exploration of multiple time series datasets organised into a hierarchical structure.

Scalability of the structure is still an issue; any large structure will not be visible within a single screen and will need ways to filter and interact with it to reduce it to a small enough visual structure to understand. The metagenomic visualization tool MEGAN (Huson et al., 2007) have expandable trees that can be explored. A later version is updated to include pie chart glyphs on top of nodes (Tipney et al., 2009), showing the abundance values for different datasets being compared.

Hyperbolic Trees (Lamping et al., 1995) display a tree within a circle, with its route at the centre expanding outwards. They mitigate the main issue of clutter that normal trees do by displaying the structure within hyperbolic space rather than Euclidean. Increasing the radius of the hyperbolic space increases its circumference exponentially, allowing for less space to be used on further out nodes, allowing the tree being laid out with less clutter than if it had been a Euclidean circle. Focus can be shifted to bring certain parts of the structure to the centre, though there still is limited space and it can be difficult to make out details of the full structure, especially at the extent of the structure.

2.3.2. Treemap

The Treemap (Shneiderman and Ben, 1992) provides a representation of a more traditional tree structure, often in the form of a series of rectangles nested inside one another to form the hierarchical structure, using a space filling algorithm to set the area of nodes based on some metric of abundance. This provides both information regarding structure and the size of each node visually without any additional information, allowing for the quick comparison of nodes to spot potential outliers or nodes of interest.

Since its hierarchy is nested internally, the Treemap is space-efficient in comparison to other methods. However nodes can suffer from cluttering as the complexity of the data increases, potentially not being rendered due to the size difference. Additionally, the original layout structure of the Treemap, slice and dice has a high width-to-height aspect ratio, resulting in long but thin rectangles which are hard compare and to interact with.

Multiple different layout structures have been developed to combat this, such as Squarified Treemaps (Bruls et al., 2000), by having nodes as squares rather than rectangle's eliminate the aspect ratio problem. Another Treemap method, designed to show stock market data (Wattenberg, 2003), partitions both vertically and horizontally to generate its structure and use measures of similarity to put similar nodes near one another. However, the layout of these methods can change dramatically from updates to the dataset, limiting their application in any system that is updated in real-time as it ruins the user's ability to remember where nodes were.

Zoomable Treemaps (Blanch and Lecolinet, 2007) follow the same layout structure as the standard Treemap but expand upon interactivity with the ability to drill down and roll up the hierarchical structure, allowing for the exploration of nodes further down the hierarchy in greater detail. Cushion Treemaps (van Wijk and van de Wetering, 1999) make use of shading to highlight the depth of the hierarchical structure.

2.3.3. Circular Treemaps

Pebble Treemaps (Haisen Zhao and Lu, 2015; Wetzel, 2003) are a variation of Treemap that uses nested circles rather than rectangles to represent nodes. The outer circle represents the root, with child nodes as the inner nested circles. While circle areas are used to show abundance, similarly to the other methods, the areas do not hold a direct one-to-one relationship to the abundance value at all levels, as the total area of a parent node is larger than the sum of the areas of its child nodes.

This results in the hierarchical structure of the method being clearer than the traditional Treemaps, as the aspect ratios of the circular nodes are unified, allowing for the hierarchical structure to be more easily perceived due to the extra space emphasising parent–child relationships. However, as a result, it is also relatively space-consuming, with lots of unfilled space within the visualization.

Haisen Zhao and Lu (2015) present variational circular Treemaps, an expansion to the method which reduces the amount of wasted space through a variational layout algorithm. Exploring the structure using drill-down and roll-up operations works well with the unified aspect ratio of the circular nodes. Additionally the method uses fisheye distortion (Wood, 1906) to support interaction with small nodes, progressively resizing nodes that are selected.

Another take on the method, the Bubble Treemap (Görtler et al., 2018) follows a similar concept to the other circular Treemaps, but eliminates the problem of wasted space further. The methods provide a more compact visualization, still using circular nodes but enclosed in a contour to show hierarchical structure. Doing so allows it to both consume less space while having the room to present additional information within nodes, in this case displaying the uncertainty of the node through using radial gradients on the contour.

Similarly to the Treemap, the leaf node patterns can be fairly easily identified in the Circular Treemap and the Bubble Treemap. Meanwhile, the diversities at other levels are harder to overview. Compared to Circular Treemap, Treemap and Bubble Treemap have more similar size representations and more clearly display the four considerably larger leaf nodes.

2.3.4. Voronoi Treemaps

Voronoi Treemaps (Balzer and Deussen, 2005) provide an alternative layout, using arbitrary polygons generated through Voronoi tessellations rather than rectangles for nodes. Doing so eliminates issues regarding to the aspect ratios problems the standard Treemap faces, as rather than identical long rectangles, the nodes and hierarchical structure all being different shapes makes them easier to distinguish and compare.

Additionally, since the nodes are all arbitrary polygons rather than rectangles, this allows for Treemaps to be generated within different display areas to suit different applications. The method is more computationally intensive than the normal Treemap; the additional freedom in the shape of individual nodes requires additional error checking to generate a structure, additionally some nodes may be sized larger or smaller than their actual abundance to avoid leaving holes in the visualization.

2.3.5. Icicle Plots

Icicle plots (Kruskal and Landwehr, 1983) are a method created originally to show the structure of hierarchically clustered data for statistical analysis. The plot displays its hierarchical structure vertically downwards, just like its namesake (though some variants show this in different orientations, such as vertically up (Brendan D. Gregg, 2011)), with parents nodes at the top of the structure and child nodes directly below.

The hierarchical structure is presented clearly, similar in appearance to a stacked bar chart, its easy to see what level a node is at and its respective children and parents' nodes. As with the Treemap, without interactivity, small nodes are difficult to read when the dataset is large and complex, especially since the Icicle plot uses relatively more space for its root node and nodes close to the root.

2.3.6. Radial Charts

Sunburst charts (Stasko and Zhang, 2000) are structured similarly to Icicle plots, using the adjacency of nodes to show parent-child relationships, but are displayed radially using polar coordinates, rather than being flat. Different levels of hierarchy are displayed outwards from the centre, providing a full overview of the hierarchical structure while still using the width to show node sizes in comparison to other nodes on the same level. In comparison to Treemaps, the hierarchical structure is more clear (Stasko et al., 2000), as again, like the Icicle plot, the path from parent to child nodes follows on from one another rather than being nested inside one another. In comparisons to other hierarchical structures, the method is generally considered aesthetically pleasing (Cawthon and Moere, 2007), while this doesn't necessarily mean the method is more efficient, it does have a positive impact in retaining user interest for using the method, having them want to engage with the method is a useful feature.

One problem with this method is it can be difficult to compare the sizes of different nodes, both in different areas of the sunburst, as nodes are arcs at different rotations, and at different levels. Physically an external node may be larger than an internal node, but the internal node will likely hold a high data value since it is higher up the hierarchical structure.

There are several other radial methods that have a similar structure to the sunburst chart. The Polar Treemap (Johnson, 1993), is an earlier method that has a similar layout to the sunburst chart, but the hierarchical structure is nested like a normal Treemap rather than based on the adjacency of nodes. The PieTree (O'Donnell et al., 2007) is an interactive pie chart that shows additional levels of hierarchy upon interaction with a node. InterRing (Yang et al., 2002) provides multiple ways for the user to distort the display space, dragging nodes and hierarchical levels to resize their width and thickness.

2.3.7. Evaluation of hierarchical visualization

A number of these methods have been compared and evaluated against one another across multiple studies. While these methods have their own strengths and weakness in different situations, there is some agreement of the effectiveness on some methods over others.

The following section was presented in in the 24th International Conference Information Visualisation (IV) (Macquisten et al., 2020), with an updated version with expanded content published as a chapter (Macquisten et al., 2022) in the book Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery (Kovalerchuk et al., 2022).

Stasko et al. (2000) compare Sunburst against Treemap within the context of file search tasks. The evaluation was measured on the correctness of results and completion time, asking users questions such as comparing the size of nodes and finding the path to nodes. The study found that Sunburst provided a greater understanding of the hierarchical structure than Treemap, especially initially, though that performance with Treemap did improve over time.

Another study by Muramalla et al. (2017) looked at Icicle and Sunburst, measuring user performance through a similar set of questions as the study by Stasko et al. (2000), looking for nodes based on size, depth within the structure and path to the node. They found that users performed better and were more efficient with Icicle plot but preferred the layout of Sunburst.

Nicholas et al. (2017) compares Icicle against Treemap, alongside Nodelink diagrams. Each method was shown at two different levels of hierarchical complexity (2 and 3 levels), and tasks consisted of counting and comparing nodes, with user performance measured by accuracy, time to completion and through eye tracking. The study found Icicle plot and Nodelink performed comparably to one another, while Treemap performed worse. Interestingly, they found through eye-tracking that Treemaps draw visual attention better than the other two despite its performance.

Long et al. (2017) compares four types of Treemap, the original, Circular Treemaps, Cushion Treemaps and 3D Treemaps, as a comparison of the effectiveness of these different tree structures. Questions were based around finding certain nodes, counting nodes and the size of the hierarchy. The study found that 3D Treemap was the best, followed by Circular Treemaps, normal Treemaps and cushion Treemaps performing the worst.

Woodburn et al. (2019) compare Treemaps, Icicle and a method they designed, a semicircular variant of sunburst called a Sundown chart. Sunburst was also compared but only within the pilot study. Their study compared performance for basic navigation tasks, node attribute (size) comparison, and understanding of hierarchical structure. Results found slight user preference in Icicle to their own method, and generally better performance for Icicle and Sundown chart than for Treemap.

These previous evaluations of hierarchical visualization methods have mainly focused on tasks directly related to the hierarchical structure, while the evaluation presented later in this Thesis (Chapter 4) focuses on the presentation of underlying data represented by node attributes (size and colour). The results from Woodburn et al. (2019) indicate that Treemap and Icicle may perform better than Sundown chart for size comparison tasks, but results were not statistically significant. These methods mostly only compared a single size of hierarchy. While Nicholas et al. (2017) compared hierarchies of different complexity, the most complex still included only three levels, which is considerably less than in many hierarchical datasets.

2.4. Visualization of Omics Data

’Omics covers various fields of biological study, such as Microbiomics, Genomics, Transcriptomics and Proteomics. Each looks to measure different types of molecules from the same set of samples. For example, within Microbiomics, a user would compare the frequencies of different bacterial species in different samples, while in Proteomics, users quantify proteins (Bersanelli et al., 2016).

Across these different fields, data is often very high dimensional and may often include thousands or even millions of unique biological entities, with a single entity of interest correlating to a single dimension in the dataset, such as Operational Taxonomic Units (OTUs), genes or enzymes. Datasets with greater than a thousand dimensions are not uncommon. The Tara Oceans expedition (Sunagawa et al., 2015), which collected data from oceans across the world, identified >40 million genes and 35,650 OTUs.

The increase in dimensionality and complexity of these datasets over the last few decades has been due to the development and advances of next-generation sequencing technologies, data collection and generation have become more efficient. At the time of the Human genome project completion in 2003, it cost about $50 million to sequence a human genome; in 2016, it cost around $1000 (NHGRI, 2021). However, this superior data collection hasn’t led to superior knowledge generation, and often the tools for analysis are not as cost-effective (Sboner et al., 2011).

As for the issues of high dimensional data in general, the scalability of the methods is the issue. Data spaces can often be sparse, as the number of samples is often relatively small and much lower than the number of dimensions. Many data analysis and visualization methods are designed mainly for data where the number of samples is considerably higher than the number of dimensions and may hence not be usable for sparse high dimensional data.

Despite the variance among the different ’omics fields, there is a fair amount of correlation in what problems different biologists are facing, mostly revolving around the comparison of entities, such as comparing groups of dimensions, multiple samples or a user-generated experimental dataset to existing public datasets.

The full scope of ‘omics analysis is vast, and building a single tool to do everything is not feasible; rather a visualization tool may be designed for a single or set of tasks, or it may be a platform for a number of different tools. For example, Qiime2 (Bolyen et al., 2018) is a suite of tools to process and visualize microbiome data. It is designed to support a third-party plugin to cover whatever functionality the user needs. Another tool Phinch (Bik and Inc., 2014), provides users with the ability to search and filter through raw data, selecting and filtering individual samples and sample groups/attributes and being able to display within a series of interactive visualizations.

Hierarchically structured data is not uncommon across various ‘omics fields (Kuznetsova et al., 2017). For example, in microbiomics, biological entities can often be classified at different taxonomic levels through the biological classification system, which generates a hierarchical description of taxonomic relationships in the data. Each hierarchical level includes a set of taxa, and each taxon consists of a set of underlying biological entities. Various hierarchical visualization methods such as dendrograms, hyperbolic trees, Treemaps and sunburst charts all are used as visualization solutions or components to these solutions to represent these structures.

Ondov et al. (2011) present Krona, an interactive metagenomics visualization tool that runs in web browsers. The visualization consists of a radial display similar to a sunburst chart, with the root of the hierarchical structure in the centre with children nodes expanding outwards. Selecting any node within the Krona chart will zoom in to that part of the structure, bringing it to the top of the hierarchy, allowing for nodes with lower abundance values to be more visible.

Hebrard and Taylor (2016) present the tool MetaTreemap as an alternative way to represent metagenomics phylogenetic trees. The main components are a Treemap and table, both show the hierarchical structure of the dataset side by side, allowing exploration and searching in either view.

MetaViz, created by Wagner et al. (2018) is a Multiview tool for statistical and visual analysis of metagenomic data. It allows the user to add various linked views to show different aspects of the data, such as PCA scatter plots, heatmaps, and sunburst charts, interactions in one view show up in all other views for linked data. The hierarchical structure of the dataset is shown through a view called FacetZoom, effectively an upside down icicle plot; nodes can be expanded to zoom in on them and see only that part of the structure.

Fernstad et al. (2011) show MicrobiVis, a tool that enables explorative dimensional reduction and visual exploration of microbial populations. A data view displays the microbial population using a visual representation of the user's choice, while a Ranking and quality view displays quality values and ranks OTUs in a Parallel Coordinates plot. The ranking makes use of multiple metrics such as abundance (total amount of an OTU across all samples) and prevalence (number of samples where the OTU is detected) which can be used in tandem with one another to provide an overall ranking of quality for OTUs.

Johansson et al. (2020) presents a study evaluating how bio-science experts perceived different quality metrics for discovering patterns in genomics data, presenting the percentage ranking of those metrics listed as 'interesting' and 'very interesting'. The metrics ranked collectively as the most interesting by participants were abundance difference of sample groups, prevalence difference of sample groups, and similarities between entities. These metrics were then divided into different categories of similar metrics with examples of their usage.

McNally et al. (2018) introduce the tool BURRITO, a multi-omics tool for visualizing microbiome taxonomical groups and gene functional data. Both the taxonomic and functional profiles have their own hierarchical structure, shown within expandable trees visualizations. Additionally, a pair of stacked bar plots show the relative abundance of the taxonomic and functional compositions across different samples. These are linked together so that interactions in one chart highlight corresponding functions or taxa in the other. Finally, a bipartite graph shows individual taxa and functions alongside the links between them.

2.5. Visualization of data diversity, distributions and statistics

There are several different methods for representing distributions, and each serves better in different situations. Statistics can be visualised in just a single/set of standard statistical visualization methods. They can be represented through some properties of the visualization, such as a colour range to show areas of low statistical interest, or as a unique visual property, such as a glyph to show statistical values.

Box plots (Tukey, 1977) display the spread and skewness groups of numerical data by means of their quartiles: minimum, first quartile, median, third quartile, and maximum. Box plots work well for comparing these quartiles against one another and other box plots, but they only provide a simple summary of the distribution of data.

Histograms (Pearson, 1895) organise groups of data points into equally sized bins that are displayed as bars showing the frequency of those groups. Compared to box plots, histograms are good for showing the overall distribution of the data but can be misleading depending on the bin sizes.

Density plots (Rosenblatt, 1956) represent distribution as numerical values, using a kernel density function to make estimates of the probability density function for a sample. They function similarly to histograms; however, the bins and shape of the plot are smooth due to the kernel density function smoothing out the noise. Depending on the parameters, the data can be distorted through the plot being too smooth or not smooth enough.

Violin plots (Hintze and Nelson, 1998) are effectively box plots with a kernel density plot mirrored on either side. This provides the benefits of a box plot and a density distribution plot. The main limitations, specifically for this project, is that they are not that compact, due to being a mirrored plot, and need more space compared to different methods if you are intending to compare multiple plots.

In the case of methods being explored by the project, they are to be used within an implicit hierarchical structure, with nodes of the method showing the aggregated distribution of values for different sample groups. In order to be usable, the statistical visualization needs to be compact enough to fit within a potentially small node, while showing multiple instances of itself for the different sample groups, either by being overlaid or semi overlaid in order to fit within the variable space available, with interaction to allow a single sample group to be brought into focus.

A study performed by Blumenschein et al. (2020), compared 20 different descriptive statistics charts, covering a range of histogram, shape and statistical-property based visualizations. Each method was evaluated in a series of analysis tasks divided between local tasks for analysing particular areas of the distribution, global tasks for analysing the whole distribution and aggregation tasks, which focus on aggregated statistical measures of distributions.

The study found that no single method was best suited for all tasks, a histogram might work well for local analysis tasks, but they are not useful for showing the statistical information for aggregated tasks. Density distributions worked best for global tasks but were less useful for local tasks. The results of the survey were used to guide the development of a hybrid chart builder for comparing two distributions, allowing users to add and remove visual components to customise the analysis process.

One aspect to consider is there isn't much literature for comparing more than two distributions at once. As Blumenschein et al. (2020) state, examples for comparing multiple distributions are limited and are often multiple charts overlaid or placed side by side.

Cluster separation metrics can be used to evaluate if groups of samples are well separated. An example of this is silhouette analysis (Rousseeuw, 1987). In silhouette analysis, the relative quality of a cluster is defined as a ratio between the within dissimilarity and the between dissimilarities of the items in the clusters. Silhouette values are generated by calculating the distances of samples within a sample group compared to the distance for a sample to samples in other sample groups. Higher Silhouette values indicate better-separated sample groups.

Another method, The Davies–Bouldin index (Davies and Bouldin, 1979) calculates an average similarity measure of a cluster with the cluster it is most similar with. It calculates the average Euclidean distance between the centroid of all clusters and their individual members. The Dunn Index (Dunn, 2008) is calculated as a ratio between the minimum inter-cluster distances over the maximum within-cluster distances.

2.6. Glyph Visualization

Within the context of data visualization, a glyph is a visual entity that represents one or more data values through graphical attributes. There is no strict definition of what a glyph is and how it differs from other visualization methods (Munzner, 2018); the term is broad as it can be used for a range of methods for different purposes with different sizes and structures (Ward, 2007).

From definitions provided by Borgo et al. (2013), a glyph is a small visual object that can be used independently to depict attributes of the data record. They are placed interdependently from each other in the display space or connected to show relationships within the data. Glyphs are a form of visual sign that differs from other forms of visual signs, such as icons, indices and symbols but can still make use of visual features of other visual signs.

Glyphs work as a method for showing multi-dimensional and multivariate data; individual dimensions and data values are mapped onto the physical attributes of the glyph, such as its position, shape, size, orientation, and colour (Ward, 2007); these mappings can be one-to-one, one-to-many or many-one.

In one-to-one mappings, each dimension is mapped to a distinct visual entity in order to take advantage of the users' domain knowledge, paring data to graphical attributes, such as the length of a line to show abundance or the colour of a node to show the intensity of a value.
One-to-many, where each dimension is mapped to several types of visual entity, redundant mapping works well for low dimensionality datasets when additional graphical attributes can be added to reduce misinterpretation of the data without overloading the user. Many-to-one mappings have several or all dimensions and attributes mapped to a single type of visual attribute, useful in situations to compare different dimensions for the same sample. An example use would be multiple small glyphs plots side by side (Fuchs et al., 2013); this allows for a quick overview of different samples.

A glyph might be a re-scaled or abstracted version of a method, such as a pie chart, bar chart or box plot. It can be an individual component of a visualization, such as a single bar of a bar chart or a data point in a scatter plot. Alternatively, the glyph can be its own method entirely, designed for the task at hand, such as the method developed by Fuchs et al. (2015), where glyphs were designed to resemble a leaf as a general option for multivariate data. Using the structure of a leaf takes advantage of its natural shape to be ascetically pleasing and intuitive to distinguish between for users. Different visual characteristics of the leaf, such as its shape, its boundary, and its venation (arrangement of veins) are used to display different potential variables or dimensions. Visual clutter and overlap of the glyphs are minimised through alternative aggregation methods. An example use of the glyphs is provided, showing forest fire data as a scatter plot with data points shown as leaf glyphs.

Polyline Glyphs (Opach and Rød, 2018) are minimised thumbnails of the polylines of a parallel coordinates plot, showing the structure of the polyline without any of the axes. These are shown as small multiple plots that allow for quick comparisons between different polylines with the overlap present when they are all within the parallel coordinates plot, whilst being dynamically linked to the PCP itself to highlight where it is and its axis values. Both of which reduce the cluttering issue present to PCP and allow for investigation of sample groups without reduction.

A glyph can be used in combination with other glyphs to form a larger visualization. Potter et al. (2010), uses a combination of methods to form a summary plot and a glyph to summarise higher-order statistics and represent uncertainty. This Summary plot is made up of an abstracted box plot to show the range of data distribution. A statistical moments plot, which uses an assortment of small glyphs to represent different moments, including mean, standard deviation, skew, kurtosis and tailing, with their position within the summary plot pertaining to their value, a histogram to show the density of distribution and a distribution fit plot.

Alongside the design of the glyph, the placement of the glyph is an important consideration. There are a number of different placement strategies for glyphs (Ward, 2002); placement strategies can be either be data-driven, where the data directly tells the glyph where it should go, such as in a scatter plot, or it can be structure driven, where some implicit or explicit relationship between dimensions or values in the data dictate where glyphs should go, such as a hierarchical structure.

One common issue of using glyphs is the implicit bias in most mappings (Ward, 2002, 2007); different attributes and relationships are easier to identify than others. This can be based on proximity, such as adjacent attributes being easier to compare than ones that are further apart, such as axes of a star glyph. It can be on that certain graphical attributes are less challenging to perceive and measure than others, such as axis length compared to the angle of an axis. Additionally, it can be based on how attributes are grouped together if multiple attributes are used to show a single dimension or feature, it's easier to understand than just using a single attribute.

2.6.1. Glyphs for high dimensional data

For high dimensional data, this becomes more challenging; regardless of the compactness and simplicity provided by glyphs, the limitations of visualization methods for large datasets and for high dimensional data are still present. Screen space is still a limited resource; visual clutter reduces the usability of methods as the data complexity increases. Alongside this, displaying too many glyphs at once results in overlap and occlusion (Ward, 2002). As such glyph design needs to consider these limitations if they are intended to be used for high dimensional data.

Cakmak et al. (2020) mitigate this through the aggregation of glyphs. The paper describes a series of glyphs used to abstract network data as a replacement for node-link diagrams. The lines connecting nodes are removed, and the nodes are replaced with two variants of circular glyph, one for single nodes and one for groups. Both types consist of a background colour to show some variable; this centre attribute is surrounded by a ring of coloured segments, with darker segments representing an area of the node which would have had a large number of connections to it at those points, while the lighter the area, the fewer connections to that part of the node, additionally a triangular arrow is used to show movement direction. The only difference is for the group cluster node. If several glyphs are in the same area of the diagram, they are aggregated into one and shown as a mini node-link diagram inside the circular glyph, showing the underlying spatiotemporal network of the group.

McNabb and Laramee (2019) present a glyph placement solution for use within multivariate maps. A limitation of choropleth maps normally is interesting values are often concentrated within a small, densely populated area of the map and resulting glyphs within those areas would either be small or suffer from overfitting. This method gets around this through the merging and aggregating of glyphs in close proximity; as the user zooms in and more space is available, aggregated glyphs split into any child glyphs they contain. The glyph itself can be selected and switched between by the user from a selection of pie charts, polar area charts, bar charts and star glyphs.

ClockMap (Fischer et al., 2012) use aggregated clockeye glyphs for hierarchical time series data within a circular Treemap. Each glyph is a circle subdivided into 24 slices representing an hour, each slice coloured to the value of the data, with no data being represented by empty space. Glyphs within a child node can be aggregated to provide a higher-level overview.

Soares et al. (2018) present an adaptive glyph for use within the hierarchical structure of a Treemap. The glyph is broken down into several overlapping layers, with the bottom layer using texture patterns, the second layer using colours, the third using geometric shapes, the fourth using a letter and the top layer using a number. The glyph scales in complexity depending on the available space within a node, first removing the text information layers, then the texture pattern layer and finally the shape layer, leaving only the colour layer for the smallest nodes. Doing so only displays visual variables that the user can perceive, reducing visual clutter.

The application of glyphs within high dimensional data is of interest to the research; while individually a glyph can't handle the scale of dimensionality the project is looking at, they can still provide good representations of smaller scales of dimensionality to show some aspect of a data subset or statistical proprieties.

Chapter 3. Methodology and Requirements

This chapter will cover he methodology decided upon for the research and how it was modified to fit the projects needs; along with the challenges presented to the project and the requirements generated through interactions and feedback from end-users.

3.1. Methodology

The primary goal of the project was the creation of methods to support the exploratory visual analysis of high-dimensional 'omics data. With said methods being designed to support the user in their analysis by allowing them to apply their domain knowledge in finding interesting subsets of data. This goal would be achieved through the creation of a software tool that provides a means to display high dimensional datasets in full, and provide the user with the tools to manipulate the data into manageable subsets suitable for their analysis tasks.

The main methodology observed throughout the project was the nested model for visualisation design and validation method described by (Munzner, 2018). The methodology as show in Figure 3.1, breaks down the design process into four nested levels, with the output from a level being in input for the on below, eventually returning to the level once its nested processes have been completed, bringing attention to any design challenges that an error in an earlier level can inflict upon the lower levels, misunderstanding user needs in the first level for example can lead to creating a solution to the wrong problem.

After an phase of gathering a solid background of the main literature and visualization challenges at hand, the project was able to organise two sets of interviews with Unilever staff, in order to generate requirements and characterise the domain problems for the Domain situation level of the model. The interviews and the resulting requirements decided upon for the data and task abstraction level, are discussed in more depth in section 3.2.

For first main divergence of the model was there wasn't a strict progression of a visual encoding phase into algorithm phase into visual encoding validation phase, as instead this happened several times throughout the project before progressing to the data and task abstraction validation phase. Chapter 4 covers how different hierarchical methods were evaluated to determine which would be the best fit for different scales of hierarchical data, while Chapter 5

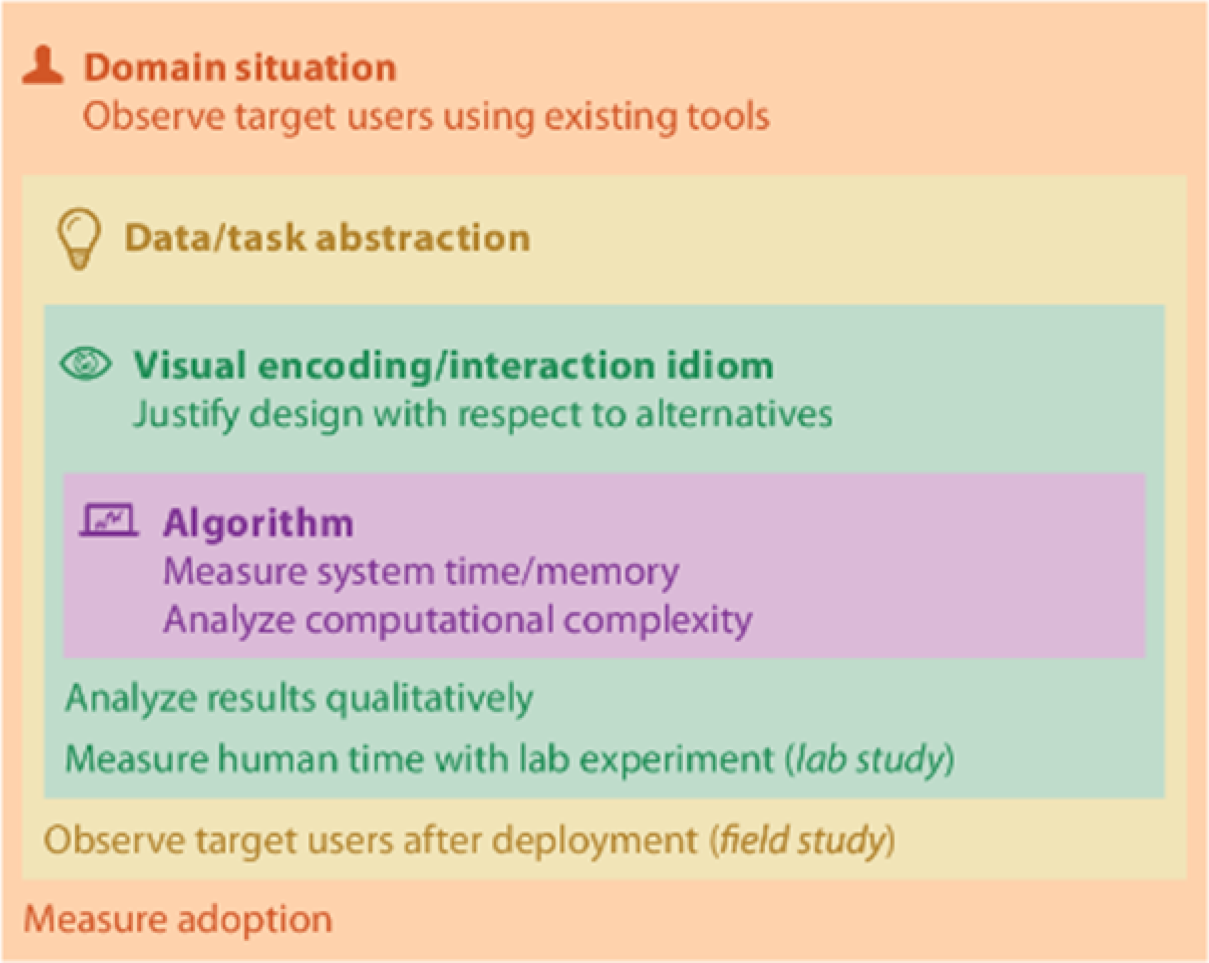

Domain situation
Observe target users using existing tools
Data/task abstraction
Visual encoding/interaction idiom
Justify design with respect to alternatives
Algorithm
Measure system time/memory
Analyze computational complexity
Analyze results qualitatively
Measure human time with lab experiment (*lab study*)
Observe target users after deployment (*field study*)
Measure adoption

Figure 3.1 The output of each level in the model becomes the input for the next, eventually leading back into its own level for validation after the nested levels have been completed. Image from (Munzner, 2018).

covers different methods of multivariate glyphs. Both of these methods had their own phases of justifying the selection of methods, working on the code to ensure the components worked with the rest of the system and an evaluation phase with a user study conducted for each method.

