UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Wilson Estécio Marcílio Júnior

Toward Interpretable and Hierarchical Methods for Dimensionality Reduction

São José do Rio Preto
2023

Wilson Estécio Marcílio Júnior

Toward Interpretable and Hierarchical Methods for Dimensionality Reduction

Tese apresentada como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Orientador: Prof. Dr. Danilo Medeiros Eler

São José do Rio Preto
2023

M319t

Marcílio Júnior, Wilson Estécio
Toward Interpretable and Hierarchical Methods for Dimensionality Reduction / Wilson Estécio Marcílio Júnior. -- São José do Rio Preto, 2023
178 f. : il., tabs.

Tese (doutorado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Danilo Medeiros Eler

1. Dimensionality Reduction. 2. Unsupervised Learning. 3. Data Visualization. 4. Hierarchical Exploration. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Wilson Estécio Marcílio Júnior

Toward Interpretable and Hierarchical Methods for Dimensionality Reduction

Tese apresentada como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora

Prof. Dr. Danilo Medeiros Eler
UNESP – Câmpus de Presidente Prudente
Orientador

Profª. Drª. Maria Cristina Ferreira de Oliveira
USP - ICMC - São Carlos

Prof. Dr. Jorge Luis Poco Medina
FGV - EMAp - Rio de Janeiro

Prof. Dr. Wallace Correa de Oliveira Casaca
UNESP - Câmpus de Rosana

Prof. Dr. Milton Hirokazu Shimabukuro
UNESP – Câmpus de Presidente Prudente

Presidente Prudente
07 de fevereiro de 2023

Para minha mãe, Raquel Rocha.

AGRADECIMENTOS

Aos meus pais, Raquel e Wilson, agradeço por todo apoio que me deram para superar os desafios da vida acadêmica e acreditar que eu posso sempre mais—vocês são as pessoas mais especiais do mundo, muito obrigado por garantir tranquilidade em muitas das vezes que eu pensei que não iria conseguir. Aos meus irmãos, Larissa e Peterson, agradeço pela parceria de sempre e pelo interesse nos assuntos do meu trabalho—fiquem sabendo que meus sobrinhos serão todos evangelizados para seguirem carreira na pesquisa, muito obrigado pelas conversas e momentos de descontração durante esse período. À minha noiva, Karla, que esteve comigo a maior parte do tempo durante o desenvolvimento deste trabalho e, querendo ou não, teve de observar uma pessoa lidando com os altos e baixos da vida acadêmica—agradeço por todo apoio, e por mudar as perspectivas quando eu me encontrava em uma *rua sem saída*. Agradeço ao meu orientador, Danilo, pela parceria (faz dez anos desde a graduação) e pela ajuda a me tornar o pesquisador que sou hoje—agradeço muito pelas oportunidades e paciência em todos esses anos. Gostaria também de agradecer aos professores que contribuíram para as atividades de pesquisa (especialmente Fernando Paulovich, Rafael Martins e Rogério Garcia) e ao pessoal do laboratório pelas diversas conversas e momentos de descontração. Por fim, agradeço aos funcionários da FCT-UNESP, que indiretamente forneceram um ambiente confortável para desenvolvimento deste trabalho. Agradeço a Deus por estar ao meu lado todo o tempo.

Este trabalho foi financiado por: Fundação de Ampara à Pesquisa do Estado de São Paulo (FAPESP), processos #2018/17881-3, #2018/25755-8; e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), processo #88887.487331/2020-00.

"There will be time, there will be time

To prepare a face to meet the faces that you meet;

There will be time to murder and create,
And time for all the works and days of hands
That lift and drop a question on your plate;
Time for you and time for me,

And time yet for a hundred indecisions,
And for a hundred visions and revisions,
Before the taking of a toast and tea."

– **T.S. Eliot, The Love Song of J. Alfred Prufrock**

RESUMO

Análise Exploratória de Dados (EDA) é uma ferramenta valiosa para descobrir novos insights a partir de dados de alta dimensão. Os procedimentos de EDA ajudam cientistas e profissionais a entender a relação entre instâncias de dados e estruturas usando metáforas visuais que enfatizam informações sobre o objetivo da análise. Amplamente empregadas para análise exploratória, técnicas de Redução de Dimensionalidade mapeiam o relacionamento presente em espaços de alta dimensão para espaços de dimensão menor, normalmente no $\mathbb{R}^2$, e permitem descobertas científicas que vão do entendimento do comportamento de redes neurais à anotação de tipos celulares. Esta tese apresenta um conjunto de abordagens para extender o poder de análise utilizando técnicas de Redução de Dimensionalidade, com o objetivo principal de ajudar no entendimento dos embeddings gerados por técnicas não-lineares, além do estabelecimento de abordagens de exploração hierárquica para análise de acordo com a demanda de informações. Tais abordagens foram resultados das seguintes atividades de pesquisa principais: (i) seleção de representativos no espaço visual com preservação do contexto e estruturas mapeadas; (ii) técnicas para análise de *clustering* utilizando contrastividade, permitindo o entendimento das diferenças entre instâncias de dados para a formação de *clusters*; (iii) explicação de forma aditiva do *layout* gerado por técnicas não lineares; (iv) técnica de redução hierárquica de dimensionalidade para análise granular de conjuntos de dados, que permite a preservação das estruturas entre níveis hierárquicos.

Palavras-chave: Redução hierárquica de dimensionalidade. Interpretação. Análise contrastiva.

ABSTRACT

Exploratory Data Analysis (EDA) is a valuable tool for discovering new insights from high-dimensional data. EDA procedures help scientists and practitioners understand the relationship between data instances and structures using visual metaphors that emphasize information about the purpose of analysis. Widely used for exploratory analysis, Dimensionality Reduction techniques map the relationship present in high-dimensional spaces to smaller-dimensional spaces, usually in $\mathbb{R}^2$, and allow scientific discoveries that range from understanding the behavior of neural networks to the annotation of cell types. This thesis presents a set of approaches to extend the power of analysis using Dimensionality Reduction techniques, with the main objective of helping to understand the embeddings generated by non-linear techniques, in addition to establishing hierarchical exploration approaches for analysis according to the demand for information. Such approaches were the result of the following main research activities: (i) selection of representatives in the visual space with preservation of the context and mapped structures; (ii) techniques for analyzing *clustering* using contrastivity, allowing the understanding of differences between data instances for the formation of *clusters*; (iii) additive explanation of the *layout* generated by non-linear techniques; (iv) hierarchical dimensionality reduction technique for granular analysis of datasets, which allows the preservation of structures between hierarchical levels.

Keywords: Hierarchical dimensionality reduction. Interpretation. Contrastive analysis.

LIST OF FIGURES

Figure 1 – Defining the grid and selecting samples. (a) A grid with cells of size k by k is used to divide the $\mathbb{R}^2$ space; (b) Only cells with at least one data point are retrieved; (c) The resulting representative set after redundancy removal. 33

Figure 2 – Filtering redundant points. In this case, with window of size $\alpha = 3$. . . 34

Figure 3 – Initial projections of the datasets and their respective sampled sets according to the sampling techniques. The color of the circles depict the different classes in the dataset. 37

Figure 4 – *Redundancy* metric measures how the sampled set is redundant by seeing how each point correlates with each other. The higher the value, the lower a set is redundant. 38

Figure 5 – *Coverage* metric to assess if a sampled set is able to represent all data structures. The standard deviation is represented by error bars. Notice that, greater standard deviation indicates that some instances are far from a representative (an instance from sampled set). 39

Figure 6 – *Silhouette coefficient* of the techniques applied for clustering data. . . . 40

Figure 7 – Runtime of SADIRE algorithm. We can see that the algorithm takes more time for sampling from high-density embeddings. 41

Figure 8 – UMAP and t-SNE projections. 42

Figure 9 – Influence of k and α parameters on the returned representative set. . . 43

Figure 10 – Initial embeddings and the layouts that result from reducing visual disorder with representative selection. SADIRE was applied to several configurations by varying the k and α parameters. 44

Figure 11 – The outcomes of the user experiments. Each line denotes a unique representative selection configuration. The heatmap cells encode how many times a representative layout was placed in that order position (visual disorder) and how many times a representative layout was assigned a rank number for each dataset (similarity). 45

Figure 12 – ExplorerTree using different sampling algorithms. On the left, the traditional scatter-plot representation. On the right, the first level of ExplorerTree using four sampling strategies. 46

Figure 13 – The cExpression pipeline. The user submits a high-dimensional dataset with $\mathbb{R}^2$ coordinates and cluster labels after dimensionality reduction (A). The contrastive information for each pair of cluster and feature is computed using the clustering labels and the high-dimensional data (B). The data generated from this process is used to create visualization approaches that highlight the unique characteristics of each cluster (C). 50
Figure 14 – Defining distinctive features. The feature's distribution (**a**) in a cluster of interest is compared to its distribution on the remaining dataset (**b**). Then, the t-Student test (**c**) returns the probability of these two distributions to be different (**d**). Finally, the features are arranged in decreasing order according to their t-score to communicate discriminative power (**e**). 52
Figure 15 – UMAP projection of the *Vertebral Column* dataset and two distinctive features for the Spondylolisthesis class. 52
Figure 16 – The cExpression visualization tool. Users provide high-dimensional points with annotated clusters and positions onto $\mathbb{R}^2$ (a). The $\mathbb{R}^2$ coordinates are used for the scatterplot layout (c), from which the analysis starts. Users click on points in a cluster to investigate their contrastive information. For instance, the bipartite graph (c) shows the distinctive terms for the dark-orange cluster. There is also an interaction mechanism to visualize the feature values on the scatterplot visualization (d, e). Finally, an overview analysis can be achieved through the heatmap encoding. 54
Figure 17 – The bipartite graph encoding feature importance. The p-values and t-values are encoded using red-to-blue and continuous color scales, respectively. It is worth noting that we display the number of decimal places after zero rather than the p-value itself. The first axis (a) represents the context of the available features, while the second axis (b) represents the focus—in this case, they are the same. The connections between the focus axis and the t-score represent the relationship between p-values and t-scores, with green representing positive values and red representing negative values. Finally, the four most distinguishing characteristics are pre-selected and displayed as a histogram plot (c). . 55
Figure 18 – Construction steps for cell-based encoding. For each cell, the sum of each feature is obtained (a). The cell is then divided based on user demand to communicate the features. The space assigned to a feature is proportional to the number of features in the cell. For example, one feature in (b), two in (c), and three in (d). 57

Figure 19 – Cell-based visualization for feature distribution on the scatterplot. After inspecting the distinctive features on the scatterplot (a), users can visualize the feature values using cell-based visualizations. For instance, when one feature is selected (b, `syrian`), a simple continuous color scale is sufficient. For more than one feature (c, `syrian, kill, protect`), cell-based encoding is employed. 58
Figure 20 – A comparison of two clusters. The visualization strategy in Fig. 17 is used to communicate information between two clusters when comparing two clusters. In essence, while the information for the first cluster (light-blue in the figure) is encoded as usual (c), the information for the second is positioned on top (a), and the context axis is positioned in the center of the visualization (b). Furthermore, the four most distinguishing features of both clusters are depicted in (d). 59
Figure 21 – Users can inspect the histograms of available features through lasso selection over the points encoding p-values. 60
Figure 22 – Focus+context operation. The visualization starts with the context and focus axes as the same (1). Then, users can select a range of the context axis to focus analysis (2), resulting in an update of the focus axis (3). . 61
Figure 23 – UMAP projection of the *news* dataset. 62
Figure 24 – After selecting the five most discriminative features, highlighting the terms `nuclear`, `plant`, and `fukushima` shows that UMAP distinguished the news articles separated the news articles related to the nuclear accident on the left side of the projection. 62
Figure 25 – Inspecting terms with lower confidence. Terms with lower confidence in the light-orange cluster are related to political aspects (a) and were positioned in a different subcluster. 63
Figure 26 – Contrastive information showing levels of understanding. The terms highlighted in red show high-level information, while the terms highlighted in blue give specific hints about the news articles. 64
Figure 27 – Inspection of `germani` and `europ`'s terms shows that the dimensionality reduction separated news articles focused on the European continent from the news articles focused on Germany. 64
Figure 28 – UMAP projection of the *tweets* about COVID-19. 65
Figure 29 – Discriminative terms for the dark-orange cluster—terms related to respiratory problems. 66
Figure 30 – Discriminative terms for the light-orange cluster. Terms indicate users tweeting their complaints about symptoms expressed during COVID-19 infection. 66
Figure 31 – Comparing two clusters with `fever` as the main subject. 67

Figure 32 – Distinctive terms for the light-blue cluster. In this case, there are more general aspects of symptoms related to fever. For instance, the term `sore`, could be related to *sore throat* or a new COVID-19 symptom. . . 68
Figure 33 – Distinctive terms from the dark-blue cluster can be considered as indicative of Dengue symptoms. 68
Figure 34 – Evaluating another term from the dark-blue cluster. The terms such as `difficulty` and `breathing` indicate the most severe COVID-19 symptoms, shortness of breath. 69
Figure 35 – Heatmap showing that the red cluster comprises the tweets where people are worried about wearing masks and coryza. 69
Figure 36 – UMAP projection of the *Vertebral Column* dataset next to the heatmap visualization. 70
Figure 37 – The class of patients with Spondylolisthesis shows greater values for all of the features, except for Pelvic Radius. 70
Figure 38 – Inspection of the distribution values of the features shows that the features are truly expressed more in the cluster of patients with Spondylolisthesis. 71
Figure 39 – Retrieving topics using cExpression. Given a bag-of-words representation of a document collection (**a**), we compute the dataset clusters (**b**). *cExpression* is executed to retrieve distinctive terms for each cluster (**c**), and the topics correspond to the first m terms of each cluster. 72
Figure 40 – cExpression shows competitive results for topics with a low number of terms (≤ 20). 73
Figure 41 – cExpression was able to surpass LDA and NMF for the majority of the number of topics. 73
Figure 42 – Time execution in logarithmic scale for retrieving distinctive features using contrastive analysis. 74
Figure 43 – ClusterShapley framework. In the "dataset annotation" component (1), a dimensionality reduction (1.a) process helps users to analyze and to annotate clusters perceived in the visual space ($\mathbb{R}^2$); then, the clusters defined in $\mathbb{R}^2$ annotate the high-dimensional dataset (1.c). In the "Shapley estimation" component (2), the annotated high-dimensional dataset is used to generate clusters probabilities for each data sample (2.a) employed for Shapley values estimation (2.b). 81
Figure 44 – Cluster definition through manual selection. Users receive a projected dataset with no clustering imposed, and then users define clusters by lassoing the groups perceived in the visual space. Notice that black circles indicate data samples not belonging to any cluster, while circles with different data hues indicate already assigned data samples. 82

Figure 45 – An overview of the prototype system. In (A), users can specify datasets and load previously generated explanations. In (B), we draw a scatterplot representation and use color to encode classes. Hovering over specific class instances generates summary representations of Shapley values, which are used to indicate importance (C). The detailed examination of Shapley values and feature importance assists analysts in comprehending the relationship between feature value and Shapley value (D). The **Importance Summary** is shown in (E), which summarizes the mean of the absolute Shapley values for each pair (class, feature). The importance overview is depicted by a heatmap with the sum of the absolute Shapley values (F). 84
Figure 46 – Summarization approach to encoding two histograms for a given feature i, $\mathcal{H}_e^i$ for feature values equal or greater than the mean (σ_i) and $\mathcal{H}_o^i$ for features values lower than the mean. The two histograms are encoded together by alternating their bins. 85
Figure 47 – A tool-tip for summarizing information from the blue cluster of Fig. 45. 85
Figure 48 – Coordination between the scatterplot and the dot plot encoding the Shapley values. Each line segment represents a point in the scatterplot representation. 87
Figure 49 – Two clusters with low separation. While the first three features show the same overall importance for these two clusters, `sepal width` plays a significant role in differentiating cluster orange (a). 88
Figure 50 – Explanations for the *Vertebral Column* dataset. There is a clear separation among patients with complications (Spondylolisthesis or Hernia) and the healthy ones. The **Importance Summary** also supports visualization of such characteristics. 89
Figure 51 – Correlation between feature values and their importance for categorizing cluster 1. The four most important features contributed to differentiate patients with Spondylolistyhesis from the others. 90
Figure 52 – Comparison between regular patients (A) and patients with Hernia (B). The difference in values for `sacral slope` was not good enough to impose a good separation on the dimensionality reduction result. 91
Figure 53 – Dimensionality reduction results for the *Red Wine Quality* dataset colored using intensity to encode quality index and using colors to depict clusters. 92
Figure 54 – Cluster 4 is well defined and distinguishable due to the salty flavor (high levels of `chrolides`) of the wines. 92

Figure 55 – Comparison between two clusters projected distant which share two important features. The difference between alcohol levels was determinant for differentiating the quality of the wine in these two clusters. 93
Figure 56 – A combination of higher values `Fixed Acidity` and lower values of `Volatile Acidity` can be unpleasant at higher levels and can explain higher quality for the data instances. 94
Figure 57 – At higher levels, `Total Sulfur Dioxide` tends to add flavor to the wine and be noticed in the nose. Such an important feature is the meaning of higher concentration of low-quality wines in cluster 1. 94
Figure 58 – Feature contributions and the dimensionality reduction result for the *Communities and Crime* dataset. 95
Figure 59 – Such community is very distinguishable from the other for having a lower percentage of houses occupied `PctHousOccup`. 95
Figure 60 – Detailed analysis of clusters 4 and 5. 96
Figure 61 – A community of immigrants due to the high percent of immigrants (`PctForeignBorn`, `PctRecImmig8`, and `PctRecImmig10`) who speak more than one language besides English (`PctSpeakEngOnly`). 97
Figure 62 – Using clustering techniques to feed ClusterShapley with different clustering results. The original dataset (`original labels`) has three labels, and we use three clustering configurations: four clusters using k-Means (`k-means (1)`), eight clusters using k-Means (`k-means (2)`), and nine clusters using Agglomerative clustering (`agglomerative`). 98
Figure 63 – Comparison of traditional dimensionality reduction (DR) techniques (A) vs. hierarchical DR techniques (B). Traditional techniques operate in a single level of detail and focus on delivering a big picture that depicts the dataset as a whole, ignoring fine-grained details of intra-cluster distributions and making it difficult to proceed with more detailed investigations. On the other hand, hierarchical DR (HDR) approaches summarize important intra-cluster information already on the top-level representations of the hierarchy imposed on the input data without affecting the layout's overall organization. 102

Figure 64 – Hierarchy construction and projection with HUMAP. The hierarchy is constructed from bottom to top. First, a k-nearest neighbor graph approximates the connection strength among data points in the high-dimensional space (A). Then, the graph structure and the connections' strengths are used to find the most visited nodes (landmarks) after several random walk steps (B), which correspond to points on superior levels (C). The representation neighborhood—local and global relationships (D)—is used for computing the similarity among landmarks (F). For projecting hierarchy levels (or subsets of them), HUMAP applies—as in UMAP—a symmetrization to the nearest neighbor graph (H) and use data points from higher levels to preserve the mental map (I). . . . 106
Figure 65 – Mental map preservation across hierarchical levels. 113
Figure 66 – Explaining dimensionality reduction results. (A) Explanation of the whole dataset provides one story about the clusters defined by the user. (B) Explanation of the top-level HUMAP representation gives an overview of the explanations that can be refined according to user demands. 114
Figure 67 – User interaction with HUMAP to get more detailed information about the data. At each level explanation (C and F), users can generate diverse insights by focusing on distinct subsets of data to investigate using HUMAP's drill-down operation. 115
Figure 68 – Hierarchical exploration of the learning weights and explanations of a CNN trained on Fashion MNIST dataset. The top hierarchy level (A) encodes the overview of the dataset. Exploring lower levels by drilling-down the hierarchy (B-F) reveals more detailed information while preserving the mental map. 116
Figure 69 – Visual analysis of the embeddings generated for top and lowest hierarchical levels using a three-level hierarchy. For each dataset, top-level embedding appears on the left, and the lowest level (whole dataset) appears on the right. 118
Figure 70 – Run-time execution in seconds using a $\log_{10}$ scale. The boxen-plots show the run-time for 20 runs of each method under the same condition – we measure the run-time to fit the hierarchy (Hierarchy fit), to embed the top level (Level 2), and to embed the whole dataset (Level 0). 120
Figure 71 – Neighborhood Preservation after embedding on $\mathbb{R}^2$. HSNE outperforms HUMAP by a great margin only for *EB* dataset. 120
Figure 72 – Evaluation of techniques' ability to represent complex structures such as clusters, manifolds, and other relationships. 121

Figure 73 – Mental map preservation measured using Procrustes Analysis. HUMAP decreases the variation when positioning points across hierarchy levels. 122
Figure 74 – Continuity values for varying neighborhoods. All of the techniques performed well at preserving clustering structures. 122
Figure 75 – Trustworthiness values showing similar information to Neighborhood Preservation values. 123
Figure 76 – Using HUMAP, UMAP, and HSNE to project subset of data. 124
Figure 77 – Mental map preservation when projecting subsets of datapoints. HUMAP shows stability across subsequent hierarchy levels. 125
Figure 78 – A sampling technique's ability to uncover the projection structures must take into account the outliers and less dense clusters. The results show that SADIRE is the most stable method among the analyzed, covering the projected dataset being taking into account all of its structures. . . 143
Figure 79 – Results for in *Execution time (s)*, *Coverage*, and *Redundancy* for CBR-ILP-IR, Shuttle, Corel-1k, Corel-5k, and Diabetes (each line) 144
Figure 80 – The overview of ExplorerTree tool. 145
Figure 81 – Choosing different dimensionality reduction techniques: **a.** IDMAP; **b.** LSP. 145
Figure 82 – Choosing different sampling techniques: **a.** SADIRE (ours); **b.** Affinity Propagation. 146
Figure 83 – Traditional scatter plot representations of dimensionality reduction results next to the first level of ExplorerTree using different sampling strategies. 147
Figure 84 – SADIRE shows slightly lower results compared to Bisecting k-means. However, SADIRE is superior over Bisecting k-means in run-time execution, an important characteristic for our scenario. 149
Figure 85 – Coherence values for different number of topics. 151
Figure 86 – Run-time execution for cExpression and ccPCA when varying the number of data points, dimensionality, and a number of clusters. 152
Figure 87 – Interpretation of two dimensionality reduction techniques using cExpression. Since the sets are used to compare the feature distributions, the result remains the same. 153
Figure 88 – Analyzing the same dataset with different clustering results. cExpression merges the unique features from the two clusters (dark-blue and dark-orange) from Figure 87. 154
Figure 89 – Results for Time, DEMaP, and Disparity for twenty executions of each HDR technique. 156
Figure 90 – Results for Neighborhood Preservation, Continuity, and Trustworthiness for twenty executions of each HDR technique. 156

Figure 91 – Results for Time, DEMaP, and Disparity for twenty executions of each HDR technique. 158
Figure 92 – Results for Neighborhood Preservation, Continuity, and Trustworthiness for twenty executions of each HDR technique. 159
Figure 93 – Projections for Embryoid Body and Mammals datasets. 160
Figure 94 – Projections for FMNIST and MNIST datasets. 160
Figure 95 – Numerical analysis for twenty executions of each HDR technique. . . . 161
Figure 96 – Mental map preservation measured using Procrustes Analysis. HUMAP decreases the variation when positioning points across hierarchy levels. 162
Figure 97 – Continuity values for varying neighborhoods. All of the techniques performed well at preserving clustering structures. 163
Figure 98 – Trustworthiness values showing similar information to Neighborhood Preservation values. 163
Figure 99 – Scalability analysis for the main steps in HUMAP algorithm. 164
Figure 100 – Scalability analysis for the main hierarchical dimensionality reduction techniques. 165
Figure 101 – Results for reproducibility following the procedures of Becht et al. (BECHT et al., 2018). 166
Figure 102 – Run-time distributions for HUMAP when generating reproducible results. 167
Figure 103 – Embeddings of the MNIST dataset for different values for “min_dist” and “n_neighbors”. 168
Figure 104 – Embeddings of the Embroyd Body dataset for different values for “min_-dist” and “n_neighbors”. 169
Figure 105 – Metrics for HUMAP with different β values. 170
Figure 106 – Projections for HUMAP with different β values. 171
Figure 107 – Embeddings of the MNIST dataset for different values for 172
Figure 108 – Embeddings of the Embroyd Body dataset for different values for . . . 173
Figure 109 – HUMAP exploration and annotation of a document collection of COVID-19 *tweets*. The top-level hierarchy level shows unlabeled data points and three major structures (A). We annotate these three clusters (B) and compute their topics computed (F). For each cluster in (B), we also project their corresponding level (and final) hierarchy to look for other patterns, annotating the dataset (C, D, E) and computing their topics. 174

LIST OF TABLES

Table 1 – Datasets used for analysis. 35
Table 2 – Time for estimating Shapley values. 100
Table 3 – Datasets used for experimentation. 118
Table 4 – The datasets used in the experiments. All the projections were performed by LSP (PAULOVICH et al., 2008), except for Shuttle, in which we used PLMP (PAULOVICH; SILVA; NONATO, 2010a). 142
Table 3 – Scores given by the participants 148
Table 4 – p-values for the running time metric. 155
Table 5 – p-values for the DEMaP metric. 157
Table 6 – p-values for the disparity metric. 157
Table 7 – p-values for the running time metric. 158
Table 8 – p-values for the running time metric. 158
Table 9 – p-values for DEMaP metric. 159
Table 10 – p-values for the disparity metric. 159

LIST OF ALGORITHMS

1 Removing redundant data points. 34
2 Shapley estimation. 83
3 Associate Landmarks. 111

CONTENTS

Resumo . . . . . **iv**

Abstract . . . . . **v**

List of Figures . . . . . **vi**

List of Tables . . . . . **xv**

List of Algorithms . . . . . **xvi**

1 INTRODUCTION **23**
1.1 Context **23**
1.2 Problems & Motivation **24**
1.2.1 Summarization & Interpretability 24
1.2.2 Hierarchical Dimensionality Reduction 25
1.3 Questions & Hypotheses **26**
1.4 Contributions **27**
1.5 Document structure **28**

2 SADIRE: SAMPLING FROM DIMENSIONALITY REDUCTION . . **30**
2.1 Related Works **31**
2.2 Sampling based on dimensionality reduction **32**
2.2.1 Selecting candidates for the representative set 32
2.2.2 Removing redundant data instances 33
2.2.2.1 Time complexity 34
2.3 Experiments **35**
2.3.1 Redundancy 36
2.3.2 Coverage 38
2.3.3 Clustering 39
2.3.4 Running time 40
2.3.5 Parameter Analysis 41
2.4 User Experiment **42**
2.5 Practical Application **45**
2.6 Final Remarks **46**

3 CONTRASTIVE ANALYSIS FOR SCATTERPLOT REPRESENTATIONS OF DIMENSIONALITY REDUCTION **47**

3.1 Related Works . . . 48
3.2 cExpression—Tool for contrastive analysis of DR results . . . 50
3.2.1 Computing constrastive information . . . 51
3.2.2 Visualization Design . . . 53
3.2.2.1 Visual and task requirements . . . 53
3.2.2.2 Visualizing the feature importance . . . 55
3.2.2.3 Visualizing feature value distribution . . . 56
3.2.2.4 Comparing clusters . . . 58
3.2.2.5 Summary View . . . 59
3.2.2.6 Interaction mechanisms . . . 60
3.3 Case studies . . . 61
3.3.1 The news dataset . . . 61
3.3.2 Exploring tweets about COVID-19 symptoms . . . 64
3.3.3 Vertebral Column . . . 69
3.4 Evaluation . . . 71
3.4.1 Coherence . . . 71
3.4.2 Run-time execution . . . 73
3.5 Final Remarks . . . 74

4 EXPLAINING DIMENSIONALITY REDUCTION RESULTS USING SHAPLEY VALUES . . . 76
4.1 Related Works . . . 78
4.2 Review of Shapley values . . . 79
4.2.1 Shapley values illustration . . . 80
4.3 Using Shapley values to explain clusters of dimensionality reduction results . . . 80
4.3.1 Dataset annotation . . . 81
4.3.2 Shapley estimation . . . 82
4.4 Visualization Design . . . 83
4.4.1 Scatterplot component . . . 84
4.4.2 Coordination between scatterplot and Shapley values . . . 85
4.4.3 Importance Summary and Importance Heatmap . . . 86
4.4.4 Analysis example . . . 87
4.5 Case Studies . . . 89
4.5.1 Vertebral Column . . . 89
4.5.2 Red Wine Quality . . . 91
4.5.3 Communities and Crime . . . 94
4.5.4 Automatic annotation with clustering techniques . . . 97
4.6 Final Remarks . . . 99
4.6.1 Supporting exploratory analysis . . . 99

4.6.2 Run-time execution . . . 99
4.6.3 Cluster-oriented analysis . . . 100

5 HUMAP: HIERARCHICAL UNIFORM MANIFOLD APPROXIMATION AND PROJECTION . . . 101
5.1 Related Work . . . 103
5.2 Hierarchical Dimensionality Reduction . . . 106
5.2.1 Landmark selection . . . 107
5.2.2 Dissimilarity definition . . . 108
5.2.3 Generating the embedding . . . 110
5.2.3.1 Subset projection . . . 111
5.2.3.2 Mental map preservation . . . 112
5.3 Case studies . . . 112
5.3.1 Supporting hierarchical explanation of dimensionality reduction results . . . 113
5.3.2 Exploring latent features . . . 116
5.4 Numerical evaluation . . . 117
5.4.1 Running-time . . . 119
5.4.2 Neighborhood Preservation . . . 119
5.4.3 DEMaP . . . 121
5.4.4 Mental map preservation . . . 122
5.4.5 Continuity . . . 122
5.4.6 Trustworthiness . . . 123
5.4.7 Projecting subsets of data . . . 123
5.5 Final Remarks . . . 124
5.5.1 Embedding initialization . . . 124
5.5.2 Generalizability and scalability . . . 125

6 CONCLUSION . . . 126
6.1 Further Discussion and Open Questions . . . 126
6.1.1 SADIRE . . . 126
6.1.2 cExpression . . . 126
6.1.3 ClusterShapley . . . 127
6.1.4 HUMAP . . . 127
6.2 Practical Implications . . . 128
6.3 Scientific production . . . 129
6.3.1 First-authored publications . . . 129
6.3.1.1 Journal articles . . . 129
6.3.1.2 Conference papers . . . 129
6.3.1.3 Preprint . . . 130
6.3.2 Co-authored publications . . . 130

6.3.2.1 Conference papers . . . 130
6.3.2.2 Preprint . . . 130

BIBLIOGRAPHY . . . 131

A SADIRE - EMPLOYING SADIRE INTO EXPLORERTREE & ADDITIONAL USER EXPERIMENTS . . . 141
A.1 Execution Time, Coverage and Redundancy . . . 141
A.2 ExplorerTree Tool . . . 143
A.2.1 Overview . . . 143
A.2.2 Using different dimensionality reduction techniques . . . 143
A.2.3 Using different sampling selection mechanisms . . . 144
A.3 ExplorerTree using different sampling strategies . . . 145
A.4 User Experiment - Evaluating sampling strategies for ExplorerTree . 146

B CEXPRESSION . . . 150
B.1 Topic evaluation . . . 150
B.2 Run-time evaluation . . . 152
B.3 Different dimensionality reduction techniques . . . 153

C HUMAP . . . 155
C.1 Hierarchical Dimensionality Reduction comparison . . . 155
C.2 Drill-down analysis . . . 158
C.3 Hierarchical Dimensionality Reduction without preserving the mental map . . . 160
C.4 Additional local/global analysis . . . 162
C.4.1 Mental map preservation. . . . 162
C.4.2 Continuity. . . . 162
C.4.3 Trustworthiness. . . . 162
C.5 Run time analysis - HUMAP steps and scalability . . . 164
C.5.1 Scalability . . . 165
C.5.2 Reproducibility . . . 166
C.6 Generalizability . . . 168
C.6.1 *min_dist* and *n_neighbors* . . . 168
C.6.2 β for local contribution . . . 170
C.6.3 Number of data points in each level . . . 172
C.7 Additional use case - Exploration of COVID-19 tweets . . . 174

1 INTRODUCTION

1.1 CONTEXT

Data visualization is critical in assisting humans in understanding natural phenomena as well as discovering ubiquitous information. Exploratory Data Analysis (EDA) (TUKEY, 1962), a set of methods and strategies for understanding results and simplifying analysis, is the primary reason scientists and other data visualization practitioners use it. EDA is typically used when the information to be uncovered is unknown.