Following the user studies and the completion of the tool, the full tool was evaluated in a study by end users, as detailed in chapter 7 for the validation phase of the data and task abstraction level.

For the final level of the model, measuring adoption wasn't something entirely feasible within the scope of the research. It would have required the tool to be created and presented to Unilever within the course of the project, allowed for enough time for different users to get used to the tool and then do a follow up study, re-evaluating users opinions of the tool and comparing it to others. While there was nothing preventing this being a consideration as further research following the conclusion of the project, this step within the model was not conducted.

3.2. Challenges and requirements

The interviews followed the learning and discover phases of the design study methodology as described by (Sedlmair et al., 2012), a nine stage methodology where every step leads back to all previous steps, allowing for the process to advance to stages even if earlier stages are not complete and going back to earlier steps later make iterative improvements. With the focus of the project being on biological data, most of the interviewees were microbiologists, but there were also biologists of other fields, data scientists, statisticians and managers.

The first set of interviews consisted of a number of one-to-one meetings at the start of the project to gain a broad overview of the issues being faced, while the latter was a more in-depth placement, involving interviews with 26 different Unilever staff, in order to generate a more detailed list of requirements.

The latter interviews took place at two Unilever research sites, Port Sunlight in Liverpool, and Unilever Research Development in Colworth. For both research sites, before starting any meetings, an informal meeting was held between myself and all the interviewees to provide an introduction to the project and to brief interviewees on what I needed from the meetings. Each meeting was scheduled for an hour, the majority of questions I asked related to the tools the staff were currently using; the sorts of data they worked with and what the main challenges, and issues they faced in using those tools and analysis of that data.

One issue with these meeting was that were organised around the convenience of when people were available, which while understandable, did result in meetings where I was interviewing two or three people at a time. In these meetings in particular I got less useful information out of the interviewees, as it was difficult at times to maintain control of the meeting as the conversations did sometimes stray to be between the participants rather than with me. The most egregious example was in a meeting with three people from the same team, while one participants was explaining a concept to me, another interrupted them and stated to be careful what they said, as I would need publish this. While it is fully understandable that they would need to maintain privacy on some aspects of their operations, I didn't really get much out of that meeting.

In the event I were to do these interviews again, I would try to organise them all myself in order to have more one-to-one meetings. Having the bulk of the meetings at the two Unilever sites was useful to directly see how users worked, but I feel it would have been fine to have some of these meetings online if that aided scheduling enough to support more one-to-one meetings.

The experience of the interviewees with visualization varied from the microbiologists who had many year's worth of experience using visualization for data analysis while others, such as the managers, who would be more versed in using visualizations to present information in meetings rather than analysis it. However, even though some of the interviewees might not have as strong grasp on the data as some of the others, their insights were useful to get a broader picture.

Meeting with a varied group of different types of users generated a broad range of ideas and requests. Some were general requests like usability features that were not directly relevant to the research contribution but of importance for a usable and useful analysis tool. Others were outside the scope of the project, such as multi-omics analysis.

It was clear that it would be a balancing act to cater to both the needs of the project and all of these different stakeholders. It would be easy to let the list of requirements run wild with every ones different wants and needs, and this would risk the tool becoming too complicated to create within the duration of the project. The focus of requirements was based therefore more so on what was commonly said by the interviewees rather than every suggestion, need or want. But despite the wide breadth of inputs, there was a fair amount of consistency in what was discussed.

Scalability of analysis was a core issue, as mentioned in Chapter 2, development of next-generation sequencing technologies have led to improved data collection, but not to direct superior knowledge generation. Any methods developed need to take into account the potential of dealing with datasets with thousands or more dimensions; in order to display the full dataset visually to the user, the data needed to be abstracted in some way that still allowed the user to explore and look in more detail at sections of the dataset.

Whilst less of a direct research concern, the technical considerations are related to this; the scale of the data is a limiting factor for processing, and the average user is unlikely to have access to a powerful machine to process large datasets. Additionally, there are other factors impacting visual scalability (Eick and Karr, 2002). Limitations in human perception can affect how much data can be displayed as human eyes and brains can only pick and process so many visual patterns. Some visual metaphors will scale better than others.

The main analysis task commonly described by the users was in comparison of entities, be it comparing groups of dimensions, multiple sample groups or an experimental dataset to public datasets. Often the criteria microbiologists look to compare is some variation of finding the abundance (how much of an entity is in the data/sample) or prevalence (how often does an entity appear across samples) of an item or items. Often this may be the most abundant or prevalent entity, but it could also be the least abundant/prevalent entity or something common enough to be interesting but not universal to all samples, for example, a bacterial species that is absent before a process but more abundant after it.

Any tool developed would need to be a discovery platform, something that allowed for and encouraged the exploration of the dataset to find new information. It was vital, therefore that the tool would highlight potentially interesting structures within the dataset to guide the user. Otherwise they might be biased to just look towards structures that proved their hypothesis. Linked to this, the tool needed to be interactive and provide on-the-fly re-computation, a common complaint regarding a good deal of visualization tools was that they were static applications that just took in data and parameters.

When it came to statistics, a common issue users found in existing tools was the lack of statistical components, requiring data to be ported into another tool, which might not be fully compatible with the data, or wait until a statistician is available. What statistical measures to include needed careful consideration, as the amount of data could lead to false positives, high dimensional data is often very sparse, with relatively few numbers of samples compared to dimensions.

Further interactions with users was conducted throughout the project, mostly in the form of monthly meetings with the projects industrial supervisor. These were expanded upon with additional online meetings with users to show the progress of the tool and receive feedback. An additional trip to Unilever was made during the second year of the project to meet face to face with users again, in the same manner of the online meetings. Further trips were planned to Unilever in a form of a months long placement sometime during 3rd and 4th year, but this was cancelled due to the COVID-19 Pandemic.

All in all, following these interviews, the research was divided into three core objectives any method developed would need to address, as described in Chapter 1.2.

1) To address high dimensionality challenges by investigating solutions using hierarchical visualization methods

2) To enable efficient exploration of extremely high dimensional data by utilising approaches of visual guidance

3) To design novel visualization to enable representation of multivariate data features in aggregated data

The next few chapters will detail the studies carried out and methods developed to meet these objectives.

Chapter 4 covers the comparison of five different hierarchical visualization methods in relation to how well they represent underlying data features. In addition, each method was tested at two sizes of hierarchical complexity, a ’large’ structure consisting of five levels, and a ’small’ structure consisting of two levels. In doing so this study informed the selection of overview and detail & subset selection methods for this project, catering to **Objective 1**.

Chapter 5 covers two studies conducted to develop and select a method to show multivariate data features in aggregated data. The Thesis presents a combination of multiple overlaid Density distribution plots and multiple semi-overlaid abstracted box plots to form a Density box plot, as a method to show the aggregated distributions of multiple sample groups within the limited space of a node within the details view, covering **Objective 3**.

Objective 2 is explored more in Chapter 6, through the combination of all the studies and additional methods, applied into one single visualization solution.

Chapter 4. Hierarchical Evaluation

The following chapter was published in the 24th International Conference Information Visualisation (IV) (Macquisten et al., 2020), with an updated version with expanded content published as a chapter (Macquisten et al., 2022) in the book Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery (Kovalerchuk et al., 2022).

The project decided upon using Hierarchical visualization methods early on for two reasons. Firstly it provided a convenient way to display both full high dimensional datasets or subsets in the same structure, no matter how high dimensional a dataset was, it could be contained within a single visualization without the need for prepossessing to reduce the dataset. From there users would be able to manipulate and filter the dataset to reduce it to a manageable subset by their own parameters based on their domain knowledge, as described in chapter 2.2.4. Additionally, hierarchical metadata was already included in the ’omics datasets the project was going to be working with, using that structure provided a direction for methods developed by the project to follow, as the taxonomic hierarchy has a biological meaning that is often relevant for the analysis.

The usability of hierarchical visualization methods and a users ability to extract meaningful information from them may often depend on the size of the hierarchy and on patterns in the underlying data. The literature was limited on this subject matter, as gone through in chapter 2.3.7, most evaluations comparing different types of methods focused predominately on their hierarchical structures.

Therefore to determine which methods were most suitable for the project, the project conducted its own study to evaluate a set of comparable hierarchical visualization methods to determine their performance when it comes to identification of features of the underlying data, whilst additionally determining which methods were suitable for analysis of hierarchical data of different sizes. The study was presented in Macquisten et al. (2020) and expanded in Macquisten et al. (2022).

The evaluation focused on five methods for comparison: Treemap, Icicle Plot, Sunburst Chart, Circular Treemap and Bubble Treemap (figure 4.1), in an online usability evaluation. The methods were considered comparable in terms of being implicit space filling methods that are

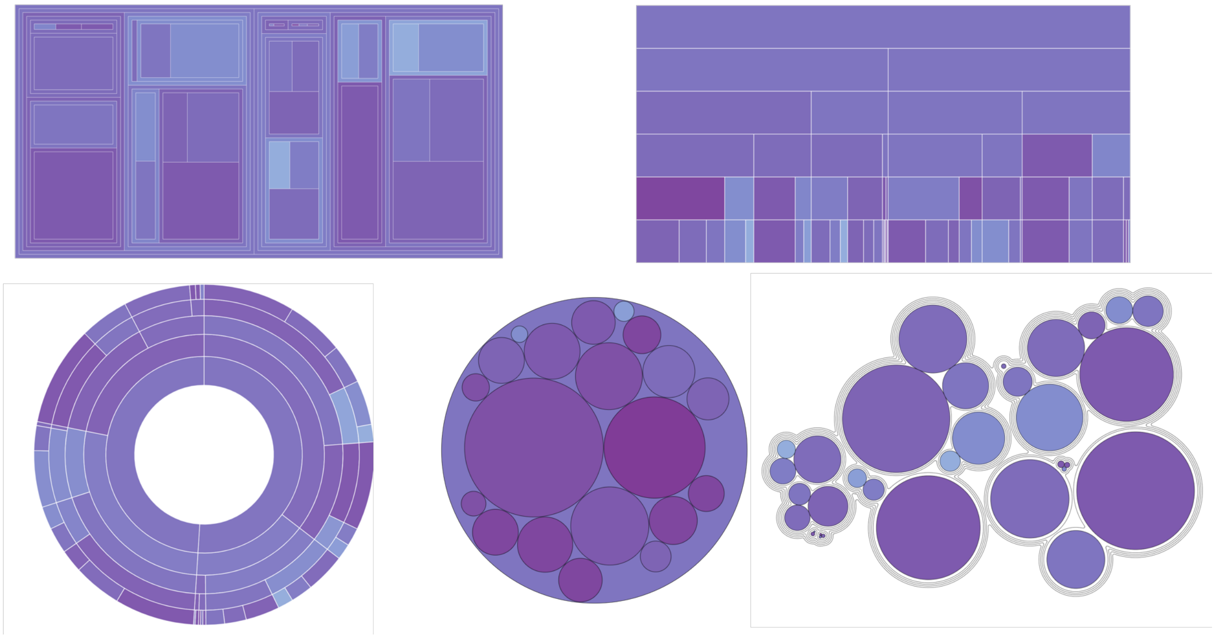


Figure 4.1 From top left to bottom right: Treemap, Icicle Plot, Sunburst Chart, Circular Treemap, Bubble Treemap

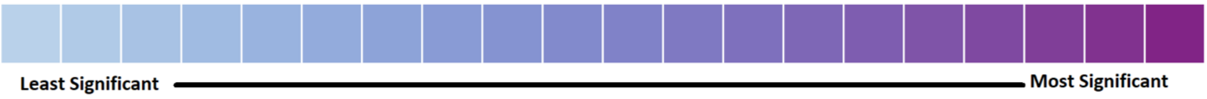


Figure 4.2 Colour scale used to show how interesting a node is, with lighter nodes being comparably less interesting than darker nodes.

able to handle relatively large hierarchies, and that are able to represent features of the data through their physical attributes.

In the study, the underlying data features were defined as a measure of abundance, which was represented by size, and a measure of interestingness (or significance), which was represented by colour. A light blue to dark purple sequential colour scheme was used, with darker colour representing a higher value of interest and with lighter values progressively showing lower values (fig. 4.2). The Colour scheme used was the '3-class BuPu' scheme from Colorbrewer2 (Brewer, 1999), which was tested for colour vision impairment using Color Oracle (Jenny and Kelso, 2006).

Each method was portrayed using two different hierarchical sizes, referred to as large and small. Large presents the full hierarchical structure the way each method traditionally does, while the small format focus on a single subsection of the hierarchical structure, only showing a parent node and its corresponding child nodes. The aim of comparing these two different sizes was to determine methods to use in the project, with the large method being used to provide overview of the entire dataset, while the small method would show only the selected data node and its immediate children nodes.

64 Synthetic datasets were generated for the experimental questions through JavaScript. Using real datasets was considered initially, but it would have taken a long time to gather enough datasets and data subsets that suited the parameters of the experiment.

Each dataset generated consisting of a random number of nodes groups, with datasets for large hierarchies consisting of six hierarchical levels with 2-5 child nodes per level. Doing so created more varied structures and prevented the generated visualizations looking all the same. While datasets generated for the small hierarchies consisted of two hierarchical levels, containing 10-20 child nodes, seeing as they wouldn't have enough levels to generate more varied structures using the previous method, it was decided to simply generate more nodes.

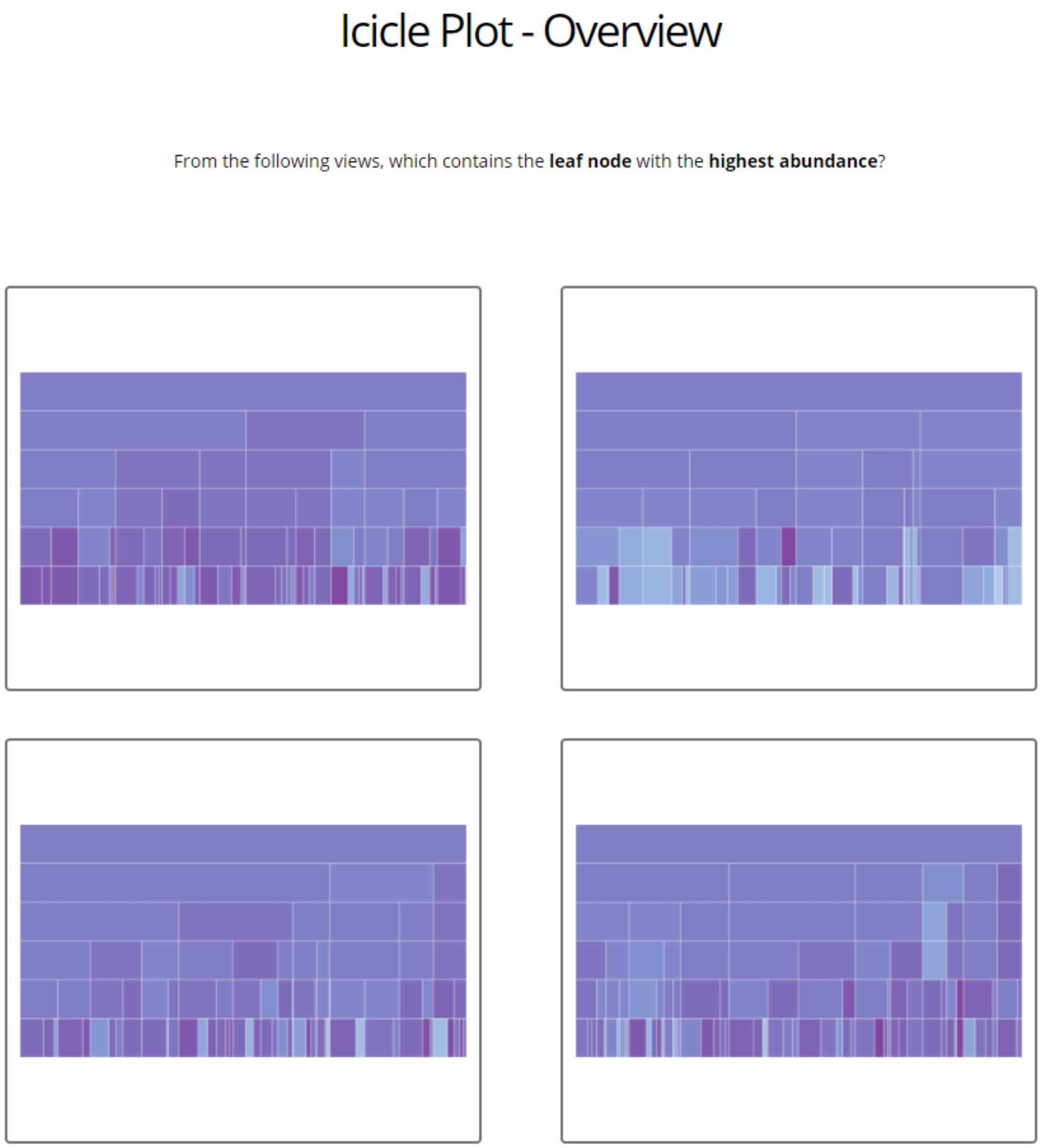
Sample group values for nodes were generated randomly as well, but were modified with overall weights for each sample group each time in order to ensure greater separation of values between sample groups and avoid sample groups becoming too homogeneous. Sample group values would be input into the previously mentioned colour interestingness metric, which used silhouette clustering (Rousseeuw, 1987) to determine if the sample group values for a node were similar, and therefore displayed lighter, or dissimilar, and displayed as a darker node, the method is explained in more detail in chapter 6.1.2. The combined sample group values were used for the size value of a node.

4.1. Tasks

Multiple choice questions were used and participant were presented with four images of a visualization method with different datasets, and asked to compare them based on the question (Figure 4.3). The following questions were asked:

- For the following views...
  1. Which contains the leaf node with the highest abundance?
  2. Which contains the node with the highest significance?
  3. On the level after the root node, which contains the node with the highest abundance?
  4. Which is the overall most significant?

While interaction may have a considerable impact on the usability of a visualization method, the focus of this study was on evaluating the most appropriate visual representation for different sized hierarchical structures. Hence, static images were used to represent the visualization methods to avoid the influence of interactive variations on performance.



Icicle Plot - Overview

From the following views, which contains the **leaf node** with the **highest abundance**?

Figure 4.3 Example of a question from the study, each question would consist of four images for participants to select.

4.2. Experimental Design and Procedure

The study was based on two hypotheses: **H1**: For large hierarchies, Sunburst would perform better due to it being space efficient and providing an easily overview-able and uncluttered representation of the hierarchy and data at the same time. **H2**: Treemap and Circle will perform best for small hierarchies, as the clutter issues they face in larger hierarchies will be mitigated, and their space efficiency can provide clear detail for a larger number of child nodes.

The experiment was designed as a within-subject study with the visualization method and display format (large/small) as the main factors. The study was conducted online, in order to standardise the range of screen sizes used, small devices, such as phones and tablets, were blocked from being used. JavaScript/D3 was used to implement the visualization methods, while the Gorilla experiment builder (www.gorilla.sc) was used to build and host the experiment (Anwyl-Irvine et al., 2020).

Participants were provided with an initial background section covering the basic concepts of hierarchical visualization, the different hierarchical size formats that would be used and an explanation of the abundance and significance/interestingness data features. Then a training section for each method was presented, questions where in the same format as the main experiment questions, however users where given the results of their training answers and were allowed to repeat each training section if they wished. Following this was the experimental phase, where each method had 16 questions, with questions divided evenly between large and small hierarchies, and each of those between abundance and significance tasks. A Latin Square procedure (Graziano and Raulin, 1993) was used to counterbalance the order users received each methods, to ensure participants received a balanced variation in ordering of methods, and to reduce the potential learning impact on the results. For each question, the accuracy of the response and the response time was recorded. Additionally, after each question, participants were asked how confident they were with their answer, using a 5 point likert scale (1 = not confident, 5 = very confident).

4.3. Evaluation Results

In total, there were 17 participants who finished the study, 8 female and 9 male, ethical approval was received prior to the study, along with participant consent. The most abundant age group was 25-34 years (47.06%), followed by 35-44 (23.53%), 18-24 (17.65%) and 45-54 (11.76%). 41.18% of participants specialised in bioscience, 29.41% specialised in Computer Science and 29.41% were in other disciplines. Participants were asked to rank their experience of 1) Data Visualization and 2) Hierarchical data, using 5 point likert scales (1 = No prior experience, 5 = Expert). 41.18% of participants listed their experience with Data Visualization as high (4 or 5) while 35.3% listed it as low (1 or 2). For hierarchical visualization, 23.5% listed their experience as high, while 64.7% listed it as low.

Retroactively running a indicative power analysis, the effect size would need to be 154.9% in order to detect with 80% power for the sample size of 17 participants used during the study. In order to have the effect size at 100% for the same detected power, the study would have needed to be conducted with 29 participants. While this would have provided more reliable statistical results, the results of this study still provide an indication of usability differences between the methods and can guide further studies in the area.

Analysis of results was divided between large and small hierarchies, with additional analysis on size and colour results for each hierarchy. The data wasn't normally distributed, so Friedman test was used for significance testing. This was followed by post-hoc tests using Wilcoxon signed rank test, with a Bonferroni correction applied resulting in a significance level set at $p < 0.005$, and pairwise comparison to identify for which combinations of visualization methods there were significant difference in the accuracy, the response time for all answers and the response time for only correct answers. The separation between response time for all answers and response time for only correct answers was made to distinguish if quick but erroneous answers may impact response times. The results of perceived confidence is reported to review if performance results agree with participant confidence. Self reported confidence is, however, an unreliable measure of performance.

4.4. Large hierarchies

Results for large hierarchies are reported in figures 4.4, 4.5 and 4.6, where medians, interquartile distances, minimum and maximum values are represented as box plots, and where red dots and white lines show the mean and standard deviation respectively. Significant differences were confirmed for overall accuracy ($\chi^2(16) = 36.078, p < 0.001$), accuracy for size ($\chi^2(16) = 42.559, p < 0.001$) and accuracy for colour ($\chi^2(16) = 22.676, p < 0.001$); as well as for overall response time of correct answers ($\chi^2(16) = 21.694, p < 0.001$) and response time of correct answers for size and colour separately ($\chi^2(16) = 18.462, p = 0.001$ and $\chi^2(16) = 26.071, p < 0.001$), though not for all response times($\chi^2(16) = 4.235, p = 375$). Post-hoc analysis, with results reported in tables 4.1 and 4.2, revealed statistically significant differences for accuracy for some visualization methods.

More specifically, the lower overall accuracy of Bubble was significant in relation to all other visualization methods. This was expected as the accuracy value for Bubble was 3, which was the lowest median accuracy value. Statistically significant differences for overall accuracy were also shown for Sunburst vs Treemap and Circle vs Treemap, where Treemap performed worse than both Sunburst and Circle. Whilst Circle performed slightly better than Sunburst with median

	Overall		Size		Colour	
Methods	Z	P	Z	P	Z	P
Icicle-Tree	-1.656	0.098	-2.345	0.019	-0.228	0.82
Sun-Tree	-3.384	**0.001**	-0.577	0.564	-3.272	**0.001**
Circle-Tree	-3.085	**0.002**	-2.951	**0.003**	-1.753	0.08
Bubble-Tree	-3.166	**0.002**	-3.247	**0.001**	-0.584	0.559
Sun-Icicle	-1.221	0.222	-2.145	0.032	-3.132	**0.002**
Circle-Icicle	-2.461	0.015	-0.794	0.427	-2.365	0.018
Bubble-Icicle	-3.086	**0.002**	-3.663	<**0.001**	-1.155	0.248
Circle-Sun	-1.383	0.167	-2.83	**0.005**	-1.645	0.1
Bubble-Sun	-3.689	<**0.001**	-3.493	<**0.001**	-3.358	**0.001**
Bubble-Circle	-3.484	<**0.001**	-3.472	**0.001**	-1.727	0.084

Table 4.1 Study 1: Accuracy results: large hierarchies. ©(2020)IEEE

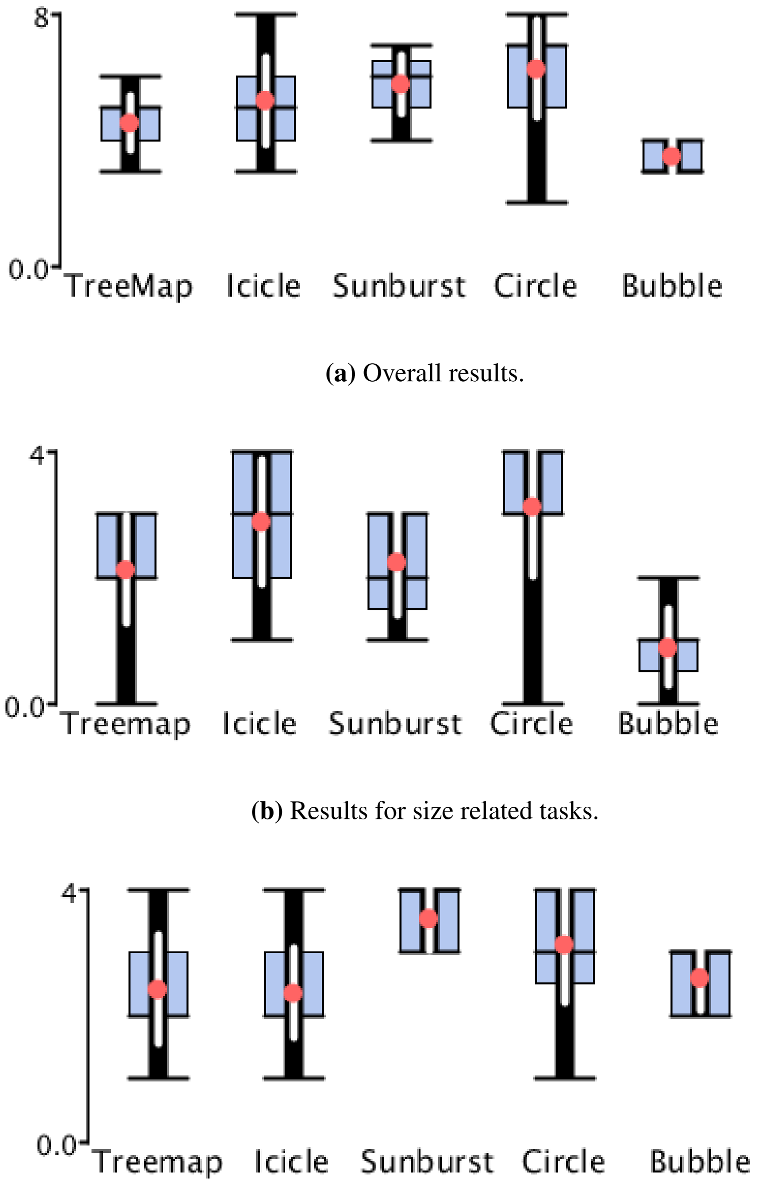
	Overall		Size		Colour	
Methods	Z	P	Z	P	Z	P
Icicle-Tree	-2.438	0.150	-1.448	0.148	-1.444	0.149
Sun-Tree	-0.118	0.906	-3.051	**0.002**	-2.249	0.025
Circle-Tree	-1.160	0.246	-1.293	0.196	-2.249	0.025
Bubble-Tree	-2.296	0.022	-2.201	0.028	-2.201	0.028
Sun-Icicle	-2.154	0.031	-3.148	**0.002**	-0.26	0.795
Circle-Icicle	-2.959	**0.003**	-1.034	0.301	-3.006	**0.003**
Bubble-Icicle	-3.432	**0.001**	-0.384	0.701	-1.16	0.246
Circle-Sun	-1.018	0.309	-2.741 0.006	-3.574	<**0.001**	
Bubble-Sun	-1.775	0.076	-1.293	0.196	-0.639	0.523
Bubble-Circle	-1.823	0.068	-1.083	0.279	-3.432	<**0.001**

Table 4.2 Study 1: Response time: correct answers, large hierarchies. ©(2020)IEEE

values of 7 and 6 respectively, there was no significant difference between them. The differences between Circle and Icicle, and Sunburst and Icicle were also not significant.

When separating size and colour related tasks, accuracy results for size where at large the same, but with slightly worse performance for Sunburst, with Circle performing significantly better than Sunburst, while there were no significant difference with Treemap. For colour related tasks, on the other hand, the accuracy of Sunburst was higher with significantly better accuracy than Treemap, Icicle and Bubble. This in part supports **H1**, although Icicle and Circle appear to perform roughly as well as Sunburst, and better for size related questions. The post-hoc tests of the overall response time for correct answers revealed significance between Circle vs Icicle and Bubble vs Icicle, with worse performance from Icicle, but no other significant differences.

For size related questions, Sunburst performed significantly worse than Treemap and Icicle While Circle performed significantly worse than Icicle; Sunburst and Bubble for colour related questions. The confidence reported by participants was fairly similar for all visualization methods, with slightly higher confidence reported for Icicle.



(a) Overall results.

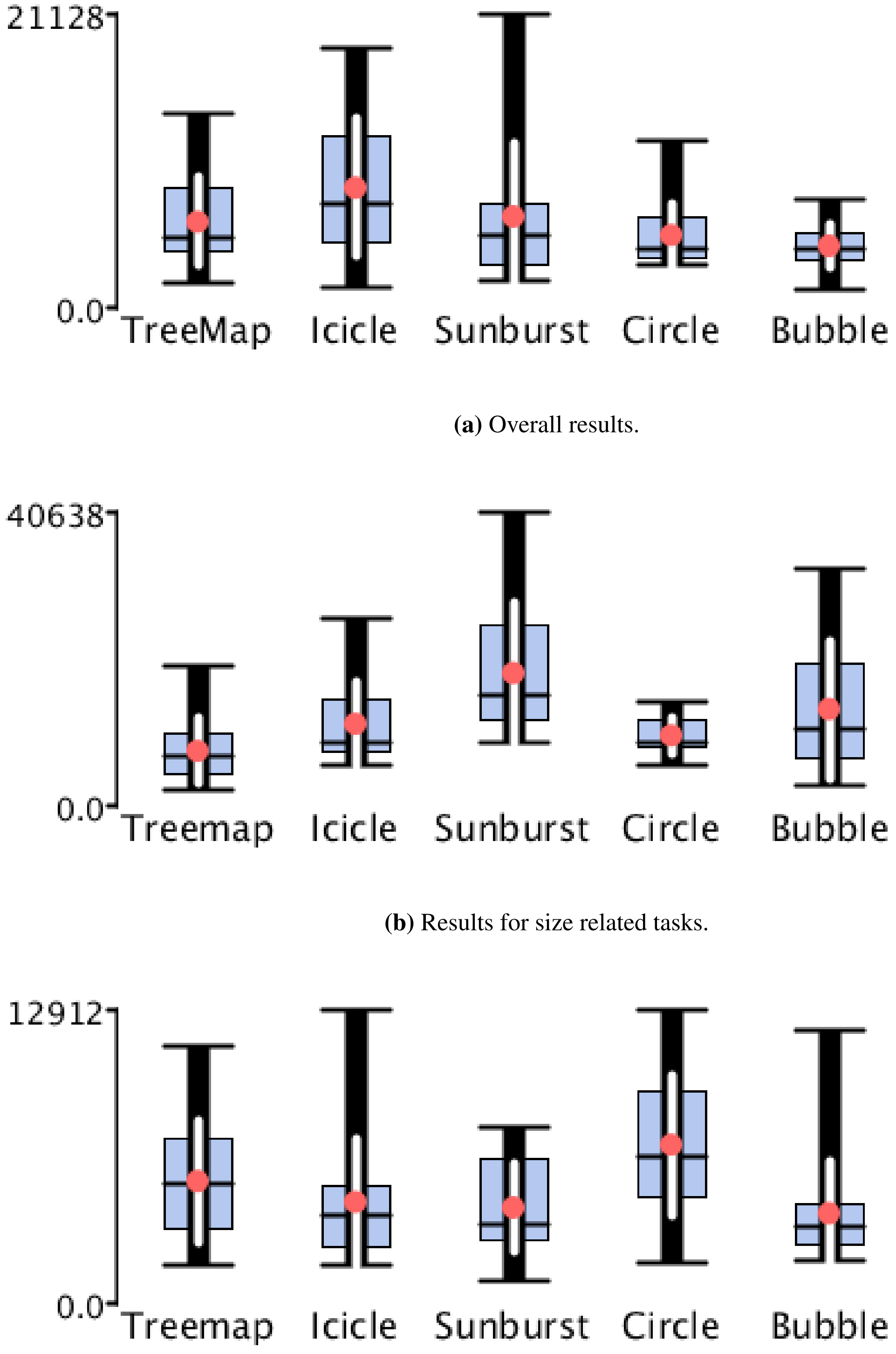
(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.4 Accuracy results: large hierarchies. ©(2020)IEEE

4.5. Small hierarchies

Results for small hierarchies are reported in figures 4.7, 4.8 and 4.9. Significant differences were confirmed for overall accuracy ($\chi^2(16) = 40.700, p < 0.001$) and accuracy for size questions ($\chi^2(16) = 48.121, p < 0.001$); for overall response time ($\chi^2(16) = 14.824, p = 0.005$), response time for size ($\chi^2(16) = 21.271, p < 0.001$) and for colour questions ($\chi^2(16) = 17.318, p = 0.002$); as well as for overall response time of correct answers ($\chi^2(16) = 36.329, p < 0.001$) and response time for correct answers for size and colour questions ($\chi^2(16) = 30.187, p < 0.001$ and $\chi^2(16) = 14.588, p = 0.006$).



(a) Overall results.

(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.5 Response time (ms): correct answers, large hierarchies. ©(2020)IEEE

Post-hoc analysis, with results reported in the tables 4.3, 4.4 and 4.5, revealed that for overall accuracy the Treemap, Icicle and Circle methods performed significantly better than Sunburst, with Treemap and Icicle performing better than Bubble, which mainly confirms **H2**. No other significant results were detected for overall accuracy. The results were comparable between overall accuracy and size related questions.

	Overall		Size	
Methods	Z	P	Z	P
Icicle-Tree	-0.461	0.623	-0.061	0.951
Sun-Tree	-3.564	<**0.001**	-3.619	<**0.001**
Circle-Tree	-1.008	0.313	-1.511	0.131
Bubble-Tree	-3.568	<**0.001**	-3.588	<**0.001**
Sun-Icicle	-3.561	<**0.001**	-3.695	<**0.001**
Circle-Icicle	-0.690	0.490	-1.998	0.046
Bubble-Icicle	-3.248	**0.001**	-3.404	**0.001**
Circle-Sun	-3.170	**0.002**	-3.36	**0.001**
Bubble-Sun	-1.406	0.160	-2.804	**0.005**
Bubble-Circle	-2.729	.006	-2.612	0.009

Table 4.3 Study 1: Accuracy: small hierarchies. ©(2020)IEEE

	Overall		Size		Colour	
Methods	Z	P	Z	P	Z	P
Icicle-Tree	-0.450	0.653	-0.686	0.492	-2.249	0.025
Sun-Tree	-2.722	0.006	-3.413	**0.001**	-1.538	0.124
Circle-Tree	-2.580	0.010	-0.052	0.959	-3.195	**0.001**
Bubble-Tree	-3.006	**0.003**	-2.215	0.027	-2.959	**0.003**
Sun-Icicle	-2.817	0.005	-3.464	**0.001**	-0.45	0.653
Circle-Icicle	-2.959	**0.003**	-0.207	0.836	-1.254	0.21
Bubble-Icicle	-3.527	<**0.001**	-1.817	0.069	-0.781	0.435
Circle-Sun	-0.639	0.523	-3.516	<**0.001**	-2.675	0.007
Bubble-Sun	-2.722	0.006	-2.897	**0.004**	-1.065	0.287
Bubble-Circle	-2.675	0.007	-2.045	0.041	-0.828	0.407

Table 4.4 Study 1: Response time: correct answers, small hierarchies. ©(2020)IEEE

Post-hoc analysis on the overall correct response time found significant differences for Bubble vs Treemap, Circle vs Icicle and Bubble vs Icicle. With shortest time for Bubble and longest time for Treemap and Icicle. For size related questions Sunburst performed significantly worse than all other methods. Treemap performed significantly worse than Circle and Bubble for colour questions.

Conversely, post-hoc analysis for overall response time for all answers showed Sunburst as requiring significantly longer to use compared to Icicle, with no other significant results detected. For size related questions, Sunburst performed significantly worse than Treemap and Icicle, while Circle performed significantly better than Treemap and Sunburst for colour related questions. Participants were confident using most methods but reported a slightly stronger confidence using Bubble.

	Overall		Size		Colour	
Methods	Z	P	Z	P	Z	P
Icicle-Tree	-1.160	0.246	-0.639	0.523	-1.728	0.084
Sun-Tree	-1.917	0.005	-3.006	**0.003**	-0.923	0.356
Circle-Tree	-0.781	0.435	-1.065	0.287	-3.243	**0.001**
Bubble-Tree	-1.302	0.193	-0.26	0.795	-1.965	0.049
Sun-Icicle	-3.290	**0.001**	-3.621	<**0.001**	-0.355	0.723
Circle-Icicle	-0.166	0.868	-1.728	0.084	-2.391	0.017
Bubble-Icicle	-0.118	0.906	-0.544	0.586	-0.402	0.687
Circle-Sun	-2.817	0.005	-2.059	0.039	-3.574	<**0.001**
Bubble-Sun	-2.391	0.17	-2.391	0.017	-0.781	0.435
Bubble-Circle	-0.260	0.765	-0.828	0.407	-2.059	0.039

Table 4.5 Study 1: Response time: all answers, small hierarchies. ©(2020)IEEE

4.6. Summary

This study contributes to visualization literature by being the first evaluation comparing the five hierarchical visualization methods: Treemap, Icicle Plot, Sunburst Chart, Circular Treemap and Bubble Treemap, in their ability to represent underlying data features for large and small hierarchies. Previous evaluations have compared subsets of these methods and have mainly focused on tasks directly related to the hierarchical structure, while this study investigated how well the methods represent underlying data through visual features.

The results presented in this study determine that for large hierarchies, Sunburst Chart and Circular Treemap had the overall best results, with Sunburst Chart performing better for colour comparison tasks, while Icicle Plot and Circular Treemap performed better for size comparison tasks. While for small hierarchies, the results suggest that Treemap, Icicle plot and Circular Treemap are the best alternatives. The results put in context of previous research clearly confirms that tasks and data structure has to be taken into consideration when selecting appropriate hierarchical visualization method.

For application in the project, the intention was to use two different hierarchical methods in conjunction with one another, a large structure to show the full dataset and provide overview, and a small structure to show the currently selected subset in detail. Sunburst was selected for the large structure, due to its performance overall and for colour comparison tasks, while Treemap was selected for the small structures. This will be discussed in more detail in Chapter 4.

This study was also the first time the project used Gorilla to build online experiments. As such it did take some time to understand how to utilise Gorilla and there was some challenges to get components working, but the support documentation and example projects were extensive enough to solve these issues, and the range of functionality proved sufficient for the tasks at hand.

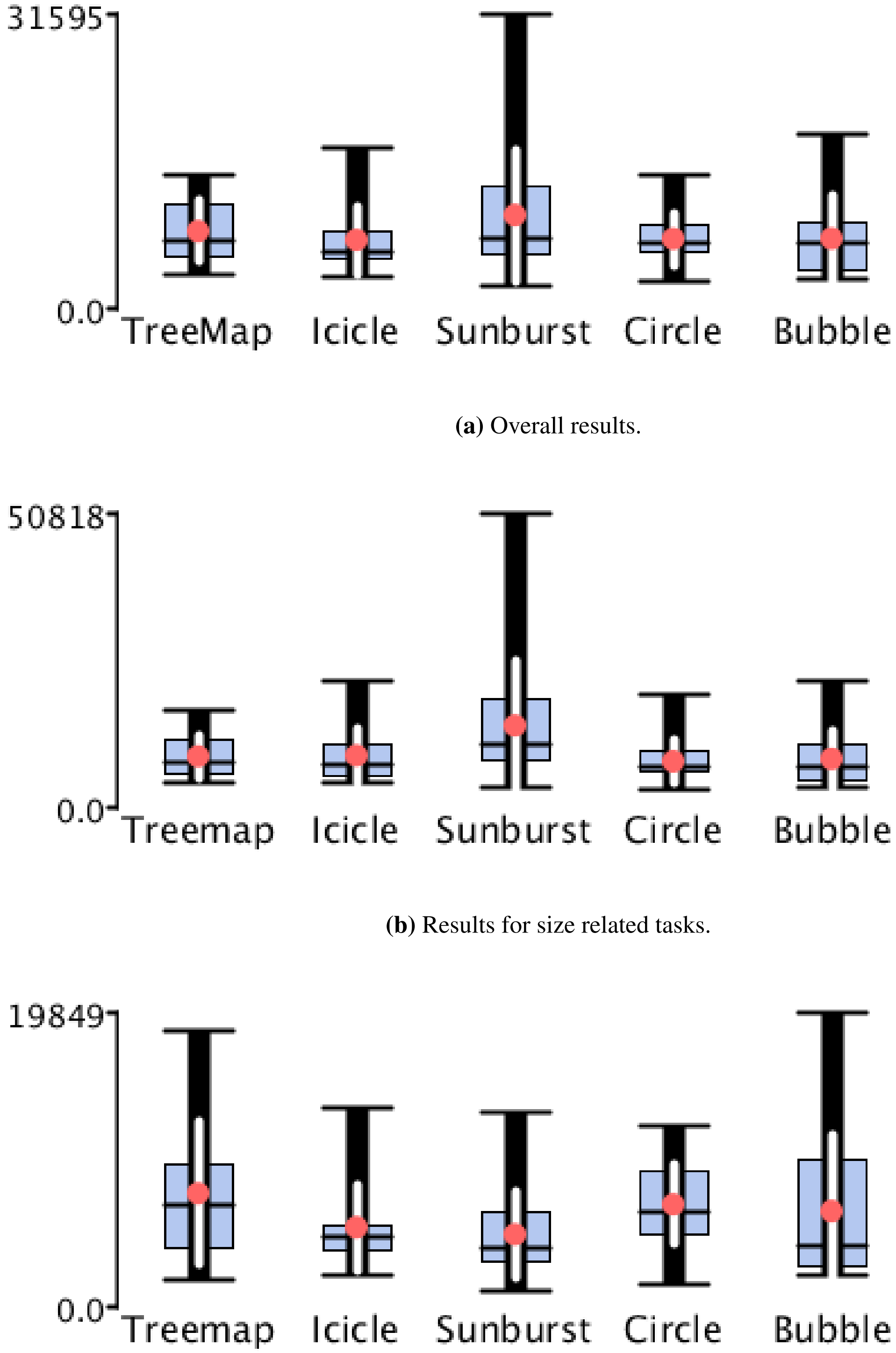
Gorilla uses a flow chart editor to structure the study, with different nodes for tasks, questionnaires, and to manage progression through the study. Among these, Checkpoint nodes track in real time how far a participant is through your study and how far they got if they stopped, randomiser nodes are used to direct users down different paths of tasks, while switch nodes allow participants to switch between different nodes, say a training task and a information page.

Each individual task uses a spreadsheet to structure the tasks and the data to be used in component questions. Since all the questions were effectively the same with different figures and text, most of the work went into getting a single question working and then replicating it with the different question data.

The resultant flow chart created for this experiment was complex, as there was many different tasks paths so that participants would get questions in different orders, and each of paths was made up of many nodes, as it felt more intuitive to break tasks down into smaller nodes. The complexity of the flowchart had no impact on the participant, as they never interact with it, but it did make it more time consuming to build and test. Regardless, the flowchart designer was intuitive to use and provided enough functionality to carry out the experiment.

Overall, the functionality within the tool was varied and powerful enough to make it a consideration to use it again for future studies, taking the experiences from this time to optimise the process for next time and push the functionality of Gorilla to do what the project needed it to do.

Additionally, it was decided that running the study online rather than in person would be more optimal. An in person study would have allowed for greater feedback, but would have left the study without the convenience of Gorilla recording every user input, along with how long participants spent on every questions and between every component of the study. If the study was more technically intense, such as if the participant was given different tools to use to do tasks, then it would have been better to do in person, but that would have required more preparation to get the tools working.

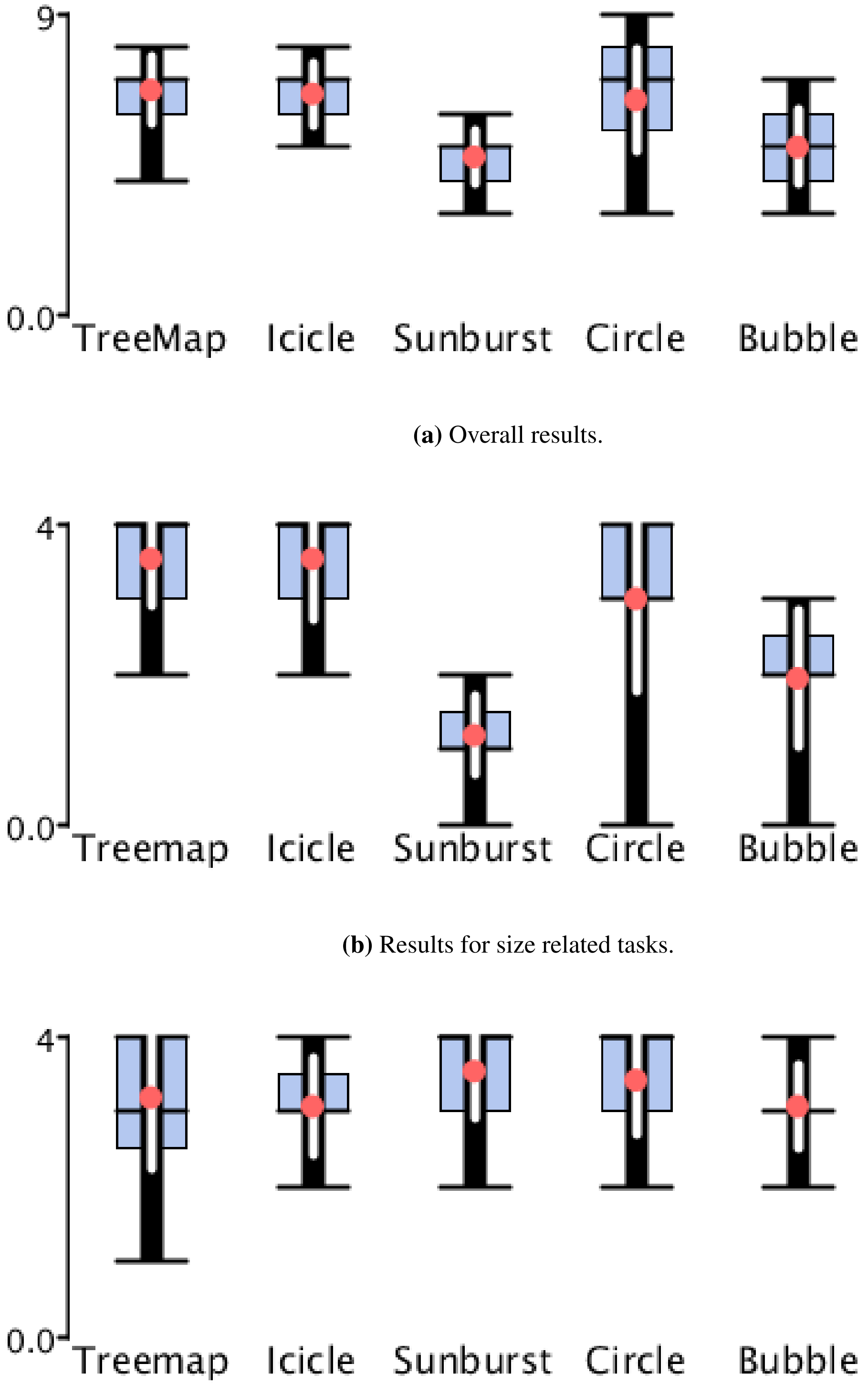


(a) Overall results.

(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.6 Response time (ms): all answers, large hierarchies. ©(2020)IEEE

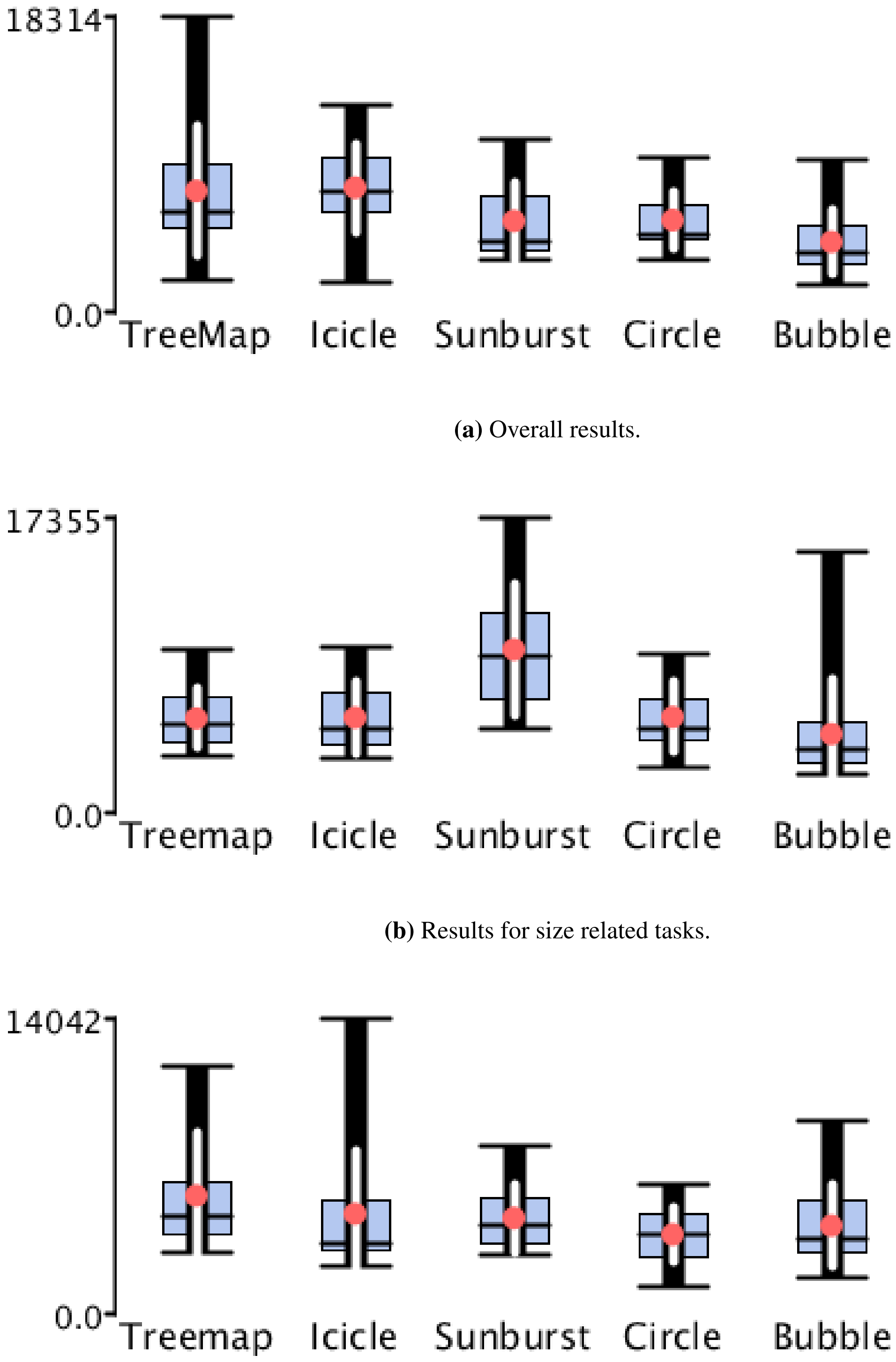


(a) Overall results.

(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.7 Accuracy: small hierarchies. ©(2020)IEEE

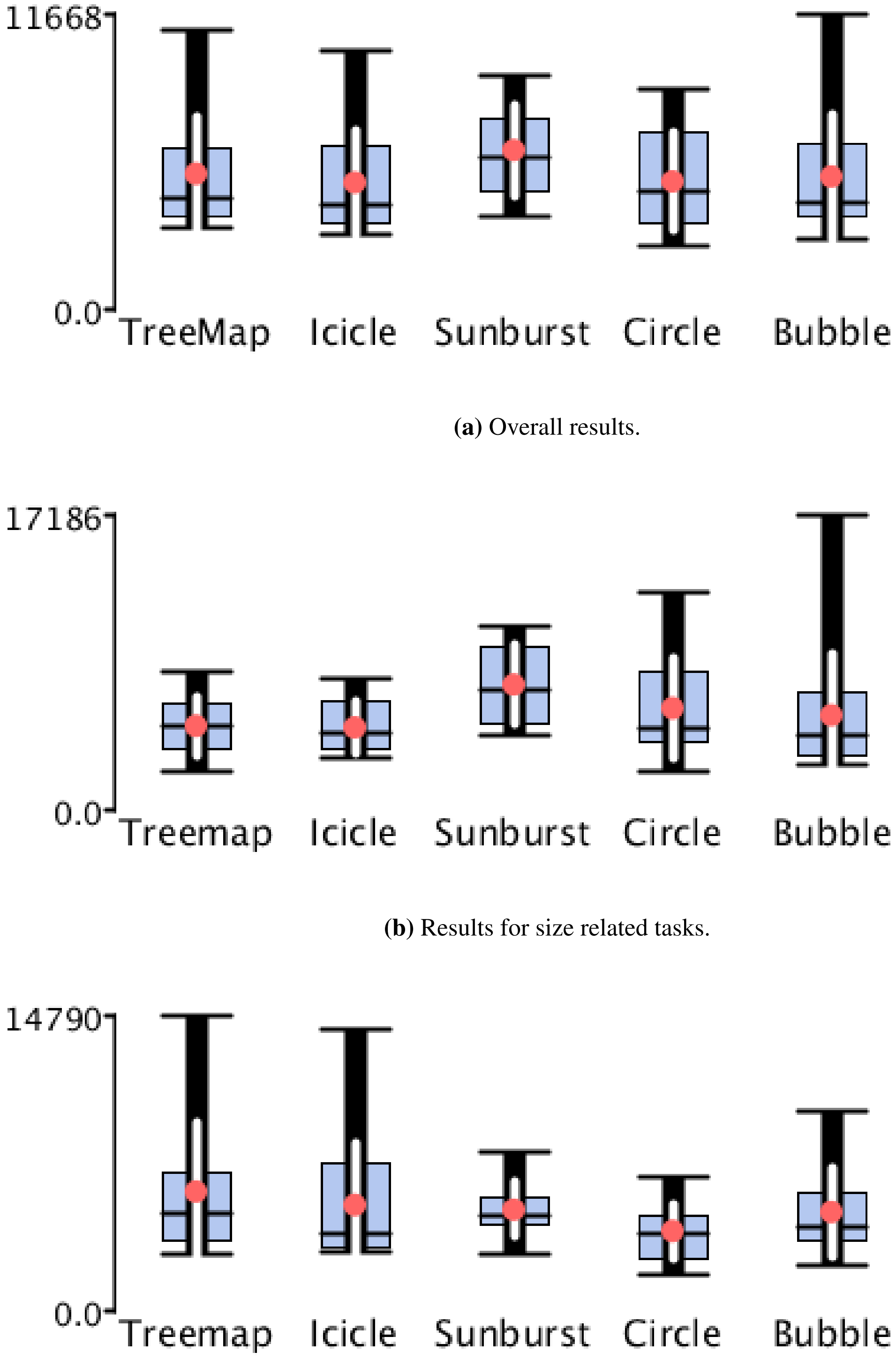


(a) Overall results.

(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.8 Response time (ms): correct answers, small hierarchies. ©(2020)IEEE



(a) Overall results.

(b) Results for size related tasks.

(c) Results for colour related tasks.

Figure 4.9 Response time (ms): all answers, small hierarchies. ©(2020)IEEE

Chapter 5. Representation of multivariate data features in aggregated data

This chapter covers the research into using glyphs to represent multivariate aggregated data, alongside two studies conducted to evaluate the developed method.

One of the core reasons, that implicit space filling hierarchical methods were selected to display the dataset was their potential to show additional information within nodes.

This would be problematic still for showing the full dataset in a single view, since only the higher levels would have space to show any information at all, but with the Treemap only showing a single level of the structure based on what node is selected in the Treemap, space is at less of a premium.

With the main interest in analysis of the data being the comparison of different sample groups, there needed to be a way to show the differences between the aggregated sample group values for every node, yet it needed to be compact enough to fit into the Treemap.

5.1. Glyph Development

Early versions of the method made use of a simple bar glyph, it showed the aggregated abundance values for each sample group for that node. While this showed off the general idea representing multivariate aggregated data and was useful to compare relative abundance, it didn't provide much in the way of statistical analysis by itself.

It was decided it would be more useful to show a distribution of the underlying data than just abundance values. Deciding what to use was an issue, as since there was often more than two sample groups, and the literature is limited on the comparison of more than two distributions Blumenschein et al. (2020).

An overlaid histogram (figure 5.1) was considered next, showing the frequency values for each sample group, overlaid on top of each other. The data was log-normalised, as otherwise the range of values for individual entities would be too vast to show within a single chart.