Exploratory Data Analysis is especially important when the dataset contains multiple attributes (or features), which is referred to as high-dimensional data. The process of evaluating high-dimensional data presents unique challenges due to its redundancy, sparsity, and complex structures. Furthermore, as data such as text, photos, and the inner weights or logits of deep learning models become more widely available, high-dimensional data analysis has become a common activity. Efforts are being made to reduce the cognitive overload required for analysis in order to comprehend and make sense of these high-dimensional landscapes as well as the inherent relationships among the data points. One popular strategy is to reduce the number of dimensions while retaining important details about the structures and similarities found in high-dimensional space.

Dimensionality Reduction (DR) techniques are commonly used to generate a representation in low dimension ($\mathbb{R}^d$) from a dataset in high dimension ($\mathbb{R}^m$). The end result may be referred to as embeddings, projections, or even DR layouts. These methods make an effort to preserve the relationship between the data points that existed in the original space. Then, while conducting exploratory data analysis, researchers use scatterplot-based representations to look for trends and other relevant information. There are numerous studies that use DR techniques for exploratory data analysis, such as understanding the learned features by ConvNets (PEZZOTTI et al., 2018) at different epochs, investigating gene expression patterns to discover new cell types (UNEN; HöLLT; PEZZOTTI, 2018), and many others. Analysts interpret the decisions made by these algorithms to determine whether the features fed into machine learning models are relevant to class separation (MARCILIO-JR et al., 2020), in addition to providing an excellent opportunity for high-dimensional data analysis.

Dimensionality reduction strategies are almost always used in high-dimensional data pipelines, but they come with their own set of challenges, such as the inability to integrate categorical data and the runtime execution of non-linear algorithms, which may be thoroughly read in recent surveys. In the following section, we delineate the problems

and motivations in the scope of this thesis.

1.2 PROBLEMS & MOTIVATION

Dimensionality Reduction (DR) techniques are widely used in a wide range of scientific fields. Embeddings encoded by scatterplots aid in the analysis of document collections (ELER et al., 2009), gene expression data (UNEN et al., 2017; SALAMON et al., 2018; HÖLLT et al., 2018; SOMARAKIS et al., 2019; KRUEGER et al., 2020), and deep learning model features (RAUBER et al., 2017; BENATO; TELEA; FALCÃO, 2018). However, non-linear techniques make it difficult to understand embeddings according to feature contributions for layout formation. Furthermore, one-level approaches, which represent all data at once, may conceal important information about the relationship between dominant structures. These issues are summarized in the two sections that follow.

1.2.1 SUMMARIZATION & INTERPRETABILITY

Scatterplot representations of embeddings become visually cluttered as the dataset size increases, limiting understanding of the relationship among data points. This limitation can be addressed by summarizing the data during analysis using common sampling techniques. Traditional techniques (VITTER, 1985; KNUTH, 1997) use probabilistic mechanisms to select samples, whereas deterministic methods, such as CSM (JOIA; PETRONETTO; NONATO, 2015), use PCA (JOLLIFFE, 1986) loadings to select samples. These techniques are problematic for scatterplot representations because probabilistic approaches are influenced by denser areas of the projection and fail to account for outliers, whereas PCA loadings are influenced by global information. The inability of these two types of sampling commonly used for scatterplot representations to convey information about class boundaries of varying sizes is a recurring pattern.

Using interpretation techniques on $\mathbb{R}^2$ is another method for improving understanding and visualization of embeddings. While many approaches have been proposed for explaining and interpreting the decisions of common machine learning models (RIBEIRO; SINGH; GUESTRIN, 2016; LUNDBERG et al., 2020), dimensionality reduction interpretation has been performed with enriching layout techniques (NONATO; AUPETIT, 2018), in which visualizations of feature distributions and other statistical charts to represent attribute information aid analysts in making sense of clusters and other patterns perceived in the visual space. Simple statistics, on the other hand, cannot fully provide the differentiation among the clusters of the embedding produced by a DR technique. To address this issue, a few approaches based on contrastive analysis (FUJIWARA; KWON; MA, 2019; LE; AKOGLU, 2019) have been proposed, with the main goal of identifying unique characteristics of the clusters perceived in the visual space $\mathbb{R}^2$. These techniques'

main limitations can be summarized as prohibitive runtime execution of high-dimension datasets (FUJIWARA; KWON; MA, 2019) or the inability to extend the analysis for different DR (LE; AKOGLU, 2019).

Rather than emphasizing the differences between the clusters, one possible solution is to assess how much each feature contributed to the embedding, that is, how much each feature contributed to the formation of clusters or other structures in the projected space (e.g., $\mathbb{R}^2$). In gene expression analysis, for example, bioinformaticians want to know which genes influence each cluster in order to annotate cell types or discover new ones (LäHNEMANN; KöSTER; SZCZUREK, 2020). Finding the contribution of these features to the dimensionality reduction result is primarily related to non-linear DR techniques (MAATEN; HINTON, 2008; MCINNES; HEALY; MELVILLE, 2018), where there is currently no way to inverse calculations and keep track of feature contributions while the algorithm is being executed (FUJIWARA; KWON; MA, 2019). Nonetheless, non-linear DR techniques are best suited for dealing with most datasets due to their ability to uncover complex structures such as manifolds.

There are a few issues with existing techniques for interpreting contributions for DR interpretation. Studies on feature values (COIMBRA et al., 2016; PAGLIOSA; PAGLIOSA; NONATO, 2016; MARCILIO-JR et al., 2020), for example, do not account for the dimensionality reduction process and instead concentrate on the reduced low-dimensional space. Other more elaborated works (TURKAY et al., 2012; JOIA; PETRONETTO; NONATO, 2015) obtain feature importance through principal components (PC). The PCs produce biased results for classes with high variation (JOIA; PETRONETTO; NONATO, 2015), and their inability to focus on local information (FUJIWARA; KWON; MA, 2019) hinders cluster-oriented analysis. These techniques cannot explain how much each feature contributed to the embedding, for example, in terms of clusters. The importance assigned to the features does not account for their contribution to clusters and other structures. Furthermore, these feature importance measures do not interact with one another to form an explanation measure. Instead, the importance of each feature is independent of the other.

1.2.2 HIERARCHICAL DIMENSIONALITY REDUCTION

Due to the increasing popularity of deep learning (RAUBER et al., 2017; BENATO; TELEA; FALCãO, 2018) applications and gene expression data analysis (UNEN et al., 2017; KRUEGER et al., 2020) with dimensionality reduction, practitioners typically seek DR techniques that can uncover complex structures (e.g., manifolds), retain clusters, and provide meaningful representations of the global relationship among clusters.

Traditional DR techniques operate at a single level of detail, representing the relationship between data points in a single low-dimensional scatterplot representation—

which limits analysis when important information is prioritized and details are added as needed. This limitation can be circumvented using hierarchical DR (HDR) techniques, which project subsets of the input dataset and allow the user to select where to focus during the analysis. The idea behind HDR techniques is to project dominant structures in areas where details about their relationship would be obscured by traditional DR techniques.

Current HDR techniques have limitations in terms of scaling thousands of data points (PAULOVICH et al., 2008), mental map preservation (PEZZOTTI et al., 2016), or are primarily aimed at mass cytometry data applications, not handling datasets with hundreds or thousands of dimensions (KUCHROO, 2020). These HDR techniques cannot fully address two major issues: a good trade-off between preserving global and local structures in the low-dimensional representation of the dataset and the mental map preservation across hierarchy levels. While preserving global and local information allows for a better understanding of clusters (local structures) and their relationships (global structures), preserving mental maps makes exploration easier by reducing the number of geometric transformations on the dataset (FUJIWARA et al., 2020). In addition to being able to deal with high dimensions and be flexible in terms of hierarchy manipulation and dataset type, HDR approaches must be able to unfold complex structures present in real-world datasets. Finally, there is a current requirement for techniques that depict complex structures while preserving global and local relationships between data samples. For example, when analyzing deep learning models (RAUBER et al., 2017; MARCíLIO-JR et al., 2020) and gene expression data (LäHNEMANN; KöSTER; SZCZUREK, 2020), the trade-off between global and local relations is critical.

This thesis focuses on the issue of encoding global and local relationships among data points during the embedding process, as well as the preservation of the mental map for hierarchical dimensionality reduction techniques.

1.3 QUESTIONS & HYPOTHESES

With regard to the aforementioned issues and the motivations behind each of them, this thesis is built around the following questions, from which we conducted differing but interconnected research activities: **Q1**. *Can non-linear dimensionality reduction techniques be interpreted for information discovery?* **Q2**. *How should information be summarized while interpreting the main causes of clustering formation?* **Q3**. *How can global and local information, as well as the mental map, be preserved when interacting with hierarchical embeddings of high-dimensional data?*

The following research activities (RAs) address each of the aforementioned questions.

RA1: We propose to adapt and advance state-of-the-art model-agnostic explainable

dimensionality reduction techniques. More specifically, we combine SHAP (LUNDBERG; LEE, 2017) with carefully designed visualization approaches to provide interpretability features as well as interaction mechanisms on scatterplot-represented embeddings. To answer **Q1**, we hypothesize that *addictive explanations of the contribution to cluster formation, combined with feature analysis, provide a more detailed understanding of the phenomenon.*

RA2: We used contrastive analysis on scatterplots to address the challenge of summarizing information and interpreting it for cluster analysis. To accomplish this, we employ statistical tests on feature values in conjunction with focus+context visualizations that enrich scatterplot representations of embeddings. Users can comprehend the positioning strategies of various dimensionality reduction techniques while retaining only a portion of the information. To specifically answer **Q2**, *we hypothesize that only a few features are required to begin exploration and interpretation of embeddings on* $\mathbb{R}^2$*, and that algorithms that can order features according to relevance can filter out a lot of irrelevant information.*

RA3: We proposed a novel hierarchical dimensionality reduction technique that uses a tunable hyper-parameter to choose between global and local preservation. In addition, the novel technique advances the state-of-the-art in preserving the mental map when projecting subsequent hierarchy levels. To specifically answer **Q3**, we hypothesize that *by using different neighborhood concepts (local and global) and fixing previously projected data points on the optimization step of the hierarchical dimensionality reduction, we will be able to reduce these specific limitations.*

Based on the the above, this thesis considers the following main hypothesis:

Main Hypothesis. *Hierarchical exploration and interpretation of embeddings improve the analysis power of dimensionality reduction techniques by assisting in identifying and emphasizing various facets present in data. While mental map preservation in hierarchical embeddings allows for context preservation throughout exploration, dimensionality reduction interpretation explains feature contributions to layout formation in the visual space.*

1.4 CONTRIBUTIONS

The main contribution of this thesis is to make progress on dimensionality reduction interpretation and hierarchical exploration. In terms of interpretation, we present two methods that use visualization approaches to highlight feature contributions for layout formation—primarily in the form of clusters. Then, to deal with large datasets, we propose a novel hierarchical dimensionality reduction technique that overcomes a few limitations of existing approaches. Despite their differences, these contributions can be used together to deal with large scale datasets while providing interpretation for their structures.

The first contribution is concerned with the information summarization of scatterplot representations for dimensionality reduction results (see Chapter 2). We make progress with a fast sampling method in visual space ($\mathbb{R}^2$) that can preserve class boundary structures while keeping outliers visible. Aside from the method itself, we also conducted experiments to investigate how changing the parameters affects the user's ability to assess the information contained. Finally, the method serves as the foundation for ExplorerTree (MARCíLIO-JR et al., 2021a), for which we conducted a user experiment by comparing it to a few other sampling techniques.

The second contribution is concerned with the visual summarization problem and the interpretation of DR results (see Chapter 3). We addressed the problem of understanding cluster formation by leveraging statistical tests on the feature values of the input datasets for dimensionality reduction. After calculating separation measures for each feature, we use a focus+context visualization tool to assist analysts in determining which features contributed the most to layout formation after dimensionality reduction. Finally, the technique enables the understanding of global positioning and intra-cluster differentiation through various interaction mechanisms.

The third contribution deals with the issue of explaining the results of dimensionality reduction (see Chapter 4). We advance the state-of-the-art by adapting SHAP to explain cluster formation after dimensionality reduction. Unlike other methods for interpreting DR results in the literature, our approach provides explanations using addictive contributions that are directly related to the distance measure used to convey similarity on the scatterplot representation. In addition to the core technique, we propose a visualization tool to aid in the exploratory and explanatory capabilities of scatterplots.

The fourth contribution advances hierarchical dimensionality reduction by employing an adaptive hyper-parameter to control the trade-off between global and local structure preservation (see Chapter 5). HUMAP employs two neighborhood concepts to capture both local and global information about manifolds, while the hierarchy is built using the Finite State Markov Chain. Finally, HUMAP preserves the mental map when projecting different hierarchy levels for fine-grained information.

Aside from the contributions discussed in the document, Section 6.3 presents other first-authored publications that were part of the research efforts during the development of this thesis.

1.5 DOCUMENT STRUCTURE

The main structure of this document is based on the research activities carried out to answer the questions in Section 1.3 and previously mentioned as our main contributions. As a result, each chapter, aside from the Introduction and Conclusion, is devoted to

detailing this thesis's research contributions. In summary, the structure of this document is as follows:

- Chapter 2 introduces the SADIRE sampling technique for scatterplot representations. It covers the following article, which is related to **Q2**:
  - **Marcilio-Jr, W.E.**, Eler, D.M. *SADIRE*[1] *a context-preserving sampling technique for dimensionality reduction visualizations.* Journal of Visualization 23, 999–1013 (2020).
- Chapter 3 introduces two techniques for visual summarization and interpretation of dimensionality reduction plots. It covers the following article, which is related to **Q1** and **Q2**:
  - **Marcílio-Jr, W.E.**, Eler, D.M., Garcia, R.E., *Contrastive analysis for scatterplot-based representations of dimensionality reduction*, Computers & Graphics 101, 46-58 (2021).
- Chapter 4 addresses the problem of interpretation of dimensionality reduction techniques. It covers the following article, which is related to **Q1** and **Q2**:
  - **Marcílio-Jr, W. E.**, Eler, D. M, *Explaining dimensionality reduction results using Shapley values*[2]*,* Expert Systems with Applications 178, 2021.
- Chapter 5 describes a novel technique for hierarchical dimensionality reduction that addresses the issues of local/global structure and mental map preservation. It covers the following article, which is related to **Q3**:
  - **Marcílio-Jr, W.E.**, Eler, D.M., Paulovich, F.V., Martins, R.M. *HUMAP*[3]*: Hierarchical Uniform Manifold Approximation and Projection*, arXiv preprint 2106.07718, 2021. Submitted to IEEE Transactions on Visualization and Computer Graphics.

This document is concluded in Chapter 6, which presents final discussions about the research efforts highlighted in this thesis. It also discusses practical implications and future research.

[1] https://github.com/wilsonjr/SADIRE
[2] https://github.com/wilsonjr/ClusterShapley
[3] https://github.com/wilsonjr/humap

2 SADIRE: SAMPLING FROM DIMENSIONALITY REDUCTION

Wilson E. Marcílio-Jr, Danilo M. Eler *SADIRE: a context-preserving sampling technique for dimensionality reduction visualizations*, Journal of Visualization, Volume 23, 2020.

Although DR techniques can represent high-dimensional data structures and relations in embedded spaces (typically $\mathbb{R}^2$), the ability to scale computational methods to massive datasets is also a frequently debated topic that is becoming increasingly important. For a long time, scientists have recognized that simply increasing raw processing power cannot guarantee method interactivity (HELLERSTEIN et al., 1999). To address this issue, sampling mechanisms are frequently used because approximate answers are just as useful as exact answers (KWON et al., 2017). The visualization community also makes use of sampling techniques, such as proposing summary visualizations to depict large amounts of data (SARIKAYA; GLEICHER; SZAFIR, 2018).

For instance, well-known sampling techniques such as Knuth's (KNUTH, 1997) and Reservoir's (VITTER, 1985) rely on probabilistic mechanisms to extract samples that depict the dominant structures of a dataset. On the other hand, a deterministic method called Column Selection Method (CSM) proposed by Joia et al. (JOIA; PETRONETTO; NONATO, 2015) was shown to be effective in selecting representative samples based on a correlation index computed using matrix decomposition. Although it outperforms commonly used sampling mechanisms, it selects non-balanced samples when classes have different coefficients of variation—in this case, important structures of the sampled dataset, such as class boundaries or outliers, are lost during the sampling process. Furthermore, when data is not uniformly distributed, random techniques (e.g., Knuth's and Reservoir's) cannot reflect all data structures, impairing visual assessment of the relationship among data points.

The problem of reducing the visual cluttering of scatterplot representations of dimensionality reduction results on $\mathbb{R}^2$ is the focus of this chapter. SADIRE[1], which stands for SAmpling based on DImensionality REduction, takes a dataset embedded in 2D space and selects a limited number of points from each cell of a grid built on the embedded space. As demonstrated in the experiments, this process reduces the dataset size while preserving the overall structure and class boundaries of a scatterplot-visualized projection. Furthermore, this chapter discusses a user experiment to better understand how users perceive the use of representative data points for visual summarization.

[1] https://github.com/wilsonjr/SADIRE

The remaining of this chapter is organized as follows: the related works are shown in Section 5.1; Section 2.2 details the sampling technique; Section 5.4 delineates the effectivity of the proposed approach throughout experiments; Section 2.4 discuss the results for the user experiment; Section 5.5 finishes the chapter with discussions.

2.1 RELATED WORKS

Sampling techniques are becoming increasingly important. They provide a means to quickly infer patterns and trends in data, in addition to reducing the burden of complex methods to execute on large datasets. Well-known sampling algorithms (TILLÉ, 2011) can be used for general purposes; that is, they can be adapted to work with most problems requiring the reduction of a dataset to a representative sample. This section focuses on sampling mechanisms designed to solve specific problems. These techniques are intended to assist computational methods and interactive approaches in scaling when dealing with large datasets.

Random sampling mechanisms are frequently used in machine learning to split datasets into train and test sets or to perform cross validation to fine-tune hyperparameters. To reduce the computational burden of performing projections, dimensionality reduction techniques frequently employ sampling. LSP (PAULOVICH et al., 2008) uses either random sampling or clustering medoids as the first stage—control points—to project the sampled instances so that the remaining dataset can be projected using interpolation. LAMP (JOIA et al., 2011) constructs a family of orthogonal affine mappings, one for each instance to be projected, using the same randomly selected control points. Other dimensionality reduction techniques, such as PLMP (PAULOVICH; SILVA; NONATO, 2010b), Pekalka's (PEKALSKA et al., 1999), and PLP (PAULOVICH et al., 2011), also use sampling as preprocessing. Sampling is also a technique for reducing cognitive overload caused by large datasets represented by traditional visualizations. In fact, such strategies are used in conjunction with DR methods. HSNE (PEZZOTTI et al., 2016) technique projects dominant structures in order to provide an exploration process based on information demand while reducing visual clutter, with Monte Carlo sampling on a Finite Markov Chain. Nguyen and Song (NGUYEN; SONG, 2016) adopt centrality cluster-based sampling approaches, which employ centrality measures to obtain more informed samples than random sampling methods. Joia et al. (JOIA; PETRONETTO; NONATO, 2015) proposed a sampling strategy based on correlation scores generated by singular value decomposition that is sensitive to class variability.

Unlike the methods discussed previously, the one presented in this chapter is based on the advances made in dimensionality reduction techniques over the years. The method is based on the assumption that DR techniques preserve similarity and class structures

in multidimensional space. As a result, unlike other techniques, our sampling mechanism can accurately reflect the data's structures and class boundaries when given a reliable projection.

2.2 SAMPLING BASED ON DIMENSIONALITY REDUCTION

When the target dimension is in visual space, dimensionality reduction results are typically represented by scatterplots, with $\mathbb{R}^2$ being the most common. This type of visualization uses spatial proximity to depict similarity, so that similar data points are close together and dissimilar data points are far apart. In order to select samples, SADIRE considers such a projection $X \subset \mathbb{R}^2$, and it can be divided into two main stages. It begins by shrinking the dataset and selecting candidates for the representative set using a grid structure on the projection plane. Second, it eliminates duplicate data instances from the sampled set. These steps are described in detail in the sections that follow.

2.2.1 SELECTING CANDIDATES FOR THE REPRESENTATIVE SET

The first stage of the sampling strategy involves sampling from an $\mathbb{R}^2$ projection. SADIRE divides the projected space by generating a grid with cells of size $k \times k$ (note that k is measured in pixels). The sampled set is then created by using only the cells with at least one data point, from which a number $m < n$ of data points can be retrieved, where n is the number of data points in each cell. While our explanation and experiments are geared toward $m = 1$, keep in mind that using $m > 1$ conveys the density of the clusters. These procedures are as follows:

1. Create a grid with cells of size k by k and divide the projected space ($\mathbb{R}^2$), where k is measured in pixels;
2. Add to the set S all the data points in each cell s that has at least one data point ($|s| \geq 1$);
3. Add to the set A, m data points of from each cell $s \in S$ (if $m = 1$, add the medoid to the set A); store the density ($|s|$) of each cell.

Fig. 1 depicts how sampling is used when $m = 1$, i.e., by retrieving one data point from each grid cell. In the first step (a), a grid structure of size k by k (in pixels) is used to group data points; then, in the second step (b), the density and m data points are stored for each cell; finally, by using the density information—explained in the following section—SADIRE reduces the redundancy of the embedding.

Because the width and height of a cell are smaller than the diagonal distance between two opposite vertices when using a grid structure to sample data points from the

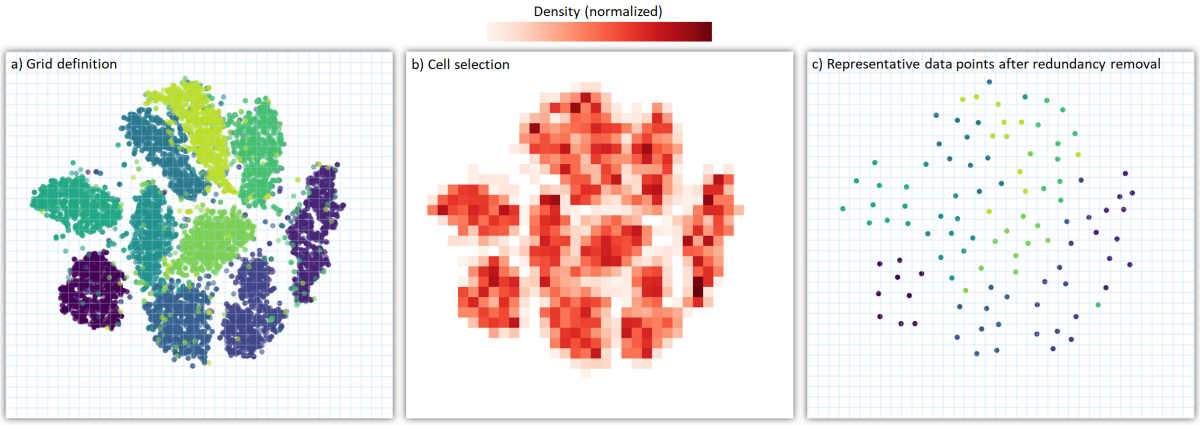


Figure 1 – Defining the grid and selecting samples. (a) A grid with cells of size k by k is used to divide the $\mathbb{R}^2$ space; (b) Only cells with at least one data point are retrieved; (c) The resulting representative set after redundancy removal.

scatterplot representation, visual connection in 45 degrees may be lost. While we argue that such a problem is insignificant enough to mislead users' exploration, the parameter k can be used to control it. Lowering the value of k yields a representative set that is more similar to the original set.

2.2.2 REMOVING REDUNDANT DATA INSTANCES

A windowing process is used in the second main step of the algorithm to reduce redundancy from neighboring cells. For a window size of $\alpha = 3$, Fig. 2 shows how the density information is used to keep useful information about the embedding structures while reducing redundancy. Starting with the densest cell (1.), all of its neighboring cells are eliminated from the competition in order to be included in the resulting representative set—the result in the scatterplot for $m = 1$ is indicated by a black arrow. The same procedure is used on the second densest cell (2.), and all of its neighboring cells are eliminated from competition. This process is repeated for all active cells in the scatterplot that were not removed.

Algorithm 1 describes the process depicted in Fig. 2, which takes as input cells ordered decreasingly by density (indicated by A). If a cell a_i was not removed from the set A (a_i.active is true in line 3), the algorithm looks for at least one cell with data points within the α-sized window centered in a_i (a_i is an outlier if it does not have neighboring cells). If neighboring cells exist, they are removed (line 7) and a_i is added to the representative set (line 6). The output of the Algorithm, G, is a set of cells that contains information about position and density.

While it is important to reduce redundancy among representative samples, outliers must be preserved on the projection plane (i.e., added to the representative set) because they can be useful in analytic tasks. Take note that we consider data points that are far

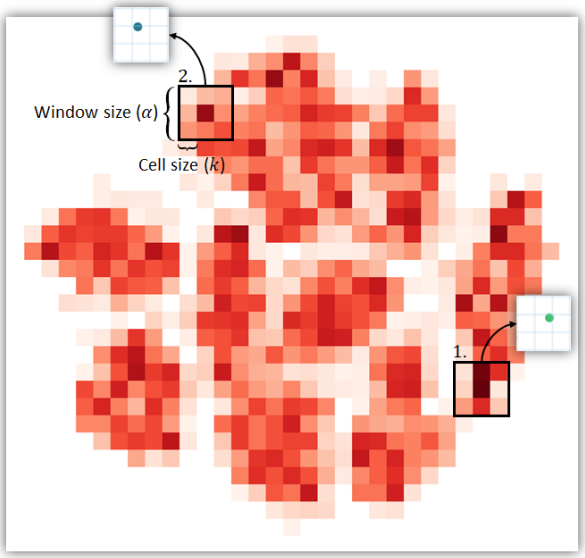


Figure 2 – Filtering redundant points. In this case, with window of size $\alpha = 3$.

Algorithm 1 Removing redundant data points.

```
1: procedure mergeInstances(A)
2:     for each a_i in A do
3:         if a_i.active then
4:             g ← cells with data points inside the window centered in a_i;
5:             if g ≠ ∅ then
6:                 G ← G ∪ a_i;
7:                 g.active ← false;
8:                 a_i.active ← false.
       return G
```

from clustered data points to be outliers. Thus, given the α-sized window used to remove redundant data points, our heuristic for detecting and adding outliers works as follows: because the data points candidates for outliers are in the active cells of set A—the data points with no neighboring cells—we select as representative data points only outliers that are truly far away from a representative data point, for example, outliers that are at least twice the distance from a representative data point (2k).

2.2.2.1 Time complexity

When all the points are far from each other and none has neighbors, the complexity of Algorithm 1 is $O(\alpha^2|A|)$, where $|A|$ is the size of the sampled set retrieved in the first step and α is the window for reducing redundant points; the complexity of searching for outliers is $O(|A|^2)$ when all the points are far from each other and none has neighbors. Because the process of creating and retrieving all points can be performed using a quad-tree, whose worst-case scenario in inserting and searching operations is known to be $O(n)$, the complexity of our method is $O(|A|^2)$. However, because of the similarity constraint mapped by DR techniques, Algorithm 1 reduces the number of instances in set A. As a

result, receiving the worst-case scenario is unlikely.