Ultimately the methods and variants of the method ended up being too difficult to make meaningful observations from, sample groups with similar values would obscure each other and

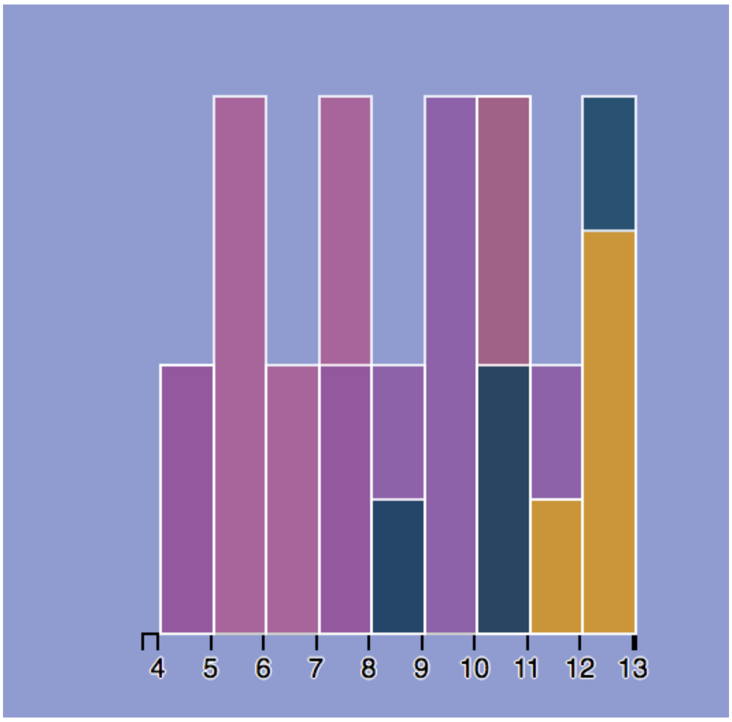


Figure 5.1 Overlaid Histogram, each sample groups bar occupies the same location for each value, with lower abundant groups being pulled to the front.

give the illusion that one didn't have a value at all, an example of this is visible in the leftmost bar in Figure 5.1 which displays the pink and purple groups overlaid with the same frequency value. And while it was intend for whatever method selected, that user interaction would be able to fade out or filter out sample groups to focus on an individual group or set of groups, the inability to allow the user to grasp any differences at a glance made the method useless. Expanding the histogram out so each sample had its own space for each frequency value was not space efficient enough for use within the Treemap.

Box plots were considered next (figure 5.2), they were useful for comparing the aggregated statistical properties of each sample group. The concern was space, comparing a two or three different sample groups within a Treemap node isn't too much of an issue, but if a user needs to compare more then either the box plots need to be made smaller, or overlaid to fit into smaller nodes.

Overlaid Density plots (figure 5.3) were explored next, it was less of an issue having the plots overlaid as they were less likely to completely obscure each other, the plots were made transparent to allow each plot to be seen. However, space limitations considerations made having an axis with lots of small text hard to read, and having just the overlaid distributions by themselves without any additional information wasn't perceived as useful.

Violin plots (Hintze and Nelson, 1998) were considered, since they are effectively a box plot within a mirrored density distribution plot. But that brings up the same issue with having

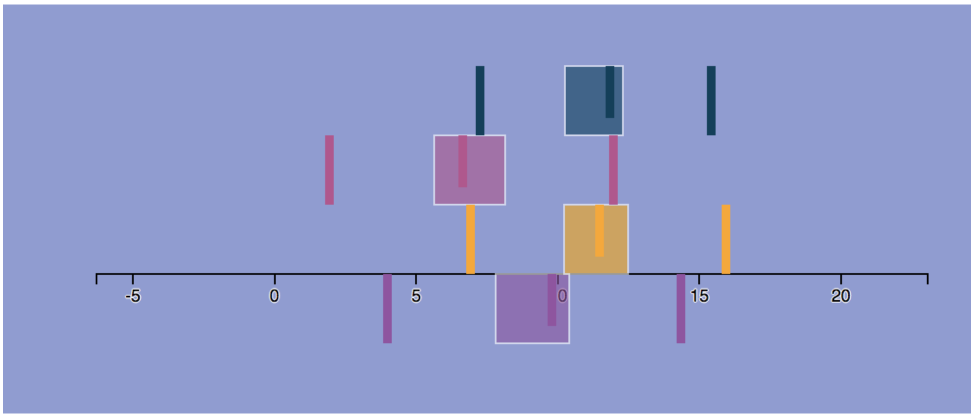


Figure 5.2 Box plots, sharing the same axis with the horizontal edges of the box representing the quartiles, the line inside the mean and the lines either side the minimum and maximum.

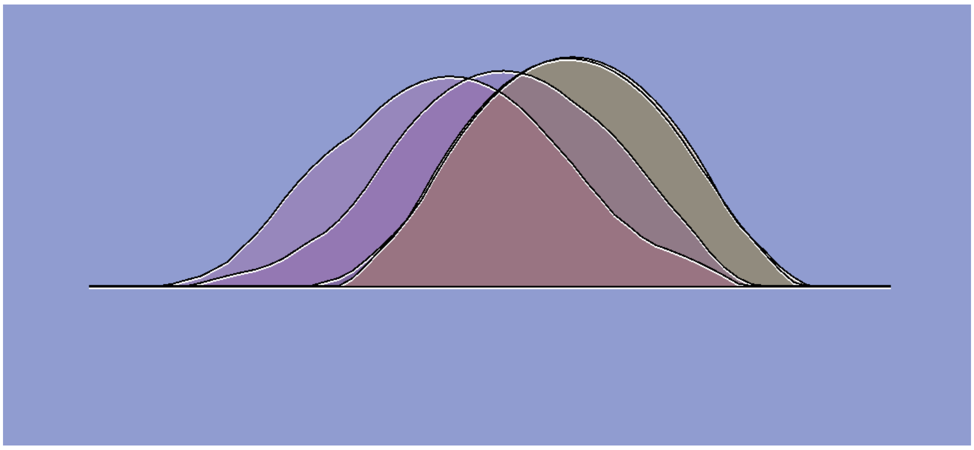


Figure 5.3 Overlaid Density plots, with each plot assigned to a different sample group.

overlaid box plots. They would need to be separated in order to fit within a Treemap node, which would be an issue in smaller nodes or if there are too many sample groups.

Following from this, the project explored adding additional information into the overlaid density distributions, using additional glyphs to show sample median and interquartile range values. This brought the same issue as with the histogram, where glyphs could overlay each other and obscure values. The method was modified so each sample group had its value glyphs on its own line, each were semi-overlaid, so that they would be reasonably compact, but differences between sample groups could be visible.

This effectively turned the visualization into a set of overlaid density distribution plots, sharing the same axis of a series of semi-overlaid abstracted box plots, as such the method was named density box plots. Two studies were conducted to evaluate the usability of the density box plots, described in detail in the following sections.

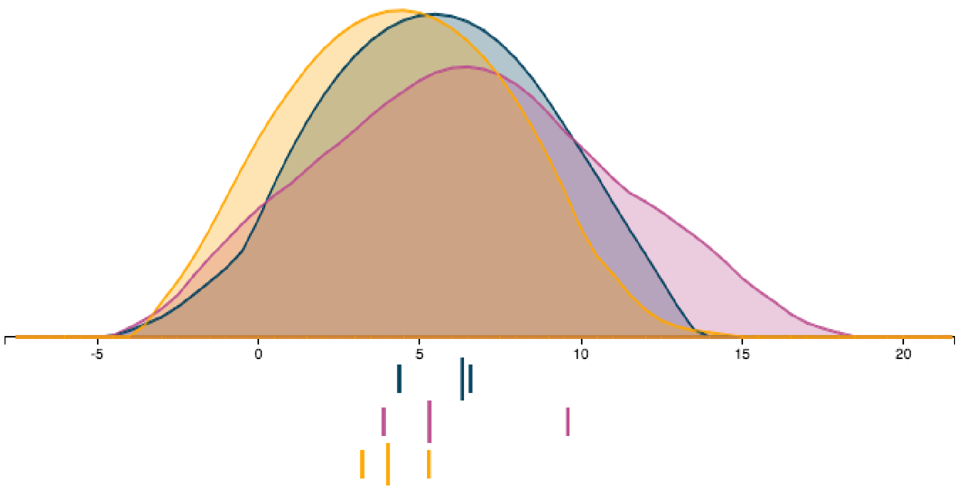


Figure 5.4 The simple method displays only the mean as a vertical line and quarterlies as smaller vertical lines for each of its box plot components in order to avoid overloading the user with information.

5.2. Study 2: Density Box Plot comparisons

This first experiment focused on the determining how to best show the Density Box Plot. The initial method had the box plot component show only the interquartile range, but the project wanted to consider a more complex version that also included the minimum and maximum values, the mean, and the skew between median and mean as well.

As such this study compared these two methods, labelled as the simple method (Figure 5.4) and the complex method (Figure 5.5). The study compared how these two methods performed in comparing different sample group statistical attributes.

64 Synthetic datasets were generated, these were used for both methods, but randomised between what questions and images they were used for to avoid users getting the same questions for both methods, split equally between three and five sample groups. As with the previous experiment, using real data was considered time consuming for study preparation and too little benefit. Instead, for each sample group had random values generated within its own set of ranges, to ensure each sample group would not generate similar values to each other.

As for the data generated for the Hierarchical study, generating random data wasn't enough and extra steps were required to make the data more realistic and avoid data points being homogeneous. Each sample group had two sets of minimum and maximum values selected manually prior to running the generation code. The first set was intended to generate a broad selection of random values, for example between 100 and 900, while the second set would create a set of random values within a smaller range, for example between 100 and 200.

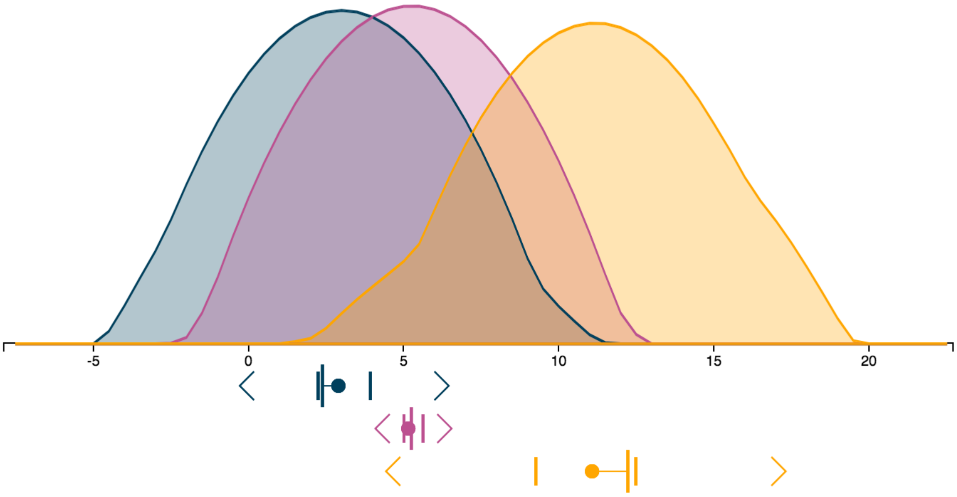
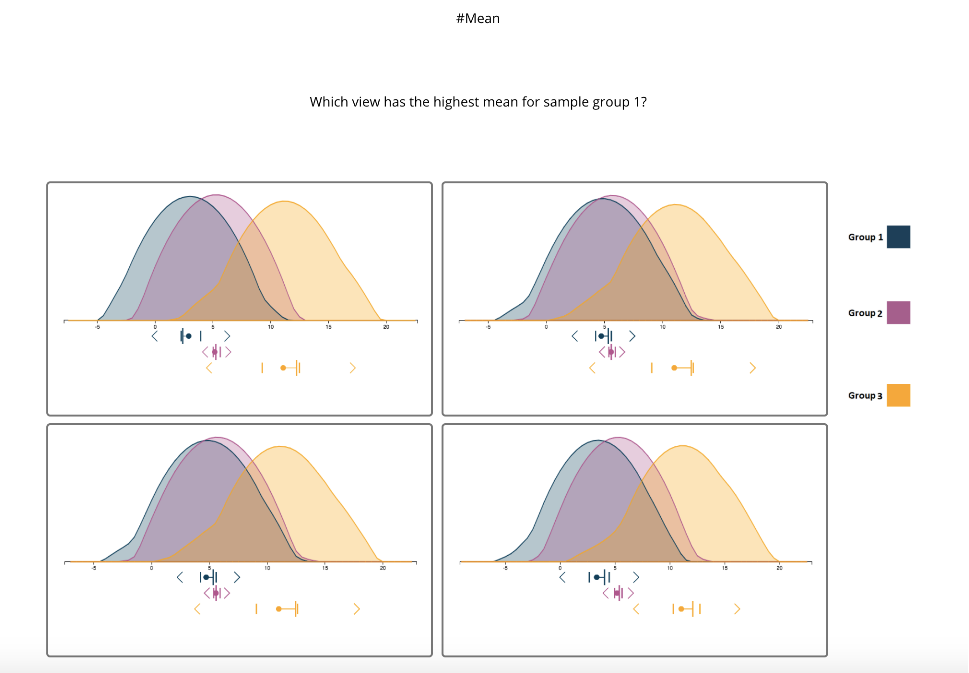


Figure 5.5 The complex method presents the same information as the simple method, but in addition, the median is displayed as a circle, the skew between mean and median shown as a horizontal line, and the arrows as the minimum and maximum values. The density distribution components remain unchanged for both methods.

This would allow for each sample group to be weighted towards different average values that might not have materialised as easily by just providing a single range, making it easier to show that the sample groups showed distinct values like they would in any real dataset. Following this the data was log normalised to be used by the kernel generation method.

The nature of the Density box plot requires a degree of interaction within the experiment, in order to filter out sample groups to focus on subsets of the image or whenever there is clutter. Whilst getting a the full interactive JavaScript tool working in some form in Gorilla would have been feasible, it likely would have been too complicated for the system to handle and risk slowing or crashing the survey. Instead, additional images were generated for each Density Box Plot image, with all bar one of the sample groups faded out. Hovering over any of the sample groups in the legend would switch out the Density Box Plot images for ones where the other sample groups were faded out (Figure 5.7), appearing visually identically to the hovering interaction that would be available in the visualization tool, allowing the participant to focus on a single sample group at a time. In total 320 Density Box Plot images were generated.

One issue with the Kernel Density components within the question images is that a number of them do not match up with the box plots, having minimum values going below 0 despite the box plot showing no values going that low. This wasn't as much as an issue for the study, as there was less interest in what the data presented and more in the comparison of different datasets, but it is a limitation of the code. The issue is caused when there are a number of sample values that are close to zero, resulting in the generated curve crossing over into a negative value due to the magnitude of its smoothing coefficient (Węglarczyk, 2018), this is detailed further in chapter 7.1.4.



#Mean

Which view has the highest mean for sample group 1?

Figure 5.6 Example question from the first glyph comparison study, consisting of four images of different datasets and a legend of the different sample groups at the side.

5.2.1. Tasks

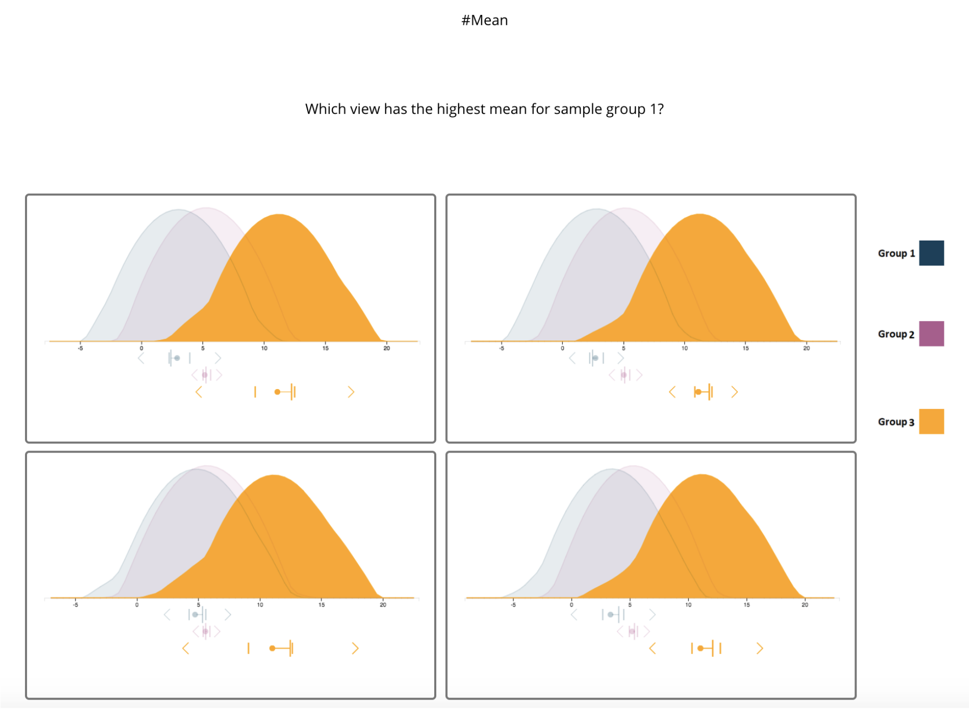
Each question consisted of 4 images, each containing a Density box plot with a different dataset. At the side of the Density Box Plots was a legend with each sample group, labelled group 1 to group N, where N equals the number of displayed sample groups (3 or 5 in this study), with a box coloured the same as its sample component in the images (Figure 5.6). The main interest was in evaluating the methods for how useful they were in presenting statistical information to the user.

The following questions were asked, with *X* being some sample group:

1. Which view has the highest/lowest mean for sample group *X*
2. Which view has the highest/lowest median for sample group *X*
3. Which view has the largest/smallest interquartile range for sample group *X*
4. Which view has the largest/smallest difference between median and mean for sample group *X*

5.2.2. Experiment Design and Procedure

The main hypothesis **H1:** The complex method would perform better for skew questions, as users only need to compare box plot components rather than the full plot.



#Mean

Which view has the highest mean for sample group 1?

Figure 5.7 Question shown in figure 5.6, with sample group 3 highlighted in the legend, updating the question images to fade out the other sample groups

The experiment was designed as a within-subject study with the visualization method as the main factor. The study was conducted online through Gorilla again, with restrictions to standardise the range of screen sizes. Participant recruitment for this study was done through the recruitment tool, Prolific (www.prolific.co), in order to gather a greater number of participants with less time spent gathering them. Candidates were pre-screened to at least have a undergraduate degree.

Using Gorilla was significantly easier this time, as I still had the study from last time to pull parts from. The resultant flowchart was much simpler and streamlined this time round, with most of the complexity moved instead into individual task design allowing for less nodes in the flowchart.

Crowd-sourcing the participants through Prolific left a risk that they would rush through the experiment only to get the reward. It was expected the study would take around 30 minutes to complete, and while it was expected for some people to finish faster or slower than that time, there was a number of participants who finished significantly quicker than would be expected, with the worst offender finishing in 2:18. To mitigate this, more participants were recruited than the experiment was aiming for, and answers were manually evaluated of anyone who finished under 15 minutes. While finishing below this time didn't necessarily mean that the participant had rushed the experiment, it was a good starting point for checking in more detail.

Since Gorilla not only records how long a participants takes to complete the experiment, but records every response they make and how long it takes between responses, its easy to see if a participant rushed through a section. This also included participants who performed fine to start with, but at some point decided to rush through the rest of the question. Be it that they lost interest part way through or were deliberately trying to increase their time to make it look like they were actually doing the experiment, Gorilla provides enough time information to determine if any results were invalid, for example if a user rushed through every question, but then waited 10 minutes at the end to make the result look good on prolific, that would be easy to spot. Additionally, its easy to see if a participant just answered the same question option repeatably, for example selecting the first image each time.

Most of these rushers were still payed, as Prolific has strict requirements rejecting submissions, participants could only be rejected for finishing too quickly if they finish three standard deviations below the mean time. However this does not require the rushers results to be used in any shape or form if deemed unacceptable to the study. While the rushers were removed early on in the data analysis, when checking their actual results they did perform worse than participants who did not rush, there is a chance that this might have impacted the results by having different methods perform better in certain ways if the rushers randomly performed better in some parts.

Its worth noting that during the hierarchical experiment from chapter 4, there wasn't the issue with rushing. That experiment was done of Gorilla only and invites were sent manually to participants rather than having prolific handle recruitment. While there was certainly more quality control needed for this study in comparison to the hierarchical study, the trade off was that the study was completed in under two hours, while the hierarchical study took a month of constantly trying to get more participants.

At the start of the experiment, an introduction section covered the overall method and the two different variations being tested, alongside an explanation of sample groups and the structure of the questions and the experiment. Each set of questions for a method was preceded with a training section. This section first provided a user with an explanation of the method and a interaction demo, showing the user how to fade out sample groups. The training section then presented 4 questions in the same format as the main section, informing the user whenever they got a question correct or incorrect, with a tally at the end showing how well they did, users could then repeat the training questions if they wished.

After the training phase for a method, participants were presented with the main questions. Each method had 16 questions, with the questions divided equally between datasets with three samples groups and five samples groups, and divided again between questions asking for the highest values and the lowest value. For each question, the accuracy of the response and the response time was recorded. Additionally, after each question, participants were asked how confident they were with their answer, using a 5 point likert scale (1 = Uncertain, 5 = Certain). With only two methods, counterbalancing was done just by switching which method a participant did each time.

	Overall	
Methods	Z	P
SimMean-CompMean	-0.363	0.717
SimMedian-CompMedian	-4.119	**<0.001**
SimIQ-CompIQ	<0.001	1.000
SimSkew-CompSkew	-3.444	**0.001**
SimTotal-CompTotal	-1.279	0.201

Table 5.1 Study2: Multivariate Accuracy

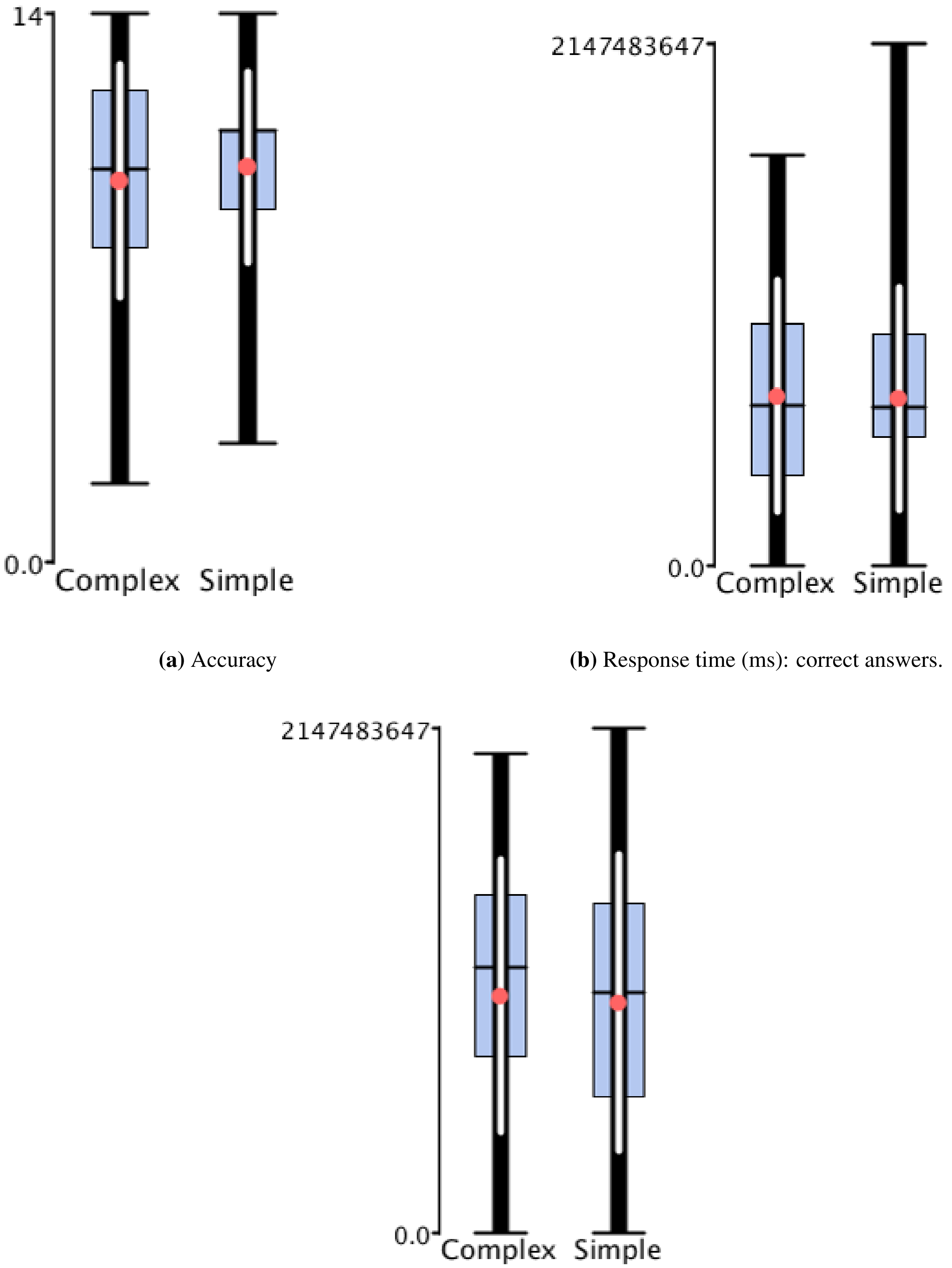
5.2.3. Evaluation Results

There were 28 participants who finished the study, 15 female and 13 male, ethical approval and participant consent was received prior to the study. The most abundant age group was 25-29 (32.14%), followed by 20-24 (28.57%), 30-34(25%), 35-39 (3.57%), 40-44 (3.57%), 45-49 (3.57%) and 60+ (3.57%). 17.86% of participants specialised in Computer Science, 10.7% in Bio Science, 3.57% in Statistics and 67.86% in other fields, such as Mathematics and Finance. The large number of other fields in comparison to the previous study is a consequence of running the study through prolific rather than through direct invitation. Participants were asked to rank their experience of 1) Data Visualization and 2) Statistical Visualization, using 5 point likert scales (1 = No prior experience, 5 =Expert). For experience with Data Visualization, 10.7% of participants put down their experience as 1, 42.86% as 2, 35.7% as 3 and 10.7% as 4, while for experience with Statistical visualizations, 21% put it down as 1, 35.7% as 2, 35.7% as 3 and 7.14% as 4.

Analysis of results was done in the same manner as in the Hierarchical visualization study, with the Friedman test being using for significance testing. This was followed by post-hoc tests using Wilcoxon signed rank test, with a Bonferroni correction applied resulting in a significance level set at p<0.01, and pairwise comparisons for each type of questions for the two methods in regards to accuracy. Results for this study are reported in Figure 5.8 where medians, interquartile distances, minimum and maximum values are represented as box plots, and where red dots and white lines show the mean and standard deviation respectively.

Retroactively running a indicative power analysis, the effect size would need to be 94% in order to detect with 80% power for the sample size of 28 participants used during the study.

Both methods performed comparably well overall, with 272/448 correct answers from all participants combined for the complex method and 282/448 for the simple. As such, the Friedman test didn't return a significant result when comparing overall accuracy ($\chi^2(28) = 0.291, p < 0.532$), average response time ($\chi^2(28) = 0.143, p < 0.705$) or average response time for correct questions ($\chi^2(28) => 0.001, p < 1.000$). When broken down to only consider correct answers for sets of similar questions, statistical results were returned for Median questions ($\chi^2(28) = 18.615 p > 0.001$) and Skew Questions



(a) Accuracy

(b) Response time (ms): correct answers.

(c) Response time (ms): all answers.

Figure 5.8 Results for Study2: Density Box Plot comparisons

$(\chi^2(28) = 12.565p > 0.001)$. As shown in figure 5.1, when followed up with Wilcoxon testing, the results again were significant for the Median in favour of the simple method and Skew results in favour of the complex method.

The simple method performed better with questions regarding to Median, we find this result strange as the symbol used to show the median is the same in both methods, its possible the addition of the glyph representing mean confused participants, but for questions regarding mean, participants performed comparably well for both methods.

For questions regarding to skew, the complex method performed better, meeting **H1**. This was to be expected, since the complex method has a direct way to compare skew. While for the simple method, the participant would need to compare the mean from the Density distribution component and the Median for the Box plot components. The confidence reported by participants was reported higher for the complex method.

5.2.4. Summary

The study compared two variations of the method Density box plot, to determine how much complexity in details would be optimal to provide.

While the Simple method performed better for the median questions, and the Complex method performed better for Skew questions, there wasn't any significant difference between the two methods. No factor was distinct enough to make a decision on what method to use on just statistics alone.

It is likely the methods were too similar to really provide significant differences, it might have been prudent to compare some of the earlier methods designed for the glyph, such as the overlaid histograms, but likely would have had to changed some of the questions.

With this in mind, the project decided to use the complex method for the next experiment. The additional glyphs provided in the complex method were determined more useful for asking a range of questions, also it wasn't clear why the method performed worse for the Median questions when both methods used the same symbols for it, but since the second experiment was going to run similar to this one, there would be a second chance to see how well the complex method handled Median questions against other simpler methods. While it was tempting to run both methods in the follow up study, it might have confused the participants too much, to have to compare two similar methods alongside different methods, whilst risking a repeat of this experiment where the two methods score comparably.

Despite that lack of a clear answer, this experiment set the path for the next. Second experiment used mostly the same structure as this one, so less preparation was needed.