2.3 EXPERIMENTS

This section evaluates SADIRE on its ability to extract representative data points that inherently reflect the structures imposed by a DR technique. *SADIRE* was compared against three sampling mechanisms: *Reservoir* (VITTER, 1985), *Column Selection Method* (*CSM*) (JOIA; PETRONETTO; NONATO, 2015), and *Knuth* (KNUTH, 1997). In the following, we provide a description of the datasets used in the experiments (see Table 1):

- *CBR-ILP-IR* is a document collection of three fields of artificial intelligence;
- *corel-1k* and *corel-5k* are image datasets of natural scenes and artificial objects;
- *segmentation* is a dataset of seven outdoor images manually segmented;
- *mammals-8k* and *mammals-20k* are datasets composed of mammal features;
- *photographers-9k* and *photographers-36k* are image datasets of different photographers[2];
- *shuttle* is a dataset of shuttle features[3];
- *fibers* is a dataset obtained from the 2009 Pittsburgh Brain Competition[4];
- *new-wavelet* is a x-ray image dataset[5].

Table 1 – Datasets used for analysis.

Dataset	Size	Dimensions	Classes	DR technique
CBR-ILP-IR	574	35	3	LSP
corel-1k	1000	150	10	LSP
segmentation	2100	19	8	LAMP
corel-5k	5000	4096	50	t-SNE
mammals-8k	8000	72	4	LAMP
new-wavelet	9000	1024	116	LAMP
photographers-9k	9077	4096	41	LAMP
mammals-20k	20000	72	4	LAMP
photographers-36k	36364	4096	41	PLMP
shuttle	43500	9	7	PLMP
fiber	250000	30	9	PLMP

[2] http://people.cs.pitt.edu/ chris/photographer/
[3] https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)
[4] http://pbc.lrdc.pitt.edu/
[5] http://ir.shef.ac.uk/imageclef/2006/

For the projection of these datasets, four dimensionality reduction techniques were used: LSP (PAULOVICH et al., 2008), t-SNE (MAATEN; HINTON, 2008), LAMP (JOIA et al., 2011), and PLMP (PAULOVICH; SILVA; NONATO, 2010b). Then, since there is no currently way to control the number of data points returned by SADIRE, its resulting set was used to define the number of data points selected by CSM, Reservoir, and Knuth. We used $k = 5$ and $\alpha = 3$ for SADIRE parameters. The experiments were performed in a computer with the following configuration: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 32GB RAM, Windows 10 64 bits.

Fig. 3 shows the $\mathbb{R}^2$ projections and the data points retrieved by each sampling technique. While the points retrieved from *Knuth* and *Reservoir* techniques are concentrated on denser areas due to their probabilistic design, *CSM* technique prioritizes elements that give the highest scores according to correlation to the original dataset—such characteristic retrieves more representatives from classes with higher variation (JOIA; PETRONETTO; NONATO, 2015). Thus, a few structures may not be preserved and the data points retrieved from the dataset are not well suited to be representatives, e.g., the sampled points of *mammals-8k*, *mammals-20k*, *shuttle*, and *fiber*. On the other hand, SADIRE reflects the boundaries imposed by the class separation and show the layout structure of the initial projection, preserving the structures imposed by the DR technique. In this sense, SADIRE resembles the structure of the dataset—if the projection technique truly reflects the high dimensional space structure (RAUBER et al., 2017). It is important to emphasize that the result of SADIRE is related to the $\mathbb{R}^2$ embedding generated by the dimensionality reduction technique, that is, the main purpose of SADIRE is to preserve the structures of the projected dataset in $\mathbb{R}^2$. In this case, the evaluation of how a method is reflecting the high-dimensional space must be performed considering the high-dimensional space and the dimensionality reduction technique.

The numerical evaluation uses the concept of representativeness (MA; WEI; CHEN, 2011) given by the metrics redundancy and coverage. While the *Redundancy* metric is used to assess how redundant is a sampled set, the *Coverage* metric assess how a sampled set can represent the information in the original set. The following sections provide analyses based on the *Redundancy* and *Coverage* of the representative sets based on DR results visualized by scatterplots. Then, SADIRE is evaluated on clustering task using *silhouette coefficient*.

2.3.1 REDUNDANCY

The idea to understand the redundancy of a dataset is to verify whether sampled point can be removed without information loss. That is, if two or more points are too close to each other, only one point can represent them due to their similarity. Equation 2.1 shows how to compute *Redundancy*, which is simply the mean of the lowest distances of

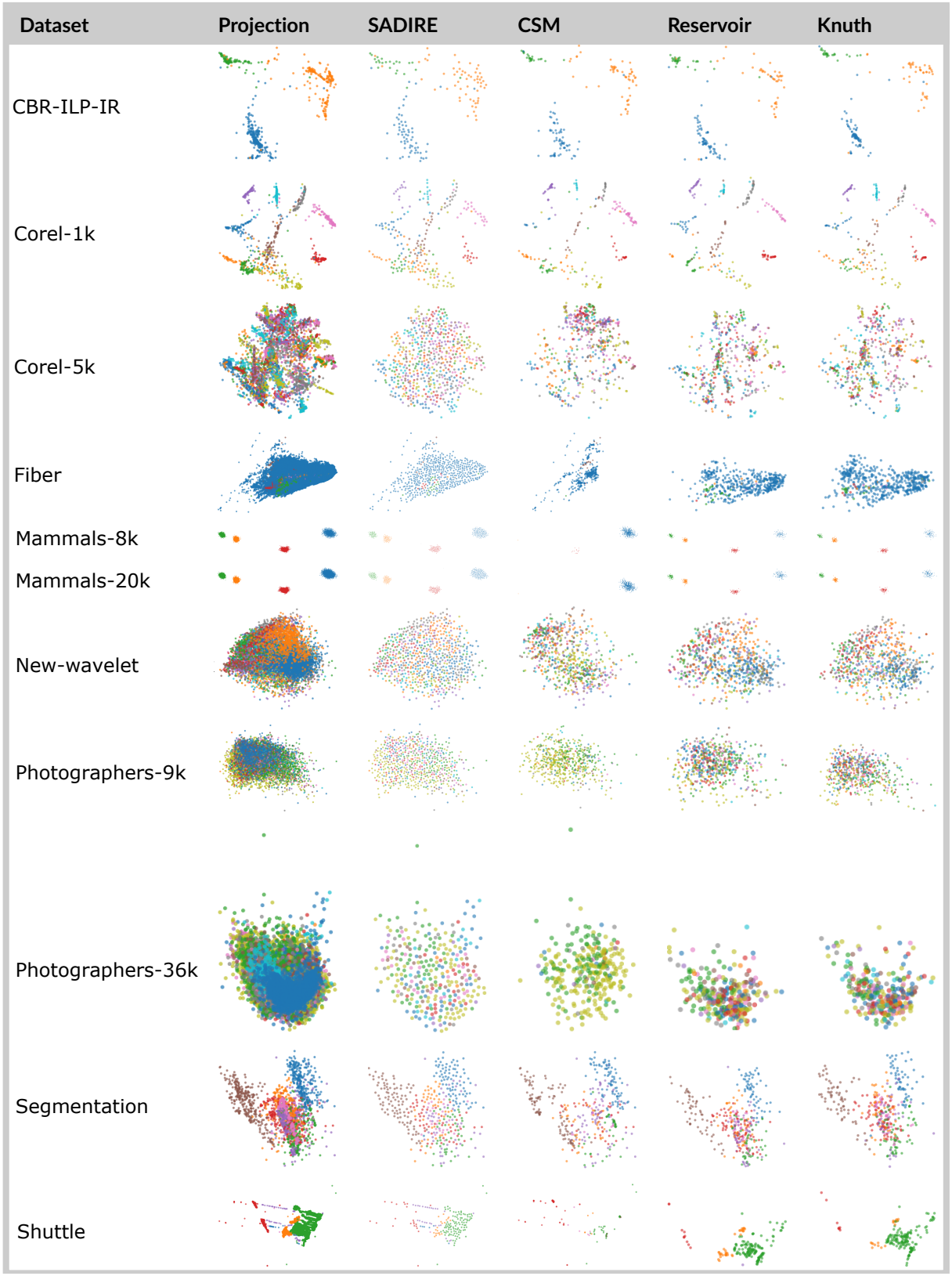


Figure 3 – Initial projections of the datasets and their respective sampled sets according to the sampling techniques. The color of the circles depict the different classes in the dataset.

each sampled point to the other points in the sampled set. In the equation, $\| \bullet \|$ stands for Euclidean distance, X is the sampled set, and x' and x are different sampled points on the visual space. Notice that higher values will encode better quality since the sampled points are far from each other.

$$\frac{1}{N} \sum_{x \in X} min_{x' \in X}(\| x' - x \|). \tag{2.1}$$

Fig. 4 shows the results for the redundancy metric. Due to their design, *Knuth* and *Reservoir* techniques perform poorly according to redundancy—the sampled points are too close to each other. Notice in Fig. 3 that the denser is the area of the projection, the greater is the number of points retrieved from these techniques. On the other hand, *CSM* technique tends to select more points from high variability classes (see sampled points of *mammals-8k*, *mammals-20k*, and *shuttle* of Fig. 3). Our technique is designed to not return redundant points, so that, it had reached the best results.

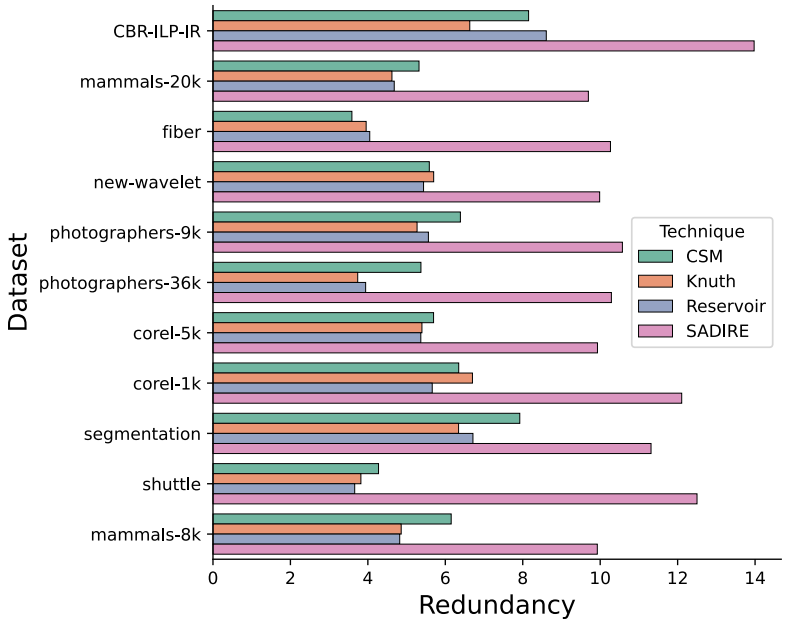


Figure 4 – *Redundancy* metric measures how the sampled set is redundant by seeing how each point correlates with each other. The higher the value, the lower a set is redundant.

2.3.2 COVERAGE

We also rely on spatial proximity to assess the coverage of a given representative set. In this case, we evaluate how far from a sampled point is a non-sampled point. Equation 2.2 shows the *Coverage* metric, where lower values encode better quality—notice

that x represent each sampled point in the sampled set (X), and y represent each non-sampled point in the original set (Y). Besides analyzing the returned value, the standard variation of the differences in Equation 2.2 is used to assess if there are points too far from the sampled set.

$$\frac{1}{N}\sum_{y\in Y} min_{x\in X}(\parallel y - x \parallel). \tag{2.2}$$

Fig. 5 shows the results for coverage metric. For a few cases, SADIRE presents (for *fiber* and *photographers-36k*, for example) higher results than *Reservoir* and *Knuth*'s. This is due to the fact that the majority of the points selected from them are concentrated in denser areas—mean of lower distances gives lower values. However, the standard deviation shows that a few points are too far from a representative (sampled point), which implies in bad results for *Knuth* and *Reservoir* techniques. *CSM* results are related to its formulation, in which there is no way to control where to focus for extracting sample points.

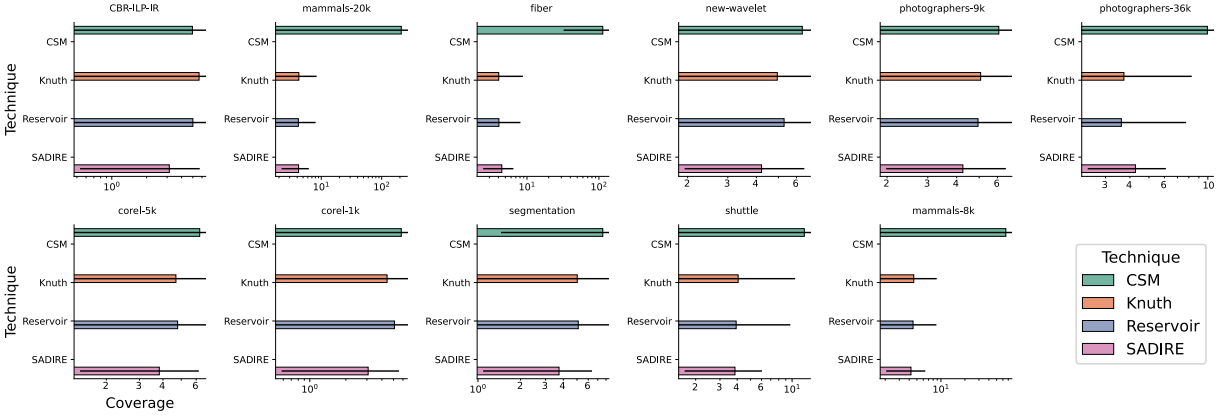


Figure 5 – *Coverage* metric to assess if a sampled set is able to represent all data structures. The standard deviation is represented by error bars. Notice that, greater standard deviation indicates that some instances are far from a representative (an instance from sampled set).

2.3.3 CLUSTERING

As in Joia et al.'s work (JOIA; PETRONETTO; NONATO, 2015), we also evaluated our technique in the clustering task.. Thus, *k-Means* (MACQUEEN, 1967) and *Expectation Maximization* (*EM*) (DEMPSTER; LAIRD; RUBIN, 1977) algorithms were applied in the *SADIRE* and *CSM* sampled points, then each non-sampled point was associated to the cluster of the closest medoid—the reason that we only compared our technique against CSM is that Joia et al. (JOIA; PETRONETTO; NONATO, 2015) have already compared CSM with another methods. The number of clusters were defined as the same number of classes of each dataset. Fig. 6 shows the *silhouette coefficient* (TAN; STEINBACH; KUMAR, 2006) for the techniques compared with the *EM* and *k-Means* algorithms, where

values range from -1 to 1 and greater values indicate better cohesion and separation among clusters.

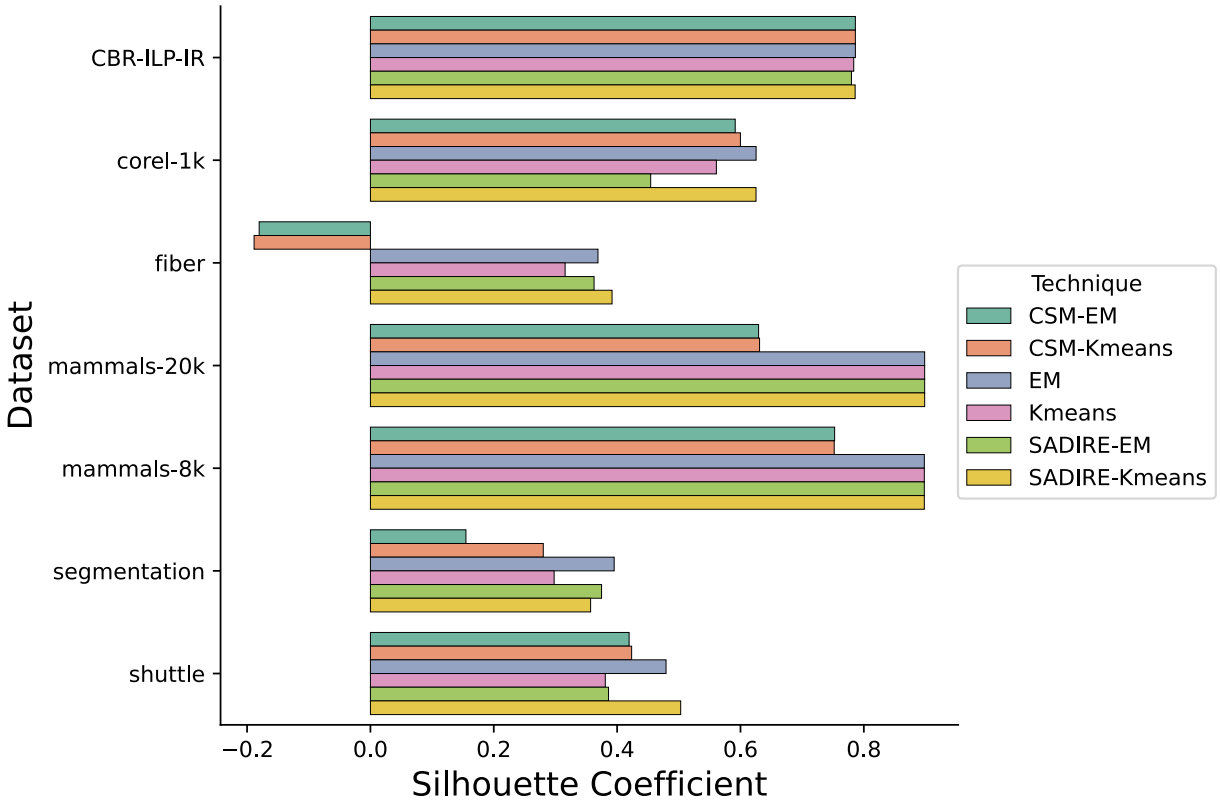


Figure 6 – *Silhouette coefficient* of the techniques applied for clustering data.

SADIRE performed similar to the plain clustering processes, reflecting the structures of these datasets. *CSM* technique, on the other hand, tends to negligence classes with lower variation, which negatively affects the clustering process.

2.3.4 RUNNING TIME

Fig. 7 shows the running time of SADIRE for the datasets in Table 1. Note that rather than the size of the datasets, the running times are more related to the density of the clusters imposed by the dimensionality reduction techniques on the $\mathbb{R}^2$ embedding—see, for example, the time taken for sampling from the *shuttle* dataset, which is the second biggest dataset in terms if size of data points. This analysis gives us the insight that although our algorithm has its greater complexity of searching for outliers in the worst-case scenario, in practice, the complexity of inserting and searching data points in a quad-tree dominates the other steps.

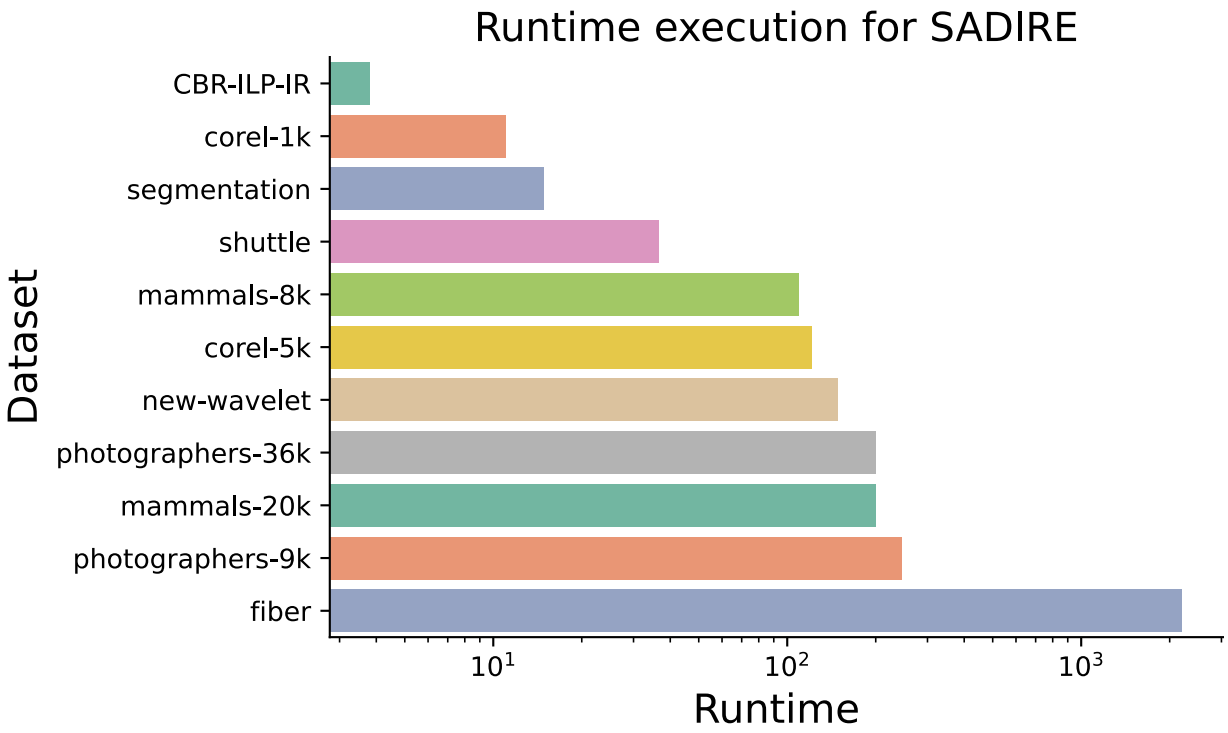


Figure 7 – Runtime of SADIRE algorithm. We can see that the algorithm takes more time for sampling from high-density embeddings.

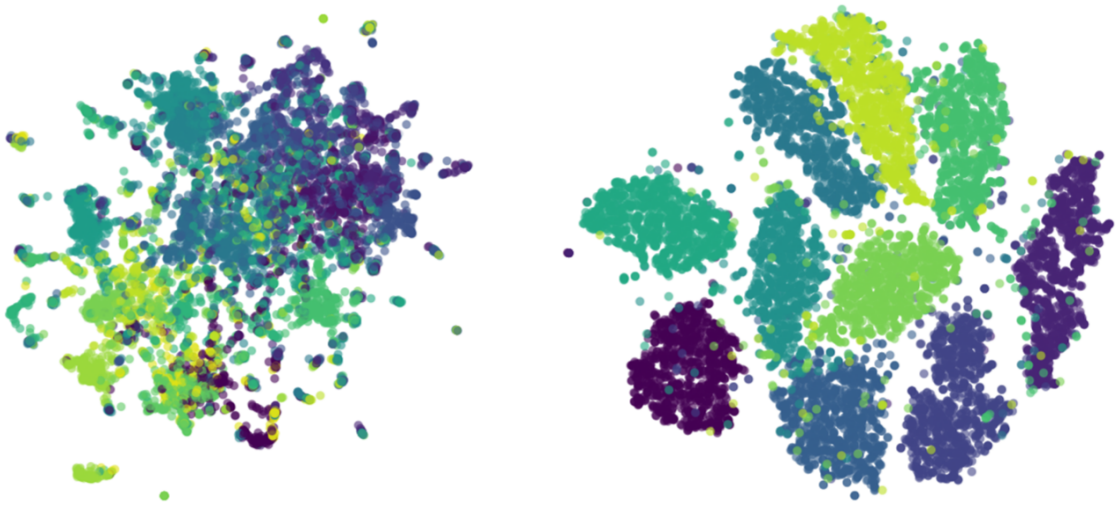
2.3.5 PARAMETER ANALYSIS

This section inspects the influence of the SADIRE's k and α parameters. We used the *20 Newsgroups* [6] and *MNIST* (LECUN; CORTES, 2010) datasets projected with UMAP (MCINNES; HEALY; MELVILLE, 2018) and t-SNE (MAATEN; HINTON, 2008), as shown in Fig. 8. The projection of the *20 Newsgroups* dataset was messier in terms of class boundaries. However, as seen in Fig. 9a, SADIRE conveys the structures and even outliers of the embedding for lower values of k and α.

Figs. 9a and 9b show the influence of the parameters k and α for the *20 Newsgroups* and *MNIST* datasets, respectively. The first characteristic is that the k and α parameters are inversely proportional to the number of representatives and the redundancy. Moreover, the layout formed by the representative set lose its similarity to the original embedding for greater values of these parameters, deceiving user exploration. The second interesting pattern is that all of the layouts using $k = 5$ show high similarity to the original embedding, while a combination of higher values for k and α produce poor layouts in terms of similarity to the original embedding. Such characteristics can indicate that k is the most critical parameter.

Depending on the application, SADIRE can be used with various combinations of parameters. However, a suggestion that balances similarity to the original embedding

[6] http://qwone.com/ jason/20Newsgroups/



(a) *20 Newsgroups* projected using UMAP. (b) *MNIST* projected using t-SNE.

Figure 8 – UMAP and t-SNE projections.

and lower redundancy is to use $k = 5$ or 10. To support our guiding, the following section provides an user evaluation on the similarity between SADIRE samples and original scatterplots.

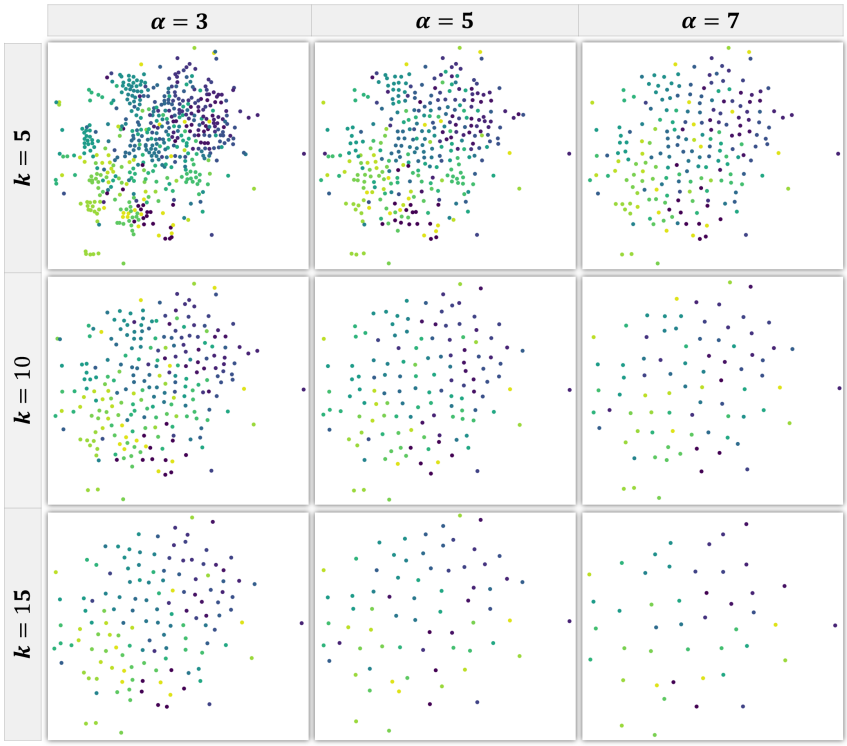
2.4 USER EXPERIMENT

In this user experiment, the representative sets were evaluated according to their similarity to the initial embeddings and according to visual clutter. For instance, we used four datasets with incremental visual disorder to perform the experiment. For each dataset, we used six different SADIRE's parameters configurations: $k = 5$, $\alpha = 3$; $k = 10$, $\alpha = 3$; $k = 10$, $\alpha = 5$; $k = 10$, $\alpha = 7$; $k = 20$, $\alpha = 5$; $k = 20$, $\alpha = 7$. The initial embeddings and layouts produced by sampling using SADIRE are shown in Fig. 10—where the circle encodes density. Due to their similarity to the other layouts, a few layout configurations were omitted, such as $k = 5$, $alpha = 3$ and $k = 20$, $alpha = 3$.

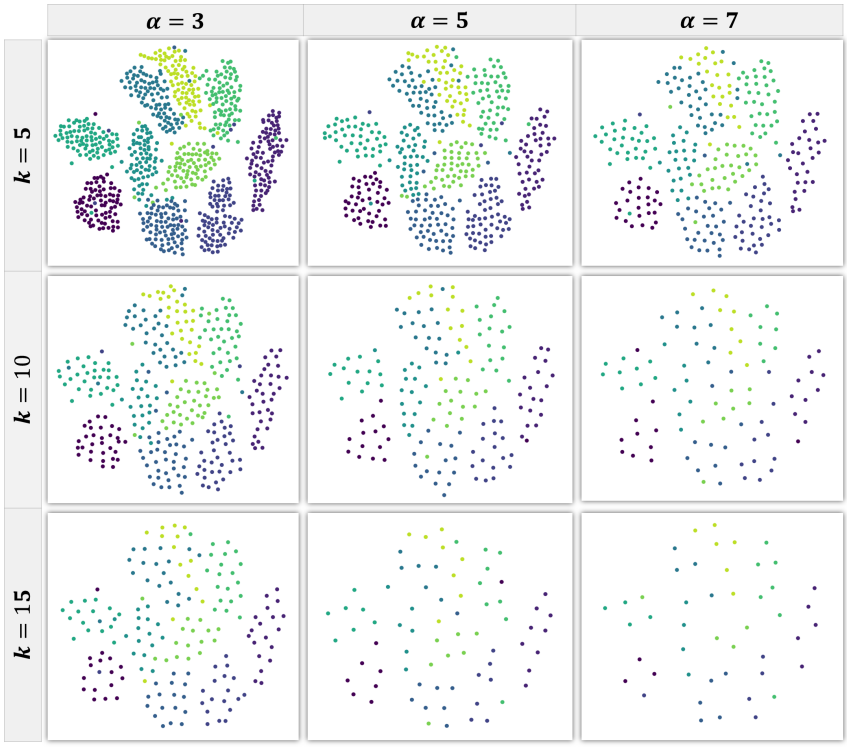
The experiment was performed by 42 participants, two graduated students and 40 computer science students. The participants were given two tasks based on the six representative layouts in Fig. 10:

T1: To classify representative layouts from the least visually cluttered to the most visually cluttered;

T2: To rate (from 0 to 10) the representative layouts according to their similarity to the initial embedding, where 0 stands for the *least* similar and 10 stands for the *most* similar.



(a) Influence of the parameters on the *20 Newsgroups* dataset.



(b) Influence of the parameters on the *MNIST* dataset.

Figure 9 – Influence of k and α parameters on the returned representative set.

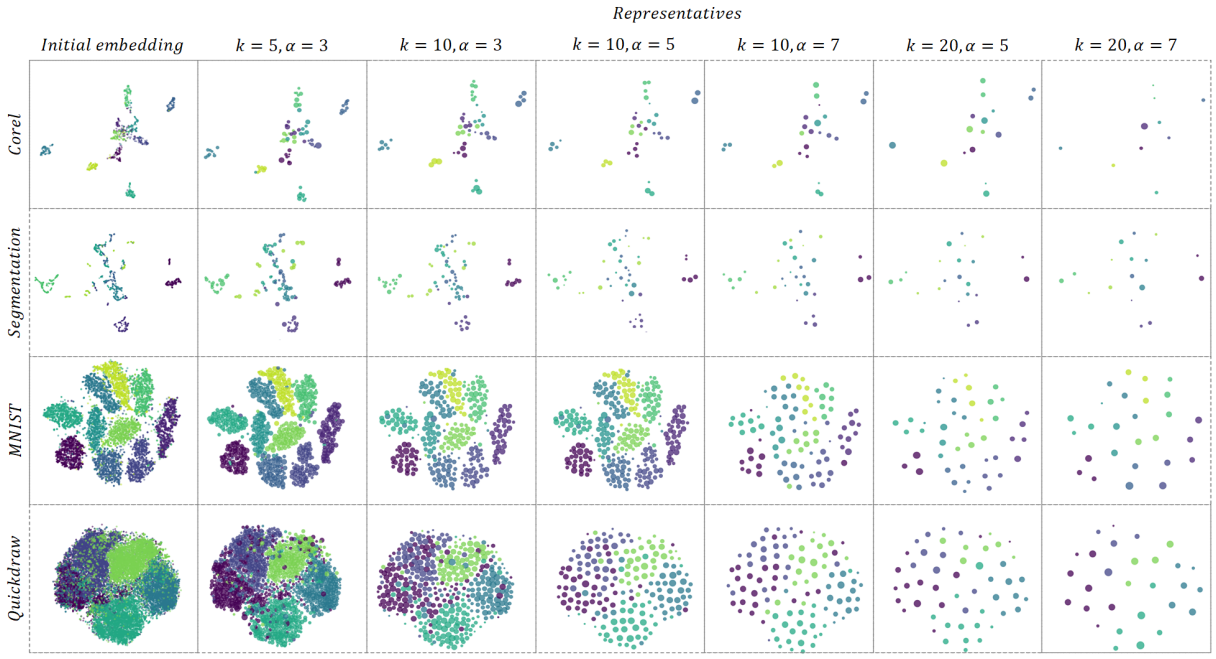


Figure 10 – Initial embeddings and the layouts that result from reducing visual disorder with representative selection. SADIRE was applied to several configurations by varying the k and α parameters.

For **T1**, the six layout configurations were presented all at once for the participants to order—we did not provide the initial embedding to avoid biasing the participants. To facilitate the rating in **T2**, the six layouts were presented to the participants one by one, following the initial embedding. Fig. 11 depicts users' perceptions of the layouts produced by the sampling technique. The rows in each heatmap represent the six different representative layouts (see Fig. 10) generated by varying SADIRE's parameters, while the columns represent the visual clutter ordering (from least to most visually cluttered) and the similarity rating. Each cell's color represents how many times a participant placed a representative layout in that position (for visual disorder) and how many times that participant rated that representative layout (for similarity). It should be noted that for the majority of the participants, a layout was assigned a position or received a similar rating.

On the heatmaps for **T1**, we can see a pattern indicating that as the number of representatives decreases, the visual disorder decreases as well, according to the participants. In other words, the visual clutter on the projection is related to the number of circles on the plane by the participants. For **T2**, we can see that the similarity of the representative layouts to the initial layout is inversely proportional, implying that as the number of data instances projected onto the plane decreases, information is lost. Given these two main patterns, representative layouts that achieve a balance between visual disorder and layout similarity are the best option. Another intriguing pattern can be seen in the elements of the heatmaps depicting visual disorder's lower triangular matrix. This pattern can be explained by the fact that users may struggle to analyze representative layouts with fewer

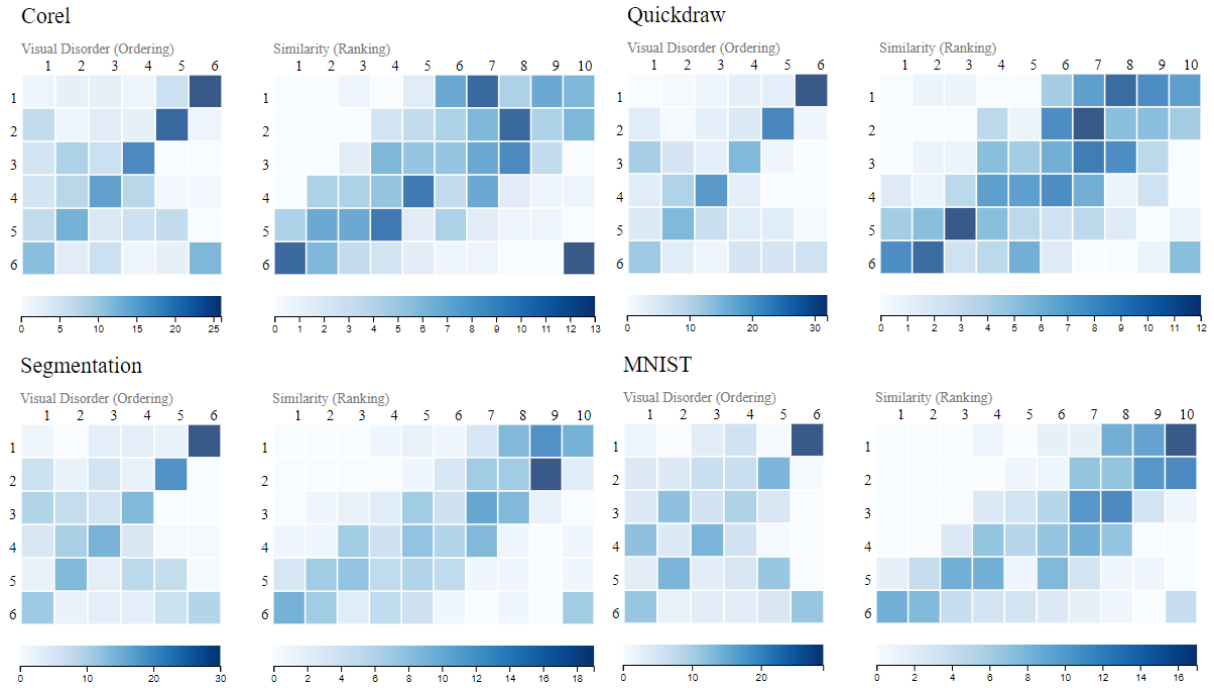


Figure 11 – The outcomes of the user experiments. Each line denotes a unique representative selection configuration. The heatmap cells encode how many times a representative layout was placed in that order position (visual disorder) and how many times a representative layout was assigned a rank number for each dataset (similarity).

data points, at which point the embeddings begin to lose information.

There is a link between coverage and layout similarity, as well as redundancy and visual clutter/disorder. According to participant responses, as the number of representatives is increased, the resulting set becomes more similar to the initial embedding (a natural observation). Furthermore, the initial embedding is better covered because it is represented by a greater number of data instances. On the other hand, we can see that in visually cluttered layouts, redundancy is higher (i.e., very similar data instances are projected close to each other). This analysis backs up our claim that a good representative selection technique imposes a balance between coverage (as layout similarity) and redundancy (as visual clutter).

2.5 PRACTICAL APPLICATION

Fig. 12 depicts SADIRE's suitability for representing the first level of ExplorerTree (MARCíLIO-JR et al., 2021a), a hierarchical scatterplot navigation approach. On the left is an LSP (PAULOVICH et al., 2008) projection of the Corel dataset, with 5,000 data points colored by class. On the right, we show the first level of ExplorerTree using the same sampling strategies on the same dataset. Unlike the other techniques, SADIRE uses sampled data points to cover the entire projected space, while the circle area accurately

depicts the densest parts of the projection. Many structures encoded by SADIRE are lost in Reservoir and CSM—these structures are highlighted in orange. Finally, bisecting k-means fails to encode density information well, aggregating data points that do not correspond to the density of the projection—see the structures highlighted in blue

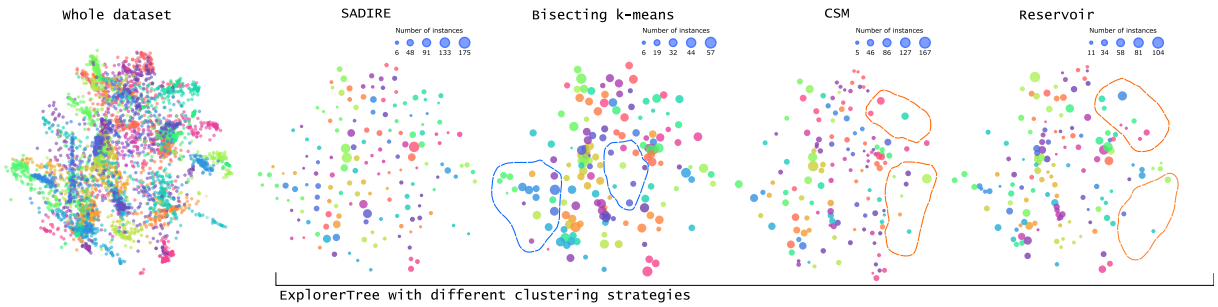


Figure 12 – ExplorerTree using different sampling algorithms. On the left, the traditional scatter-plot representation. On the right, the first level of ExplorerTree using four sampling strategies.

Finally, in Appendix A, the results and rationale for using SADIRE in the ExplorerTree technique are documented, along with additional user experiments.

2.6 FINAL REMARKS

Sampling mechanisms are an important tool for dealing with increasing data availability, the high complexity of computational methods, and the interaction requirements of user-centered approaches.

In the preceding sections, we show how our sampling method can be used to generate meaningful representative sets of visual representations of dimensionality reduction (DR) results. As a result, SADIRE is dependent on the ability of DR techniques to represent the underlying data structures in multidimensional space. This means that our technique is appropriate for visual tasks in which datasets are represented by $\mathbb{R}^2$ embeddings and a thorough examination of the DR technique and feature space has been performed.

Another intriguing feature of our proposal is that the sampling mechanism can be modified to convey cluster density—this can be accomplished by returning a percentage of data points (or simply more than one data point) from each cell (see Section 2.2.1). Finally, the parameters k and α can be used to adjust the size and redundancy of the representative set.

3 CONTRASTIVE ANALYSIS FOR SCATTERPLOT REPRESENTATIONS OF DIMENSIONALITY REDUCTION

Wilson E. Marcílio-Jr, Danilo M. Eler, Rogério E. Garcia *Contrastive analysis for scatterplot-based representations of dimensionality reduction*, Computers & Graphics, Volume 101, 2021.

Analysts inspect clusters using scatterplot representations of Dimensionality Reduction (DR) results to understand data nuances and features' contributions to embedding in the projected space. Contrastive analysis (FUJIWARA; KWON; MA, 2019; LE; AKOGLU, 2019) is a promising strategy for analyzing DR results that involves understanding how features differentiate and contribute to the formation of clusters in the visual space ($\mathbb{R}^2$). Such an approach has a wide range of practical applications. A tool for labeling textual data, for example, would benefit from contrastive analysis to highlight the differences among the clusters (e.g., topics)—these clusters can represent candidates for new classes due to their distinct characteristics. Another important application is determining which characteristics describe two distinct groups of patients following medical experiments (ABID et al., 2018).

Only a few studies provide contrastive analysis (FUJIWARA; KWON; MA, 2019; LE; AKOGLU, 2019). Instead, the literature presents a small number of studies that support the analysis of DR results via visualization of global information (TURKAY et al., 2012; JOIA; PETRONETTO; NONATO, 2015; COIMBRA et al., 2016; STAHNKE et al., 2016; PAGLIOSA; PAGLIOSA; NONATO, 2016), emphasizing the importance of features (attributes) imposed on the embedded space. The principal components (PCs) from PCA (JOLLIFFE, 1986) are used in the main approaches to determine which features contribute to cluster formation (TURKAY et al., 2012; JOIA; PETRONETTO; NONATO, 2015). These contributions, however, do not highlight the clusters' distinct characteristics. ccPCA (FUJIWARA; KWON; MA, 2019), on the other hand, uses contrastive PCA (ABDI; VALENTIN, 2007) to find contrastive information (i.e., the unique characteristics) for each cluster. The main limitation of ccPCA stems from PCA, specifically the prohibitively long run-time execution time for high-dimensional datasets—it takes $O(m^3)$ on the number of dimensions. ContraVis (LE; AKOGLU, 2019), another approach, is only applicable to textual data and cannot assist in the interpretation of DR layouts because it is already a dimensionality reduction approach. Furthermore, the study does not present a strategy for comprehending the contributions of the terms to the embedding organization.

In this chapter, we will discuss cExpression, a method for analyzing DR results that employs contrastive analysis and a carefully designed visualization tool. Specifically, cExpression uses statistical variables (p-values and t-scores) to determine the most distinguishing features of clusters (t-scores) and the level of confidence in the results (p-values). Although the t-test is a common method for feature selection in machine learning and cell type definition in bioinformatics, it has not been extensively studied for analyzing dimensionality reduction results. Users can interact with scatterplot representations of high-dimensional datasets in the visualization design to visualize the cluster summaries—designed after the definition of several requirements. To communicate the relationship between t-scores and p-values, we use focus+context interaction on a bipartite graph. The focus+context interaction allows users to explore more information while inspecting the histograms of small groups of features. A heatmap representation of the most distinguishing features of each cluster also aids in providing an overview of the structures. Finally, we propose an encoding strategy for communicating the distribution of feature values in the scatterplot representation at the same time.

cExpression, as demonstrated in the numerical experiments, can be applied to a variety of data types and is scalable to handle large datasets with thousands of dimensions. While the computational components of our approach help generate contrastive information quickly, the visualization design is simple and effective for analyzing even complex textual data.

In summary, the contributions of this chapter are:

- A strategy to analyze and interpret dimensionality reduction through clusters using contrastive analysis;
- Novel visualization strategies to analyze the relationship among statistical variables and simultaneously visualize various features in the scatterplot;

The chapter is organized as follows: Section 5.1 presents the related works; Section 3.2 delineates our methodology accompanied by motivation and the visualization design; Section 5.3 shows the case studies; Section 3.4 shows numerical evaluation; Section 4.6 presents discussions about the work.

3.1 RELATED WORKS

Layout enrichment strategies combine visualization approaches and valuable information extracted from data on high-dimensional space concerning low-dimensional representations, typically on $\mathbb{R}^2$, to aid in the analysis of dimensionality reduction (DR)

results. Using bar charts and color encoding to understand three-dimensional projections (COIMBRA et al., 2016), for example, or encoding attribute variation using Delaunay triangulation to assess neighborhood relations on $\mathbb{R}^2$ (SILVA et al., 2015). Another interesting work, proposed by Tian et al. (TIAN et al., 2021), extends Silva et al.'s (SILVA et al., 2015) by aggregating the explanatory mechanisms using a visual analytics system supported by a local explanation method (variance ratio) that describes the dimensionality of local neighborhoods. To understand distortions in the projection process, Probing Projections (STAHNKE et al., 2016), for example, displays error information by displaying a halo around each circle in a scatterplot with interaction mechanisms. The majority of the works visualize attribute variability (PAGLIOSA; PAGLIOSA; NONATO, 2016), neighborhood and class errors (MARCILIO; ELER; GARCIA, 2017), or quality metrics (KWON et al., 2018) using traditional statistical charts. Martins et al. (MARTINS et al., 2014), for example, use space-filling techniques to assist users in reasoning about the impact of parameterized projections on neighborhood preservation and other quality aspects.

Other techniques identify significant features in data clusters. Linear Discriminative Coordinates (WANG et al., 2017), for example, employ LDA (IZENMAN, 2008) to generate cohesive clusters by discarding the least important features. Joia et al. (JOIA; PETRONETTO; NONATO, 2015) use PCA to identify the most important features using a simple matrix transposition, and then visualize feature names as word clouds within each cluster region—this approach is influenced by classes with higher variation. Our approach, on the other hand, overcomes these issues by comparing distribution differences and returning the confidence levels associated with such differences. Turkay et al. (TURKAY et al., 2012) propose using the principal components computed by PCA to obtain the representative features of MDS (KRUSKAL; WISH, 1978). The issue with using PCA is that it focuses on the global characteristics of datasets, making it difficult to analyze the unique characteristics of clusters when we use contrastive analysis.

Fujiwara et al. (FUJIWARA; KWON; MA, 2019) proposed the contrastive cluster PCA (ccPCA) technique, which finds the most important features for each cluster in a projection. Fujiwara et al.'s approach differs from that of Joia et al. and Turkay et al. in that it demonstrates which features significantly contribute to cluster differentiation. The main limitation of ccPCA is its prohibitively long run-time execution for datasets with many dimensions, such as document collections (LOPES et al., 2007; WANG et al., 2013; ELER et al., 2018; FELIX; DASGUPTA; BERTINI, 2018), gene expressions (MURIE et al., 2008; ROBINSON; MCCARTHY; SMYTH, 2009; RITCHIE et al., 2015; HÖLLT et al., 2018), or filter activations (MARCíLIO-JR et al., 2020). Finally, there is no consistent way to visually correlate the value distribution with the feature contribution returned by contrastive PCA. Unlike ccPCA, cExpression successfully analyzes datasets with very high dimensionality while producing more consistent results. Furthermore, because it is based on well-known statistical principles, it produces more interpretable results. Another

interesting work, proposed by Le et al. (LE; AKOGLU, 2019), combines topic extraction with visualization to bring the distinct characteristics of document collections to light. Although their technique is effective at distinguishing clusters, it cannot be used to analyze other dimensionality reduction approaches and is restricted to textual data. cExpression is independent of dimensionality reduction techniques and can be used to analyze a wide range of data types.

The section that follows covers cExpression, which includes a method for retrieving contrastive information as well as a visualization tool.

3.2 CEXPRESSION—TOOL FOR CONTRASTIVE ANALYSIS OF DR RESULTS

Fig. 13 presents the workflow for using cExpression to interpret clusters after dimensionality reduction before delving into details in the following sections. First, the user must preprocess a high-dimensional dataset by using a dimensionality reduction technique and annotating the visual space clusters, which results in the state (A). Then, for each pair of feature and cluster, we use the t-test to compute a measure of deviation (t-score) and associated confidence level (p-value). For example, the state (B) in Fig. 13 represents two distributions associated with the feature f_2, one for the cluster 0 and another for the rest of the dataset. The t-test comparison answers the question, "Is this feature a unique characteristic for cluster 0?" Finally, the data generated by this process is fed into a visualization tool (C) that aids in the exploratory process by using coordinated views.

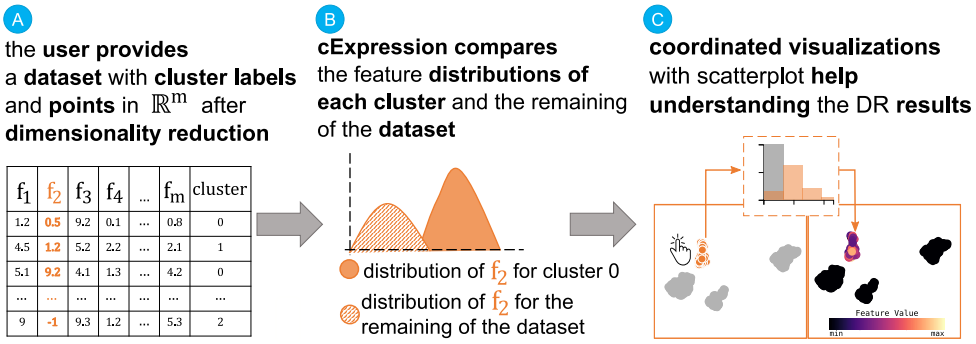


Figure 13 – The cExpression pipeline. The user submits a high-dimensional dataset with $\mathbb{R}^2$ coordinates and cluster labels after dimensionality reduction (A). The contrastive information for each pair of cluster and feature is computed using the clustering labels and the high-dimensional data (B). The data generated from this process is used to create visualization approaches that highlight the unique characteristics of each cluster (C).

3.2.1 COMPUTING CONSTRASTIVE INFORMATION

cExpression consists of comparing value distributions of different clusters using a t-test to determine if the differences between these distributions are significant enough to describe clusters. Given a dataset X, let X_f^c and $X_f^{c'}$ be the feature f values for data observations in the cluster c and the feature f values for data observations **not** in the cluster c (c')—c and c' are visual space clusters from which users want to find unique characteristics. The summary statistics are then computed to obtain the t value for the t-test: average and variance for X_f^c and $Y_f^{c'}$, defined as $\overline{X_f^c}$, $\overline{X_f^{c'}}$ and $Var(X_f^c)$, $Var(X_f^{c'})$. Finally, the t-statistic is computed as follows:

$$t = \frac{\overline{X_f^c} - \overline{X_f^{c'}}}{\sqrt{\frac{Var(X_f^c)}{|X_f^c|} + \frac{Var(X_f^{c'})}{|X_f^{c'}|}}} \tag{3.1}$$

in which $|\bullet|$ stands for the size of the sets. With the t-statistic, we use a statistical library to calculate the p-value. The p-value denotes the likelihood that the distribution of values in cluster c for feature f (X_f^c) and the distribution of values not in cluster c for feature f ($X_f^{c'}$) are equal. When a cluster c and a feature f are fixed, smaller p-values have high importance because they indicate (in probabilistic terms) that the distributions are less likely to be equal—t-scores are measures of standard deviation. In this case, extremely high or extremely low (negative) t-scores are associated with extremely small p-values and are found in the t-tails, distribution's as illustrated in Fig. 14(**d**). The variance-based explanation (SILVA et al., 2015; TIAN et al., 2021), a similar approach to generating explanations, computes the relative variance of a neighborhood across the entire dataset, emphasizing how dimensions contribute to similarity within local neighborhoods—such an explanation approach emphasizes which dimension makes their neighbors similar, whereas the t-score is a measure to understand how far two distributions deviate from each other. The main difference between these two approaches is how they determine the relevance of these features. While the t-score compares disjoint sets, the variance-based explanation considers neighborhood contribution across the entire dataset.

To interpret clusters using contrastive analysis, we iterate the preceding steps for each pair of clusters and feature of the dataset. Then, t-scores are used as a ranking metric. It is critical to emphasize that these clusters must be perceived in visual space in order for the insight generated by the exploratory process to be consistent with the organization depicted by the scatterplot.

Fig. 14 depicts the idea of a fixed cluster and a fixed feature. The distributions (**b**) of values for the cluster and feature are generated using the dataset in **a.**—red encodes the feature distribution of data samples within the cluster, while gray encodes the distribution of data samples outside the cluster. Based on these distributions, we used the t-Student test

(**c**) to determine the probability (p-value, **d**) to observe the t-score when the distributions are similar. Finally, ordering the cluster's features in descending order of absolute t-score will result in the most distinguishing features of the cluster being placed on top.

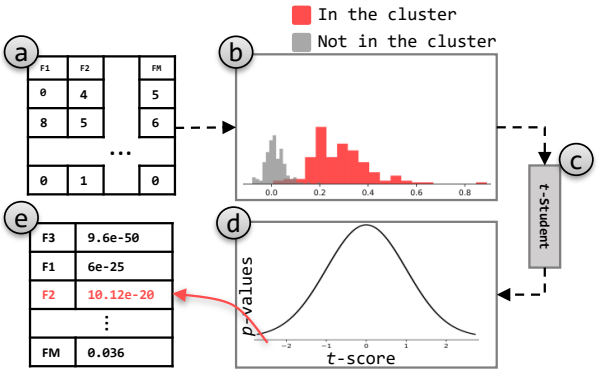


Figure 14 – Defining distinctive features. The feature's distribution (**a**) in a cluster of interest is compared to its distribution on the remaining dataset (**b**). Then, the t-Student test (**c**) returns the probability of these two distributions to be different (**d**). Finally, the features are arranged in decreasing order according to their t-score to communicate discriminative power (**e**).

Consider applying the concept to a multivariate dataset. Fig. 15 (left) depicts a UMAP projection of the Vertebral dataset, which includes 310 data instances described by six biomechanical features derived from the shape and orientation of the pelvis and lumbar spine: `pelvic incidence`, `pelvic tilt`, `lumbar lordosis angle`, `sacral slope`, `pelvic radius`, `d. of spondylolisthesis`. The patients of interest have spondylolisthesis, a spine disorder in which a bone (vertebra) slides forward over the bone below it.

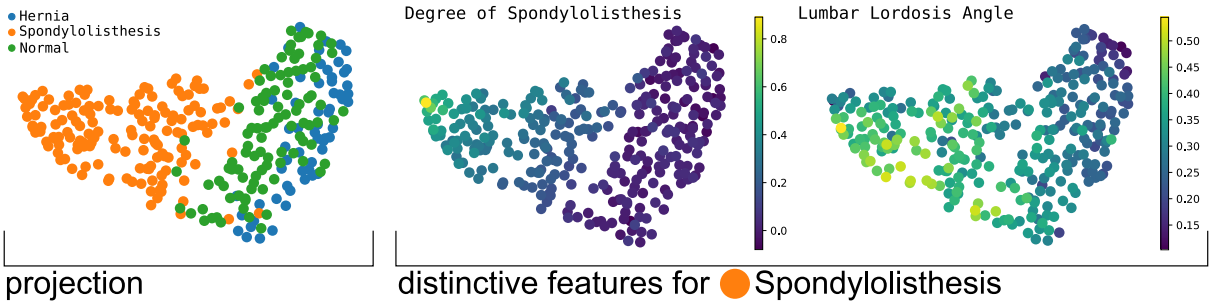


Figure 15 – UMAP projection of the *Vertebral Column* dataset and two distinctive features for the Spondylolisthesis class.

The `degree of spondylolisthesis`, is a critical feature for those with Spondylolisthesis. It is reasonable to believe that an algorithm looking for important features would choose the `degree of spondylolisthesis` to define such a class. To investigate cExpression on this task, we look at the distribution of values for the key features throughout the projection. The two most important ones retrieved by cExpression are shown in Fig. 15 (right). The scatterplot of normalized feature values confirms that the degree of

spondylolisthesis is a viable candidate for a distinguishing feature. In other words, data instances from class ● have significantly higher values for this feature than data instances from classes ● and ●.

3.2.2 VISUALIZATION DESIGN

We designed a few visualization strategies to analyze dimensionality reduction results with cExpression. The tool in Fig. 16 displays two main views: scatterplot and heatmap. To begin, users must supply the high-dimensional dataset and the points after dimensionality reduction ($\mathbb{R}^2$), as well as the cell size for the cell-based encoding visualization and the number of features automatically inspected using the summary visualization (**a**). The scatterplot representation of the projected dataset (**b**) is used to interact with the bipartite graph (**c**), which represents the relationship between the statistical variables. In Fig. 16, for example, the bipartite graph displays information for the cluster ●. To inspect the scatterplot representation distribution, users can toggle the distribution plots (**d** and **e**). The heatmap view (**f**) summarizes the feature importance, with the t-scores encoded by a diverging color scale from purple to green. The heatmap's cell sizes represent the number of decimal places in the associated p-value. In the heatmap visualization, high t-scores (high color saturation) are associated with lower p-values (larger tile sizes). In the following section, we explain the decisions about visualization encodings by first defining the task and visual requirements.

3.2.2.1 Visual and task requirements

To create our visualizations, we follow the layout enrichment (NONATO; AUPETIT, 2018) strategy of scatterplot representations after dimensionality reduction. As a result, the following task requirements (TRs) assist us in comprehending cluster formation in DR results (projection):

- **TR1:** Compare a cluster against the remaining of the dataset;
- **TR2:** Compare two selected clusters.