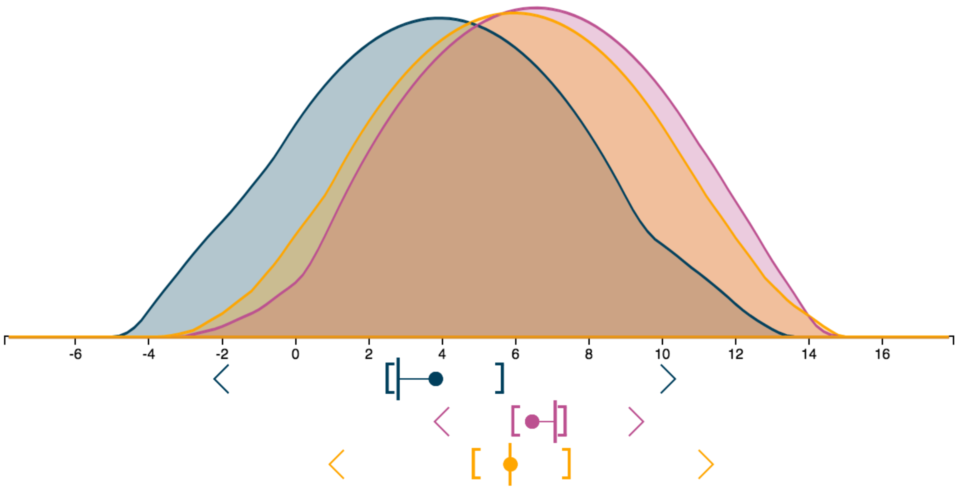


Figure 5.9 Density Box Plots in the same manner as the complex method from the previous study, but with a slight modification so that the quarterlies are square brackets rather then horizontal lines, in order to differentiate them more from the mean.

5.3. Study 3: Density Box plots: Statistical visualization method comparison

With the complex method becoming the selected way to display Density box plots, the main part of the experiment could commence, the comparison of Density box plots (Figure 5.9) to other methods to confirm if they were worth applying to the project or not.

The methods chosen to compare against were Box plots (Figure 5.10) and Density distribution plots (Figure 5.11). Both are common methods used to display statistics and distributions, with box plots being useful for highlighting median and quartile values, while Density distributions are good for showing differences in distribution shapes (Blumenschein et al., 2020). Additionally, they were the components parts of Density box plots in the first place, and if either method alone performs better than the Density box plot method, then the additional complexity added through merging the two would be questionable. The study compared how these methods performed in comparing different sample group statistical attributes. Consideration was made to compare the overlaid histogram as well, but it was considered too visually cluttered for the questions at hand.

40 Synthetic datasets were generated, again used for each method, but used in different questions and for different images to avoid learning effects. The random generation code used for this study was identical to the one described in Chapter 5.2.

Interaction was done in the same manner of the previous study, hovering over sample groups in the legend would fade out other sample groups in the question images.

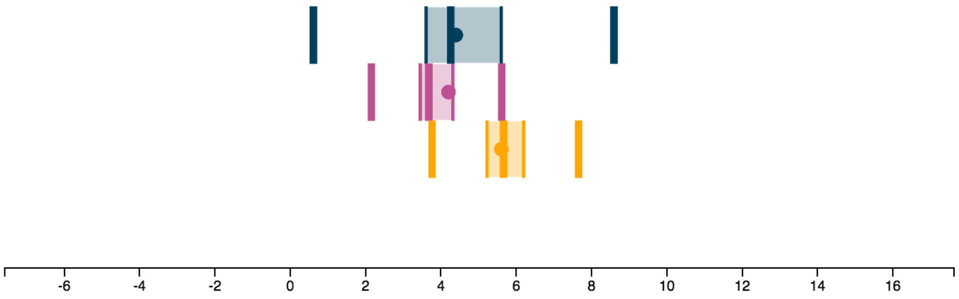


Figure 5.10 Box plots with quarterlies are represented as either side of the transparent box, with mean show as the horizontal line within each box, median shown as the circle, and the horizontal lines on either side of the box being the minimum and maximum.

5.3.1. Tasks

Each question was a multi choice question consisting of four images of a method with different dataset. At the side of the question was a legend with each sample group, labelled group 1 to group N, where N equals the number of displayed sample groups (three or five in this study), with a box coloured the same as its sample component in the images.

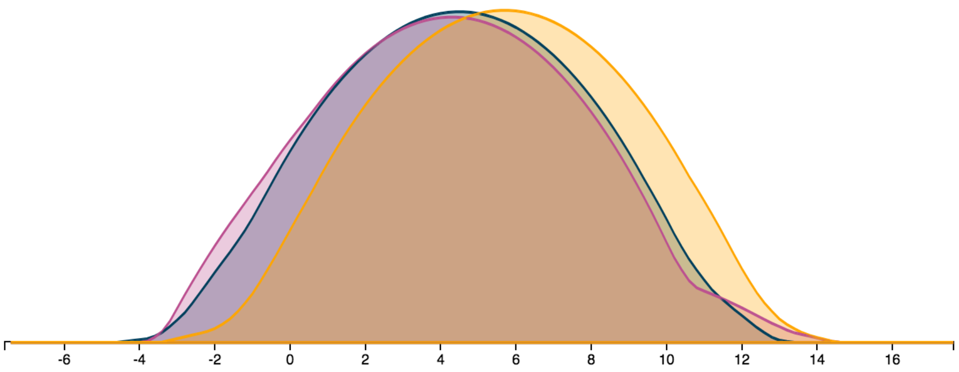


Figure 5.11 Density Distribution plots with no additional quartile information shown.

The following questions were asked, with *X* being some sample group:

1. Which view has the highest/lowest mean for sample group *X*
2. Which view has the highest/lowest median for sample group *X*
3. Which view has the largest/smallest difference between median and mean for sample group *X*
4. Which view has the largest/smallest interquartile range for sample group *X*
5. Which view overall has the most skewed/most normal (least skewed) distributions across sample groups?

5.3.2. Experiment Design and Procedure

The study was based on two hypotheses: **H1:** Density Box Plot would perform better than the other methods overall, as it offers more support in answering a wide array of questions. **H2:** Density Distribution plot would perform worse than the other methods, as working out questions such as the mean and median requires looking closely at the distributions, rather than looking for a glyph.

It was also of interest to see if the Density Distribution plot would perform as poorly as it did for Median questions as it did in the previous Glyph study.

The experiment was designed as a within-subject study with the visualization method as the main factor. The same structure was used from the previous experiments, developing the methods in JavaScript/D3, using Gorilla to build and run the experiment, and with Prolific being used to gather participants.

The same issues with participants rushing the study to get the rewards. The same actions were taken to evaluate user results to determine if they had been sincere or not.

At the start of the experiment, an introduction section covered the overall method and the three different variations being tested, alongside an explanation of sample groups and the structure of the questions and the experiment. Each set of questions for a method was preceded with a training section. This section first provided a user with an explanation of the method and a interaction demo, showing the user how to fade out sample groups. The training section then presented 5 questions in the same format as the main section, informing the user whenever they got a question correct or incorrect, with a tally at the end showing how well they did, users could then repeat the training questions if they wished.

After the training phase for a method, participants were presented with the main questions. Each method had ten questions rather than the 16 used in the previous study. It was decided that with more methods, the study would take too long and there would be more attrition of participants.

Questions were divided equally between datasets with three samples groups and five samples groups, and divided again between questions asking for the highest values and the lowest value.

For each question, the accuracy of the response and the response time was recorded. Additionally, after each question, participants were asked how confident they were with their answer, using a 5 point likert scale (1 = Uncertain, 5 = Certain). With three methods to compare, a Latin Square procedure (Graziano and Raulin, 1993) was used to counterbalance the order users received each methods, to ensure participants received a balanced variation in ordering of methods, and to reduce the potential learning impact on the results.

5.3.3. Evaluation Results

There were 35 participants who finished the study, 18 female and 17 male, ethical approval and participant consent was received prior to the study. The most abundant age group was 18-24 (34.29%), followed by 25-29 (25.72%), 30-34(17.14%), 40-44 (8.57%), 35-39 (5.71%), 45-49 (2.86%), 50-54(2.86%) and 60+ (2.86%). 17.14% of participants specialised in Computer Science, 5.71% in Bio Science, 5.71% in Statistics and 71,43%. As with before, the disparity between Computer Science, Bio Science and Statistics with the Other fields is due to the nature of Prolific allowing participants to sign up rather than directly inviting them. Participants were asked to rank their experience of 1) Data Visualization and 2) Statistical Visualization, using 5 point likert scales (1 = No prior experience, 5 =Expert). For experience with Data Visualization, 14.29% of participants put down their experience as 1, 34.29% as 2, 28.57% as 3 and 22.86% as

	Overall	
Methods	Z	P
BoxMean-CombiMean	-2.782	**0.005**
DenMean-CombiMean	-2.815	**0.005**
DenMean-BoxMean	-0.270	0.787
BoxMed-CombiMed	-1.414	0.157
DenMed-CombiMed	-4.310	**<0.001**
DenMed-BoxMed	-3.828	**<0.001**
BoxSkew-CombiSkew	-1.732	0.083
DenSkew-CombiSkew	-4.082	**<0.001**
DenSkew-BoxSkew	-2.985	**0.003**
BoxIQ-CombiIQ	-2.500	0.012
DenIQ-CombiIQ	-3.334	**0.001**
DenIQ-BoxIQ	-4.309	**<0.001**
BoxDist-CombiDist	-2.071	0.038
DenDist-CombiDist	-1.883	0.060
DenDist-BoxDist	-3.984	**<0.001**
BoxTotal-CombiTotal	-2.822	**0.005**
DenTotal-CombiTotal	-4.846	**<0.001**
DenTotal-BoxTotal	-3.885	**<0.001**

Table 5.2 Study3: Multivariate Accuracy

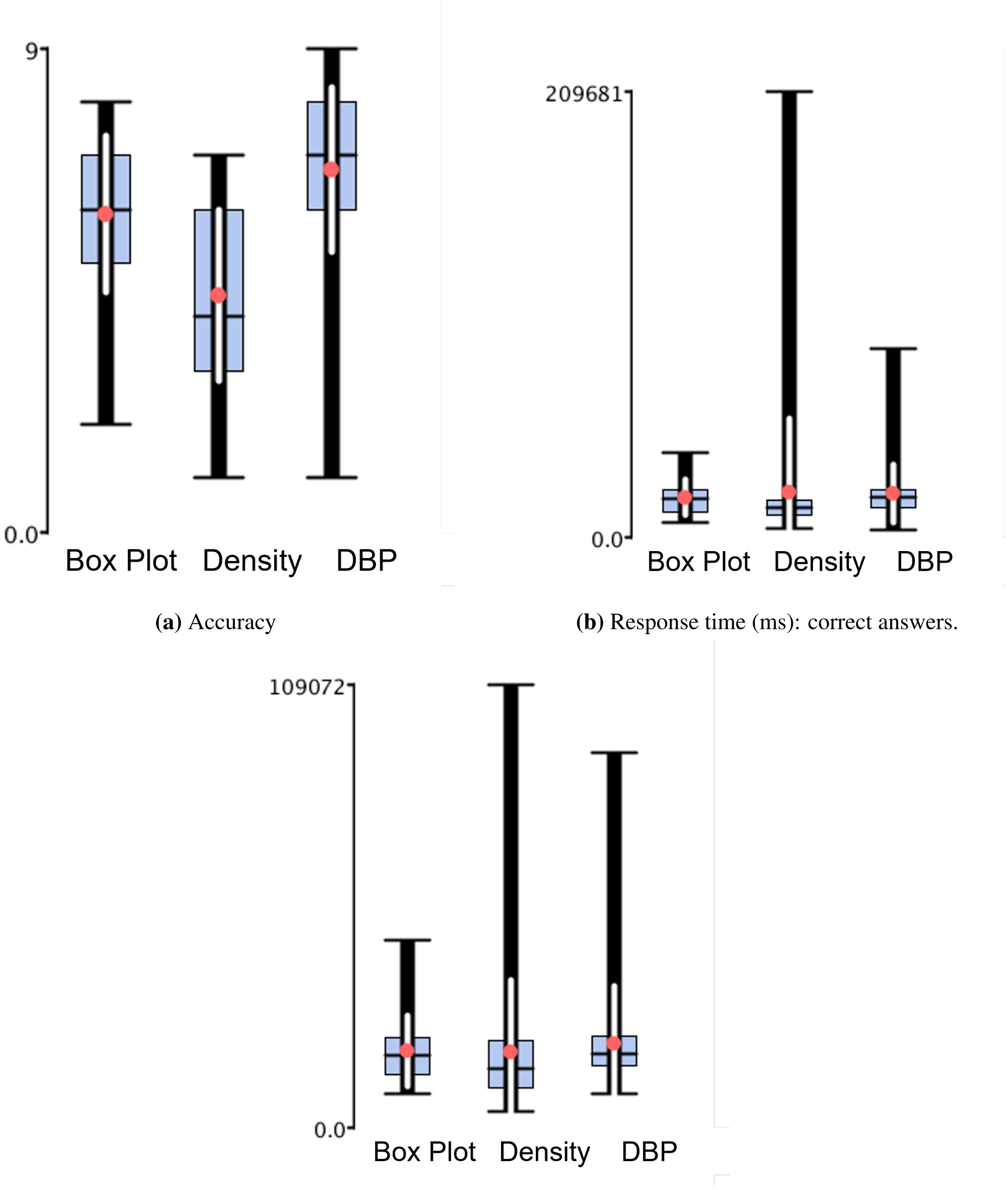
4, while for experience with Statistical visualizations, 17.14% put it down as 1, 37.14% as 2, 37.14% as 3 and 11.43% as 4.

Analysis of results was done in the same manner as in the Hierarchical previous studies, with the Friedman test being using for significance testing. This was followed by post-hoc tests using Wilcoxon signed rank test, with a Bonferroni correction applied resulting in a significance level set at p<0.0083, and pairwise comparisons for each type of questions for the two methods in regards to accuracy. Results for this study are reported in Figure 5.12 where medians, interquartile distances, minimum and maximum values are represented as box plots, and where red dots and white lines show the mean and standard deviation respectively.

Retroactively running a indicative power analysis, the effect size would need to be 85% in order to detect with 80% power for the sample size of 35 participants used during the study.

Results were more varied this time, for correct answers, the Boxplot Scored 207/360, Density distribution 154/360 and Density box plot 236/360, with significance testing returning ($\chi^2(35) = 35.812, p < 0.001$). Results for correct time ($\chi^2(35) = 4.343, p < 0.114$) and time ($\chi^2(35) = 4.343, p < 0.114$) didn't return any significant results.

Post-hoc testing, as seen in Appendix table 5.2 detected significant differences for accuracy for certain questions. For overall accuracy, Density Box plots performed better than Box Plots for Mean questions, Skew questions and for Overall results, while performing better than Density Distribution plots for Mean questions, Median Questions, Interquartile range questions, Skew



(a) Accuracy

(b) Response time (ms): correct answers.

(c) Response time (ms): all answers.

Figure 5.12 Results for Study 3: Density Box plots: Statistical visualization method comparison

questions and for Overall results, confirming **H1**. Box Plot performed better than Density Distribution plots for Median questions, Interquartile Range questions, Skew questions and for Overall results. Density Distribution plots only performed significantly better in one category, that being for skewed distribution results in relation to the Box plots. It was the one set of questions where participants scored higher for Density Distribution plots than for the other methods, overall meeting **H2**, since the only set of questions it performed well in were the ones that did not benefit from glyphs.

5.3.4. Summary

The study compared three methods, Box Plots, Density Distribution Plots and Density Box plots, in order to determine which would be most optimal for representing overlaid multivariate distributions.

The results presented in this study determine that Overall, Density Box plot proved better for representing overlaid multivariate distributions, and was at least comparable to the other methods for individual sets of questions. Following this Box plot was the second best method, while Density Distributions performed the worst.

The poor performance of the current version of Density Box plot in the previous study for Median questions, does not seem to have repeated itself here.

As a main result, this research contributes a novel visualization method that enables identification of multivariate aggregated sample groups; the efficiency and effectively of this novel visualization was demonstrated through two usability studies.

As for application within the project, the Density box plot will be used within nodes of the Treemap, alongside other views, to display multivariate aggregated distributions and support the comparison of different aggregated sample groups. This will be discussed in greater detail in Chapter 4.

The studies were designed and ran in late 2020, with the Covid-19 pandemic influencing a good deal of design decisions. Doing studies in person at this time was not viable, so running the studies online this time was an even more attractive option than it was for the Hierarchical study. Again I feel using to Gorilla experiment builder worked well for these experiments, with the lessons I learned from constructing the hierarchical experiment, I was able to make progress quickly this time.

The experiences with Prolific were more varied. The platform complemented Gorilla well, allowing for each experiment to be done in a few hours. However, more time was spent on quality control of participants that was not needed when manually inviting participants. This trade off was acceptable, as each experiment with data processing took two-three weeks each, whereas the Hierarchical experiment took a month on data gathering alone. Additionally the risk of getting rushing participants was mitigated by inviting more participants than was required with the expectation that some would not be suitable, and if that didn't get enough usable results, the experiment could be continued until enough .

Initial planning for the experiment was carried out just before Covid-19 hit, and at the time there was a consideration to run these experiments in person as a placement at Unilever, since the methods were interactive by nature and getting the methods to work on Gorilla would have been challenging. If the study had gone down this path, there likely would have only been a single set of experiments for the convenience of doing a single placement. Rather than having multiple choice tasks, the experiment would have participants compare the different methods within the context of them being used within a small hierarchical structure. This experiment likely would have provided better feedback, at the cost of not having Gorilla record all the timing variables, but in this situation the experiment would have preformed better in person, if that had been an option.

Chapter 6. HieraVis design and implementation

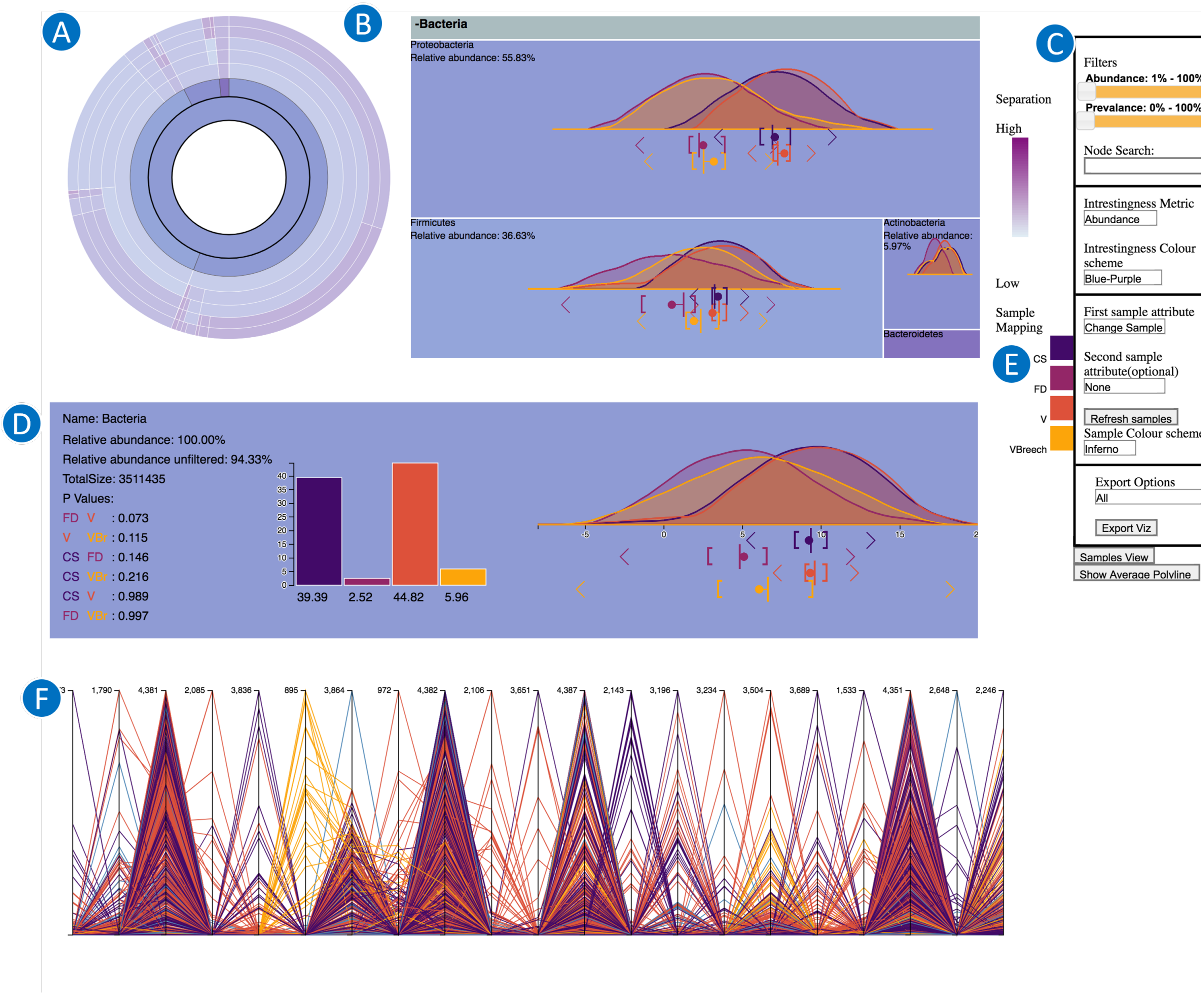
This section will cover the implementation of a visualization tool, HieraViz, to demonstrate the utility of the methods and concepts covered in the previous chapters.

6.1. Visualization System

HieraViz (Figure 4.1 was designed with the information seeking mantra in mind (Shneiderman, 1996), Overview first, zoom and filter, then details-on-demand. By presenting the user with ways of displaying and manipulating the full dataset through the Sunburst Chart (Chapter 6.1.1) and Multiple Small Plots (Chapter 6.1.7), users are presented with an overview to start exploring from. The Treemap (Chapter 6.1.1) provides a Zoomed in view of data subsets selected by the user, while filtering is done through the legend (Chapter 6.5). For expanded details, the Analysis view (Chapter 6.1.5) provides an extended view of the currently selected Treemap node, while the Parallel Coordinates view (Chapter 6.1.6) allows for the comparison of the subset of selected entities.

Additionally, HieraViz was developed around multiple coordinated views (Baldonado et al., 2000), with the core of the tool being the linked Sunburst-Treemap views, providing both an overview of the full data set in the Sunburst, and a detailed zoomed in view of a single data node and its children in the Treemap. With the Analysis View, Parallel Coordinates and Multiple Small plots views providing supporting information.

Implementation of HieraVis was done with JavaScript/D3 for the client and NodeJS for the server handling backend data processing. JavaScript, and specifically D3.js, have an extensive suite of options of creating dynamic visualizations, additionally it was expected that the final implementation would need to run online to be usable on end user machines, which were expected to be laptops that likely could not handle processing large datasets by themselves. NodeJS was selected to handle the data processing, be it pre-processing of the data, or sending a subset of the data whenever the user filtered the dataset. NodeJS was selected both for the convenience of working with both JavaScript for client and server, and because its a powerful framework for server-side development. Many of the implementations created during the user studies in Chapter 4 and Chapter 5 were reused and modified in the construction of this tool.



Filters
Abundance: 1% - 100%
Prevalance: 0% - 100%
Node Search:

Intrestingness Metric
Abundance

Intrestingness Colour scheme
Blue-Purple

First sample attribute
Change Sample

Second sample attribute(optional)
None

Sample Colour scheme
Inferno

Export Options
All

Name: Bacteria
Relative abundance: 100.00%
Relative abundance unfiltered: 94.33%
TotalSize: 3511435
P Values:
FD V : 0.073
V VBr : 0.115
CS FD : 0.146
CS VBr : 0.216
CS V : 0.989
FD VBr : 0.997

Figure 6.1 Main HieraViz view showing: (A) Overview of dataset in Sunburst Chart, (B) Selected subset in Treemap, (C) filtering options, (D) Details of selected node in Analysis view, (E) Sample group mapping, (F) Entity subset in Parallel Coordinates plot.

6.1.1. Sunburst Treemap view

As detailed in Chapter 4, the use of hierarchical representations provided a way to display a full high dimensional dataset within a single view, without the need to extensive prepossessing to reduce the dataset to something the user can understand. This allows for reduction and subset selection to be done on the fly within the tool, allowing the user to make use of their domain knowledge to inform decisions.

In the case of HieraViz, the primary exploration and analysis of the high dimensional data is performed in a pair of coordinated hierarchical views, one for overview and one for detail, as shown in figure 6.2. The overview method shows the full hierarchical structure of the dataset using Sunburst, while the details view only shows the currently selected taxon and its local hierarchical structure.

The combination of Sunburst and Treemap was chosen based on their performance in the usability study, and as they complement each other well, with the Sunburst able to present the

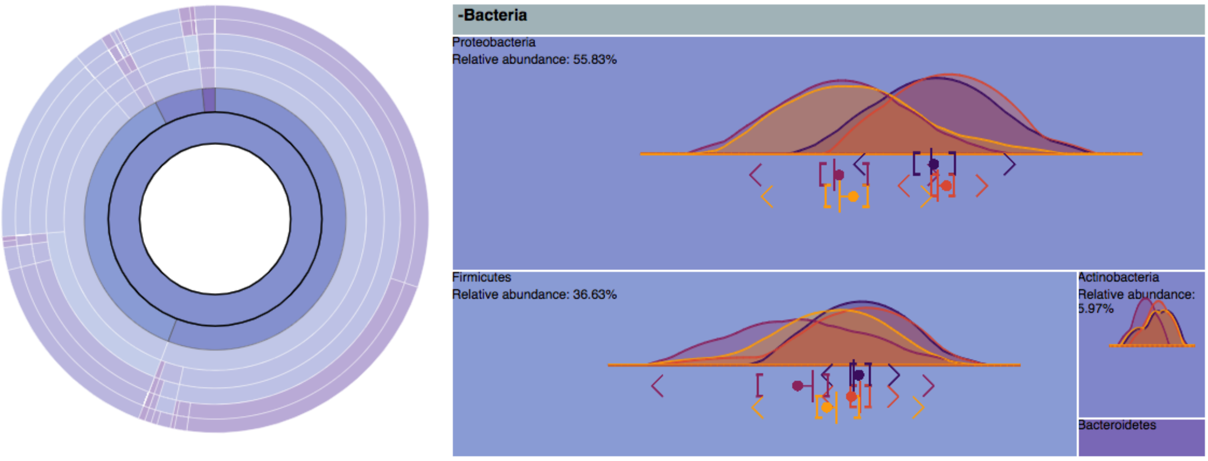

-Bacteria
Proteobacteria
Relative abundance: 55.83%
Firmicutes
Relative abundance: 36.63%
Actinobacteria
Relative abundance: 5.97%
Bacteroidetes

Figure 6.2 Linked Sunburst Treemap view, the node *Bacteria* is selected in the sunburst, with the Treemap showing its subset of child nodes.

full scope of the current dataset, but only shows broad patterns, while the Treemap allows for expanded details on individual subsets. Sunburst performed well for large hierarchies, and was selected over Icicle plot and Circular Treemap as overview visualization. While both Icicle and Circular Treemap performed comparably well in the study and performed better for size related tasks, Sunburst performed better for colour related tasks. With exploration in the tool being supported through use of a colour metric, the performance for colour tasks was considered more important.

Treemap, Icicle Plot and Circle Packing performed comparably well for small hierarchies in the evaluation. Of these, Treemap is the most space efficient, while Circle Packing and Icicle dedicate more space to showing the hierarchical structure. Displaying hierarchy is less relevant in the detailed view in this case, since only a small number of hierarchical levels are shown and with main focus being the individual nodes. The potential drawback of the hierarchical representation in Treemap not being as clear is also overcome by combining and linking it to Sunburst, which represents hierarchy clearly and allows the Treemap to be less cluttered.

In this combined hierarchical visualization, Sunburst provides an easily understandable overview of the hierarchical structure (Figure 6.3.a). It is also able to highlight and draw attention to patterns of potential interest in the hierarchy through colouring, as exemplified in the usability study. Through this it can provide guidance for exploration and allows for fast identification of taxon that may be particularly interesting to explore in more detail.

Hovering over a node in the sunburst (Figure 6.3.b) shows the path of nodes above the chart, while a tool tip shows the abundance count. Clicking on any node will make it the selected node in all other views, the node itself in the sunburst is highlighted, and all other nodes other than those that form the path to the selected node, or are its child nodes are faded out. Selecting the centre of the sunburst returns the view to the root of the structure and restores all nodes (Figure 6.3.c).

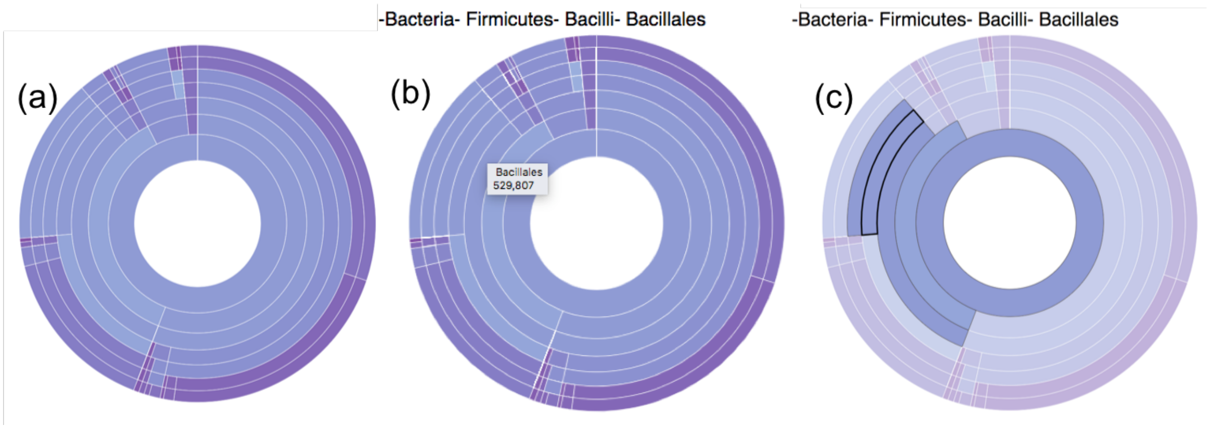

(a)
-Bacteria- Firmicutes- Bacilli- Bacillales
(b)
Bacillales
529,807
-Bacteria- Firmicutes- Bacilli- Bacillales
(c)

Figure 6.3 Sunburst chart selection: (a) Basic Sunburst Chart (b) Hovering over a node (c) Selected node

The Treemap will show whatever node is currently selected in the Treemap, with the current path shown above, though unlike the sunburst, this path only changes when the subset is changed, rather than just mousing over a different node (Figure 6.2). Child nodes are shown within the Treemap, each sized based on their abundance values. Navigation of the dataset can also be performed in the Treemap, going down the structure by one level by selecting any of the child nodes, while going up by one level by selecting the node path.