The task requirements (TRs) assist users in comprehending the projection structures by comparing how clusters differ. To accomplish these tasks, we define the minimum requirements for the visualization tool:

- **VR1:** Visualize the importance of a feature;
- **VR2:** Visualize the distribution of values of the comparing components;
- **VR3:** Know the feature name;

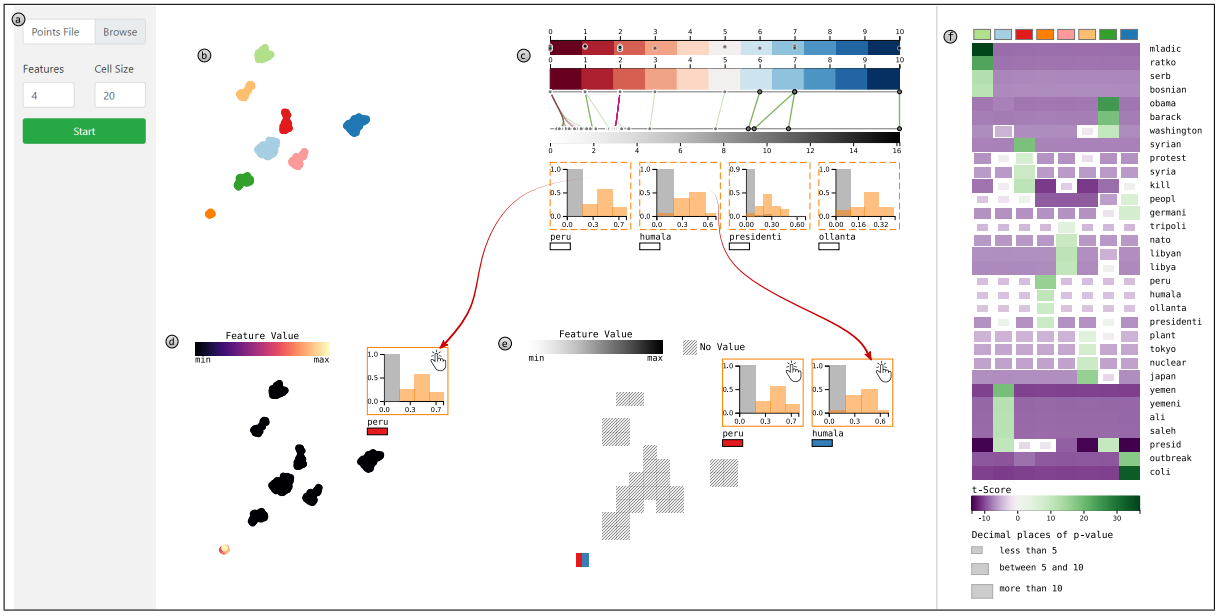


Figure 16 – The cExpression visualization tool. Users provide high-dimensional points with annotated clusters and positions onto $\mathbb{R}^2$ (a). The $\mathbb{R}^2$ coordinates are used for the scatterplot layout (c), from which the analysis starts. Users click on points in a cluster to investigate their contrastive information. For instance, the bipartite graph (c) shows the distinctive terms for the dark-orange cluster. There is also an interaction mechanism to visualize the feature values on the scatterplot visualization (d, e). Finally, an overview analysis can be achieved through the heatmap encoding.

- **VR4:** Assess how trustful the result is;
- **VR5:** Understand the organization of features distribution on the embedding.

During exploratory data analysis, these visualization requirements (VRs) are examined together. Users understand intra-cluster differences from the rest of the dataset (VR2) and how their values are distributed throughout the DR layout (VR5) by knowing the most discriminative features for a cluster (VR1 and VR3). VR4, which is related to p-values, assists users in understanding the statistical significance of the difference.

Finally, because tasks TR1 and TR2 are not novel when it comes to scatterplot visualizations (SEDLMAIR; MUNZNER; TORY, 2013; MICALLEF et al., 2017; SARIKAYA; GLEICHER, 2018), other works address the same issue. One similar study (RAUBER et al., 2015) uses feature selection to re-project high-dimensional data onto $\mathbb{R}^2$ to induce cluster separation and aid in classification tasks. Instead, we use contrastive analysis to explain which features are important for cluster formation—we want to improve understanding, not the projection itself.

3.2.2.2 Visualizing the feature importance

We visualize the importance of the features as their position over a horizontal axis to compactly represent the VR1-4 requirements, while providing distribution plots for each visualized feature. The visualization design is shown in Fig. 17, where the color hue for the distribution plots corresponds to cluster ●, and the operation illustrated in Fig. 16 (c).

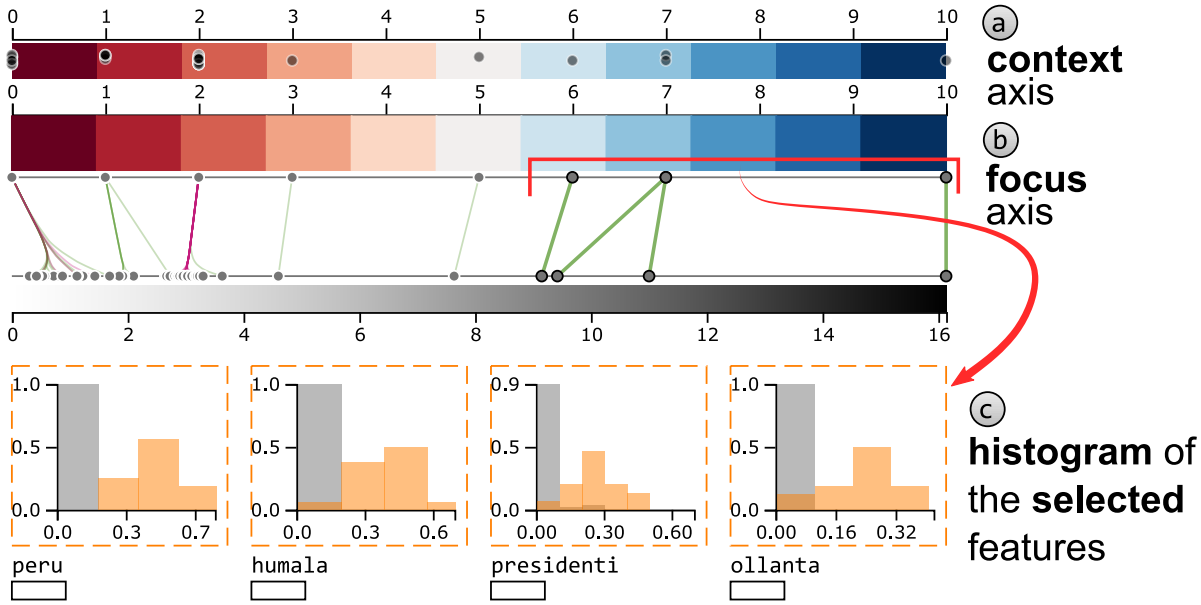
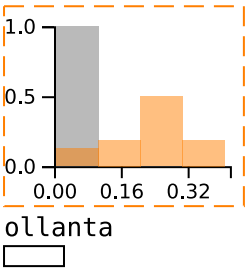


Figure 17 – The bipartite graph encoding feature importance. The p-values and t-values are encoded using red-to-blue and continuous color scales, respectively. It is worth noting that we display the number of decimal places after zero rather than the p-value itself. The first axis (a) represents the context of the available features, while the second axis (b) represents the focus—in this case, they are the same. The connections between the focus axis and the t-score represent the relationship between p-values and t-scores, with green representing positive values and red representing negative values. Finally, the four most distinguishing characteristics are pre-selected and displayed as a histogram plot (c).

A line segment representing the relationship between p-values and t-scores encodes the importance of the features, where the gray bar in Fig. 17 represent the absolute value for the t-score. The information presented on axes (**a**) and (**b**) in Fig. 17 is the same, namely the number of decimal places after zero for each p-value. These two axes are central our focus+context interaction—please, refer to Section 3.2.2.6 for a detailed description of the interaction mechanisms. The gray circles in the context axis (**a**) represent the analyzed features, and their horizontal position encodes the p-values; because there is no information to encode in the vertical position, we assign a random value to reduce overlapping among circles. We can see that this design meets requirements VR1 and VR4. However, we must still consider other visualization requirements in order to differentiate between features and understand the confidence in feature differentiation. Thus, the bipartite graph visualization in Fig. 17 shows the $min(m, 50)$ most important features for reducing overplotting, where

m represents the dataset's dimensionality. We focus on a subset of features independent of dataset dimensionality because the most useful information is concentrated in the top few. Furthermore, the use cases demonstrate how this approach is appropriate even for textual data, which typically contains hundreds of dimensions.

A red-to-blue color scale assists users in determining feature confidence—blueish colors encode greater confidence (or lower p-values). We encode the difference between distributions (t-scores) using another axis in addition to the confidence in the difference represented by the p-values. The t-score represents the difference in feature distributions between the analyzed cluster and the rest of the dataset. The relationship between these statistical variables is depicted by line segments, and the color represents the t-score signal (pink for negative and green for positive). When these line segments are not selected, they are drawn with greater transparency, and the edge bundling algorithm is used to reduce over-plotting (see the effect on the left side of Fig. 17). The rectangles below the feature names, as shown in the following section, aid in dynamically assigning color hues to features when exploring their distribution on the scatterplot representation.



A distribution plot enriched with other information satisfies the requirements for VR2 and VR3. A histogram is used to encode the value distribution (see Fig. 17(**c**)). The y-axis encodes the relative frequency of the bins, as in Fujiwara et al.'s work (FUJIWARA; KWON; MA, 2019), that is, the number of data observations with a particular value divided by the number of data observations in the cluster (for histograms of the cluster) or outside the cluster (for histograms of the remaining data). The color bars (and borders) of the distribution plots are the color of the corresponding cluster. The remaining dataset's distribution plots of values, on the other hand, receive gray. In Fig. 16, for example, the distribution plots (**c**) have the same color ( ) as the selected cluster (**b**).

3.2.2.3 Visualizing feature value distribution

Visualizing the distribution of feature values aids in comprehending the impact of features on the embedding. In other words, in Fig. 16 (d) and (e), the user interacts with the tool to visualize the feature values `peru` and `humala`. For a single feature, a continuous color map solves the problem (see Fig. 16 (d)). A larger number of features, on the other hand, necessitates more attention to maintain the similarity relations and analysis power of proximity-based scatterplot representations (e.g., see Fig. 16 (e)). So, inspired by Sarikayra et al.'s (SARIKAYA; GLEICHER; SZAFIR, 2018) work to visualize classification metrics of protein surfaces, we discretize the projected space in cells of fixed size (in pixels) to plot the contribution of each feature to the cell. The construction of the encoding for three features and only one cell is shown in Fig. 18.

Fig. 18(**a**) depicts a cell in the projected space with a hypothetical number of 16 data points, with color encoding serving as the most distinguishing feature for each data point (●, ●, ●). Because the markers only change according to the hue, a continuous color scale is a simple yet effective option for inspecting only one feature (**b**). The issue arises when inspecting multiple features while attempting to keep the organization imposed by the dimensionality reduction technique on the resulting layout. Thus, similar to the SADIRE (MARCíLIO-JR; ELER, 2020) sampling technique, we use the grid structure to group similar data points and divide the space based on the number of features expressed inside each cell. After adding feature ● to the inspection, the cell is divided and colored based on the color-encoding of the features in (**c**). Furthermore, color opacity is used within the cell to communicate feature expression—the most opaque segment for a feature indicates the highest expression for that feature. The same happens when the feature ● is added to the analysis (**d**) when the cell is divided into three parts.

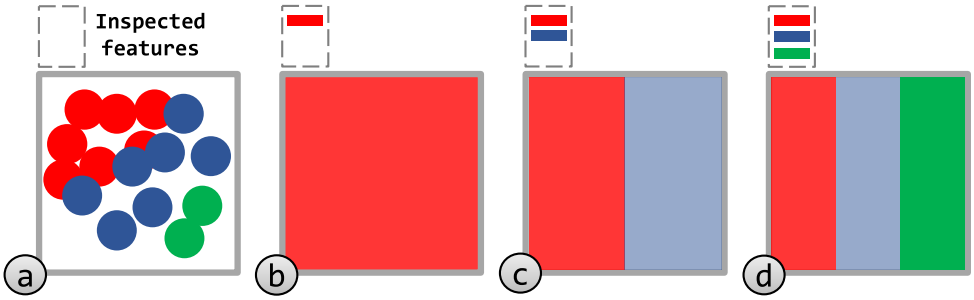
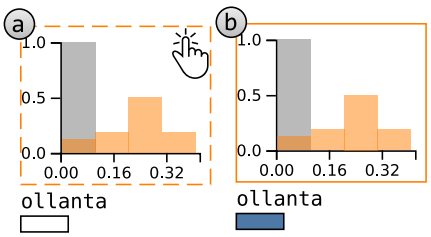


Figure 18 – Construction steps for cell-based encoding. For each cell, the sum of each feature is obtained (a). The cell is then divided based on user demand to communicate the features. The space assigned to a feature is proportional to the number of features in the cell. For example, one feature in (b), two in (c), and three in (d).

The encoding strategy discussed above is depicted in Fig. 19. We chose four clusters (**a**). Following an examination of cluster ●, the selection of `syrian` shows where the term is most frequently expressed (note that the colors representing the cluster and the term itself are unrelated) using a color scale, allowing users to investigate where the documents discuss `syrian`. The aggregated encoding communicates the expression level in regions defined by cells of a fixed size (defined by users) to visualize term co-expression. The space inside each cell is divided by the number of features being examined.



The steps in Fig. 19 are accomplished by interacting with the distribution plots. Borders with straight lines indicate feature inspection, as shown in Fig. 16(**d-e**), and users perform combinations to visualize multiple values. It is worth noting that cells with no data observations expressing any feature will not be visible in the visualization if the opacity strategy discussed above is used. We used a texture (▨) to fill these cells and to maintain the

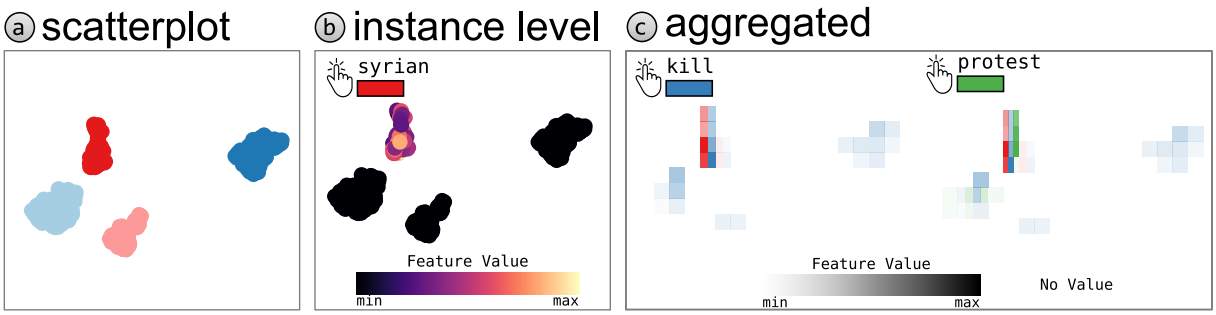


Figure 19 – Cell-based visualization for feature distribution on the scatterplot. After inspecting the distinctive features on the scatterplot (a), users can visualize the feature values using cell-based visualizations. For instance, when one feature is selected (b, `syrian`), a simple continuous color scale is sufficient. For more than one feature (c, `syrian, kill, protect`), cell-based encoding is employed.

general structure of the projected dataset. Finally, each cell's division corresponds to the number of expressed features in the cell—that is, if three features are chosen but only two are expressed, the cell is divided into two equal parts.

Another important aspect of cell-based encoding to consider is the trade-off between computational performance and the quality of explanations related to cell size in pixels. Finding which points belong to which cell becomes faster as cell size increases, although the resemblance to the original scatterplot decreases. The inverse is also true. As the cell size decreases, the cell-based encoding approaches the scatterplot structure. However, it increases the computational burden, and a small cell size also makes distinguishing colors in the visualization difficult. To address these issues, we use a KD-tree to efficiently identify which points belong to a cell and a cell size of 20x20 pixels—we find it to be a good compromise between feature distribution visualization and scatterplot structures. Finally, before generating the visualizations, users can specify the cell size (see Fig. 16).

3.2.2.4 Comparing clusters

Aside from comparing a cluster of interest to the rest of the dataset, understanding the differences between two clusters is useful. For example, to comprehend why subclusters are formed.

On the scatterplot view, users select two clusters of interest for comparison. The contrastive information is retrieved in this case from two arbitrary clusters, c_a and c_b. Fig. 20 depicts an enhanced version of the visualization in Fig. 17 in which the information about each cluster is mirrored—the color scale representing the p-values is constructed based on the clusters with the longer range. When comparing clusters, the information provided to contrast clusters to the remaining dataset is presented to both comparison clusters (such as distribution plots, p-values, and t-scores axes).

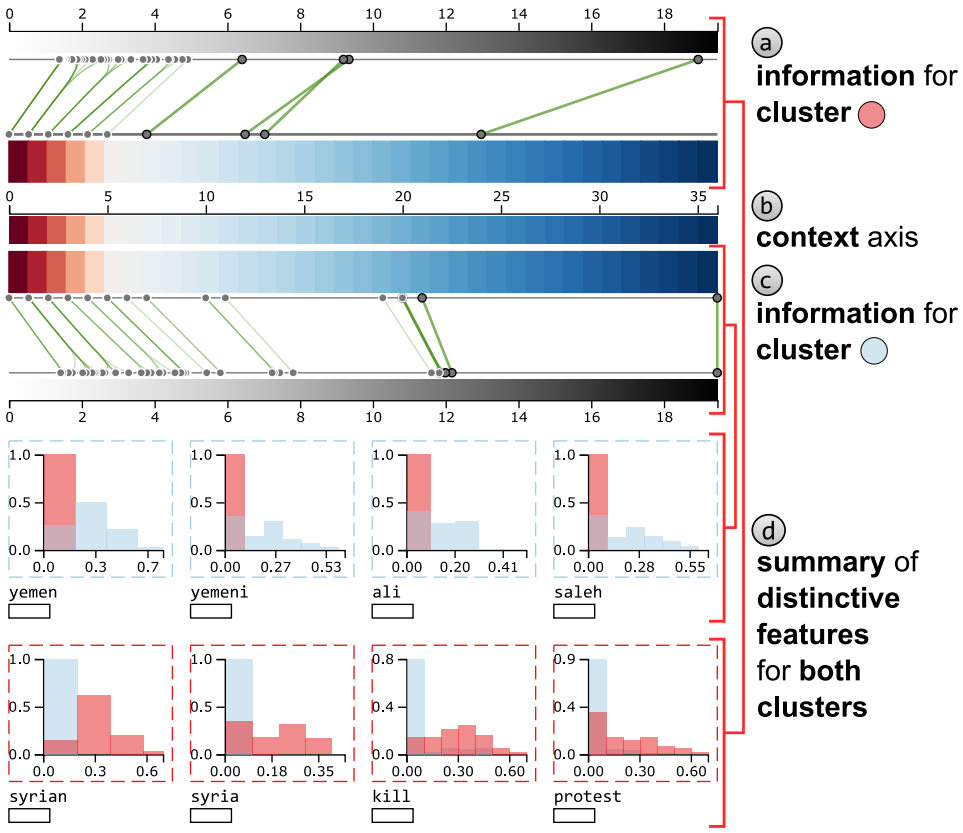


Figure 20 – A comparison of two clusters. The visualization strategy in Fig. 17 is used to communicate information between two clusters when comparing two clusters. In essence, while the information for the first cluster (light-blue in the figure) is encoded as usual (c), the information for the second is positioned on top (a), and the context axis is positioned in the center of the visualization (b). Furthermore, the four most distinguishing features of both clusters are depicted in (d).

3.2.2.5 Summary View

Although our technique emphasizes differences in the dataset even for a reasonable number of features, it would be interesting to learn more about a cluster structure and visualize the importance of additional features for some datasets. As shown in Fig. 16(**f**), a heatmap provides an overview of the feature importance and visualizes the statistics used by our technique. The t-score is encoded by a purple to green color scale, and the p-values are encoded by the tile dimensions. It is important to remember that t-scores are standard deviation measures, whereas p-values are probabilities. Both statistics correspond to the standard t-distribution. This distribution associates standard deviations with probabilities and allows t-scores and p-values to be associated with significance and confidence. In this case, Fig. 16(**f**) shows how the distinguishing features aid in explaining each cluster via contrastive analysis. That is, no green tiled features are shared across columns.

3.2.2.6 Interaction mechanisms

When inspecting a cluster, the four most distinguishing features are displayed, as shown in Fig. 21. Users may, however, inspect other features based on their p-value. As shown in Fig. 21, such an interaction is carried out using a selection box (smallest box). The visualization of their respective distribution plots thoroughly inspects all of the corresponding features inside the box. To communicate which features are selected, the line segments of the bipartite graph are also updated with color opacity.

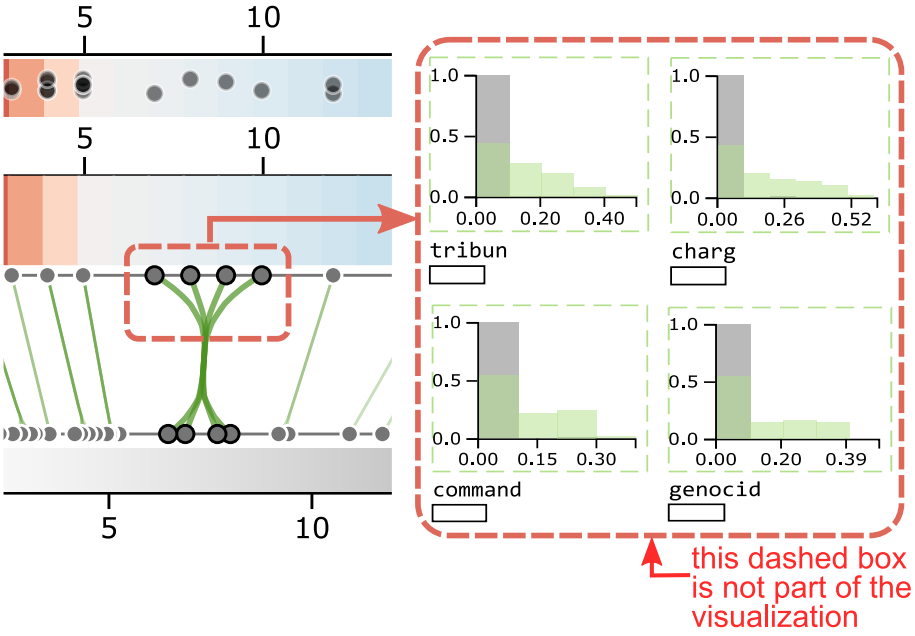


Figure 21 – Users can inspect the histograms of available features through lasso selection over the points encoding p-values.

Another unique feature of our tool is the ability to color-code the features being examined in the scatterplot. While users can freely visualize a large number of distribution plots, humans can only distinguish a small number of colors effectively (WARD, 2002). Users can toggle the distribution plots with a maximum of ten features displayed at once. Fig. 19(**c**) shows the results of this operation for the terms `kill` and `protest` in the scatterplot representation using color encoding.

The selection mechanism helps users to understand other aspects of the clusters by inspecting many features. When these features have a similar p-value, the visual space dedicated to features may be insufficient. As shown in Fig. 22, we use focus+context (MUNZNER, 2015) interaction on the p-value axis to reduce such a problem. On top of the p-value axis, another axis represents the context, or all of the available information. To begin with, the focus axis corresponds to the context axis (1). Then, as shown in the figure by a red arrow, users specify a range where the focus axis will be defined (2). The focus option makes the p-value axis to display only the information contained within the selection box (3). Finally, the changing in focus makes the visualization to update the

distribution plots.

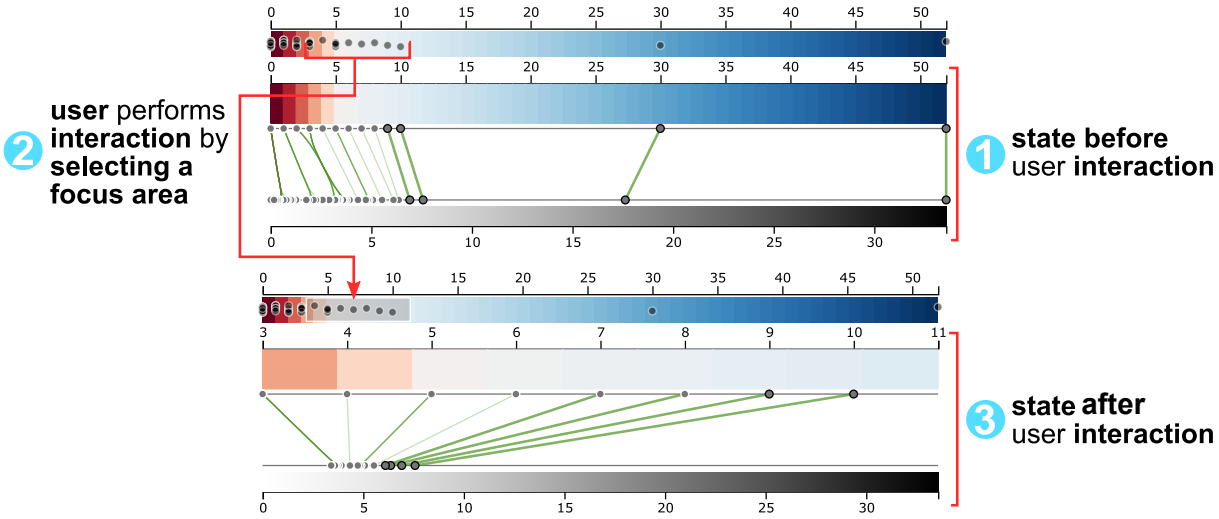


Figure 22 – Focus+context operation. The visualization starts with the context and focus axes as the same (1). Then, users can select a range of the context axis to focus analysis (2), resulting in an update of the focus axis (3).

3.3 CASE STUDIES

To validate the proposed technique, we explore three datasets. The first consists of a document collection of news articles from 2011. The second provides an extensive analysis of a dataset of tweets about COVID-19. Then, the last case study consists of a multivariate dataset representing patients with vertebral problems.

3.3.1 THE NEWS DATASET

In this first case study, we inspect a document collection of 495 news articles in English available in RSS format by Reuters, BBC, CNN, and Associated Press agencies. Fig. 23 shows the UMAP (MCINNES; HEALY; MELVILLE, 2018) projection of the dataset color-coded based on the Leiden (TRAAG; WALTMAN; ECK, 2018) algorithm. We used the first 40 Principal Components (PCs) of the dataset to compute the neighborhood graph ($k = 15$) for the UMAP technique. The resolution parameter of the Leiden algorithm was set to 0.3.

Looking at the most distinctive terms presented in Fig. 24, and recalling that the news dataset contains articles from 2011, cluster ■ represents news articles of the earthquake that hit Japan in 2011. The terms show that the incident on the nuclear plant of Fukushima I is present in almost all of the news articles due to the earthquake. Further, the news articles related to the nuclear power plant incident are concentrated in just one part of the cluster. The projection technique successfully uncovered a subcluster of news articles referring to the same aspects of the earthquake.

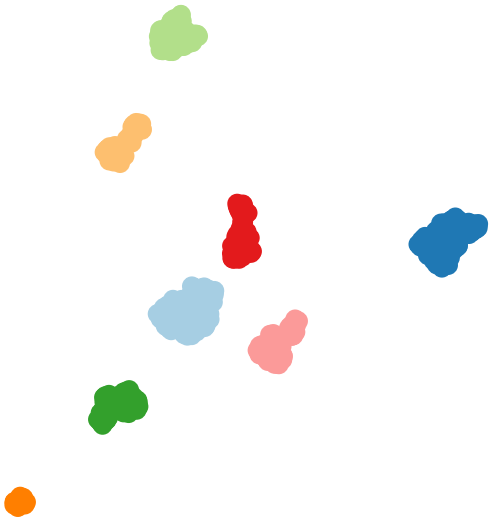


Figure 23 – UMAP projection of the *news* dataset.

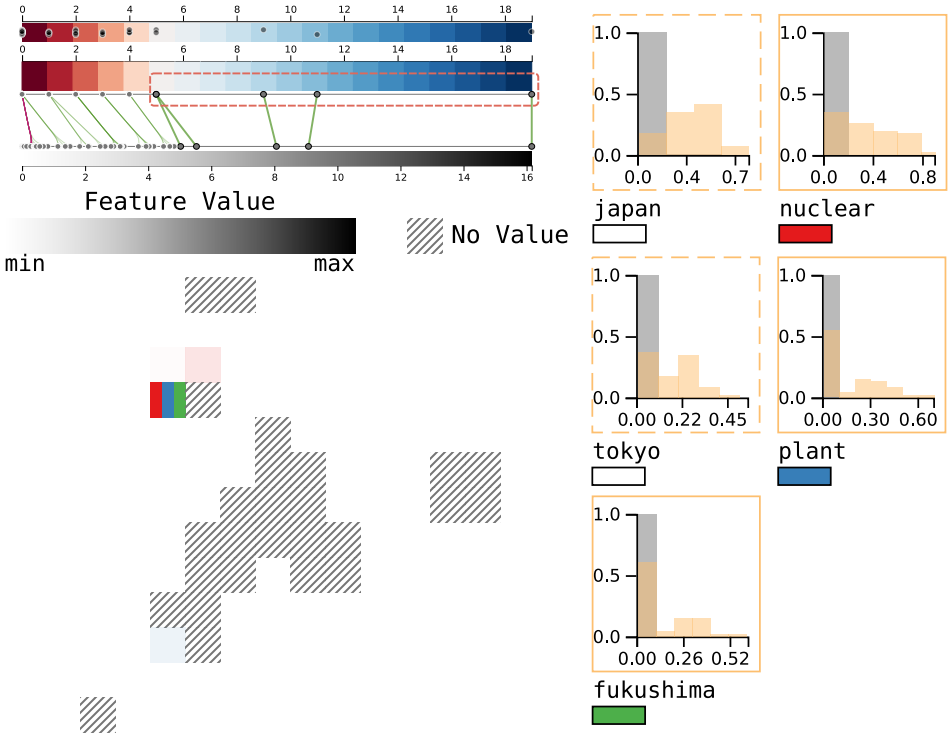


Figure 24 – After selecting the five most discriminative features, highlighting the terms `nuclear`, `plant`, and `fukushima` shows that UMAP distinguished the news articles separated the news articles related to the nuclear accident on the left side of the projection.

To further explore cluster ■, we focus on the terms that were not assigned much confidence (with p-values between 10^{-3} and 10^{-5}) and then select a few terms, as illustrated in Fig. 25 (a). In this case, the terms refer to former Japan's primer-minister, Naoto Kan, who resigned his role after the crisis provoked by the earthquake and the consequent incident in Fukushima I. The terms `naoto`, `kan`, `prime`, `minist`, `pm`, and `japanes` further support this idea.

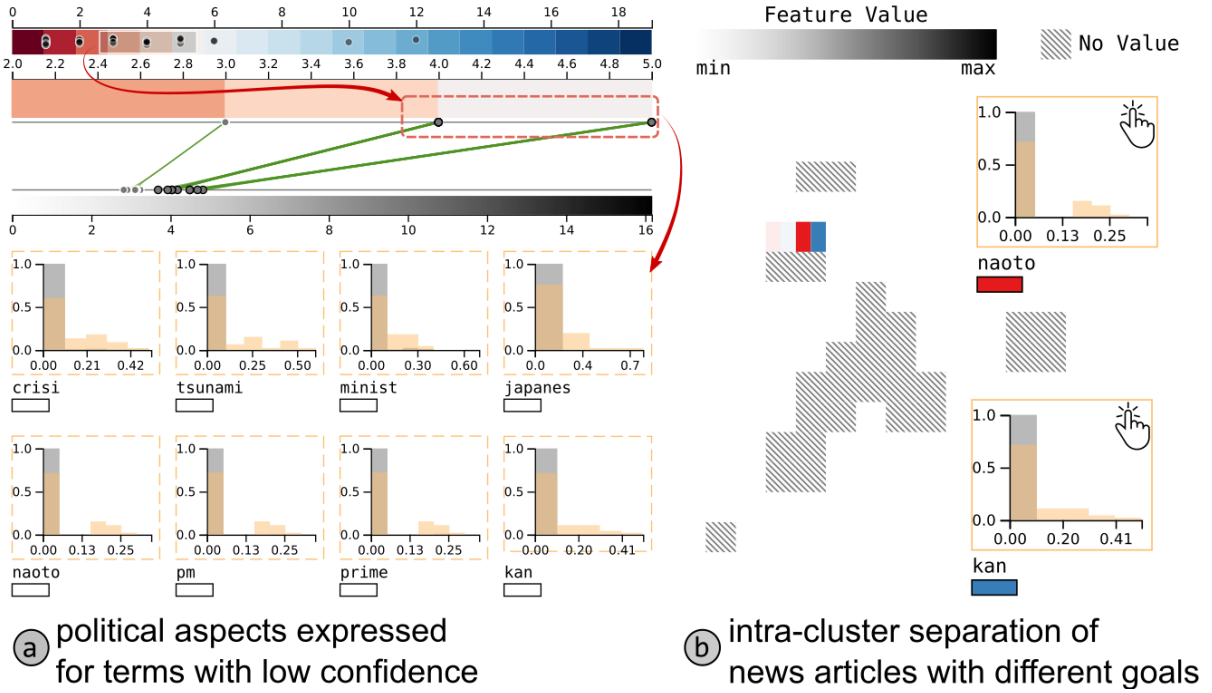


Figure 25 – Inspecting terms with lower confidence. Terms with lower confidence in the light-orange cluster are related to political aspects (a) and were positioned in a different subcluster.

Fig. 25 (b) shows the scatterplot representation of the expression intensity of the terms `naoto` and `kan`, which validates our approach to understand the organization imposed by the dimensionality reduction technique. While news articles regarding the Fukushima I incident were positioned on the bottom of the cluster, news articles about the former primer-minister were positioned on the top of the cluster.

Proceeding to cluster in Fig. 26, there are many terms with high confidence that could be used to understand the main topics of the news articles. The first terms (highlighted in red) correspond to Ratko Mladić, a former Serbian military officer, head of the Serbian Republic Army during the Bosnian War between 1992-1995. The other group represents more specific information about the news articles contained in the cluster. The terms refer to the prison of Mladić due to war crimes. More specifically, the terms `tribun` and `hagu` refer to the fact that Mladić was extradited to The Hague city in the Netherlands to respond to his crimes in the International Court of Justice.

We finish the analysis by inspecting cluster , as illustrated in Fig. 27. From the terms highlighted as discriminative, such a cluster corresponds to news articles of a strain of Escherichia Coli O104:h4 bacteria outbreak in northern Germany from May to June 2011. The majority of the news articles mention `coli` and `outbreak`. While the terms `europ` and `germani` express on opposite sides of the cluster could indicate that some news articles focus specifically on Germany, and others referring to the whole of Europe. The other terms describe the whole cluster: `health`, `deadli`, and `infect`.

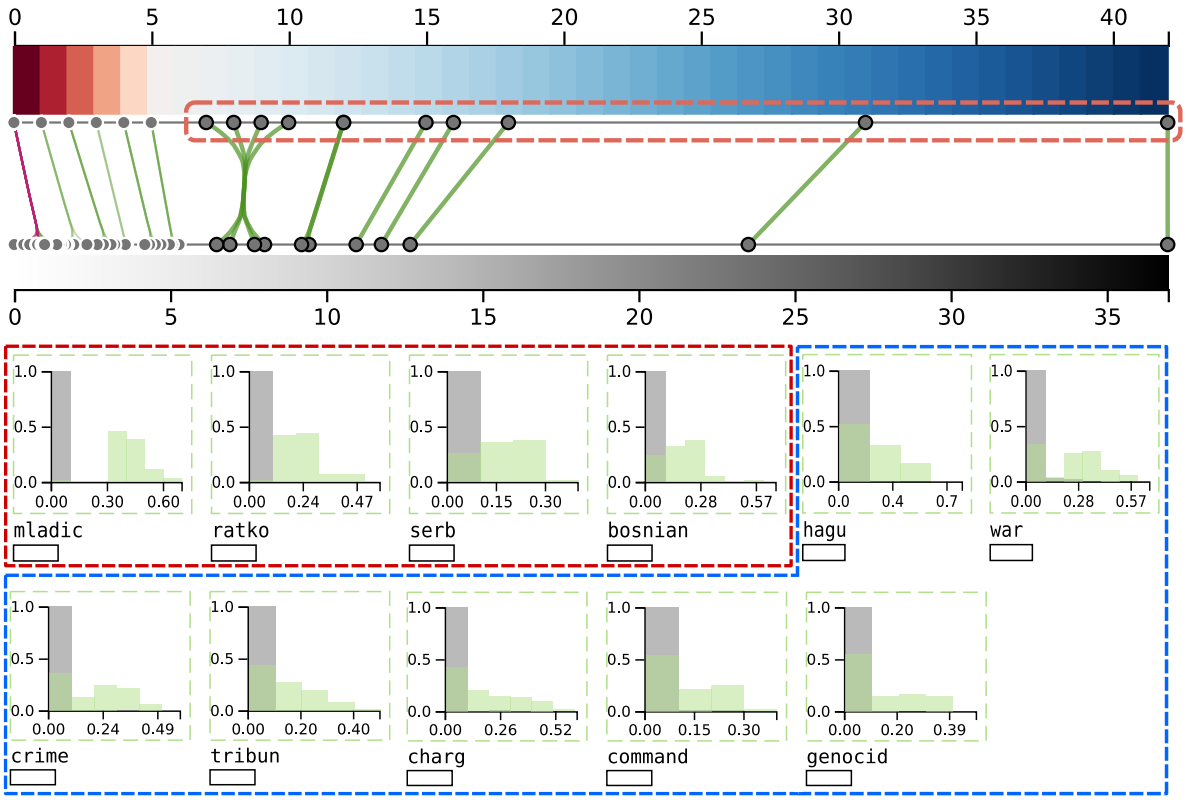


Figure 26 – Contrastive information showing levels of understanding. The terms highlighted in red show high-level information, while the terms highlighted in blue give specific hints about the news articles.

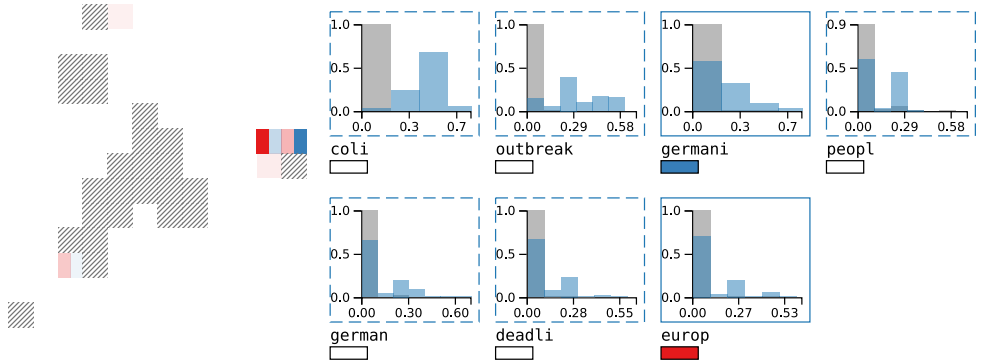


Figure 27 – Inspection of `germani` and `europ`'s terms shows that the dimensionality reduction separated news articles focused on the European continent from the news articles focused on Germany.

3.3.2 EXPLORING TWEETS ABOUT COVID-19 SYMPTOMS

For this use case, we aim to analyze a complex document collection of *tweets* about COVID-19 common symptoms (fever, high fever, cough, dry cough, difficulty breathing, shortness of breath) retrieved from São Paulo state (Brazil) from March 2020 to August 2020. To create the dataset, we retrieved tweets mentioning one of the symptoms discussed above; then, we classified the *tweets* according to their relevance.

We used the BERT (DEVLIN et al., 2018) language model to train a classifier and

to select only relevant *tweets*. Ten thousand tweets were manually classified and BERT was employed for the classification of 30 thousand tweets. Relevant *tweets* are those with serious comments about COVID-19 and where there is a chance of infection. Non-relevant *tweets* correspond to news, jokes, and other informative comments. Further, to inspect the dataset projected on $\mathbb{R}^2$, we used the UMAP technique with the nearest neighbor graph set as 20. Finally, the clusters were manually defined, as shown in Fig. 28.

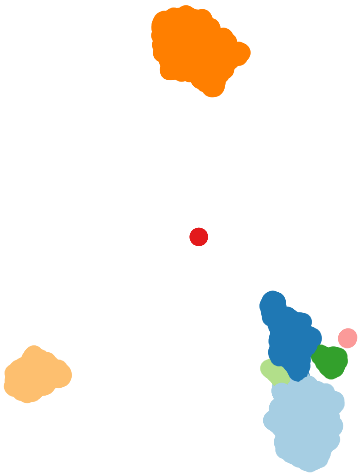


Figure 28 – UMAP projection of the *tweets* about COVID-19.

The projection of the tweets shows four main clusters: red, dark orange, orange, and a cluster divided into five other subclusters (, , , ,). To analyze the projection, we follow the strategy of analyzing the separated greater clusters and then proceeding to the very cohesive cluster .

Fig. 29 shows that cluster is mainly related to tweets about the shortness of breath, one of the most severe COVID-19 symptoms—the term `chest`, for example, could indicate people describing how they were feeling and where the sensations are occurring in their body. Another interesting aspect of such a cluster is the term `anxiety`, which indicates a problem arising from the necessity of social isolation during the pandemic. In fact, during data reading, many tweets corresponded to people asking whether a COVID-19 infection or anxiety crisis caused their shortness of breath.

Proceeding to cluster , we focus on the terms with p-value $\leq$ 1e-5. Fig. 30 shows that such a cluster is related mainly to the *dry cough* symptom of COVID-19 due to the presence of terms `cough` and `dry`—the term `sneeze` is also present when Twitter users were describing their symptoms. The other terms (`annoying` and `each`) usually consist of phrases commonly identified in the tweets, such as, (directly translated from informal Portuguese) "I am with an annoying cough..." or "It is each cough...".

Finally, we analyze the subclusters on the bottom right of Fig. 28. As shown in Fig. 31, both of clusters and refer to characteristics of fever symptom—see how the terms `face`, `fever`, `body`, and `to-measure` are expressed in cluster while the terms

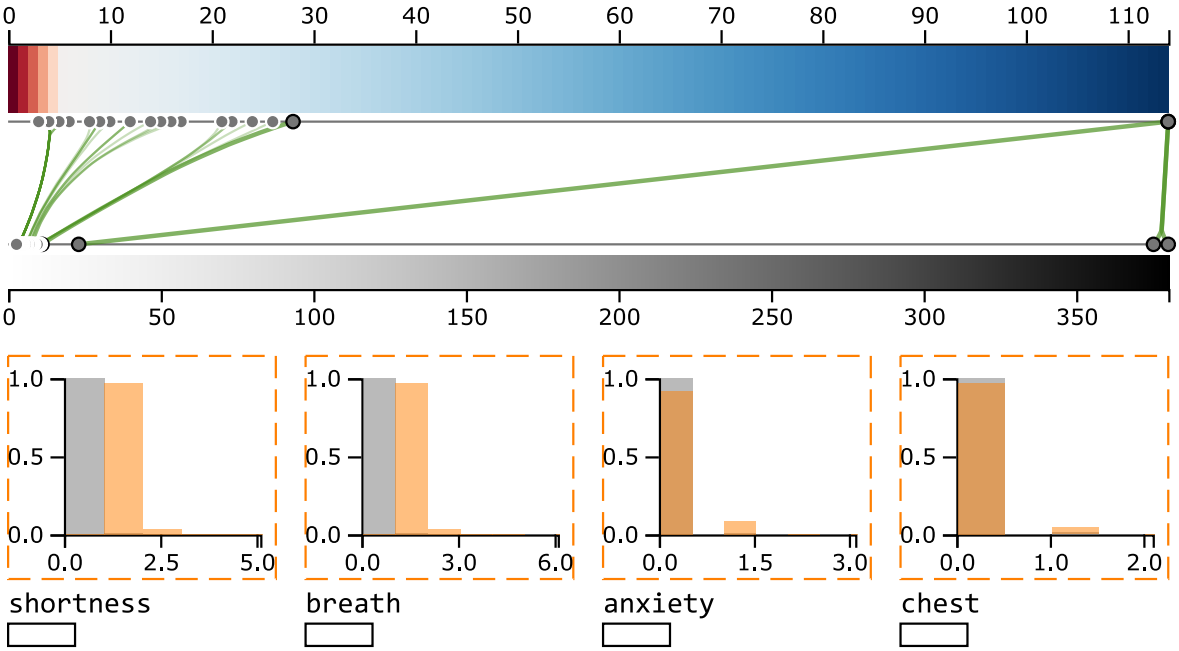


Figure 29 – Discriminative terms for the dark-orange cluster—terms related to respiratory problems.

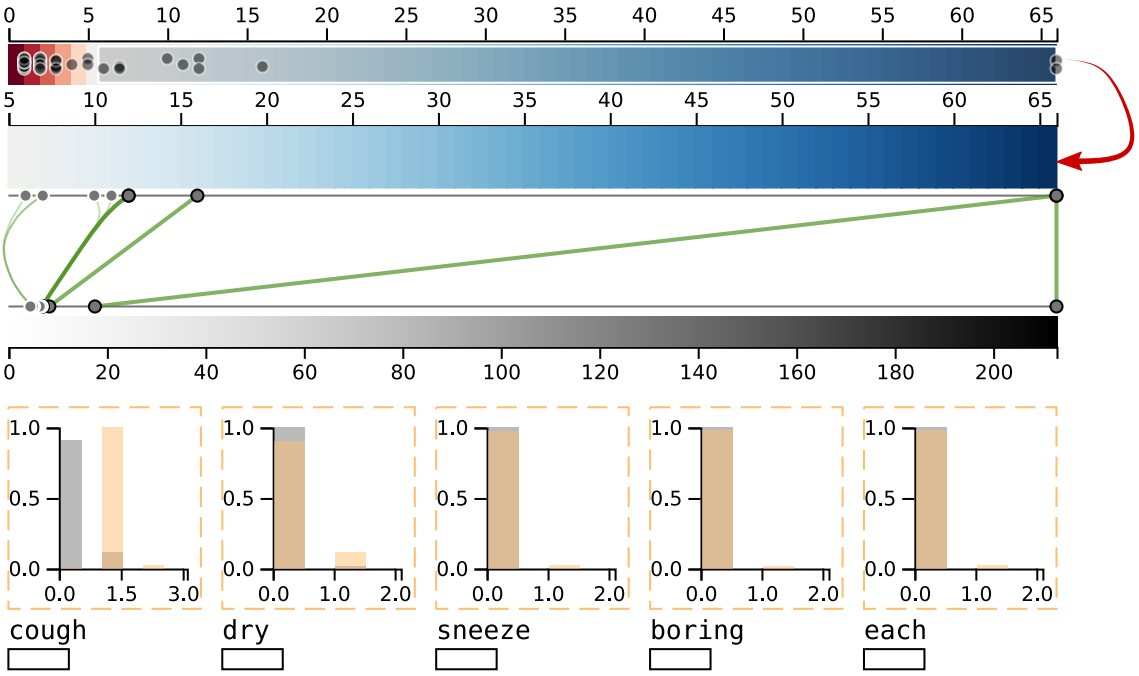


Figure 30 – Discriminative terms for the light-orange cluster. Terms indicate users tweeting their complaints about symptoms expressed during COVID-19 infection.

`I-think`, `fever`, `getting`, `thermometer`, and `hot` are expressed in cluster ■. Particularly for cluster ■, the terms indicate users that had started to feel febrile when they posted the symptom on Twitter. Such insight could be significant to reveal which cities are presenting people with developing symptoms of COVID-19. That is, regulatory policies could be made based on the geolocalization of the *tweets*.

Cluster ■ indicates more general aspects about the symptoms related to fever.

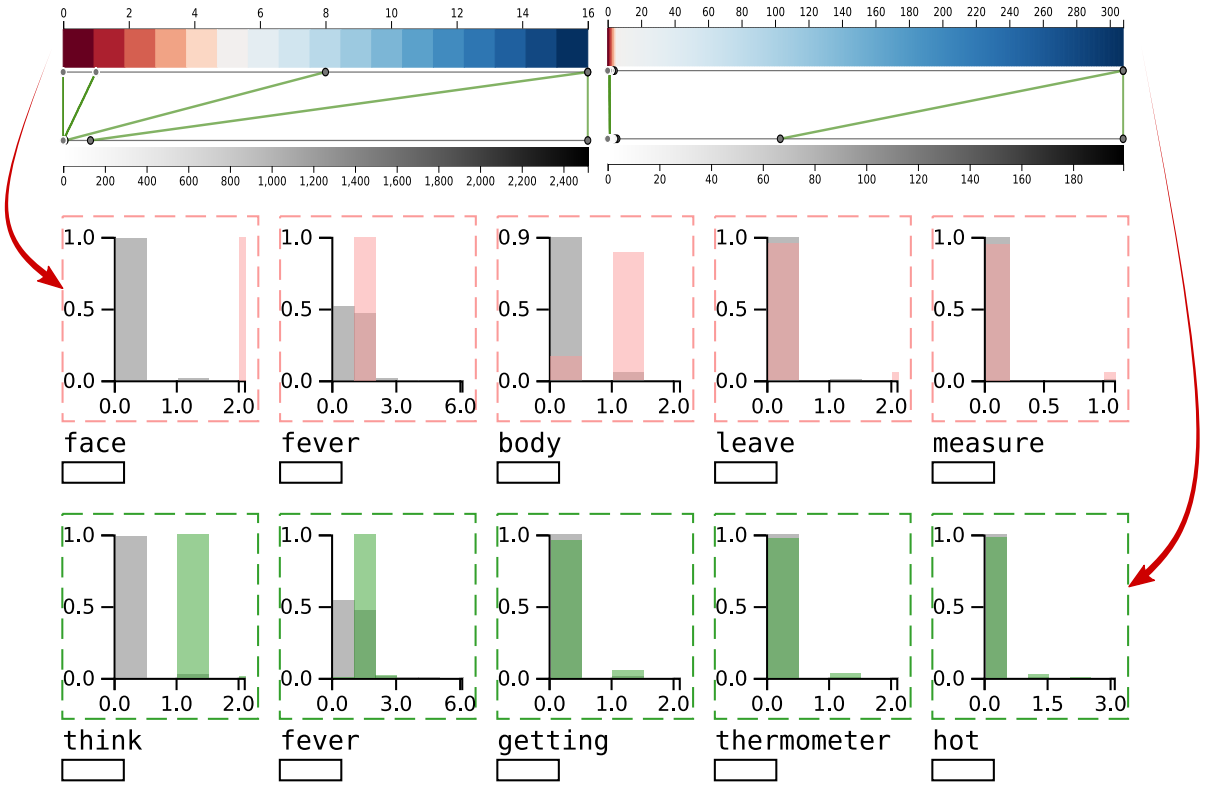


Figure 31 – Comparing two clusters with `fever` as the main subject.

Fig. 32 shows that the discriminative terms are related to fever—notice that we omitted the terms `sick`, `was`, `dipyrona`, `pain`, `night`, `to-be`, and `bath` since they are terms used to create the phrases. An interesting aspect of this cluster is the presence of the term `sore`, an uncommon symptom.

Unlike the other clusters, cluster ■ presents many different levels of understanding. Firstly, Fig. 33 shows that although one of the five most discriminative refers to fever, such a cluster might represent Dengue symptoms. That is, while the citizens were posting how they were worried about COVID-19, the symptoms indicated on the tweets match with those used to describe Dengue infection (`pain`, `head`, `throat`, `body`, and `fever`), a common disease faced in Brazil.

Finally, Fig. 34 shows that a severe symptom of COVID-19 is also present in cluster ■: `difficulty breathing`. Besides that, other terms indicate common symptoms of seasonal flu, such as `coryza`. Finally, the term `tiredness` reinforces the idea of Dengue symptoms or consists of a uncommon COVID-19 symptom.

In this case study, the tweets presented three main topics: clusters ■ is related to respiratory problems induced by COVID-19 infection, as well as related to anxiety crisis due to long periods of social isolation; cluster ■ is related to *dry cough*, a common COVID-19 symptom; finally, the cluster with subclusters has a lot of aspects regarding high fever, where each one of the subcluster (■, ■, ■, ■, ■) has its particularities. Finally, to investigate cluster ■, the heatmap shown in Fig. 35 shows much of the very

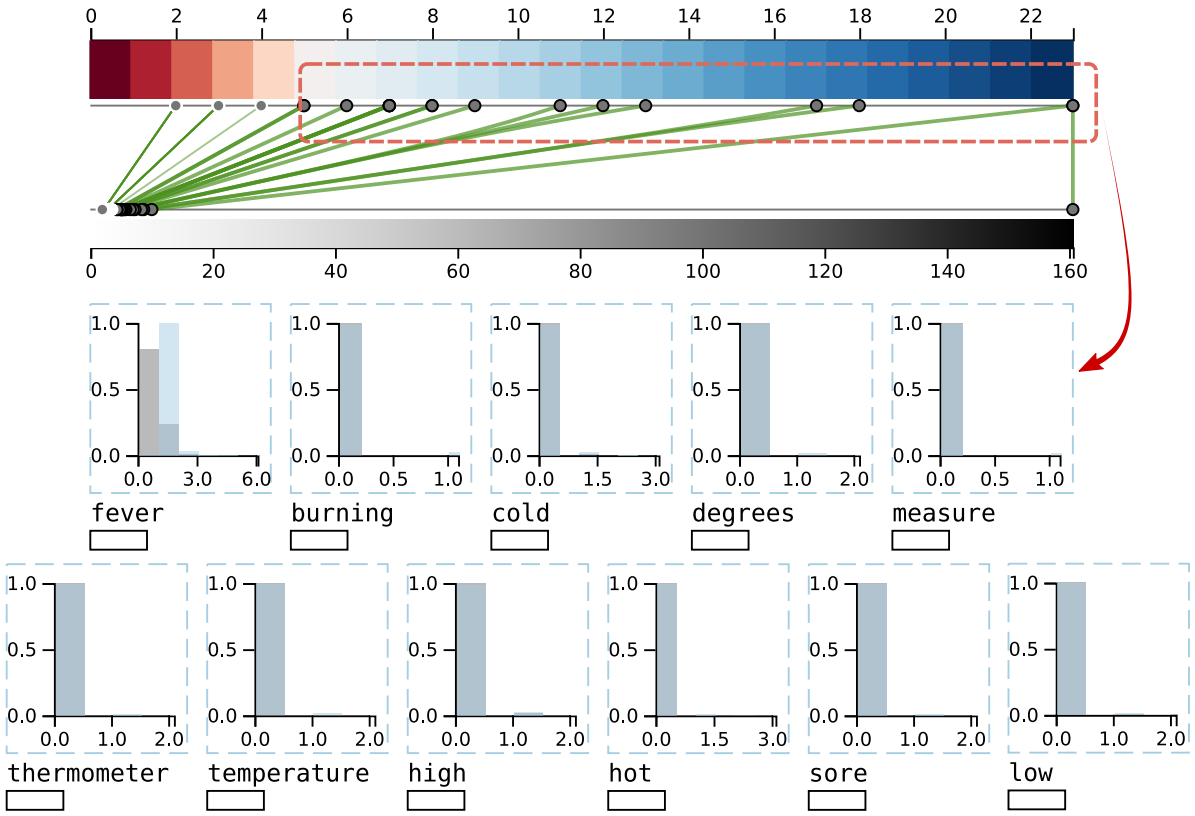


Figure 32 – Distinctive terms for the light-blue cluster. In this case, there are more general aspects of symptoms related to fever. For instance, the term `sore`, could be related to *sore throat* or a new COVID-19 symptom.

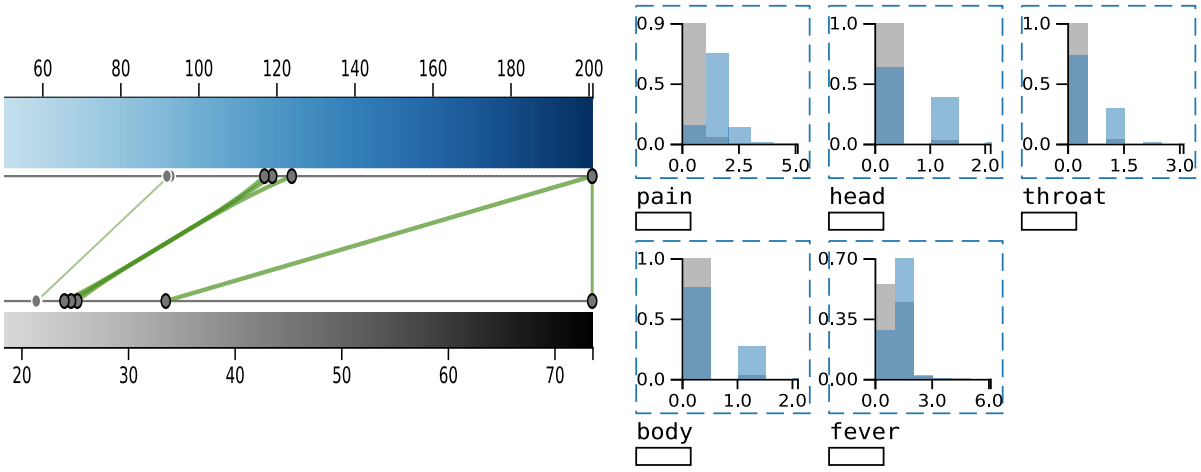


Figure 33 – Distinctive terms from the dark-blue cluster can be considered as indicative of Dengue symptoms.

cohesive cluster is comprised of tweets concerning terms of worrying (see `worry` and `risk`) about coryza and wearing masks.

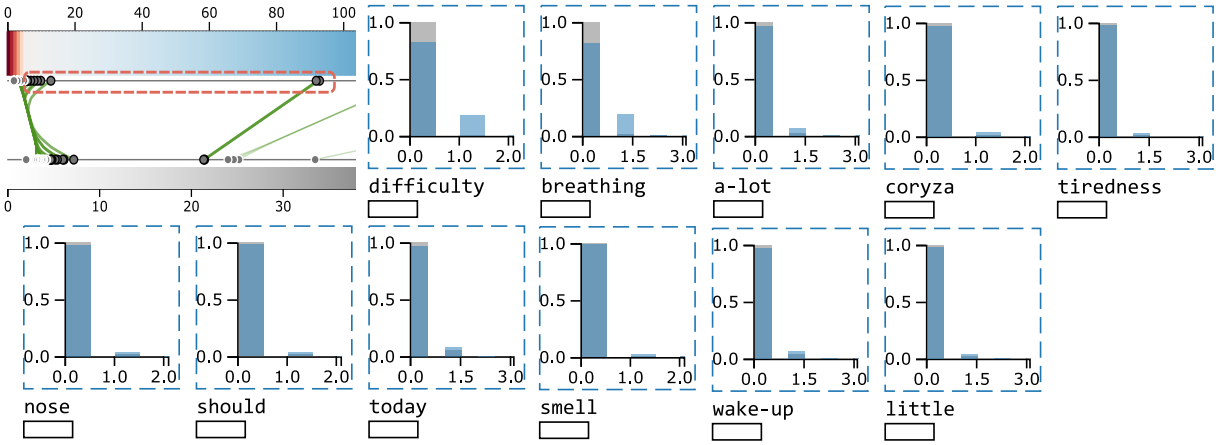


Figure 34 – Evaluating another term from the dark-blue cluster. The terms such as `difficulty` and `breathing` indicate the most severe COVID-19 symptoms, shortness of breath.

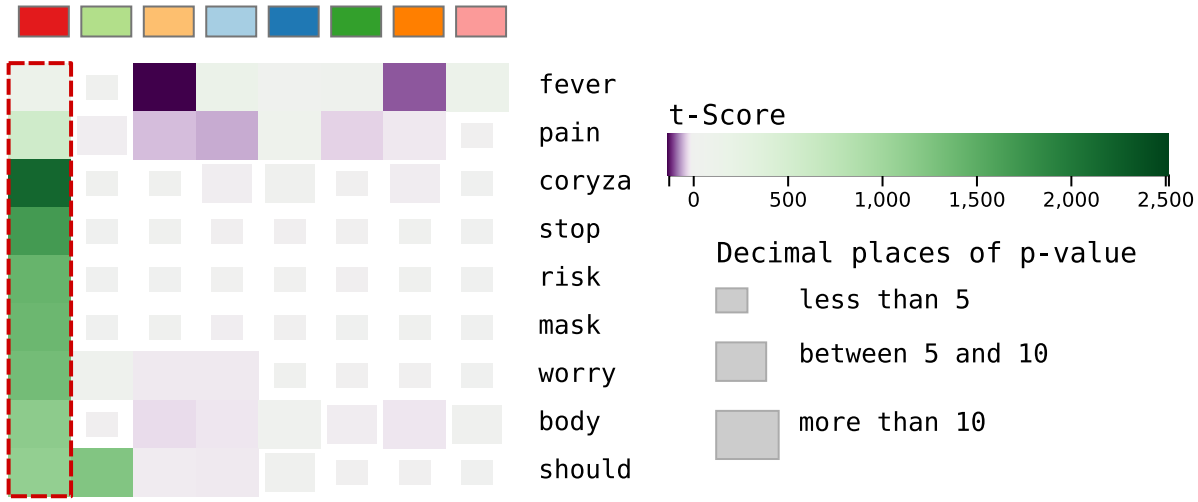


Figure 35 – Heatmap showing that the red cluster comprises the tweets where people are worried about wearing masks and coryza.