Within each Treemap node, the user is presented with a density box plot, showing the aggregated distributions of entities within the currently selected sample groups, alongside the node name and its relative abundance percentage in relation to the dataset. Relative abundance is used over raw counts due to the potential differences in scale between the counts values of entities, the data is compositional, an entity whose samples add up to hundreds of counts is not easily comparable to one with tens of thousands.

The combination Sunburst-Treemap method was presented in a chapter ((Macquisten et al., 2022)) in the book Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery (Kovalerchuk et al., 2022), a demonstration of how a large hierarchical structure being supported by a small hierarchical structure could be used to complement each other in the exploration of high dimensional data.

6.1.2. Colour Metrics

Taking advantage of the space provided within nodes to display information, nodes in both the Sunburst Chart and Treemap are coloured based on the separation of their sample groups, via silhouette clustering (Rousseeuw, 1987). In silhouette clustering, the relative quality of a cluster is defined as a ratio between dissimilarity within sample group and the dissimilarities of the items between clusters 6.4.

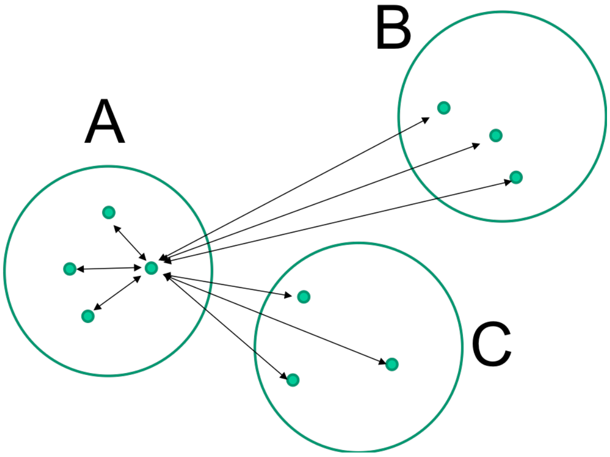


Figure 6.4 Silhouette clustering: For a data point in cluster A, the average distance between itself and each other data point in its cluster is calculated, in the relation to the average distance between it and data points in other clusters.

In the case of microbial data, this means that for each sample the average abundance or prevalence difference to samples in its own group, $a(x_{i,j})$ is calculated, as well as the average difference to samples in other groups, $d(x_{i,j}, G)$, where G is the other sample group. Equation 6.1 shows how the silhouette value, $s(x_{i,j})$, of a sample is calculated, where $b(x_{i,j}) = min_{G \neq A}(d(x_{i,j}, G))$ is the average difference to the most similar sample group.

$$s(x_{i,j}) = \frac{b(x_{i,j}) - a(x_{i,j})}{max(a(x_{i,j}), b(x_{i,j}))} \tag{6.1}$$

This results in a silhouette value for each sample where 1 indicates high similarity to its own group, while a value of 0 or lower indicates higher similarity to samples in other groups. The aggregated silhouette value of all samples provides a QM with high value if sample groups are different and hence well separated, and a lower value if sample groups are less different and hence not as well separated.

There are some issues with Silhouette Clustering, for example a cluster might be well separated from other clusters but very close to a single other cluster might end up with a similar metric value to one where all the clusters are relatively medium distance from each other. This may be fine in some cases, as through further exploration, users can compare the colour metric to other aspects of the visualization, such as the Density Box Plots in the Treemap or the Histogram in the Analysis view and see when these scenarios occur. But it is a limiting factor that the metric alone cannot tell apart situations like these.

There are alternative methods that could have been used, such as the Davies-Bouldin index (Davies and Bouldin, 1979) takes a ratio between the within cluster scatter and the separation between pairs of clusters. Or the Dunn Index (Dunn, 2008), which takes a ratio between the minimum inter cluster distances over the maximum within-cluster distances. This implementation Silhouette works as an example method for the concept of colouring by class separation, and there may be other methods that provide different separation results.

The metric can be configured to use abundance values or prevalence values, both are quality metrics for highlighting data of potential interest in 'omics data (Johansson et al., 2020). Users can switch between the metrics on the fly to see their impacts across the different sample groups.

In the colour scheme, darker colours represent higher abundance/prevalence and clearer separation of abundance/prevalence between sample groups. Correspondingly, lighter colours represent lower values of separation. A legend bar is displayed at the side of the Treemap (figure 6.5C), showing the range of colour values from low to high, updating if the colour scheme is changed.

By default, the colour metric uses the light blue to dark purple colourscheme '3-class BuPu' scheme from Colorbrewer2 (Brewer, 1999), this can be switched for alternative colour schemes: light green to dark green '3-class Greens'; light green to blue '3-class GnBu'; light orange to dark red '3-class OrRd'; light blue to dark blue '3-class PuBu' and a greyscale colour scheme. While these provide different options for users to suit their preferences, more importantly these methods cater towards users with colour vision impairment.

6.1.3. Filtering and data manipulation

At the side of the Sunburst-Treemap view is a series of panels and functions to make changes to the loaded dataset and modify the visualizations. Ultimately, the dataset is always going to be too large for a user to understand when presented in full, but by having these filter and data manipulation options, the user is given allot more control over the exploration and analysis process. Subsets of data and sample options can be selected and changed by the user as they explore the dataset, without the need to reload the data or the tool. Preprocessing to remove portions of the data can instead be left to the in-tool options.

The full dataset can be filtered using sliders for abundance and prevalence (Figure 6.5A). Both metrics are provided, as while an individual entity might be highly abundant, it might not be very prevalent across all samples.

Users can set a percentage range for both sliders, cutting out data values below or above certain thresholds. While most microbiomics datasets are normally filtered to cut out the bottom 5% of values, sometimes its more interesting to look for values that at are less common, for example an entity that has a high abundance value but a low prevalence value might indicate that the samples it in may be outliers.

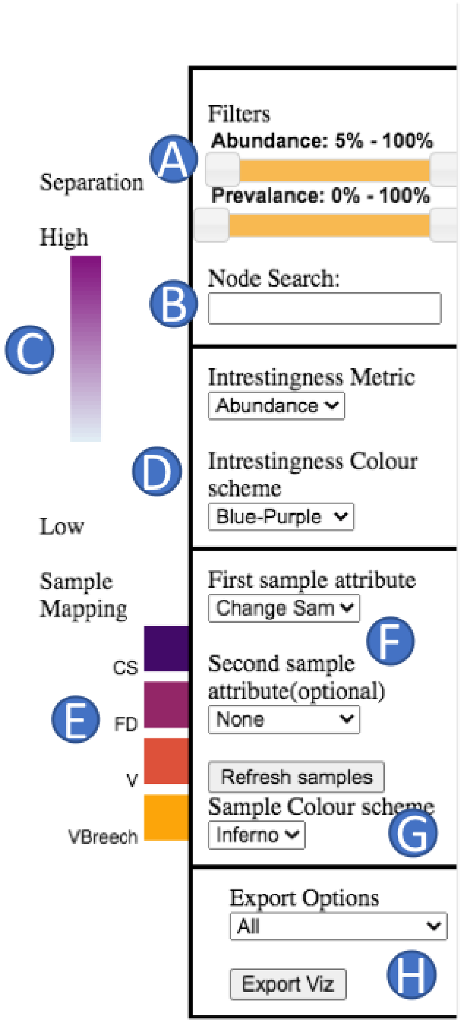

Filters
Abundance: 5% - 100%
Prevalance: 0% - 100%
Node Search:
Intrestingness Metric
Abundance
Intrestingness Colour scheme
Blue-Purple
First sample attribute
Change Sam
Second sample attribute(optional)
None
Refresh samples
Sample Colour scheme
Inferno
Export Options
All
Export Viz

Figure 6.5 Sidebar for filtering options, with: (A) Showing the abundance and prevalence filters. (B) Search function. (C) Colour range for the colour separation metric. (D) Colour separation options. (E) Sample group legend. (F) Sample group selection options. (G) Sample group colour scheme. (H) Export options.

A search function is provided through an auto-fill text box (figure 6.5B). As the users fill this out, the box presents a scrollable dropdown with all auto-filled paths to nodes in any level of the hierarchical structure. Hovering over any of these highlights the node in the Sunburst chart, selecting any of them loads it into the main view with path highlighted.

Below this are the previously mentioned options for changing the colour separation metric and scheme (figure 6.5D).

A sample mapping legend shows the current selection of sample groups, each with a coloured box showing the sample colour used throughout the visualization (figure 6.5E). Hovering over any of these sample boxes fades out all other sample groups in all views, so to focus only on that sample group. Selecting any of the sample boxes changes the box to black and filters that sample group out of the dataset and all views, selecting it again brings it back into the dataset.

To the side of this are options for changing the sample group (Figure 6.5F). Users can select either a single sample attribute, or select two and merge them together for when a single group doesn't express differences in the dataset well enough, with these new sample groups comprising of entities belonging in both sample components. For example in Figure 6.6 the two sample groups Visit (A and B) and Product (C and D) each individually have two sample groups, resulting in Four possible combined sample groups (AC; AD; BC and BD).

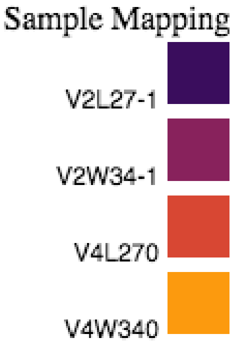


Figure 6.6 Sample group Visits consists of two Sample groups, A:V2 B:V4 and Sample group Product also consists of two groups, C:L27 D:W34. These are merged to form four new sample groups.

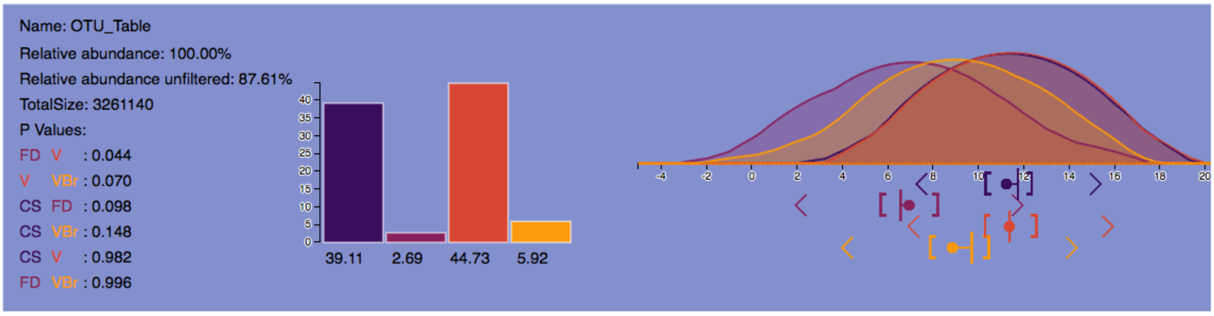
Below this, there is an option to change the scheme used for the sample groups (figure 6.5G), the default option is the 'Inferno' scheme from the Matplotlib colourmaps (Smith and van der Walt, 2015), with the 'Vidris' and 'Cividis' (Nuñez et al., 2018) schemes being provided as alternatives. These schemes were selected to provide options for users with colourblindness, while in addition to being a different set of schemes to the ones used for background colour to avoid colours getting mixed up. There are still combinations of background and sample colour schemes that do clash to some degrees, but that is in part due to the number of different options provided for both sets of schemes.

Finally, at the bottom of the sidebar are export options (figure 6.5H), allowing users to select which part or parts of the visualization they want exporting as a PNG.

6.1.4. Density Box plot

As detailed in the glyph development studies in chapter 3, a Density Box plot glyph is used to show the aggregated distribution of abundance values for each sample group within each Treemap node.

The glyph is scaled with the size of the node, in smaller nodes only the top component is shown. Interaction with the sample group filters at the side can be used to filter in and out sample groups from all the plots, while hovering over a sample group will fade out the others, allowing the user to focus on a single sample group and compare across charts. Data is log transformed, due to the extreme range of abundance values that can be provided from the aggregated entities. Many



Name: OTU_Table

Relative abundance: 100.00%

Relative abundance unfiltered: 87.61%

TotalSize: 3261140

P Values:

FD V : 0.044

V VBr : 0.070

CS FD : 0.098

CS VBr : 0.148

CS V : 0.982

FD VBr : 0.996

Figure 6.7 The Analysis View extends the currently selected node in the Treemap with additional information, including a histogram of sample group relative abundance distribution, P values, and additional text information.

entities will have a single abundance count detected across the whole dataset and will likely be filtered out by the user, but the two most abundant entities could still have a large difference in abundance between them.

6.1.5. Analysis view

Despite all the filtering options available, along with the Treemap only showing a very small subset of the entire dataset, there is still the issue that a small enough node within the Treemap, won't be able to show all of the information displayed that a large node can. To combat this, an expanded Analysis view is used to show the currently selected node in full (figure 6.7).

The user can change what node is selected by right clicking a node within the Treemap. In addition to showing the normal node information in full, the analysis view also displays P values for the aggregated values of each combination of sample group currently not filtered out. The current method for generating P values is the Tukey's range test (Equation 6.1.5), which returns P values for each pairwise combination of sample group values, with Y_A being the larger mean being compared and Y_B being the smaller mean, while SE is the standard error of the means. This is just one example of a method to generate P values provided as a demonstration and could be swapped out for other methods, or expanded to allow multiple methods to switch between.

$$q_s = \frac{Y_A - Y_B}{SE}$$

(6.2)

In addition, a histogram is displayed showing the relative abundance values of each sample group. While the histogram considered in Chapter 5 was deemed not useful enough by itself in the Treemap view as it didn't provide much in the way of statistical analysis by itself, it did provide a clear way to see the values of individual sample groups, which is helpful to understand the background colour metric values further. A node that displays a dark background colour signifies separations between the sample groups, and from the histogram the user gets a clear value. This can be done in some degree in the Density Box plot still, as the different samples will still be separated in that, just without a direct value. The Overlaid histogram discussed in Chapter 5 was not used for this view, as it was deemed too complicated due to it being difficult to distinguish between the different overlaid sample bars.

6.1.6. Parallel Coordinates

A parallel coordinates view 6.8 is displayed below the analysis view, displaying the currently selected data subset of entities selected in the Sunburst-Treemap, updating as the subset does, extending the analysis from the other views. Individual samples are show as polylines, coloured by the sample group they are associated with, while leaf node entities are shown as the individual axes, showing the relative abundance profiles of samples across the subset.

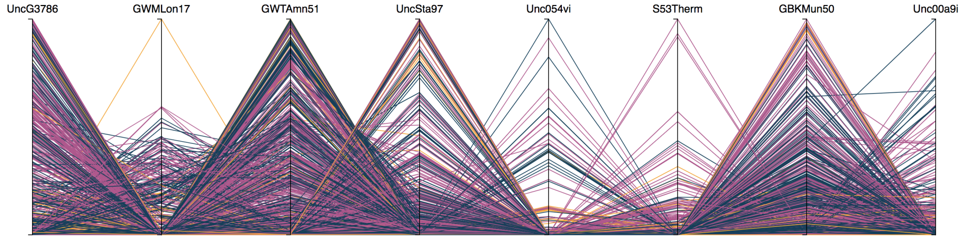


Figure 6.8 Parallel Coordinates plot, each axis is an individual dimension in the currently selected subset, while each polyline is a different sample, coloured by its sample group.

Hovering over a polyline shows its sample name and fades out the rest of the samples to show its values more clearly. Hovering over any of the sample groups in the legend will fade out samples not in those groups. An average polyline for each sample group can be toggled in and out, fading out the rest of the sample polylines.

Hovering over an axe shows its genus name and species name and fades out the other axis names. Again like in the Treemap, when showing the names of the individual entities, the same names can appear for different entities that have different parent geneses, so having the both names is important for identification. By default when not interacting with an axe, each axis shows just its species name for space reasons. If there are more than 10 axis, names won't show unless the user hovers over one.

6.1.7. Multiple Small Plots

While the tool allows for the changing of sample metadata on the fly and for the filtering of sample groups, the default view is limited to showing a single combination of sample groups at once due to the limited screen space.

To overcome this, HieraViz makes use of Multiple Small Plots (figure 6.9), which provides an overview of each possible combination of sample groups side by side. This aids the user in identifying a meaningful starting point for the exploration process by allowing for a quick comparison of patterns in the different data subsets.

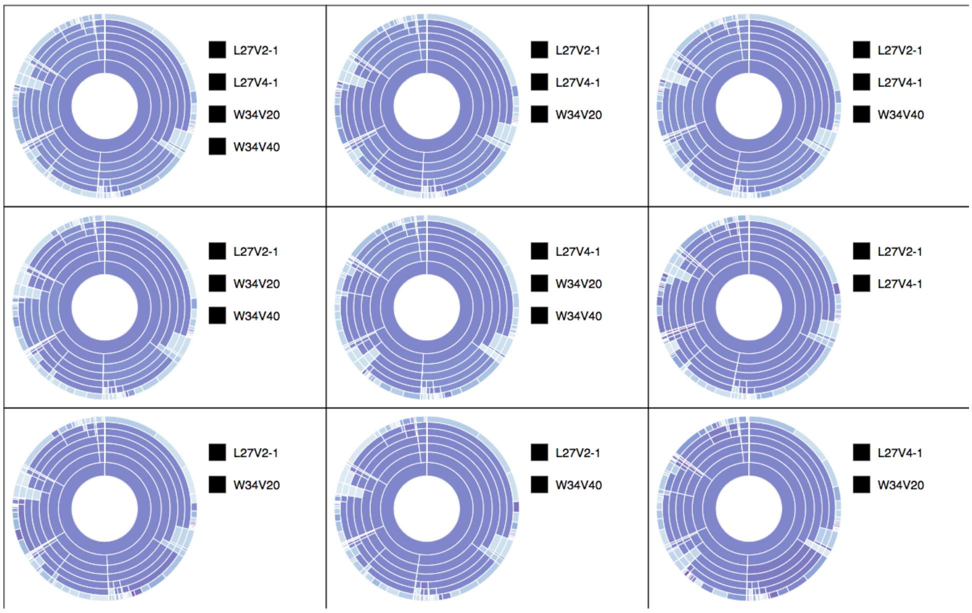


Figure 6.9 Multiple Small Sunburst charts are generated for each combination of sample group and laid out next to one another.

The Multiple Small Plots view uses sunburst charts to keep things consistent with the Sunburst-Treemap view, using the same structure, colour metrics and filters. This consistency was important, as stated by Qu and Hullman (2018) in their consistency constraints, with the same data being shown, it should be encoded in the same way, as so the viewers attention is not drawn to visual changes given a constant data view across views. While also maintaining the consistence between these views reduces the complexity of learning the tool and allows for users to make comparisons easier (Baldonado et al., 2000).

The view can be filtered to ignore combinations of sample groups below a set number. Hovering over any node in any plot, will highlight its location in all plots, showing differences between sample combinations as nodes are at different sizes; have different colour metric values and are in different locations from the sunburst ordering nodes highest to lowest abundance for each segment. Selection of any individual node will load its plot into the Sunburst-Treemap view with all its sample parameters and with the path to the node highlighted.

Chapter 7. User Testing

This section covers a user study carried out to evaluate HieraViz, including the results and what feedback would be implemented.

7.1. User testing

In order to evaluate the tool and by extension the research, a series of one-to-one semi-structured interviews were carried out. For this assessment, the primary goals were to determine user acceptance of the tool and methods, alongside generating feedback for updating the tool.

Restrictions due to COVID-19 prevented these interviews from being organised in person, so instead the plans were adapted to be carried out online. This also allowed for participants to not be constrained by location, with participants being split between various locations in both UK and India.

The interviews were run through Microsoft Teams, primarily as Microsoft Teams has a share option that allows other people in the session to have their own cursor and interact with the hosts machine, allowing participants the interaction they would otherwise had during an in person interview. A drop in performance of the tool was expected from running the tool over Microsoft teams, but from initial pilot test runs of the interviews, it wasn't a significant issue.

7.1.1. Participants

Five participants took part in the study. Three of the participants were microbiologists at Unilever R&D, two of them were PhD students from the Translational and Clinical Research Institute at Newcastle University. From the participants, three were female and two were male, and their backgrounds were all in bio-science.

Two additional studies were carried out as pilot test runs prior to main set of interviews in order to refine the study, both test participants were PhD students researching visualization in the school of computing at Newcastle university. Participants for the study were recruited through our collaborators at Unilever and at the University. Prior to each interview, participants were sent a consent form to sign an return and ethical approval was received from the university before each study was carried out. Alongside this they were presented with a form with a series of heuristics (Forsell and Johansson, 2010) to evaluate HieraViz to read before the interview and fill out after.

7.1.2. Interviews

Interviews were done with a single interviewee at a time. Each interview began with going over the structure of the interview. This was followed by the main demonstration, going through each view and function of the tool in turn. The same structure was used between interviews for consistency, but participants were able to ask questions at anytime, alongside interacting with the tool themselves if there was any aspect they wanted further understanding of. Following the demonstration, participants were given a chance to ask any additional questions and further interact with the tool if needed. Each interview lasted between 30 minutes to 1 hour. Each demonstration used the same dataset, a microbiomics multi-site study, comparing mouth, skin and scalp data, consisting of 488 samples and 1254 dimensions.

Following the interview, the participants were instructed to fill out and return the heuristics form (Appendix section A.1). The heuristics form consisted of 14 questions (table 7.1), with each heuristic rated with a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree), additionally each heuristics had a space for users to make comments. Heuristics were split into groups with relation to what view or functionality they referred too, with each group of heuristics had an attached description of the method they related too, alongside a screenshot or diagram to aid participants.

7.1.3. Results

The study performed well, running the tool through Microsoft teams did provide some issues in regards to performance, but not by a significant degree and each study finished within the allotted one hour time. The results can be seen in table 7.2 and figure 7.1.

In order to process the results, a modified version of the score aggregation metric described by Wall et al. (2019) was used. Individual heuristics are grouped into guidelines based on their similarity, which in turn are grouped into top level components, in their case, Insight, Time, Essence and Confidence. The likert score of each heuristic is used as s_h, for each guideline, its score is generated by averaging each of its associated heuristic values, with j being the number of associated heuristics $s_g = \frac{1}{j}\sum_i^j s_{h,i}$. With this, each top level component has its score generated by averaging its k guideline scores: $s_c = \frac{1}{k}\sum_i^k s_{g,i}$. The final visualization score is generated by averaging all the top level components.

Number	Component	Heuristic
1	Sunburst	The Sunburst Hierarchical structure is intuitive.
2	Sunburst	The Sunburst highlights taxon of interest to guide exploration.
3	Treemap	Zoom/filter – The Treemaps local hierarchical structure is intuitive.
4	Sunburst-Treemap Combined view	Concretize Relationships – Selecting a node in either view loads it into the other view and highlights the path in the sunburst, the relationship between the two views is intuitive.
5	Sunburst-Treemap Combined view	The size of the views and nodes is sufficient
6	Multivariate sample group comparison	The interestingness metric provides a good guide to which nodes are interesting to look at.
7	Multivariate sample group comparison	Both the Density Box plots and Colour separation metric shows differences between sample groups within a single node and between nodes. These views complement each other well in determining why a node is interesting
8	Multivariate sample group comparison	The use of multiple overlaid glyphs supports the ease of sample group comparison?
9	Analysis view	Multivariate Explanation – The relationship between histogram values and colour interestingness metric (background colour) is intuitive.
10	Analysis view	Multivariate Explanation - The relationship between P values and colour interestingness metric (background colour) is intuitive.
11	Analysis view	Multivariate Explanation - The relationship between histogram values and Density box plot is intuitive.
12	Parallel Coordinates	Details – The relationship between axes of Parallel Coordinates plot and the selected taxon in the other views is intuitive.
13	Multiple Small Plots	Overview – Comparing and interacting with the multiple sunburst charts provides a useful starting point for exploration.
14	Colour options	Was able to find colour combinations that were usable.

Table 7.1 Heuristics used within the study, the full set of questions with accompanying information is present in the appendix form A.1

	P1	P2	P3	P4	P5
H1	5.00	4.00	5.00	4.00	4.00
H2	4.00	5.00	4.00	4.00	4.00
H3	4.00	4.00	5.00	4.00	4.00
H4	5.00	4.00	5.00	5.00	5.00
H5	3.00	4.00	5.00	4.00	4.00
H6	5.00	4.00	5.00	5.00	4.00
H7	5.00	4.00	5.00	5.00	5.00
H8	4.00	3.00	5.00	5.00	5.00
H9	4.00	2.00	5.00	5.00	3.00
H10	2.00	4.00	4.00	4.00	3.00
H11	4.00	3.00	5.00	4.00	4.00
H12	5.00	4.00	5.00	3.00	5.00
H13	5.00	5.00	5.00	5.00	3.00
H14	4.00	4.00	5.00	5.00	3.00

Table 7.2 Scores of the five participants, with participants as columns and heuristics as rows.

The advantage of this approach is that it doesn't favour any single component, guideline or heuristics over the other. While designed to compare multiple visualizations, it is usable for evaluating a single method. A modification of the method is used for the user results, by changing how our heuristics are divided between groups and only using two levels of evaluation rather than three. Heuristics were grouped into components based on their similarity, with Heuristics 1-5 covering the Sunburst-Treemap method, 6-8 covering multivariate sample group methods, 9-11 covering comparisons of multivariate sample group methods, with 12-14 covering additional methods.

	Sunburst-Treemap	Multivariate sample group methods	Comparisons of Multivariate sample groups	Additional views
P1	4.20	4.67	3.33	4.67
P2	4.20	3.67	3.00	4.33
P3	4.80	5.00	4.67	5.00
P4	4.20	5.00	4.33	4.33
P5	4.20	4.67	3.33	3.67

Table 7.3 Score Aggregation the five participants, with the the first four columns showing the aggregated groups, and the final 'Average' column showing the aggregated value of the other columns.

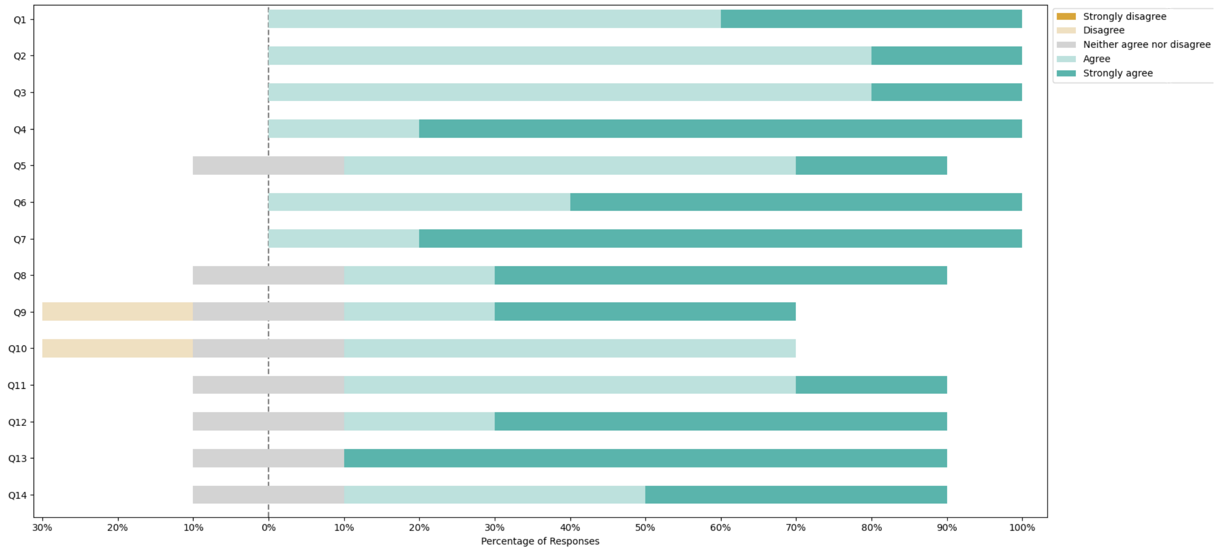


Figure 7.1 Bar chart of Table 7.2 likert responses.

The results of the score aggregation are shown in figure 7.3. The general consensus among participants was that they liked the tool, which can be seen though the overall results being positive, with the lowest average group score received being 3.0, equating to a neutral response, and the lowest total average for a participant being 3.8. Scores for each group were relatively consistent between participants.

7.1.4. Feedback and Discussion

The overall reaction to HieraViz was positive, with interviewees liking the tool. Most feedback given, though not all, was minor, with small changes or additions requested to certain components. The following section will cover details on feedback from the interviews, divided between the individual components of HieraViz.

Sunburst-Treemap

Overall the combination of Sunburst chart and Treemap to show different aspects of the dataset was well received. The Sunburst (Figure 6.2) was generally recognised from tools previously used by the interviewees, such as Krona plots Ondov et al. (2011) and generally understood how it functioned. Comparisons to other sunburst tools led to feedback from interviewees asking for additional information with the sunburst chart, such as showing the current path for whatever node was being hovered over, alongside name labels and relative abundance value. While this information is provided in the treemap, and space limitations of the sunburst prohibit much to be added directly, providing this either above the sunburst or when hovering over a node was an easy addition.