3.3.3 VERTEBRAL COLUMN

In this case study, we aim to analyze a multivariate dataset. The *Vertebral Column* dataset (DUA; GRAFF, 2017) is composed by 310 data instances described by six features derived from the shape and orientation of the pelvis and lumbar spine: `pelvic incidence`, `pelvic tilt`, `lumbar lordosis angle`, `sacral slope`, `pelvic radius`, and `grade of spondylolisthesis`. Fig. 36 shows the UMAP projection of the *Vertebral Column* dataset with the heatmap view. The projection is colored according to the ground truth classes: class ■ stands for regular patients, class ■ stands for patients with Spondylolisthesis—a disturbance of the spine in which a bone (vertebra) slides forward over the bone below it; the class ■ stands for patients with Hernia. Notice a clear separation of Spondylolisthesis class (■) among the others. The heatmap view shows that class ■ is different from the others for having all of the feature values (except `P. Radius`) with higher values, as

indicated by the green cells.

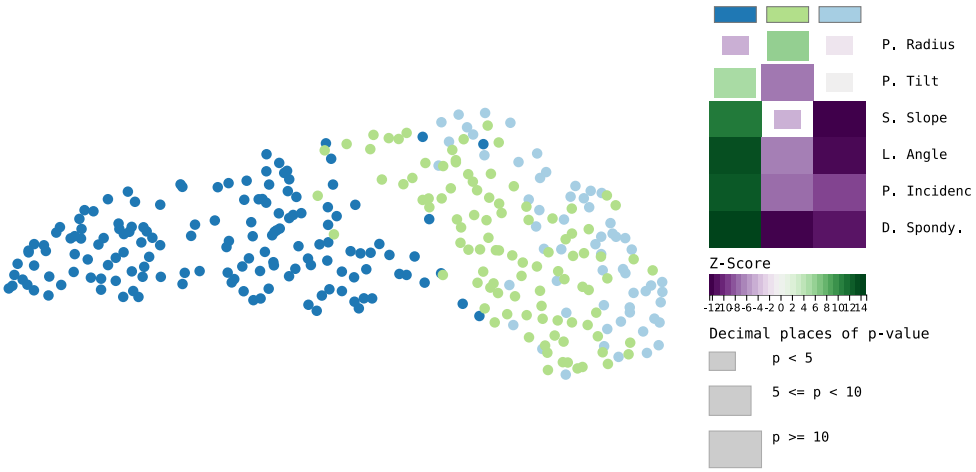


Figure 36 – UMAP projection of the *Vertebral Column* dataset next to the heatmap visualization.

To investigate more about the class of patients with Spondylolisthesis (■), the selection of the features with high confidence on their distinction aspects (that is, features whose p-values are lower than 10^{-5}) shows that the distribution plots present the values skewed to the right, as illustrated in Fig. 37. The distribution plots also show how data instances of patients with Spondylolisthesis present higher values for `D. Spondylolisthesis`, `L. Angle`, `P. Incidence`, and `S. Slope`.

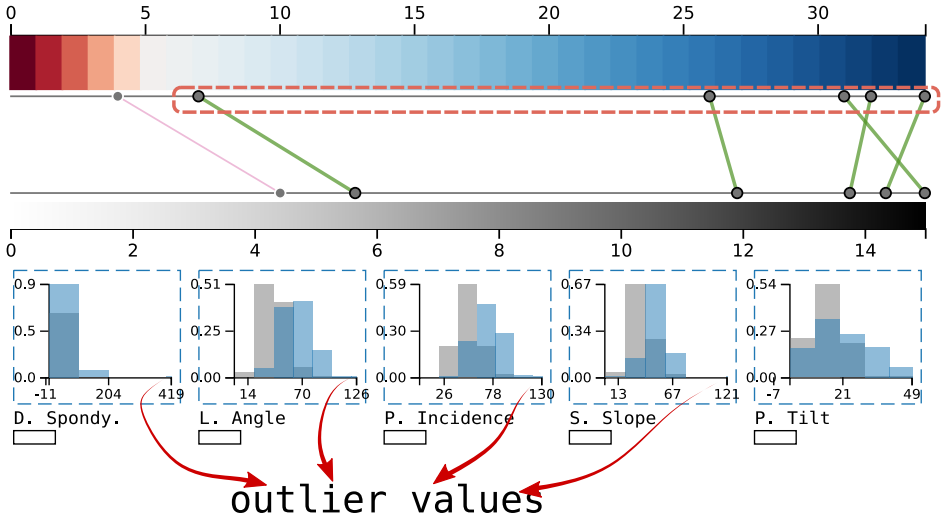


Figure 37 – The class of patients with Spondylolisthesis shows greater values for all of the features, except for Pelvic Radius.

The scatter plot encoding with the intensity bars communicate where the data instances corresponding to the higher values were positioned by dimensionality reduction technique. Fig. 38 shows the scatterplot encoded as intensity bars. Note more intensity as the data instances approach to the left of the projection, i.e., more distant from classes ■

and . Such a characteristic validates our approach by helping at understanding how the dimensionality reduction technique organized the data instances on the projected layout.

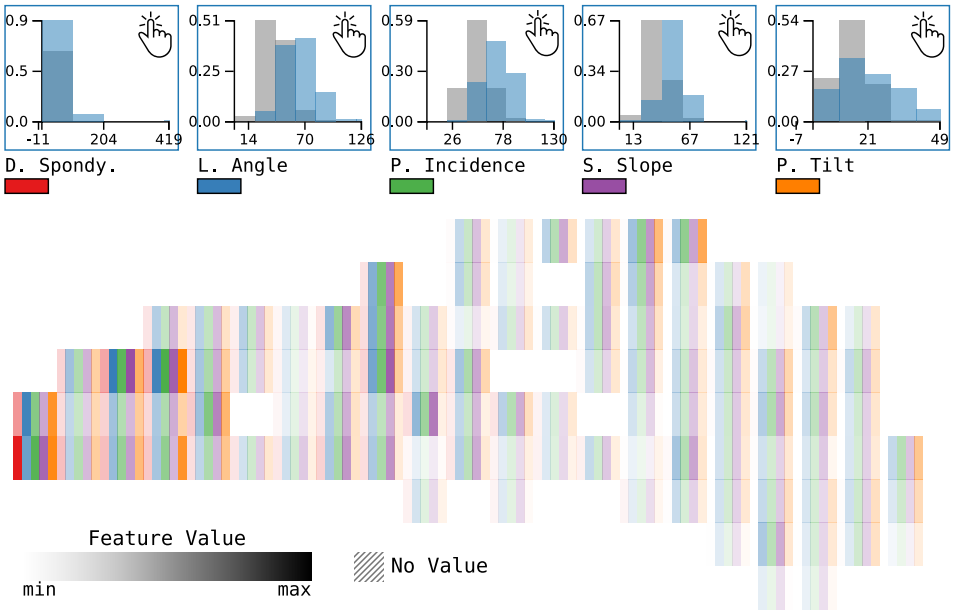


Figure 38 – Inspection of the distribution values of the features shows that the features are truly expressed more in the cluster of patients with Spondylolisthesis.

Lastly, the distinctive features returned by our approach reflect what happens when patients are diagnosed with Spondylolisthesis. That is, according to (LABELLE et al., 2005), pelvic incidence, sacral slope, pelvic tilt, and lumbar lordosis angle are prone to be higher in patients with developmental spondylolisthesis. Moreover, the lower values for sacral slope (`S. Slope`) is a known of centralistic herniation (TEBET, 2014), which could explain the slight separation between the classes of healthy patients ( ) and patients with Hernia ( ).

3.4 EVALUATION

To further testify *cExpression* to support analysis of dimensionality reduction results, we compare it against well-known topic extraction techniques using the *coherence* metric. Finally, we assess run-time execution of *cExpression* and *ccPCA* (FUJIWARA; KWON; MA, 2019) upon various dimensionality values for a document collection.

3.4.1 COHERENCE

A useful way to analyze *cExpression* results is by assessing how the terms selected from a document collection describe the clusters, similarly as in topic extraction tasks. Fig. 39 shows the pipeline for such a task.

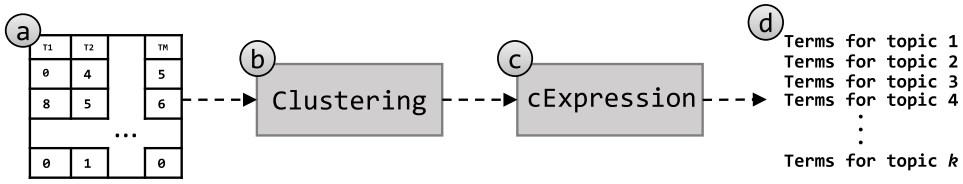


Figure 39 – Retrieving topics using cExpression. Given a bag-of-words representation of a document collection (**a**), we compute the dataset clusters (**b**). *cExpression* is executed to retrieve distinctive terms for each cluster (**c**), and the topics correspond to the first m terms of each cluster.

After creating a bag-of-words representation (**a**.) of a document collection, we apply a clustering algorithm to group documents based on similarity (**b**.). Then, *cExpression* returns the terms in each cluster that are not likely to appear in the other clusters. As explained in Section 3.2.1, these terms are ordered based on their contrastive score—the first m terms are returned to compose the topics of each cluster (**d.**). We use the *Topic coherence* (RöDER; BOTH; HINNEBURG, 2015) metric to evaluate the robustness of the terms returned by *cExpresion*. This metric is applied to the top m terms of a topic and consists of the average of the pairwise word-similarity scores of the topic. A good model will generate coherent topics with high topic coherence scores. Thus, the topic extraction technique **A** is better than the technique **B** if it has a greater score.

The evaluation was performed using three datasets: 495 news articles (205 terms) and 40794 tweets related to COVID-19 symptoms (295 terms), and the *20newsgroups*[1] dataset. For the *20newsgroups*, we use a subset comprising the classes *alt.atheism*, *comp.graphics*, *comp.windows.x*, *rec.motorcycles*, *sci.electronics*, *sci.med*, *talk.politics.guns*, *talk.politics.misc*, *talk.religion.misc*. After preprocessing the dataset—removal of English stop-words and terms appearing in a proportion of documents below 0.5—the dataset resulted in a bag-of-words representation of 4828 documents by 3347 terms. Finally, we compared *cExpression* against the well-known and commonly used topic extraction approaches, LDA (HOFFMAN; BACH; BLEI, 2010) and NMF (CICHOCKI; PHAN, 2009).

Fig. 40 shows the coherence when varying the number of terms in each topic for the *news* and *covid-19* datasets. *cExpression* presents a slight advantage for topics with size below 20 and emphasizes the terms that are not expressed in other clusters—terms of topic A (cluster A) are not likely to appear in other topics. So, for a lower number of terms, *cExpression* emphasizes the differences while showing good coherence.

To analyze the `20newsgroups` dataset, besides varying the number of terms, we also varied the number of topics returned from the algorithm. Fig. 41 shows that the results for cExpression—encoded by deviation of the line plots—remains stable for different

[1] http://qwone.com/ jason/20Newsgroups/

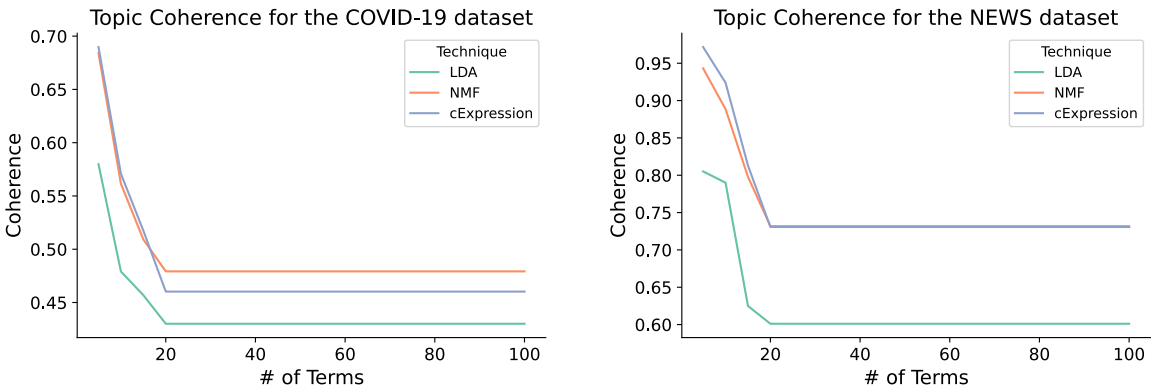


Figure 40 – cExpression shows competitive results for topics with a low number of terms (≤ 20).

number of clusters. Appendix B presents these results in separated plots.

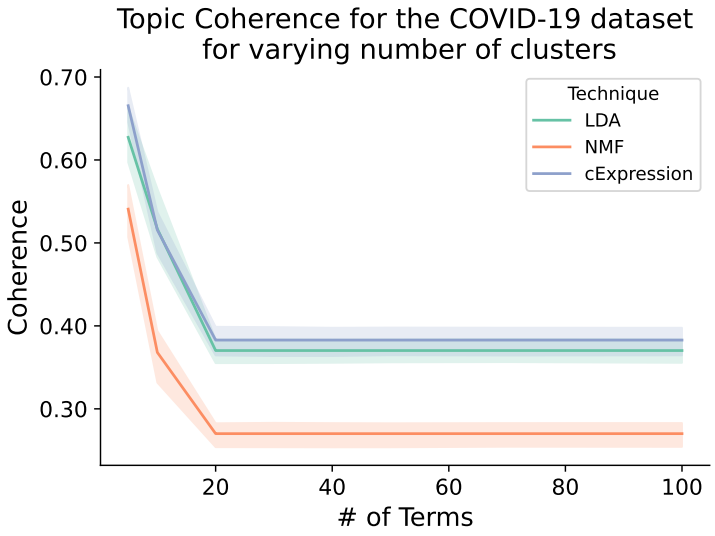


Figure 41 – cExpression was able to surpass LDA and NMF for the majority of the number of topics.

3.4.2 RUN-TIME EXECUTION

A technique must cope with high dimensionality to successfully analyze real-world data. Thus, we analyze the run-time execution of *cExpression* and *ccPCA* by varying the number of features and samples to perform analysis—we focus only on the techniques used to retrieve contrastive information. The experiment was performed in a dataset composed of 40794 tweets about COVID-19 symptoms (see Section 5.3 for a detailed description of a smaller version of it). Then, to evaluate the techniques, we generated different versions of the dataset by ranging the number of features between ten and 2000 and the number of samples between 2000 and 40000. Fig. 42 shows that, for eight clusters, *cExpression* is faster than *ccPCA* when we augment the dataset dimensionality. Appendix B (Fig. 86) also shows run-time executions for different number of clusters presenting the same pattern

of Fig. 42, which testify the superiority of *cExpression* over *ccPCA* regarding run-time execution.

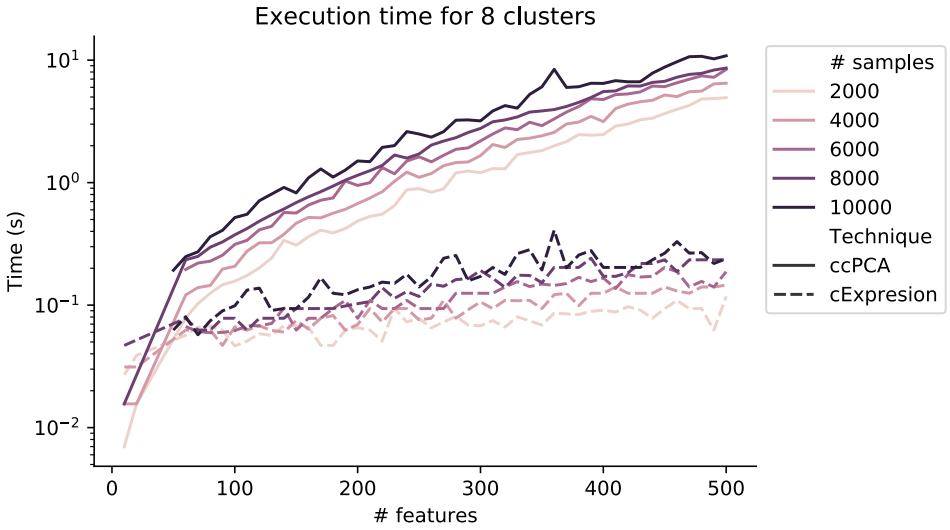


Figure 42 – Time execution in logarithmic scale for retrieving distinctive features using contrastive analysis.

3.5 FINAL REMARKS

The use of contrast analysis for dimensionality reduction enables the understanding of how clusters differ in the projected space. Other aspects of our approach are discussed further below.

Cell-based encoding of scatterplot. To facilitate local and global analyses among data points, the distribution of feature values visualized on the scatterplot helps analysts gain insights about the datasets and the characteristics of a dimensionality reduction technique to organize the projected space. The cell-based encoding also reduces change blindness since users are not required to switch between color scales to visualize the feature value distribution on the scatterplot.

Usability In the use cases, cExpression's strengths were assessed in order to better understand cluster formation after dimensionality reduction. However, we intend to evaluate it in terms of usability via user experiments in order to understand how it benefits analysis in well-defined tasks.

Analyzing different DR techniques. The way cExpression's results differ for different DR reduction techniques is an important aspect. Appendix B includes two additional analyses of this characteristic. The results show that if the clusters are the same—that is, the same data points in the same clusters—the organization of the projected space does not matter, which can be explained by the fact that the same data subsets to cExpression. We intend to define the subsets using k nearest neighbors, which will

make the results more dependent on the projection structures rather than the projection clusters.

Although our work addresses important problems in the literature, there are few other aspects that need further research.

Visual scalability of the number of clusters.In Section 3.2.2.2, we discuss visual scalability with respect to dataset dimensionality. Another point of view is related to the number of clusters; the more clusters there are, the more iterations the user must perform to cover the entire dataset. While we do not address this issue in this document, our methodology is fast enough that it does not add a burden to the analysis. Furthermore, we intend to investigate how multilevel dimensionality reduction strategies (PEZZOTTI et al., 2016; MARCíLIO-JR et al., 2021a; MARCíLIO-JR et al., 2021b), such as overview first and details on-demand operations, can help to alleviate the analysis.

Visual scalability of the number of features. Users can better distinguish among different classes when visualizing the distribution of a larger number of features on the scatterplot view by augmenting the area of the boxes to avoid the space dedicated to visualizing feature intensity. Such an increase, however, may hinder the context preservation between cell-based visualization and scatterplot. The number of simultaneous features visualized can also affect users' ability to distinguish between feature intensities. We intend to investigate into glyph-based visualizations to encode more information, such as using a star plot and color encoding to visualize feature values and colors, respectively.

Assumptions for the t-Student test. Although the t-test provides information about the dataset's discriminative power, we have assumed that the samples have normal distributions. Furthermore, computing the t-statistic can be difficult because variance estimates can be skewed by features with very low variance (JEANMOUGI et al., 2010). As a result, these characteristics are associated with a high t-statistic and incorrectly chosen discriminative terms (TUSHER; TIBSHIRANI; CHU, 2001). Another disadvantage is that it has low statistical power when applied to small sample sizes (MURIE et al., 2008). One remedy to these limitations is already built into our tool. It enables users to investigate the distribution of values in the scatterplot representation using the color scale or to visualize multiple features at the same time using the encoding.

Selection of clusters in the visual space. As discussed in this chapter, we concentrate on explaining visual clusters, or clusters perceived in visual space. For trustworthy and consistent exploratory analysis, the input data containing the cluster labels must match the clusters in the visual space. When this is not the case and different cluster labels are assigned to different clusters in the visual space, the cExpression technique will fail to identify truly distinct features because the feature value distribution will be similar (see Section 3.2.1). This limitation, however, is easily overcome by preprocessing the input file with consistent cluster label assignment.

4 EXPLAINING DIMENSIONALITY REDUCTION RESULTS USING SHAPLEY VALUES

Wilson E. Marcílio-Jr, Danilo M. Eler, *Explaining dimensionality reduction results using Shapley values*, Expert Systems with Applications, Volume 178, 2021.

After dimensionality reduction, researchers perform exploratory analysis on datasets to look for patterns and additional relevant details (DR). Numerous studies have used DR techniques for exploratory data analysis, including understanding how CNNs learn features over time (PEZZOTTI et al., 2018) and examining gene expression patterns to identify novel cell types (UNEN; HöLLT; PEZZOTTI, 2018). Although DR techniques provide a great opportunity for high-dimensional data analysis, analysts must interpret the decisions made by these algorithms to determine whether the embeddings encode information in the high-dimensional space. Understanding embeddings, for example, assists machine learning practitioners in assessing the quality of feature spaces in terms of class separation (MARCILIO-JR et al., 2020).

One possible solution for interpreting dimensionality reduction techniques is to analyze the features' contributions to the embedding, assessing how much each feature contributed to the formation of clusters or other visible structures in the projected space (e.g., $\mathbb{R}^2$). In gene expression analysis, for example, bioinformaticians want to know which genes contribute to a cluster in order to annotate cell types or discover new ones (LäHNEMANN; KöSTER; SZCZUREK, 2020). Finding the contribution of these features to the dimensionality reduction result is primarily a problem for non-linear DR techniques (MAATEN; HINTON, 2008; MCINNES; HEALY; MELVILLE, 2018), where there is currently no way to inverse calculations and keep track of feature contributions while the algorithm is running (FUJIWARA; KWON; MA, 2019). Non-linear DR techniques, on the other hand, are best suited for dealing with most datasets due to their ability to uncover complex structures.

Existing DR interpretation techniques have a few weaknesses. Studies on feature values (COIMBRA et al., 2016; PAGLIOSA; PAGLIOSA; NONATO, 2016; MARCILIO-JR et al., 2020), for example, do not account for the dimensionality reduction process and instead concentrate on the reduced low-dimensional space. Other more elaborated works (TURKAY et al., 2012; JOIA; PETRONETTO; NONATO, 2015) retain feature importance through the principal components (PC). The PCs produce biased results for

classes with high variation (JOIA; PETRONETTO; NONATO, 2015), and their inability to focus on local information (FUJIWARA; KWON; MA, 2019) hinders cluster-oriented analysis. A robust approach known as ccPCA (FUJIWARA; KWON; MA, 2019) employs contrastive PCA (ABID et al., 2018) to comprehend DR results based on information specific to each cluster. ccPCA emphasizes the differences between each cluster using contrastive analysis. For example, for a dataset of classification-related machine learning papers, it would return the information that distinguishes them, such as the classification method. These techniques, however, cannot explain how much each feature contributed to the DR result. The importance assigned to the features does not account for their contribution to clusters and other structures. Furthermore, these feature importance measures do not interact with one another to form an explanation measure. Instead, the significance of each feature is unrelated to the other.

This chapter introduces a new method, called ClusterShapley[1], for interpreting dimensionality reduction results, as well as carefully designed visual metaphors. We propose a novel methodology for explaining feature contributions in cluster formation in dimensionality reduction results. We interpret DR results using the features' contributions in an additive way to show how much each feature contributes to the resulting projection in visual space, using Shapley values (SHAPLEY, 1953) to derive explanations ($\mathbb{R}^2$). Aside from explaining DR results and supporting analysis with Shapley value similarity, our methodology allows extrapolation from feature contribution to feature importance in cluster formation. The ClusterShapley method assists analysts in understanding the decisions made by DR techniques after projection. Finally, we propose summary visualizations for Shapley values, as well as Kernel Density Estimation (ROSENBLATT, 1956; PARZEN, 1962) to aggregate highly correlated feature contributions.

The case studies demonstrate how feature contributions can explain interesting patterns and provide insights into medical and social datasets. Then, using ClusterShapley, we discuss the implications of our work by outlining potential applications. Finally, we would like to emphasize that this is the first study to use Shapley values to explain dimensionality reduction results in a cluster-oriented analysis. Summarily, the contributions detailed in this chapter:

- A methodology for applying Shapley values to explain dimensionality reduction results upon a cluster-oriented analysis;
- Summary visualizations to encode feature contributions based on Shapley values;
- Categorization of important features in datasets about pathologies and patients in different conditions.

[1] https://github.com/wilsonjr/ClusterShapley

The following sections are organized as follows: we delineate the related works in Section 5.1, a brief background on Shapley values is provided in Section 4.2, we explain our methodology and the visualization design in Section 4.3, validation of our approach is performed through case studies in Section 5.3.

4.1 RELATED WORKS

To improve interpretation capabilities of dimensionality reduction (DR) techniques, researchers provide additional information to these methods' results. A few works include visual information such as bar charts and color encoding to interpret three-dimensional projections (COIMBRA et al., 2016) or encode attribute variation using Delaunay triangulation to assess neighborhood relations in two-dimensional projections (SILVA et al., 2015). Probing Projections (STAHNKE et al., 2016), for example, depicts error information by displaying a halo around each circle in the scatterplot representations besides providing interaction mechanisms to understand distortions in the projection process. The majority of the works use traditional statistical charts to visualize attribute variability (PAGLIOSA; PAGLIOSA; NONATO, 2016), neighborhood and class errors (MARCILIO; ELER; GARCIA, 2017), or quality metrics (KWON et al., 2018).

More related to our work are the techniques that try to find important features given clusters of data samples. For example, the Linear Discriminative Coordinates (WANG et al., 2017) use LDA (IZENMAN, 2008) to produce cohesive clusters by discarding the least important features. (JOIA; PETRONETTO; NONATO, 2015) use PCA to find the most important features by a simple matrix decomposition and visualize feature names as word clouds within each cluster region. Although useful and fast, classes with high variation influence the result. Another work, proposed by (TURKAY et al., 2012), also uses PCA's principal components to obtain the representative features of a multidimensional scaling (KRUSKAL; WISH, 1978) projection. Recently, (FUJIWARA; KWON; MA, 2019) proposed a cluster contrastive PCA (ccPCA) technique that finds the most important features for a given cluster in contrast with the other clusters in a DR result. Fujiwara et al.'s approach is different from Joia et al.'s and Turkay et al.'s works. It provides a way to understand which features positively contribute to the differentiation of clusters.

These works cannot explain how much a feature contributes to the dimensionality reduction result. That is, the feature importance does not show its contribution to the dimensionality reduction result.

Our work advances the state-of-the-art interpreting dimensionality reduction results by showing how each feature contributes to the results. It explains the position of the data samples on the embedding through the combination of feature contribution.

4.2 REVIEW OF SHAPLEY VALUES

When explaining machine learning models, one critical consideration is how to explain predictions without considering the model itself. According to (STRUMBELJ; KONONENKO, 2014), the critical component of a model-agnostic explanation is the contribution of features to prediction. In other words, the impact of each dataset feature explains the prediction. Let f be a machine learning model and x be a dataset X instance. The feature's situational importance (ACHEN, 1982) (Equation 4.1) computes how a specific value influences a prediction. It is the difference between the feature's contribution when its value is x_i and its expected contribution, where β_i denotes the global importance of the i-th feature.

$$\varphi_i(x) = \beta_i x_i - \beta_i E[X_i]. \tag{4.1}$$

The situational importance signal for feature i indicates a positive, negative, or no contribution. Although such contributions are simple to compute for additive models (such as linear regression models), they are difficult to compute for complex models due to feature interaction. The conditional expectation of a model's prediction (STRUMBELJ; KONONENKO, 2014), defined in Equation 4.2, takes into account all subsets of features (Q).

$$f_Q(x) = E[f|X_i = x_i, \forall_i \in Q]. \tag{4.2}$$

Using Equation 4.2, Shapley values, a concept from coalitional game theory (SHAPLEY, 1953), can be used to measure each feature's contribution to model prediction (LUNDBERG; LEE, 2017). Shapley values are calculated by averaging each feature permutation on the conditional expectation of a model prediction, or by measuring the change in prediction after including a feature in Equation 4.2. Thus, in the equation Equation 4.3 (LUNDBERG et al., 2020), Q represents all feature permutations, y_i^q is the set of all features that come before feature i in the permutation q, $y_i^q \cup i$ is the union of the set of all features that come before feature i in the permutation q and the feature i itself, and $|Y|$ corresponds to the number of input features for the model.

$$\phi_i(f, x) = \sum_{q \in Q} \frac{1}{|Y|!}(f_{(y_i^q \cup i)}(x) - f_{(y_i^q)}(x)) \tag{4.3}$$

To compute Shapley values for a dataset with n features, 2^n model evaluations would be required, making it prohibitively expensive for moderate numbers of n. As a result, we employ KernelSHAP (LUNDBERG; LEE, 2017) to approximate Shapley values via a linear regression model.

4.2.1 SHAPLEY VALUES ILLUSTRATION

Consider a synthetic scenario to demonstrate how Shapley values aid in understanding the contributions to a model's prediction (MOLNAR, 2019). Assume we trained a machine learning model to predict house prices, and it predicts $300,000 for a specific house. Our model used the following features to make such a prediction: `pets allowed`, `one-bedroom`, 100 m2 `size`, and two `bathrooms`.