Relating to the space limitations of the sunburst, one interviewee asked for more ways to manipulate the sunburst chart, specifically being able to zoom in to see the outermost nodes. Currently the filtering options do allow to see the low abundance nodes, by filtering out nodes above a certain abundance value to only see those below. But more ways to see these smaller nodes, either through zooming in the Sunburst to see certain levels, or options to switch how the Sunburst partitions data to be something other than abundance. Investigating appropriate methods for doing so is subject for future work.

Feedback for the treemap mostly referred to the contents. At the time of the interview, raw reads were used to show the count of each node, however due to the different scales of data, comparing these reads wasn't a helpful or accurate comparison, instead the values shown needed to be relative abundance, the percentage value of abundance in relation to the full dataset. This had been applied in various parts of the tool already, but not in all places it should have been. Regardless it was an easy update.

Other minor feedback covered how the names entity's were shown, specifically the outermost leaf node names of the hierarchical structure. Firstly, there was a request that the names be italicised to be consistent with how bacterial entities were commonly written. Additionally it was commented that there could often be allot of duplicate names for entities across dataset with different parent/genus names, so having both the genus and species name for the outermost leaf nodes would be beneficial.

Group Separation

One request was for the option to set the node background to be white, purely for the purpose of being able to use the image of the treemap and the density box plots, with the background being consistent so the main focus can be on the other elements of the treemap. This isn't that relevant for the data exploration and analysis aspects of the tool, but as an additional option for accessibility and data presentation, it is a very easy addition.

Expanding on this, one comment was to remove the colour metric from the analysis view altogether: "*I really like the layout and the information, not sure if the background colour helps. The colour helps in the above combinations (Combined View) to select which taxon to look at but not sure it is need here.*". This might be good as an option, as mentioned before with setting the background to white, though removing it entirely wouldn't necessarily be good. Having the Analysis views node have the same background colour as its Treemap and Sunburst Chart nodes maintains consistency between the view.

Another request was to explore other colour patterns for interestingness, such as a 'rainbow spectrum' rather than light to dark, which would be a diverging colour scheme. This wasn't considered, the colour schemes all selected for the colour metric were sequential ones, having a different colour for every single value of the metric would be confusing to make out, especially if in between values (say between 0.2 and 0.3) were a transition between those two values colours. Additionally using sequential schemes for the colour metric separates them for the diverging colour maps used for the sample group colours. Additionally, as stated by (Borland and Taylor, 2007) a rainbow colour map in particular is problematic, its lack of perceptual ordering results in the data being obscured through its uncontrolled luminance variation, requiring more effort for the user to interrupt which colour value attributes to which metric value.

Density Box plot

General response to the Density Box plot (Figure 5.9) was positive, interviewees were familiar with the component methods. There was some initial confusion with the method being overlaid, but once the full method and ways to interact with it were demonstrated, they understood it.

Some users stated a preference for other methods, with one interviewee stating "*I personally prefer boxplots overlapped with scatter plots respect to having a density distribution, as I find it more intuitive for understanding the number of samples in each group and how they are distributed.*"

The overlaid nature of the Density Distribution component was accepted and understood, but some of the interviewees were confused by the semi-overlaid box plot components, commenting it would be less overwhelming if they were more separated. When shown that they could interact with the sample legend to hide and filter out individual samples, they did find it less overwhelming. But modifying the Density Box plot layout to expand these components out when there is space. Another suggestion was the adding of individual data points into the abstracted box plot components, or as a separate box plot.

The main issue pointed out as that the Density box plots sigmod curve would go below 0 on the X axis sometimes or wouldn't matching up with the box plot values, despite there never being any negative values in the data. As stated by Węglarczyk (2018), the kernel estimate can have a degree of leakage into negative values when part of the sample lies near 0 and the magnitude of the smoothing coefficient enables crossing. This scenario could easily come up within the data, as many samples can have low values from having only a few detected counts within the dataset. A potential solution mentioned was to replace the Kernel function with a asymmetric kernel function instead, such as a gamma kernel (Chen, 2000), which always generates kernels that are always non-negative.

Additionally whenever a sample group only had a single value, all the glyphs for that sample group would be placed at the same point, which may be problematic if users mistake a sample with a series of very close values for a sample with a single value, replacing the box plot glyphs with a single glyph in these scenarios would be ideal.

Another concern was that there was too much ’white space’ within the Treemap and Analysis view, that the Density Box Plots could be increased in size to fill up more of the area.

Finally, it was requested that an axis was added to the Treemap’s Density Box plot. While an axis wasn’t added to it originally, it is inconsistent with the Density Box plot in the Analysis view having one.

Multiple Small Plots

Reception to the Multiple Small Plots (Figure 6.9) was well liked, one interviewees stated "*Very nice way in which to look at multiple comparisons in a single analysis / screen. Can easily determine the potential differences between groups.*". The same comments provided in regards to the sunburst chart were also made here, providing labels, paths and relative abundance values. Since the two methods are effectively identical for the Sunburst coding component, any updates made to one can be made to the other without many differences.

Beyond a repeat of those issues, one interviewee was concerned about the comparing multiple sample groups. *"Some potential difficulties when comparing 3 or more groups. Are differences suggesting that all 3 groups are different to each other or if 1 group is different to the other two. Would need to delve into the data further to examine but can be done via these plots depending on how they are arranged."*

In this situation, the group separation metric only tells you there is a difference for plots with three or more sample groups, not which groups are different to each other. But that is where the user would select the plot showing potential differences and explore it further within the Sunburst-Treemap view, where separation of individual sample groups can be compared by looking at the Density box plots.

Parallel Coordinate Plot

While some interviewees hadn’t had much prior experience with Parallel Coordinate Plots (Figure 6.1.6), the method was understood once explained. As one Interviewee stated "This is a simple way to determine potential outliers in the dataset. Of most interest is average values for each group as this can help to distinguish potential differences between groups."

One addition suggested was to use the Parallel Coordinates plot as a way to filter out individual samples from the dataset. Currently there is no method to remove sample from the dataset, beyond through filtering or manually editing the dataset. Considering how HieraViz is supposed to provide users with the ability to explore and manipulate the dataset without the need for prepossessing, such an addition would be good for a future version.

Data

One comment was that it might be a good idea to give the user the option to transform or normalise the data at the beginning, rather than all data being log-transformed as the default for the Density Box Plot. For example, allowing the user to choose centred log ratio transformation, arcsin square-root etc.

The way P values were generated was questioned by a few users. The most common comment being that P values should be generated through a centred log ratio approach. As mentioned previously, the current method of generating P values through the Tukey's range test is replaceable with other methods.

There was some discussion regarding using False Discovery rate Q values instead or alongside P values, but other interviewees who mentioned them felt they would not be necessary as long as the P value method was good enough.

7.2. Summary

The study evaluated the method HieraViz, a tool for the exploratory analysis of high dimensional hierarchical data through the combination of Sunburst Chart and Treemap. The study had users evaluate the tool through a series of evaluation heuristics and participant feedback. Despite the limitations presented by having to do the interviews online, the overall process went well.

Responses to HieraViz were overall positive, both from feedback and from score aggregation results of the Heuristics. Overall, there wasn't any feedback that would require massive fundamental changes to the tool. Most of it was small changes, such as modifying the way P Values are generated, or additions such as showing more information when interacting with the sunburst chart.

Larger updates that the project will consider for future iterations of HieraViz include expanding the amount of interactions that can be done with the Sunburst Chart, most notably being able to switch into a 'Zoom in' mode to see lower levels of the hierarchical structure to make smaller nodes more visible. In this instance the Treemap would still show the selected data node with its immediate children nodes, the only change being the Sunburst would be more flexible. Additionally, adding a help page to provide users with guidance of the various methods needs to be added.

There was a few limitations of the way the study was conducted. There was a risk that some interviewees might not provide fully honest feedback, feeling constrained talking directly to the developer of the tool. While it is possible this could be mitigated by having someone else do the study in my place, this however wouldn't have been very practical, as it would require training someone else in both using the tool and the experiment. In the end however, the last thing the interviewees want is for the tool to not work, they are the people who would be using this tool in the end, and if it doesn't do enough they I feel they are more likely to say something about it.

As was the case during the requirements gathering phase of the project, the opinions of a single interviewee might not be enough to validate changes to the tool. To blindly follow each piece of feedback would risk making changes that only suited that one interviewee, or were not beneficial to the tool. As such, feedback that is provided should only be considered if it makes sense to include in the tool, the more people who provide the same feedback, the more likely it is worth considering.

Of course, having to do the study online was a big limitation, and while steps were taken to ensure users had a similar experience to what they would have had in person aided in mitigating the issue, it is still a notable limitation. In the event I was able to conduct the experiment the way I wanted, without the restrictions imposed by COVID-19, not only would the study have been conducted in person, but it likely would have been structured different. With the option for users to directly interact with the tool in person, the interview would have ideally contained a task phase alongside the demo and heuristic form. While interaction within the tool was possible in the online study, having to actually use the tool to answer questions would allow interviewees a better understanding of HieraViz's functionality.

While the feedback wasn't significantly impacted by doing the study online, there was some issues caused in getting heuristic forms back and accessing the recordings of the meetings. The recording issue was caused by Unilever security, requiring the interviewees to record meeting and send the recording, the impact was minimal though, as the recordings were only necessary for additional note taking. The heuristic forms on the other hand often were sent after the meeting rather than during it, risking the participant forgetting comments they had. Had the meeting been done in person, it would be easier to enforce the forms being filled out then and there.

7.2.1. Comparison to other methods

As mentioned during Chapter 2.2, visualization methods were evaluated for high dimensional data in their ability to support the exploration process. Of key interest were the following metrics: The scale of dimensionality the method could handle; The options for interaction provided; and how engaging the method was. In table 7.4, HieraViz is compared against methods evaluated in the literature review for its performance in these metrics.

In terms of dimensionality, most of the datasets used in testing were in the scale of hundreds or thousands of dimensions, with the largest dataset used in testing was the Tara oceans dataset

(Sunagawa et al., 2015), at 35,650 dimensions. Modifications to the server might allow for a greater capacity for dimensionality, but this is the scale the project has worked at.

On the interactivity front, HieraViz keeps the user in the loop throughout the exploration process, allowing for the user to explore and manipulate the data themselves, with multiple different ways to select and filter dimensions and samples into more manageable subsets. The user can make iterative changes response to updated views without needing to reload the dataset or the tool, and can reload the full dataset should they need to explore different options. The main limiting factor in the current iteration is that base dataset needs to be edited outside of the tool, and any changes that cannot be handled by filtering or changing which sample groups the data is divided into would require the tool to be reloaded.

Paper Title	Method	Dimensionality	Interactivity & User Engagement
HieraViz	Hierarchical data exploration	36550	Iterative exploration, Filter Metrics
Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data (Tatu et al., 2012)	Subspace Clustering	44	Iterative exploration
Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces (Jackle et al., 2018)	Subspace Clustering	8	Iterative exploration
Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data Zhou et al. (2016)	Subspace Clustering, Dimensional Reconstruction	221	Iterative exploration
Interactive Dimensionality Reduction Through User-defined Combinations of Quality Metrics (Johansson and Johansson, 2009)	Quality Metrics	100	Iterative exploration, Quality Metrics
Visual Exploration of Microbial Populations (Fernstad et al., 2011)	Quality Metrics	227	Iterative exploration, Quality Metrics
Brushing Dimensions – A Dual Visual Analysis Model for High-dimensional Data Turkay et al. (2011)	Brushing linked views	7129	Iterative exploration
SeekAView: An intelligent dimensionality reduction strategy for navigating high-dimensional data spaces Krause et al. (2017)	Subspace exploration	147	Iterative exploration, Quality Metrics
High-dimensional data analysis with subspace comparison using matrix visualization Wang et al. (2019)	Subspace exploration	20	Iterative exploration
Representative factor generation for the interactive visual analysis of high-dimensional dataTurkay et al. (2012)	Representative factor generation	357	Iterative exploration
Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis Turkay et al. (2017)	Progressive computations	77	Iterative exploration

Table 7.4 Comparison of HieraViz to different tools, covering the method used, the highest dimensionality mentioned and how the user interacted and engaged with the method

Chapter 8. Conclusion

8.1. Thesis Summary

The increasing scale of dimensionality demands methods better catered towards supporting its exploration. Older methods designed for lower scales of high dimensionality struggle to handle this increasing scale, from issues relating to the sparsity of data from the curse of dimensionality, screen limitations and human limitations in understanding high dimensionality.

Automated methods of processing high dimensional datasets into lower dimensional representations have their place, for when the analysis question is well defined and the user knows what they are looking for. But in cases where the question is unknown or broad, these methods without support inhibit the exploration process.

The objective of this project was the creation of a method to support the exploration and analysis of this high dimensional data. In particular, the focus was on providing the full dataset for users to explore and manipulate on the fly, being able to selected data subsets based on their domain knowledge, rather than relying on automated reduction to determine what data is shown. ’Omics data has been the main focus of the project, though the methods are adaptable to other forms of high dimensional hierarchical data.

From the project, the work into evaluating large and small hierarchical structures was published in the 24th International Conference Information Visualisation (IV) (Macquisten et al., 2020), with an updated version with expanded content covering the use of Sunburst chart and Treemap as a linked large and small hierarchical view to demonstrate how they could be used to explore high dimensional data, published as a chapter (Macquisten et al., 2022) in the book Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery (Kovalerchuk et al., 2022).

8.2. Research Questions

Addressing the first research question, *"To address high dimensionality challenges by investigating solutions using hierarchical visualization methods"*, a online user study in Chapter 4 was conducted to compare five different hierarchical visualization methods, Treemap, Icicle

Plot, Sunburst Chart, Circular Treemap and Bubble Treemap, for their ability to represent underlying data features in both large and small hierarchical structures. The results determined that for large structures, Sunburst Charts and Circular Treemaps had the best results, While for small structures, Treemaps, Icicle Plots and Circular Treemaps. This was the first study comparing these five methods together, while previous studies had compared some combinations of these methods, their main focus was on the methods hierarchical structure, while the study presented here focused on the representation of underlying data features.

Based on the study results, the project went on to use a combination of Sunburst Chart and Treemap to display high dimensional hierarchical datasets, as shown in Chapter 7.1.4. With the Sunburst Chart displaying either the full dataset or a filtered subset, while the Treemap displayed the subset of the currently selected node in the Sunburst, along with its immediate child nodes.

For the second research question: "*To enable efficient exploration of extremely high dimensional data by utilising approaches of visual guidance*" the application of the methods determined through the user studies alongside other methods covered in chapter 6.1 address this question. The combination of Sunburst and Treemap provides the basis for exploration, this is supported through the group separation colour metric, guiding exploration through displaying if nodes have well separated sample groups or not.

This is augmented with options to filter the dataset based on abundance and prevalence, alongside being able to filter out individual sample groups. Additional views support the exploration and analysis process, with the Analysis view providing a zoomed in of the currently selected treemap node, the Parallel Coordinates plot showing the current subset of data entities with sample values for each dimension, while the Multiple Small plots view allows for the comparison of different combinations of sample groups side by side.

Addressing the third research question: "*To design novel visualization to enable representation of multivariate data features in aggregated data*", the project developed the method Density Box plots, the combination of multiple overlaid Density Distribution plots and semi-overlaid Box plots, with each set of plots belonging to a single sample group.

Two online studies were conducted to evaluate the Density Box plot against other methods in their ability to compare multivariate statistical attributes. The first study in Chapter 5.2 compared two variations of Density Box plot against each other, with differences being in the level of complexity displayed in their box plot components. This study yielded no significant results, the methods likely being too similar.

The second study in Chapter 5.3 compared the complex version of the Density box plot, against Box plots and Density distribution plots. This experiment did yield significant results, with Density Box plot preforming better than Box plots and Density Distribution plots, while Box plots performed better than Density Distribution plots.

Following this, in chapter the project went on to use the Density Box plots within the nodes of the Treemap and with the Analysis view. These would show the aggregated values of different sample groups for the data subset within the node they were displayed.

8.3. Limitations

The core limitation of all the studies conducted was their online nature, this took away a degree of control that would have been present in a physical study, requiring additional measures to ensure the studies were conducted properly. The tools present in Gorilla and Prolific allowed for a great deal of control over how the experiment ran and who was able to participate, which mitigated some of the issues from running these studies online.

Additionally, the first study presented in chapter 4 was conducted with only 17 participants, so the sample size is slightly small. Ideally the project was hoping for at least 20, but had issues gathering participants. The other two studies did not have this issue, as rather than gathering participants manually, they were recruited through prolific, this however resulted in a wide range of different kinds of participants, some which needed to be manually removed from the study.

The study presented in chapter 5.2 was limited by comparing only two methods, both of which were too similar to one another. It may have just been prudent to have done the third study by itself with this prior study, but the lessons learned from this study streamlined preparations for the third study.

It was intend to test HieraViz with different types of ’omics data beyond microbiomics, however suitable datasets were not found in time. There was issues in converting these datasets into a usable hierarchical structure and limited time shifted the projects focus to other matters.

8.4. Future work

Following on from the user study presented in chapter 7.1, the project will look to implement the feedback generated through the study as and when it is appropriate. As mentioned some of the large changes include expanding the ways users can manipulate the sunburst chart to be able to better see small nodes.

Other considerations for changing the methods used to generate P values and for generating the group separation metric values. Ultimately the best solution may be to add several methods and let users switch between them based on their preferences. Additionally HieraViz could do with supporting documentation to assist users in understanding the methods included.

There is also further potential for additional publications of the work done throughout the project. The studies into methods of multivariate data features in aggregated data from Chapter 5 could be submitted as a conference paper in the same manner as the Evaluation of Hierarchical Visualization for Large and Small Hierarchies paper.

This was planned as a submission originally in 2021, however the sudden offer to expand the Evaluation of Hierarchical Visualization for Large and Small Hierarchies paper to be a chapter in the book Integrating Artificial Intelligence and Visualization for Visual Knowledge Discover was considered a better opportunity, so writing the paper on multivariate data features was put off to focus on the book chapter, but the potential remains for a future submission.

Chapters 6 and 7 can be rewritten as a technical paper covering HieraViz and the user study to be submitted to a visualization journal.

There are a few options for which conferences and journals these papers could be submitted to. There is the option of again submitting to the IEEE Vis conference. But alongside this, the EuroVis conference, IEEE Transactions on Visualization and Computer Graphics, and Information Visualization Journal are all options.

There is also the option of writing a manuscript covering HieraViz and its application to ‘omics data could be submitted to more bioinformatics focused journal, such as BMC BioInformatics or PLOS One.

For the project itself, there is the potential for the work presented within this Thesis to be expanded upon in a follow up project. If the project was put forward as another PhD, I would propose the following research questions:

What is the limit on dimensionality that can be supported within HieraViz and how might it be expanded?

Whilst in theory, the methods developed for displaying and exploring high dimensional datasets should be able to handle any size of dataset, it is limited by what the code and the machine it is running on can handle. There are also practical limitations of how many dimensions can be supported, as eventually it will be too difficult to represent within the visual structure, even with the current options for filtering.

The project would determine the limit of what the method can support and how the methods might be expanded to allow for these more complex datasets to be displayed and analysed. On the technical side, depending on how far in the future this project would take place, this may entail reprogramming HieraViz from JavaScript/D3 and Node.JS into something else.

How could the approach be expanded to support multi-omics analysis?

One of the early considerations of the project was to create a tool that could be used to support multi-omics analysis, but this was written off as it could have easily been a separate project unto itself. Regardless one future direction would most certainly be how the method could be used within multi-omics analysis, the interest in comparing different types of omics data is only going to increase as time goes on and there is going to be an increasing demand for more tools that can enable exploring multi-omics data. This would also include exploring how many different kinds of hierarchical data could be supported within the approach, how many changes would need to be made to accommodate them, and should method support them at all?

How could HieraViz be expanded to support greater data manipulation?

Currently, a Sunburst Chart and Treemap are used as the core method to display and navigate datasets, supported by options to filter out parts of the data or sample groups, alongside additional views to show different aspects of the data. The new project would explore how this could be expanded, be it with more alternate views or expansions to the methods present. From the feedback of the user studies alone, there was a few suggestions on how this could be improved, such as allowing the sunburst chart to zoom in on individual levels to highlight smaller nodes that are difficult to see otherwise.

This also includes improvements to the pre-processing stage of the tool, which doesn't allow for direct changes to the dataset without re-uploading to the server. Having a way to handle CSV files and directly edit them within the tool, would allow for changes to be made by the user without having to reload everything. This may also including exploring how the tool would link to other workflows, specifically in expanding the pre-processing step to allow for the outputs of other tools that handle the conversion of raw data into formatted data, to be used as an input for HieraViz directly.

Appendix A. Appendix

A.1. HieraViz: User Study form

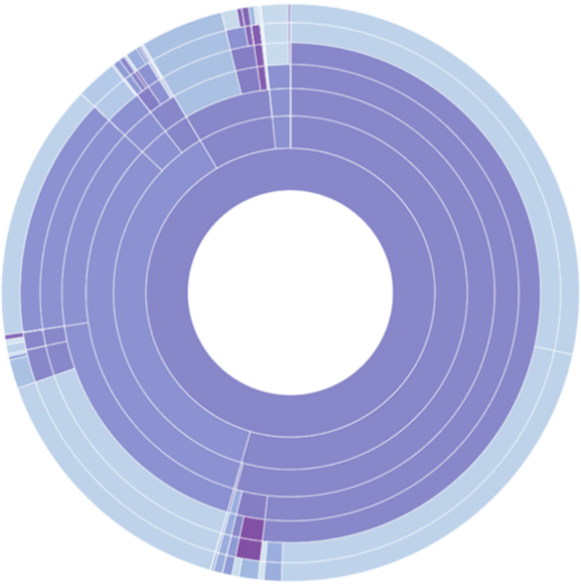
The goal of this study is to evaluate the tool HieraVis, a JavaScript based webapp designed to support the exploration and visual analysis of high dimensional hierarchically structured data. The first phase of the study will consist of a demonstration of HieraViz functionality. While the second phase is a semi-structured interview and discussion where you will have a chance to test out the tool yourself and give feedback. Presented here are a list of visualization heuristics that you will be asked to use to evaluate the tool, each heuristic is measured through a Likert scale. A description of each component is listed below alongside its heuristics, each component will also be explained during the demonstration. While the discussion is recorded, if you wish to provide any additional written feedback, there is space below each heuristic.

Exploration view

The primary exploration and analysis of the high dimensional data is performed in a pair of coordinated hierarchical views, one for overview, which shows the full hierarchical structure of the dataset, and one for detail, showing the local structure of the currently selected data point.

Sunburst

The overview method shows the full hierarchical structure of the dataset using Sunburst. The root of the hierarchical structure is show at the centre of the sunburst, with child nodes expanding outward, with outermost nodes representing species.



Heuristic 1. Overview – The Sunburst Hierarchical structure is intuitive

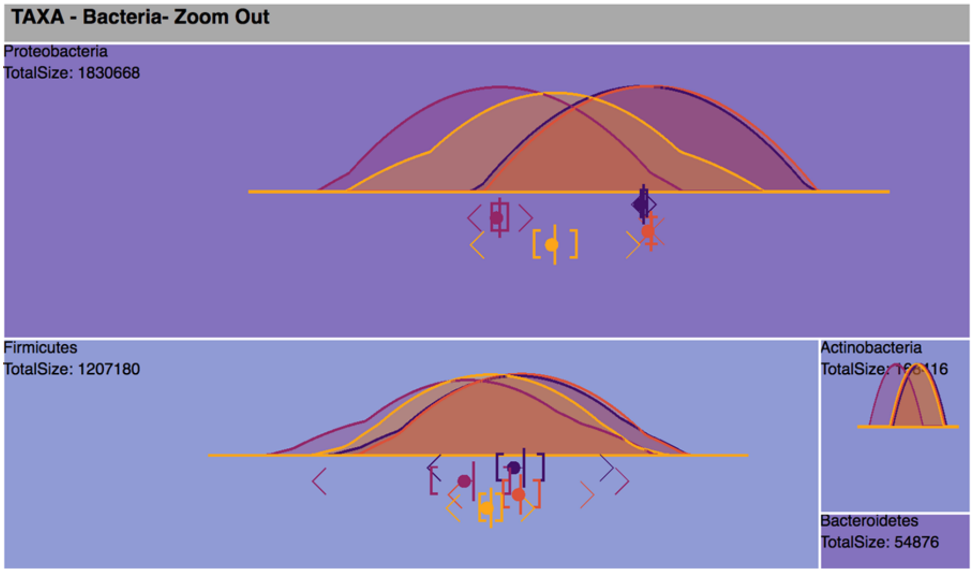
Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Heuristic 2. Overview – The Sunburst highlights taxon of interest to guide exploration.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Treemap

Treemap is used for the details view, the bar at the top shows the name of the currently selected taxon, alongside the path to it. Below that, the child taxon of the currently selected node are show, sized relative to their abundance values.

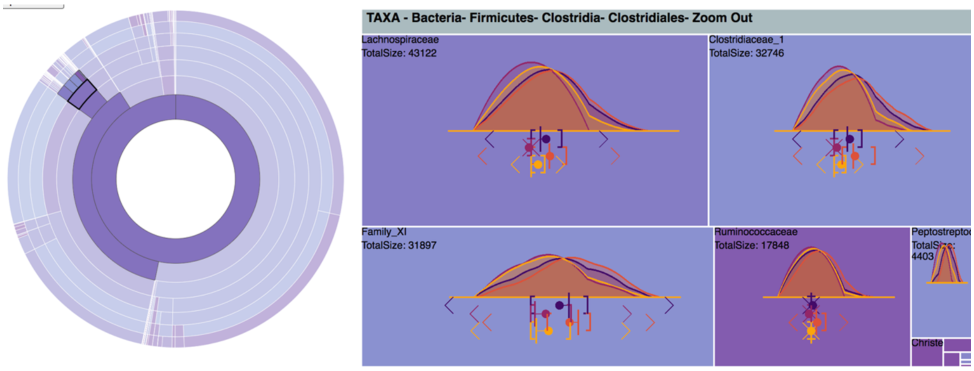


Heuristic 3. Zoom/filter – The Treemaps local hierarchical structure is intuitive.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Combined view

The Overview and Details views are designed to complement each other. The overview aims to provide general understanding of patterns of potential interest in the hierarchy. This provides a guide for exploration and allows fast identification of taxa that may be particularly interesting to explore in more detail. A taxon of interest can then be selected in the overview, resulting in its path in the hierarchical structure being highlighted. The details view displays only the taxon selected in the Overview and its child taxa. This provides a less cluttered view where the relationships and differences between child taxa can be better understood and patterns easier compared, since space is available for each individual child in the data subset.



Heuristic 4. Concretize Relationships – Selecting a node in either view loads it into the other view and highlights the path in the sunburst, the relationship between the two views is intuitive.

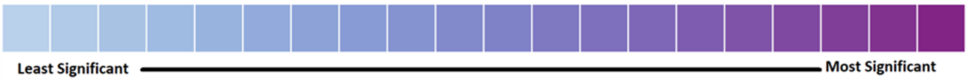
Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Heuristic 5. The size of the views and nodes is sufficient

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Multivariate sample group comparison

Nodes in both views are coloured using a series of interestingness metrics, In the colour scheme, darker colours represent higher interestingness in terms of higher abundance/prevalence/correlation, clearer separation of abundance/prevalence between sample groups, or strong correlation or high similarity of biological entities within a taxon. Correspondingly, lighter colours represent lower interestingness values.

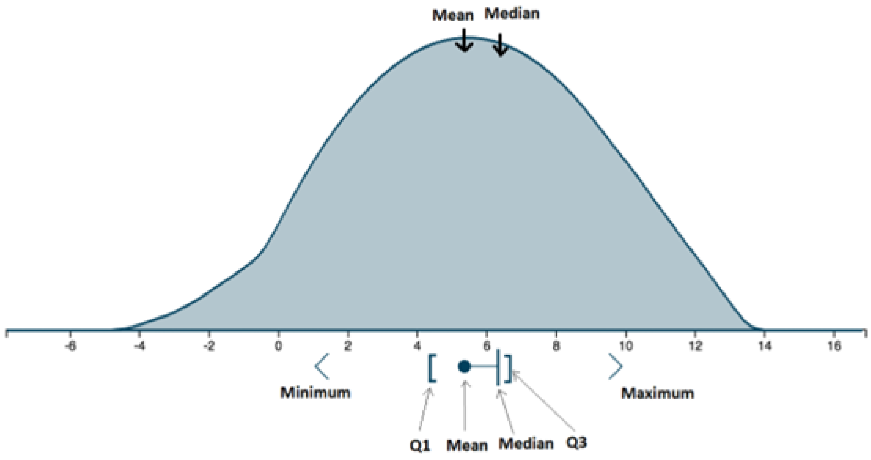


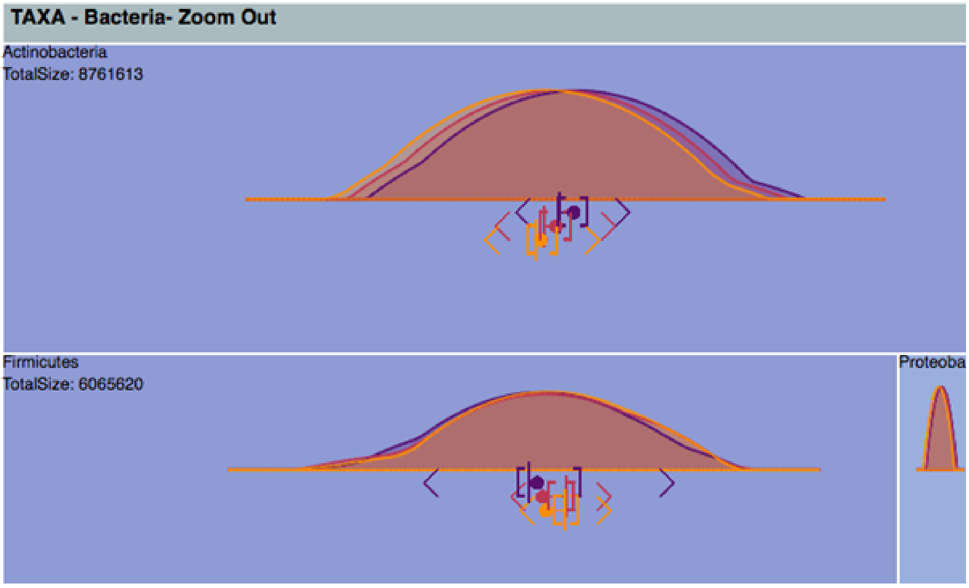
Heuristic 6. The interestingness metric provides a good guide to which nodes are interesting to look at.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Density Box Plot Glyph

Density box plots are a distributions glyph used to compare multivariate sample data within a limited space. The top component of the glyph is a density distribution plot to show the overall distribution of the sample data. The bottom component of the glyph consists of an abstracted box plot. Each sample group has its own Density Box plot, which is overlaid on the same axis to support comparison between sample groups, individual glyphs can be filtered out or focused on.





Heuristic 7. Both the Density Box plots and Colour separation metric shows differences between sample groups within a single node and between nodes. These views complement each other well in determining why a node is interesting.

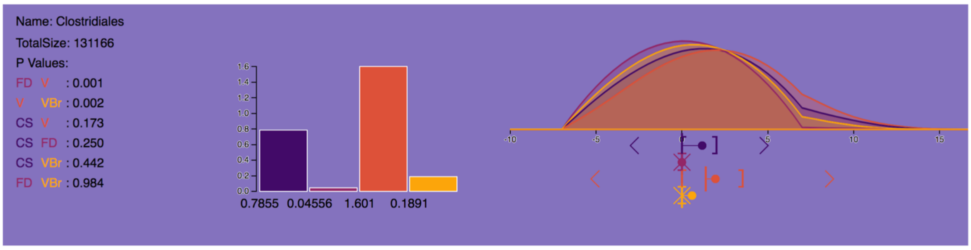
Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Heuristic 8. The use of multiple overlaid glyphs supports the ease of sample group comparison?

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Analysis view

This view provides expanded details on the currently selected node, that otherwise would not fit in the variably sized treemap. Alongside the details and density box plot shown in a normal treemap node, there is a list of P values for each sample group combination for that node, alongside a relative abundance histogram.



Name: Clostridiales
TotalSize: 131166
P Values:
FD V : 0.001
V VBr : 0.002
CS V : 0.173
CS FD : 0.250
CS VBr : 0.442
FD VBr : 0.984

Heuristic 9. Multivariate Explanation – The relationship between histogram values and colour interestingness metric (background colour) is intuitive.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Heuristic 10. Multivariate Explanation - The relationship between P values and colour interestingness metric (background colour) is intuitive.

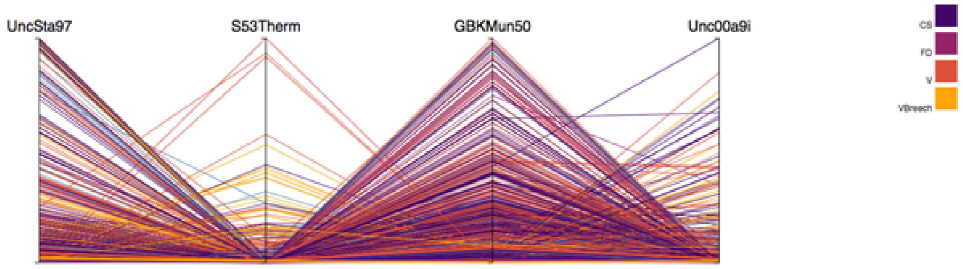
Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Heuristic 11. Multivariate Explanation - The relationship between histogram values and Density box plot is intuitive.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Parallel Coordinates

The Parallel Coordinates view shows a subset of the species of the taxon that is currently selected in the Sunburst and displayed in the Treemap. With samples as polylines and the selected species as individual axes, the Parallel Coordinates shows the relative abundance profiles of samples across the subset. The plot is updated as the selected subset is changed within the hierarchical views. The polylines in the Parallel Coordinates are coloured based on sample groups, to support comparison of sample group profiles.

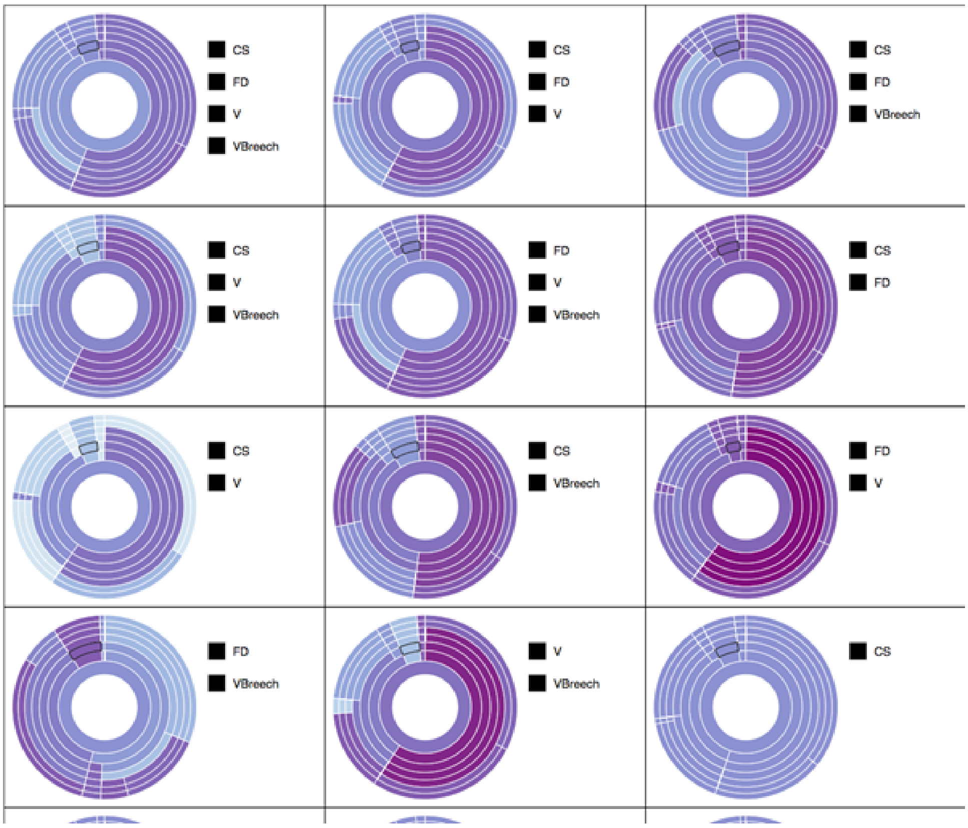


Heuristic 12. Details – The relationship between axes of Parallel Coordinates plot and the selected taxon in the other views is intuitive.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Multiple Small Plots

The Multiple Small Plots view presents a sunburst chart for each possible combination of sample groups side by side, allowing for a quick comparison of patterns in the different sample subsets without manually checking each one. The Multiple Small Plots has the same set of filters as the normal view, on top of only showing subsets of sample combinations.

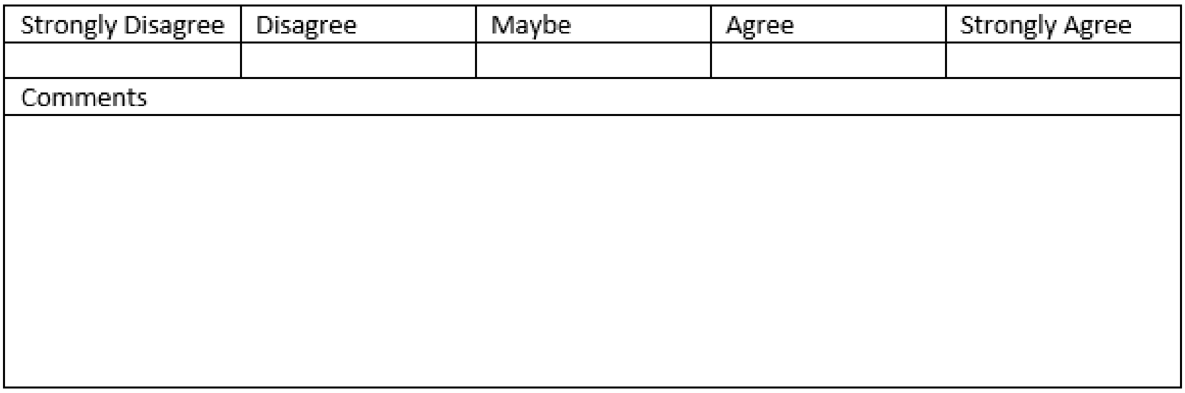


Heuristic 13. Overview – Comparing and interacting with the multiple sunburst charts provides a useful starting point for exploration.

Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

Colour options

Both the interestingness metric and the sample groups have several sets of separate colour options Heuristic 14. Was able to find colour combinations that were usable.



Strongly Disagree	Disagree	Maybe	Agree	Strongly Agree
Comments				

References

Albuquerque, G., Eisemann, M., Lehmann, D. J., Theisel, H., and Magnor, M. (2010). Improving the visual analysis of high-dimensional datasets using quality measures. In *VAST 10 - IEEE Conference on Visual Analytics Science and Technology 2010, Proceedings*.

Anand, A., Wilkinson, L., and Dang, T. N. (2012). Visual pattern discovery using random projections. In *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*.

Andrews, K., Osmić, M., and Schagerl, G. (2015). Aggregated Parallel Coordinates: Integrating Hierarchical Dimensions into Parallel Coordinates Visualisations. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, pages 1–4, New York, New York, USA. ACM Press.

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., and Evershed, J. K. (2020). Gorilla in our midst: An online behavioral experiment builder. *Behavior Research Methods*, 52(1):388–407.

Baldonado, M. Q., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for using multiple views in information visualization. *Proceedings of the Workshop on Advanced Visual Interfaces*, pages 110–119.

Balzer, M. and Deussen, O. (2005). Voronoi Treemaps. In *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pages 49–56.

Baumgartner, C., Plant, C., Kailing, K., Kriegel, H. P., and Kröger, P. (2004). Subspace selection for clustering high-dimensional data. In *Proceedings - Fourth IEEE International Conference on Data Mining, ICDM 2004*, pages 11–18.

Becker, R. A. and Cleveland, W. S. (1987). Brushing scatterplots. *Technometrics*.

Behrisch, M., Blumenschein, M., Kim, N. W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H., Schreck, T., Weiskopf, D., and Keim, D. A. (2018). Quality Metrics for Information Visualization.

Bersanelli, M., Mosca, E., Remondini, D., Giampieri, E., Sala, C., Castellani, G., and Milanesi, L. (2016). Methods for the integration of multi-omics data: mathematical aspects. *BMC Bioinformatics*, 17:15.

Bertini, E., Tatu, A., and Keim, D. (2011). Quality metrics in high-dimensional data visualization: An overview and systematization.

Bik, H. M. and Inc., P. I. (2014). Phinch: An interactive, exploratory data visualization framework for -Omic datasets. *bioRxiv*, page 009944.

Blanch, R. and Lecolinet, E. (2007). Browsing Zoomable Treemaps: Structure-Aware Multi-Scale Navigation Techniques. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1248–1253.

Blumenschein, M., Zhang, X., Pomerenke, D., Keim, D. A., and Fuchs, J. (2020). Evaluating Reordering Strategies for Cluster Identification in Parallel Coordinates. Technical Report 3.

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C., Al-Ghalith, G. A., Alexander, H., Alm, E. J., Arumugam, M., Asnicar, F., Bai, Y., Bisanz, J. E., Bittinger, K., Brejnrod, A., Brislawn, C. J., Brown, C. T., Callahan, B. J., Caraballo-Rodríguez, A. M., Chase, J., Cope, E., Silva, R. D., Dorrestein, P. C., Douglas, G. M., Durall, D. M., Duvallet, C., Edwardson, C. F., Ernst, M., Estaki, M., Fouquier, J., Gauglitz, J. M., Gibson, D. L., Gonzalez, A., Gorlick, K., Guo, J., Hillmann, B., Holmes, S., Holste, H., Huttenhower, C., Huttley, G., Janssen, S., Jarmusch, A. K., Jiang, L., Kaehler, B., Kang, K. B., Keefe, C. R., Keim, P., Kelley, S. T., Knights, D., Koester, I., Kosciolek, T., Kreps, J., Langille, M. G., Lee, J., Ley, R., Liu, Y.-X., Loftfield, E., Lozupone, C., Maher, M., Marotz, C., Martin, B. D., McDonald, D., McIver, L. J., Melnik, A. V., Metcalf, J. L., Morgan, S. C., Morton, J., Naimey, A. T., Navas-Molina, J. A., Nothias, L. F., Orchanian, S. B., Pearson, T., Peoples, S. L., Petras, D., Preuss, M. L., Pruesse, E., Rasmussen, L. B., Rivers, A., Michael S Robeson, I., Rosenthal, P., Segata, N., Shaffer, M., Shiffer, A., Sinha, R., Song, S. J., Spear, J. R., Swafford, A. D., Thompson, L. R., Torres, P. J., Trinh, P., Tripathi, A., Turnbaugh, P. J., Ul-Hasan, S., Hooft, J. J. v. d., Vargas, F., Vázquez-Baeza, Y., Vogtmann, E., Hippel, M. v., Walters, W., Wan, Y., Wang, M., Warren, J., Weber, K. C., Williamson, C. H., Willis, A. D., Xu, Z. Z., Zaneveld, J. R., Zhang, Y., Zhu, Q., Knight, R., and Caporaso, J. G. (2018). QIIME 2: Reproducible, interactive, scalable, and extensible microbiome data science.

Borgo, R., Kehrer, J., Chung, D. H. S., Maguire, E., Laramee, R. S., Hauser, H., Ward, M., and Chen, M. (2013). Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. *Eurographics State of the Art Reports*, pages 39–63.

Borland, D. and Taylor, R. M. (2007). Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications*, 27(2):14–17.

Brendan D. Gregg (2011). Flame Graphs.

Brewer, C. A. (1999). Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics, American Statistical Association*, pages 55–60. American Statistical Association Alexandria, VA.

Bruls, M., Huizing, K., and van Wijk, J. J. (2000). Squarified Treemaps. pages 33–42.

Cakmak, E., Schäfer, H., Buchmüller, J., Fuchs, J., Schreck, T., Jordan, A., and Keim, D. A. (2020). MotionGlyphs : Visual Abstraction of Spatio-Temporal Networks in Collective Animal Behavior. *Computer Graphics Forum*, 39(3).

Cawthon, N. and Moere, A. V. (2007). The Effect of Aesthetic on the Usability of Data Visualization. Technical report.

Chen, S. X. (2000). Probability density function estimation using gamma kernels. *Annals of the Institute of Statistical Mathematics*, 52(3):471–480.

Cuenca, E., Sallaberry, A., Wang, F. Y., and Poncelet, P. (2018). MultiStream: A Multiresolution Streamgraph Approach to Explore Hierarchical Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3160–3173.

Davies, D. L. and Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

de Carvalho Pagliosa, L. and Telea, A. C. (2019). RadViz++: Improvements on radial-based visualizations. *Informatics*, 6(2):16.

Dinkla, K., Westenberg, M., Timmerman, H., van Hijum, S., and van Wijk, J. (2011). Comparison of Multiple Weighted Hierarchies: Visual Analytics for Microbe Community Profiling. *Computer Graphics Forum*, 30(3):1141–1150.

Dunkler, D., Sánchez-Cabo, F., and Heinze, G. (2011). Statistical analysis principles for Omics data. *Methods in molecular biology (Clifton, N.J.)*, 719:113–131.

Dunn, J. C. (2008). Well-Separated Clusters and Optimal Fuzzy Partitions. *https://doi.org/10.1080/01969727408546059*, 4(1):95–104.

Eick, S. G. and Karr, A. F. (2002). Journal of Computational and Graphical Statistics Visual Scalability Visual Scalability. *Journal of Computational and Graphical Statistics*, 11(1):22–43.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*.

Espadoto, M., Martins, R. M., Kerren, A., Hirata, N. S. T., and Telea, A. C. (2019). Towards a Quantitative Survey of Dimension Reduction Techniques. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Fernstad, S. J., Johansson, J., Adams, S., Shaw, J., and Taylor, D. (2011). Visual exploration of microbial populations. In *2011 IEEE Symposium on Biological Data Visualization (BioVis).*, pages 127–134. IEEE.

Fernstad, S. J., Shaw, J., and Johansson, J. (2013). Quality-based guidance for exploratory dimensionality reduction. *Information Visualization*, 12(1):44–64.

Fischer, F., Fuchs, J., and Mansmann, F. (2012). Clockmap: Enhancing circular treemaps with temporal glyphs for time-series data. Technical report.

Fischer, R. A. and Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*.

Forsell, C. and Johansson, J. (2010). An heuristic set for evaluation in Information Visualization. *Proceedings of the Workshop on Advanced Visual Interfaces AVI*, pages 199–206.

Friendly, M. and Denis, D. (2005). The early origins and development of the scatterplot.

Fuchs, J., Fischer, F., Mansmann, F., Bertini, E., and Isenberg, P. (2013). Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 3237–3246, New York, New York, USA. ACM Press.

Fuchs, J., Jäckle, D., Weiler, N., and Schreck, T. (2015). Leaf Glyph - Visualizing Multi-dimensional Data with Environmental Cues. In *Proceedings of the 6th International Conference on Information Visualization Theory and Applications*, pages 195–206. SCITEPRESS - Science and and Technology Publications.

Görtler, J., Schulz, C., Weiskopf, D., and Deussen, O. (2018). Bubble Treemaps for Uncertainty Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):719–728.

Graziano, A. M. and Raulin, M. L. (1993). *Research methods: A process of inquiry, 2nd ed.* HarperCollins College Publishers, New York, NY, US.

Grinstein, G., Trutschl, M., and Cvek, U. (2001). High-dimensional visualizations. *Proceedings of the Visual Data Mining Workshop, KDD*, pages 1–14.

Haisen Zhao and Lu, L. (2015). Variational circular treemaps for interactive visualization of hierarchical data. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 81–85. IEEE.

Halford, G. S., Baker, R., McCredden, J. E., and Bain, J. D. (2005). How many variables can humans process? *Psychological Science*.

Hebrard, M. and Taylor, T. D. (2016). Metatreemap: An alternative visualization method for displaying metagenomic phylogenic trees. *PloS one*, 11(6):e0158261.

Hintze, J. L. and Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. *American Statistician*, 52(2):181–184.

Hoffman, P., Grinstein, G., Marx, K., Grosse, I., and Stanley, E. (1997). DNA visual and analytic data mining. In *Proceedings of the IEEE Visualization Conference*.

Hoffman, P., Grinstein, G., and Pinkney, D. (1999). Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in conjunction with the 8th ACM Internation Conference on Information and Knowledge Management, NPIVM 1999*.

Huson, D. H., Auch, A. F., Qi, J., and Schuster, S. C. (2007). MEGAN analysis of metagenomic data. *Genome Research*, 17(3):377–386.

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*.

Jackle, D., Hund, M., Behrisch, M., Keim, D. A., and Schreck, T. (2018). Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces. Technical report.

Jenny, B. and Kelso, N. V. (2006). Designing maps for the colour-vision impaired. *Bulletin of the Society of Cartographers*, 40(1-2):9–12.

Johansson, S. and Johansson, J. (2009). Interactive dimensionality reduction through user-defined combinations of quality metrics. In *IEEE Transactions on Visualization and Computer Graphics*.

Johansson, Fernstad, S., Macquisten, A., Berrington, J., Embleton, N., and Stewart, C. (2020). Quality Metrics to Guide Visual Analysis of High Dimensional Genomics Data. *https://eprints.ncl.ac.uk*.

Johnson, B. (1993). Treemaps: Visualizing Hierarchical and Categorical Data. *Journal of Experimental Psychology: General*.

Johnstone, I. M. and Titterington, D. M. (2009). Statistical challenges of high-dimensional data.

Jolliffe, I. T. (2002). *Principal Component Analysis*.

Kovalerchuk, B., Nazemi, K., Andonie, R., Datia, N., and Banissi, E. (2022). Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery. 1014.

Krause, J., Dasgupta, A., Fekete, J. D., and Bertini, E. (2017). SeekAView: An intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. In *IEEE Symposium on Large Data Analysis and Visualization 2016, LDAV 2016 - Proceedings*.

Kruskal, J. B. and Landwehr, J. M. (1983). Statistical computing: Icicle plots: Better displays for hierarchical clustering. *American Statistician*, 37(2):162–168.

Kuznetsova, I., Lugmayr, A., and Holzinger, A. (2017). Visualization methods of hierarchical biological data: A survey and review. In *International series on information systems and management in creative eMedia*.

Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, pages 401–408, New York, New York, USA. ACM Press.

Long, L. K., Hui, L. C., Fook, G. Y., and Zainon, W. M. N. W. (2017). A Study on the Effectiveness of Tree-Maps as Tree Visualization Techniques. In *Procedia Computer Science*, volume 124, pages 108–115.

Macquisten, A., Smith, A. M., and Fernstad, S. J. (2022). Hierarchical Visualization for Exploration of Large and Small Hierarchies. pages 587–612.

Macquisten, A., Smith, A. M., and Johansson Fernstad, S. (2020). Evaluation of Hierarchical Visualization for Large and Small Hierarchies. *Proceedings of the International Conference on Information Visualisation,*, 2020-Septe:166–173.

McNabb, L. and Laramee, R. S. (2019). Multivariate Maps-A Glyph-placement algorithm to support multivariate geospatial visualization. *Information (Switzerland).*

McNally, C. P., Eng, A., Noecker, C., Gagne-Maynard, W. C., and Borenstein, E. (2018). BURRITO: An interactive multi-omic tool for visualizing taxa-function relationships in microbiome data. *Frontiers in Microbiology*.

Munzner, T. (2018). *Visualization Analysis and Design*.

Muramalla, S., Altarawneh, R., Humayoun, S. R., Moses, R., Panis, S., and Ebert, A. (2017). Radial vs. rectangular: evaluating visualization layout impact on user task performance of hierarchical data. *IADIS International Journal on Computer Science and Information Systems*, 12(2):17–31.

NHGRI (2021). The Cost of Sequencing a Human Genome - National Human Genome Research Institute (NHGRI).

Nicholas, H. M., Liebold, B., Pietschmann, D., Ohler, P., and Rosenthal, P. (2017). Hierarchy Visualization Designs and their Impact on Perception and Problem Solving Strategies. *Proceedings of the International Conference on Advances in Computer-Human Interactions*, (c):93–101.

Nuñez, J. R., Anderton, C. R., and Renslow, R. S. (2018). Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PLOS ONE*, 13(7):e0199239.

Ondov, B. D., Bergman, N. H., and Phillippy, A. M. (2011). Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics*, 12(1):385.

Opach, T. and Rød, J. K. (2018). Augmenting the usability of parallel coordinate plot: The polyline glyphs. *Information Visualization*.

Oppenheim, A. N. and Torgerson, W. (1961). Theory and Methods of Scaling. *The British Journal of Sociology*.

O'Donnell, R., Dix, A., and Ball, L. J. (2007). Exploring the PieTree for Representing Numerical Hierarchical Data. In *People and Computers XX — Engage*.

Pearson, K. (1895). Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:343–414.

Potter, K., Kniss, J., Riesenfeld, R., and Johnson, C. R. (2010). Visualizing summary statistics and uncertainty. *Computer Graphics Forum*.

Qu, Z. and Hullman, J. (2018). Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):468–477.

Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *https://doi.org/10.1214/aoms/1177728190*, 27(3):832–837.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Sboner, A., Mu, X., Greenbaum, D., Auerbach, R. K., and Gerstein, M. B. (2011). The real cost of sequencing: higher than you think! *Genome Biology*, 12(8):125.

Schulz, H.-J. (2011). Treevis.net: A Tree Visualization Reference. *IEEE Computer Graphics and Applications*, 31(6):11–15.

Schulz, H. J., Hadlak, S., and Schumann, H. (2011). The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411.

Schulz, H. J. and Schumann, H. (2006). Visualizing graphs - A generalized view. In *Proceedings of the International Conference on Information Visualisation*, pages 166–173. IEEE.

Sedlmair, M., Meyer, M., and Munzner, T. (2012). Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *The Craft of Information Visualization*, pages 336–343. IEEE Comput. Soc. Press.

Shneiderman, B. and Ben (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99.

Smith, N. and van der Walt, S. (2015). Matplotlib Colormaps.

Soares, A. G. M., dos Santos, D. H., Barbosa, C. L. R., Goncalves, A. S., dos Santos, C. G. R., Meiguins, B. S., and Miranda, E. T. C. (2018). Visualizing Multidimensional Data in Treemaps with Adaptive Glyphs. In *2018 22nd International Conference Information Visualisation (IV)*, pages 58–63. IEEE.

Stasko, J., Catrambone, R., Guzdial, M., and McDonald, K. (2000). An evaluation of space-filling information visualizations for depicting hierarchical structures. *International journal of human-computer studies*, 53(5):663–694.

Stasko, J. and Zhang, E. (2000). Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*, pages 57–65. IEEE.

Sunagawa, S., Coelho, L. P., Chaffron, S., Kultima, J. R., Labadie, K., Salazar, G., Djahanschiri, B., Zeller, G., Mende, D. R., Alberti, A., Cornejo-Castillo, F. M., Costea, P. I., Cruaud, C., d'Ovidio, F., Engelen, S., Ferrera, I., Gasol, J. M., Guidi, L., Hildebrand, F., Kokoszka, F., Lepoivre, C., Lima-Mendez, G., Poulain, J., Poulos, B. T., Royo-Llonch, M., Sarmento, H., Vieira-Silva, S., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., coordinators, T. O., Bowler, C., Vargas, C. d., Gorsky, G., Grimsley, N., Hingamp, P., Iudicone, D., Jaillon, O., Not, F., Ogata, H., Pesant, S., Speich, S., Stemmann, L., Sullivan, M. B., Weissenbach, J., Wincker, P., Karsenti, E., Raes, J., Acinas, S. G., and Bork, P. (2015). Structure and function of the global ocean microbiome. *Science*, 348(6237):1261359.

Tatu, A., Maaß, F., Färber, I., Bertini, E., Schreck, T., Seidl, T., and Keim, D. (2012). Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*, pages 63–72.

Tipney, H. J., Leach, S. M., Feng, W., Spritz, R., Williams, T., and Hunter, L. (2009). Leveraging existing biological knowledge in the identification of candidate genes for facial dysmorphology. In *BMC bioinformatics*, volume 10 Suppl 2.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley Pub. Co., Reading Mass.

Turkay, C., Filzmoser, P., and Hauser, H. (2011). Brushing Dimensions - A Dual Visual Analysis Model for High-Dimensional Data. *IEEE transactions on visualization and computer graphics*, 17(12):2591–2599.

Turkay, C., Kaya, E., Balcisoy, S., and Hauser, H. (2017). Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*.

Turkay, C., Lundervold, A., Lundervold, A. J., and Hauser, H. (2012). Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*.

Van Der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2625.

van Wijk, J. J. and van de Wetering, H. (1999). Cushion treemaps: visualization of hierarchical information. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 73–78.

Wagner, J., Chelaru, F., Kancherla, J., Paulson, J. N., Zhang, A., Felix, V., Mahurkar, A., Elmqvist, N., and Corrada Bravo, H. (2018). Metaviz: interactive statistical and visual analysis of metagenomic data. *Nucleic Acids Research*, 46(6):2777–2787.

Wall, E., Agnihotri, M., Matzen, L., Divis, K., Haass, M., Endert, A., and Stasko, J. (2019). A heuristic approach to value-driven evaluation of visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):491–500.

Wang, J., Liu, X., and Shen, H. W. (2019). High-dimensional data analysis with subspace comparison using matrix visualization. *Information Visualization*, 18(1):94–109.

Ward, M. O. (2002). A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization. *Information Visualization*, 1(3-4):194–210.

Ward, M. O. (2007). Multivariate Data Glyphs: Principles and Practice. In *Handbook of Data Visualization*, pages 179–198. Springer Berlin Heidelberg.

Wattenberg, M. (2003). Visualizing the stock market.

Węglarczyk, S. (2018). Kernel density estimation and its application.

Wetzel, K. (2003). Pebbles-using circular treemaps to visualize disk usage. *URL: http://lip. sourceforge. net/ctreemap. html*, 2.

Wood, R. W. (1906). XXIII. Fish-eye views, and vision under water. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 12(68):159–162.

Woodburn, L., Yang, Y., and Marriott, K. (2019). Interactive Visualisation of Hierarchical Quantitative Data: An Evaluation. In *2019 IEEE Visualization Conference, VIS 2019*, pages 96–100.

Wright, E. M. and Bellman, R. (1962). Adaptive Control Processes: A Guided Tour. *The Mathematical Gazette*, 46(356):160.

Yang, J., Ward, M. O., and Rundensteiner, E. A. (2002). InterRing: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, volume 2002-Janua, pages 77–84. IEEE Comput. Soc.

Ying-Huey Fua, Ward, M., and Rundensteiner, E. (2003). Hierarchical parallel coordinates for exploration of large datasets.

Zhou, F., Li, J., Huang, W., Zhao, Y., Yuan, X., Liang, X., and Shi, Y. (2016). Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In *IEEE Pacific Visualization Symposium*, volume 2016-May, pages 128–135. IEEE.

Zhou, Z., Ye, Z., Yu, J., and Chen, W. (2018). Cluster-aware arrangement of the parallel coordinate plots. *Journal of Visual Languages and Computing*, 46:43–52.