Our synthetic scenario's average prediction was 290 thousand dollars. In our case, Shapley values indicate how much each feature contributed to the prediction in comparison to the average. The difference in contributions for the prediction (300 thousand dollars) and the average prediction (290 thousand dollars) is explained by Shapley values, which is ten thousand dollars. Finally, one possible solution to this problem could be: `pets allowed` contributed 40 thousand dollars, `one-bedroom` contributed -60 thousand dollars, `size` of 100 m^2 contributed 10,000, and two `bathrooms` contributed 20 thousand dollars. Take note that the total of these values is $10,000.

Now, following the idea discussed to formulate Equation 4.3, the Shapley value for a feature consists of the average contribution of such a feature according to all possible feature permutations. To estimate the Shapley value for the `pets allowed` feature, one has to compute the prediction with: `pets allowed`; `pets allowed` and `size` of 100 m^2; `pets allowed`, `size` of 100 m^2, and `two bathrooms`, and so on—the feature `pets allowed` also has to be removed for each permutation. Finally, by removing a feature for the prediction, the idea in practice is to assign a random value (from another data sample) to the "removed" feature.

Due to its exponential nature, the application of Shapley values is infeasible for high-dimensional datasets. Thus, as discussed previously, we use a technique called KernelSHAP (LUNDBERG; LEE, 2017) that estimates Shapley values through a linear regression upon sampled permutations.

4.3 USING SHAPLEY VALUES TO EXPLAIN CLUSTERS OF DIMENSIONALITY REDUCTION RESULTS

This section describes how to define clusters for computing explanations with Shapley values. Then, we outline the visualization system that will be used to assist analysts in interpreting embeddings.

Fig. 43 depicts the two main components used to generate explanations following dimensionality reduction. The first component, *Dataset annotation* (1.), defines the clusters for explanation in the second component, *Shapley estimation* (2.). Instead of displaying the Shapley values as a measure of importance to the dimensionality reduction

result, visual metaphors are used to assist exploratory analysis.

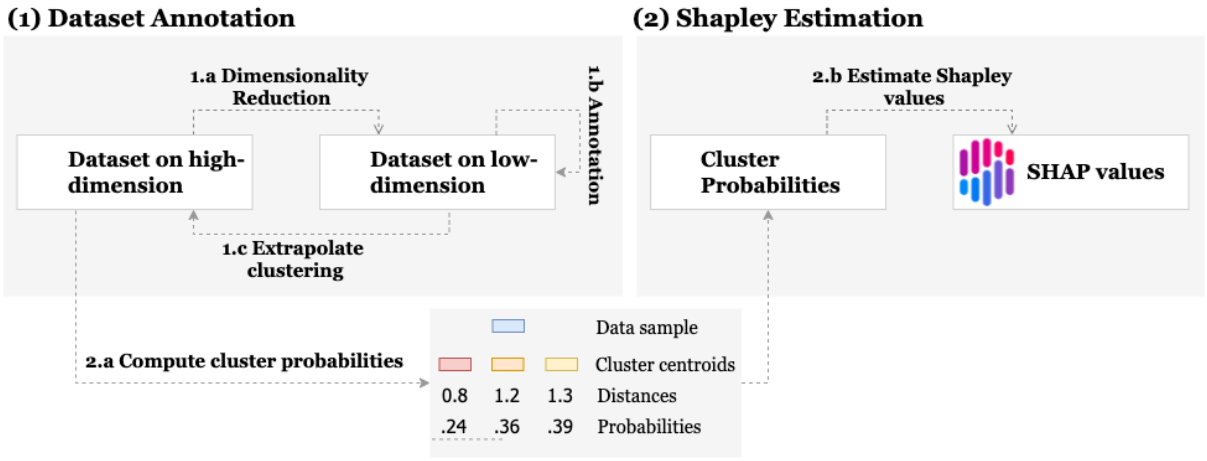


Figure 43 – ClusterShapley framework. In the "dataset annotation" component (1), a dimensionality reduction (1.a) process helps users to analyze and to annotate clusters perceived in the visual space ($\mathbb{R}^2$); then, the clusters defined in $\mathbb{R}^2$ annotate the high-dimensional dataset (1.c). In the "Shapley estimation" component (2), the annotated high-dimensional dataset is used to generate clusters probabilities for each data sample (2.a) employed for Shapley values estimation (2.b).

4.3.1 DATASET ANNOTATION

We rely on interpreting the clusters formed in the projected space to explain dimensionality reduction (DR) results ($\mathbb{R}^2$). The goal of DR techniques is to reduce the dimensionality of a high-dimensional dataset ($\mathbb{R}^m$) to a low-dimensional dataset ($\mathbb{R}^d$) while preserving (as much as possible) the data structures—$d = 2$ is commonly used to visualize the outcome of a DR technique in which the visual proximity of data points encodes similarity. For instance, clusters are rapidly perceived in the visual space ($\mathbb{R}^2$) since humans quickly notice groups of visual markers (BERTIN, 1983).

To understand the DR technique's decisions that result in the embedding, we use $\mathbb{R}^2$ clusters to annotate the dataset in high-dimensional space. In this case, users can freely define clusters using mouse interaction, as shown in Fig. 44, where the black color encodes data samples that have not been assigned to any cluster. When using labeled datasets, users may choose data samples from different classes projected on the same cluster by using such an approach. Because the goal is to understand and explain the clusters in the visual space, such an idea is reasonable and consistent with our proposal.

Of course, labeled datasets and clustering techniques (KAUFMAN; ROUSSEUW, 2005) are two other common ways to define clusters. We want to define as many clusters as the number of clusters perceived in the visual space using one of these three strategies. For example, in the projection of Fig. 44, seven clusters could be defined (as shown for a study case in Section 4.5.3).

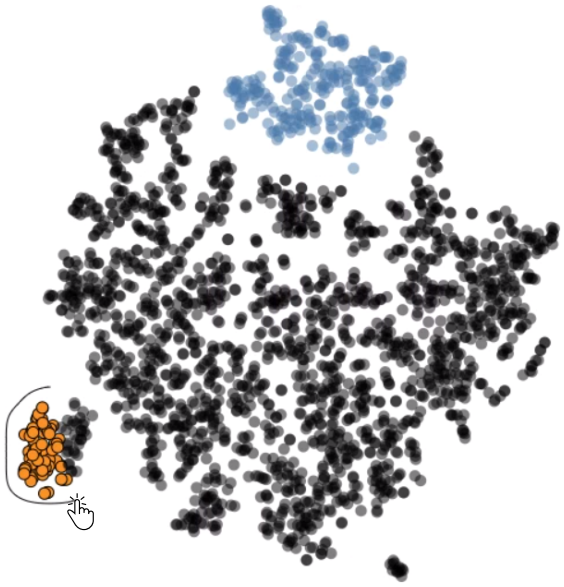


Figure 44 – Cluster definition through manual selection. Users receive a projected dataset with no clustering imposed, and then users define clusters by lassoing the groups perceived in the visual space. Notice that black circles indicate data samples not belonging to any cluster, while circles with different data hues indicate already assigned data samples.

4.3.2 SHAPLEY ESTIMATION

Shapley values are estimated for each data sample after the cluster definition. Thus, based on the cluster definition discussed in the previous section, a model f returns the prediction similarities for a data sample x.

We measure the distance from x to each cluster centroid to return the prediction similarities for a data sample x—note that we use "prediction similarities" because lower values indicate high-similarity between x and cluster centroids. Fig. 43 (2.a—bottom) depicts this procedure for three cluster centroids (■, ■, ■) and, as a result, a three-dimensional distance vector. The distances are then converted into similarities using L1 normalization. The Shapley estimator (in our case, KernelSHAP (LUNDBERG; LEE, 2017)) uses these similarities to generate the explanations discussed in Section 4.2. While estimating the Shapley values with KernelSHAP accounts for the majority of the dataset, we only compute the estimation for 20% of it. This procedure will generate a matrix of dimensions $n \times M$ for each cluster, where n represents 20% of the dataset size and M represents the dataset's dimensionality—each cell i, j of the matrix will contain the Shapley value of the data point i for the feature j.

Algorithm 2 depicts the Shapley value estimation process in greater detail. First, the method splits the dataset into training and test sets (Line 8) to compute Shapley values for the test set (Line 1) while fitting the technique to the training set (Line 9). To create a Shapley estimation instance, we must provide a function that returns the

prediction probabilities. This function (cluster_probability) computes the distance from a data sample x to the cluster centroids (Lines 4 and 5) and then returns the prediction probabilities (prediction similarities in our case) in Line 6.

Algorithm 2 Shapley estimation.

```
1: procedure CLUSTER_PROBABILITY(x ∈ ℝ^m, X ∈ ℝ^m)
2:     C ← get_centroids(X)
3:     D ← ∅
4:     for i ∈ |C| do
5:         D ← D ∪ (‖ x − C_i ‖)
6:     return D/ ‖ D ‖_1
7: procedure SHAPLEY_ESTIMATION(X ∈ ℝ^m)
8:     X_train, X_test ← split(X, 0.2)
9:     SE ← KernelSHAP(cluster_probability, X_train)
10:    SV ← SE(X_test)
11:    return SV
```

Because we are applying L1 to distances, lower probabilities indicate better cluster cohesion. As we will see later, negative Shapley values indicate that a feature contributed to cluster cohesion, whereas positive Shapley values influence cluster non-cohesion. This idea is consistent with the visual space, where visual proximity encodes similarity.

Finally, we emphasize that any technique, such as t-SNE (MAATEN; HINTON, 2008), LSP (PAULOVICH et al., 2008), or UMAP (MCINNES; HEALY; MELVILLE, 2018), can perform the dimensionality reduction step. When using different dimensionality reduction techniques, different clusters may be perceived in the visual space. As a result, labeling these clusters (see Section 4.3.1) would be the first step in putting ClusterShapley to use.

The estimated Shapley values correspond to the contribution of each feature to the embedding. As a result, a feature with high absolute Shapley value contributes significantly to the cluster formation of the projected dataset. Each data point used for Shapley value estimation in this case contains the corresponding Shapley value. Negative Shapley values indicate that the feature contributes to cluster formation, while positive Shapley values indicate that the feature does not.

4.4 VISUALIZATION DESIGN

Fig. 45 shows a prototype system that uses ClusterShapley to explain the *Iris* (DUA; GRAFF, 2017) dataset embedding. The system is composed of three components. Users provide datasets to generate explanations and load stored explanations in the first component (A). In the second component, a scatterplot representation of DR results is drawn (B). A visual summary of the Shapley values for the four most important features is shown in the tooltip (C) next to the blue cluster. Users can also click on a circle to see a

more detailed analysis of the Shapley values, as shown in the image below (D). Dot plots and aggregated Kernel Density Estimation (ROSENBLATT, 1956; PARZEN, 1962) of the absolute sum of Shapley values are used in the detailed analysis. The **Importance Summary** is depicted in (E), where users evaluate the mean values of the four most important features for each class and their contribution to characterization of the clusters depicted by bar-plots. Finally, we present a heatmap of the sum of Shapley values in the (F). Each component is described in detail in the sections that follow.

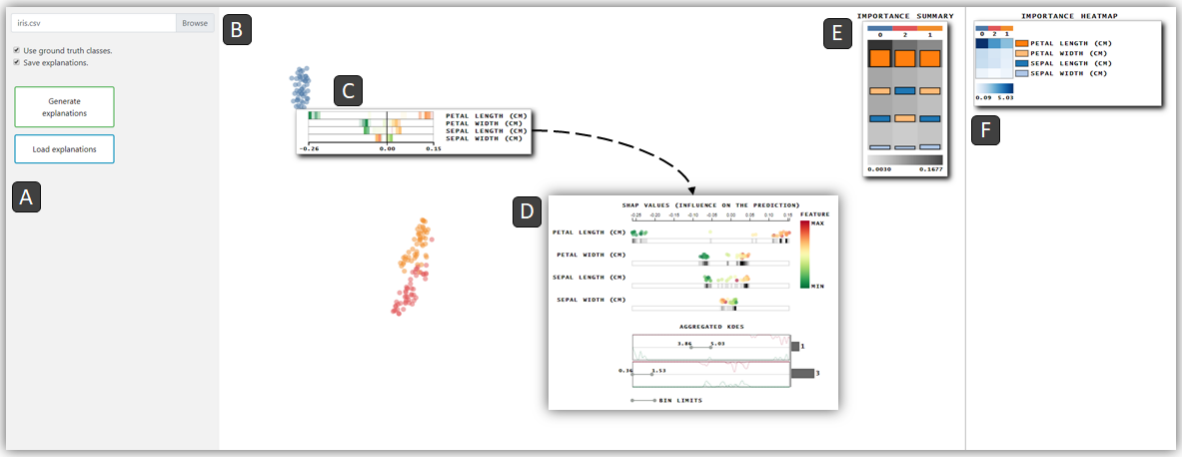


Figure 45 – An overview of the prototype system. In (A), users can specify datasets and load previously generated explanations. In (B), we draw a scatterplot representation and use color to encode classes. Hovering over specific class instances generates summary representations of Shapley values, which are used to indicate importance (C). The detailed examination of Shapley values and feature importance assists analysts in comprehending the relationship between feature value and Shapley value (D). The **Importance Summary** is shown in (E), which summarizes the mean of the absolute Shapley values for each pair (class, feature). The importance overview is depicted by a heatmap with the sum of the absolute Shapley values (F).

4.4.1 SCATTERPLOT COMPONENT

To provide a rapid assessment of Shapley values, we use a tool-tip with summarized information about the interactions between feature values and Shapley values when users hover over circles of a particular class. The process of summarizing the Shapley values and feature values works as follows. $\mathcal{H}_e^i$ and $\mathcal{H}_o^i$ are the histograms created from the Shapley values for the feature i, where $\mathcal{H}_e^i$ stores information for values equal or greater than the mean (σ_i) and $\mathcal{H}_o^i$ stores information lower than the mean of feature values for feature i. Knowing that Shapley_{min} and Shapley_{max} represent the lowest and highest Shapley values for the visualized features, we divide the histograms $\mathcal{H}_e^i$ and $\mathcal{H}_o^i$ into the same number of bins, with Shapley_{min} and Shapley_{max} as bin limits.

Color saturation encodes the density of each histogram—green for $\mathcal{H}_e^i$ and red for $\mathcal{H}_o^i$—and the bins of the two histograms are drawn sequentially one after the other. That is, while $\mathcal{H}_e^i$ takes the even positions, $\mathcal{H}_o^i$ takes odd positions. Fig. 46 illustrates this procedure. If only one histogram has a density greater than zero for a given bin, the even and odd positions will be used.

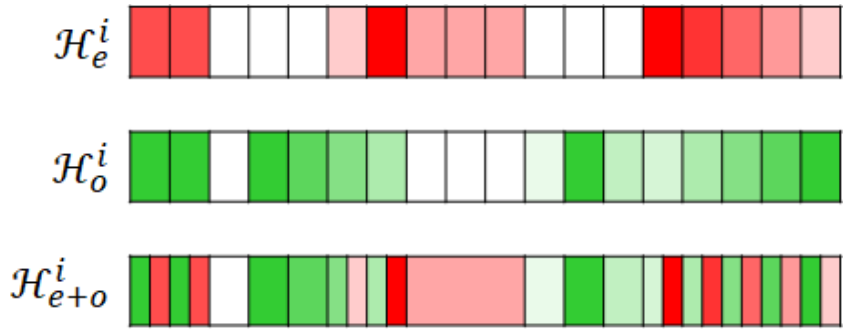


Figure 46 – Summarization approach to encoding two histograms for a given feature i, $\mathcal{H}_e^i$ for feature values equal or greater than the mean (σ_i) and $\mathcal{H}_o^i$ for features values lower than the mean. The two histograms are encoded together by alternating their bins.

The summarized information from real data looks like in Fig. 47, where reddish colors encode feature values above the mean and greenish colors encode feature values below the mean. The feature value's contribution is encoded using the position x axis, which corresponds to the Shapley values—the contribution is proportional to the distance from zero, as indicated by a vertical line segment. The ordering in the representation indicates the overall importance of the feature, i.e., `petal length (cm)` is the most important, followed by `petal width (cm)`, and so on.

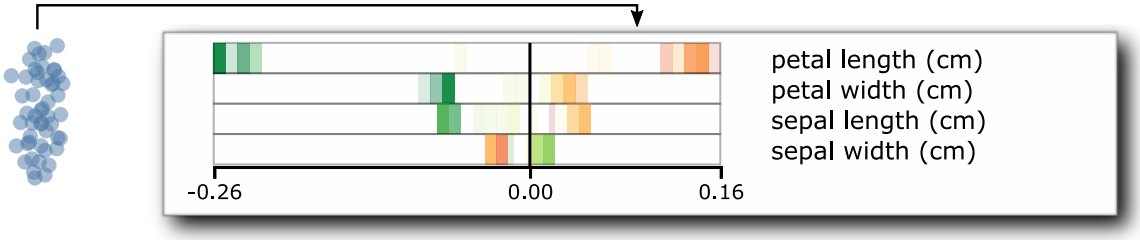


Figure 47 – A tool-tip for summarizing information from the blue cluster of Fig. 45.

4.4.2 COORDINATION BETWEEN SCATTERPLOT AND SHAPLEY VALUES

Understanding how the combination of feature values affects the embedding requires analyzing the correlation between feature values and Shapley values. Thus, a detailed analysis shows up when users click on a circle representing a particular class. Fig. 48 shows dot plots of the four most important features to support such a detailed analysis. For

instance, the four most important features, in order of importance, are `petal length (cm)`, `petal width (cm)`, `sepal length (cm)`, and `sepal width (cm)` It's worth noting that the more the absolute Shapley values (encoded as horizontal position) deviate from zero, the more influence the feature has in characterizing the cluster. As a result, the density bars beneath each dot plot aid in determining how much the feature values contribute to cluster characterization.

We visualize the subset of data samples used to compute Shapley values in the dot plot representation. Each circle represents a data point's feature value. While color represents the feature value, the positions of the points represent the Shapley values of the inspected cluster. Finally, assume the dataset contains more than four features. In that case, they can also be inspected using a histogram based on the absolute sum of their Shapley values, as shown in Fig. 48 (**Aggregated KDEs**). The tool shows only the bins with elements (as defined by the bin limits), while the bars indicate how many features are in each bin limit. In Fig. 48, for example, there is one feature inside [3.86, 5.03) and three features inside [0.36, 1.53). Each bin's aggregation is encoded by two Kernel Density Estimation (KDE) curves: the red curves encode feature values equal to or greater than the mean, and the green curves encode feature values less than the mean. Here, `petal length (cm)` is contained within the bin with limits [3.86, 5.03), and `petal width (cm)`, `sepal length (cm)`, and `sepal width (cm)` are contained within the bin with limits [0.36, 1.53).

The coordination mechanism between the scatterplot and the Shapley values assists users in determining which features contributed to cluster cohesion and which features contributed to cluster dispersal during the detailed inspection. Fig. 48 depicts the outcome of these operations after lassoing data points from the blue cluster. On the dot-plot, polylines connecting their feature values show how feature combinations influence cluster formation.

4.4.3 IMPORTANCE SUMMARY AND IMPORTANCE HEATMAP

Color intensity depicts the mean of absolute Shapley values for the four most important features in the **Importance Summary** (see Fig. 45 (E)). The contribution of each feature is depicted by bar height, and the feature values are encoded by color. Users can assess the heterogeneity of the importance of features for different clusters by positioning the feature bar with different colors.

The **Importance Heatmap** (see Fig. 45 (F)) shows the absolute sum of the Shapley values for each feature to provide an overview of the features' contributions. As in Fujiwara et al's (FUJIWARA; KWON; MA, 2019) approach, after computing the absolute sum for each pair of features and clusters, we apply hierarchical clustering to the heatmap's rows and columns to order the cells and facilitate cluster identification. The clustering

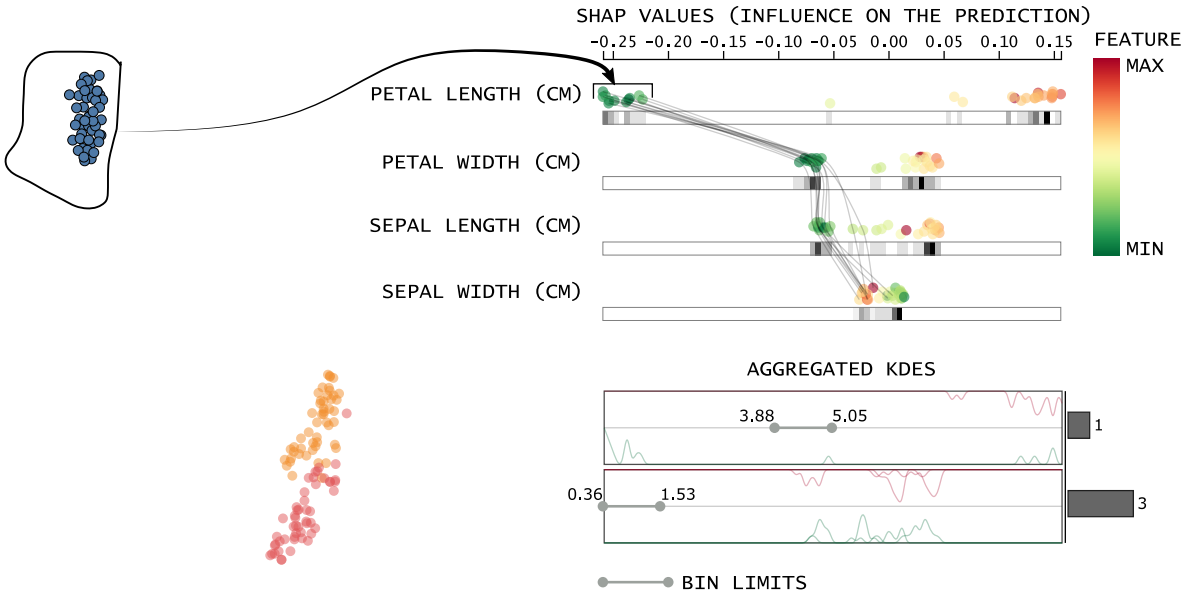


Figure 48 – Coordination between the scatterplot and the dot plot encoding the Shapley values. Each line segment represents a point in the scatterplot representation.

leaves are then optimally ordered to produce more understandable results, with heatmap values that are similar placed close together.

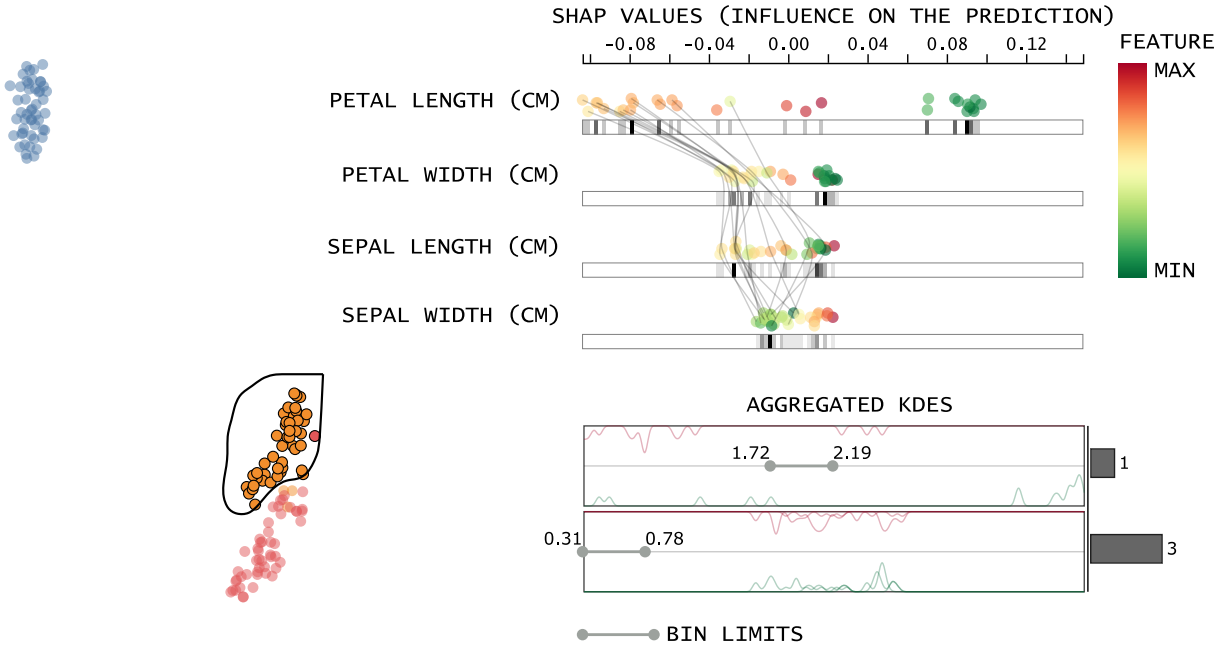
4.4.4 ANALYSIS EXAMPLE

To familiarize readers with the ClusterShapley analysis, we examine the explanations for an Iris dataset embedding. Furthermore, we use an icon (such as ■) and the cluster's indices to make reading easier.

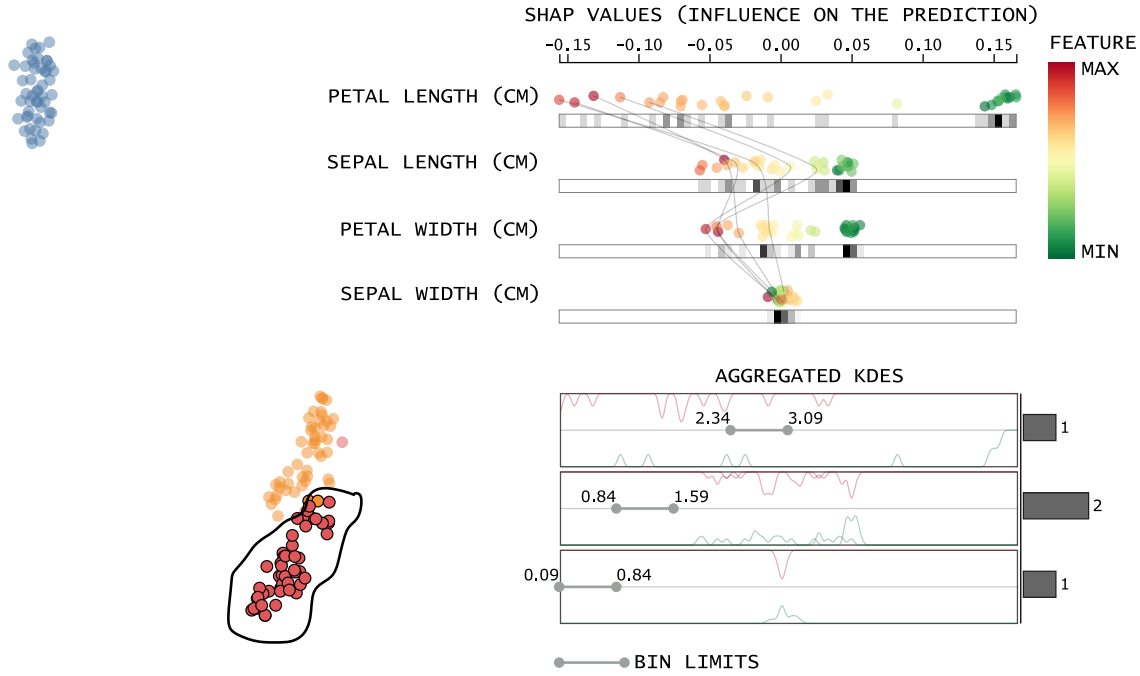
Fig. 45 shows the separation of cluster 0 ■ from the other clusters. According to the **Importance Summary**, `petal length` has a much greater influence on the cluster. The summary visualization of Shapley values in Fig. 45 (C) provides an indication of how feature values influence cluster formation, i.e., there is a clear separation of lower and higher `petal length` values. The coordination of the scatterplot and dot plot in Fig. 48 shows that instances of cluster 0 ■ are very different from the others due to lower values of `petal length`, `petal width`, and `sepal length`. The negative Shapley values indicate that their feature values facilitated cluster cohesion.

Clusters 1 ■ and 2 ■ (shown in Fig. 49) have similar `petal length` patterns (the most important feature). Higher values for this feature are required for cohesion in both clusters (see negative Shapley values). Furthermore, while the features of `sepal length` and `petal width` take different positions in the importance ordering, their overall importance appears to have little effect when the density plot patterns are considered. Finally, unlike clusters 0 ■ and 2 ■, `sepal width` has a significant influence on cluster 1 ■, where lower values help characterize the clusters—as seen in Fig. 49a by the higher

density for green areas of the dot plot.



(a) Feature values for cluster orange seems to be moderate (yellowish colors).



(b) Feature values for cluster red seems to be higher if compared with the features of orange cluster.

Figure 49 – Two clusters with low separation. While the first three features show the same overall importance for these two clusters, `sepal width` plays a significant role in differentiating cluster orange (a).

4.5 CASE STUDIES

In the case studies, ClusterShapley is used for exploration of datasets from different scenarios. First, we provide empirical evidence that Shapley values help to generate insights about pathologies and patients in different conditions. Then, ClusterShapley is used to emphasize the differences among the quality indices of red wines. Finally, we analyze a social dataset.

All of the embeddings were generated using t-SNE (MAATEN; HINTON, 2008), on a computer with the following configuration: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 32GB RAM, Windows 10 64 bits.

4.5.1 VERTEBRAL COLUMN

In this first case study, we analyze a dataset containing six biomechanical features. The *Vertebral Column* dataset (DUA; GRAFF, 2017) is composed by 310 instances described by six features derived from the shape and orientation of the pelvis and lumbar spine: `pelvic incidence`, `pelvic tilt`, `lumbar lordosis angle`, `sacral slope`, `pelvic radius`, and `grade of spondylolisthesis`. Fig. 50 shows the projected instances, colored based on the ground truth classes: class 2 for normal patients, class 0 for patients with Hernia, and class 1 for patients with Spondylolisthesis—a disturbance of the spine in which a bone (vertebra) slides forward over the bone below it. There is a clearly separation of class 1 among the remaining data points.

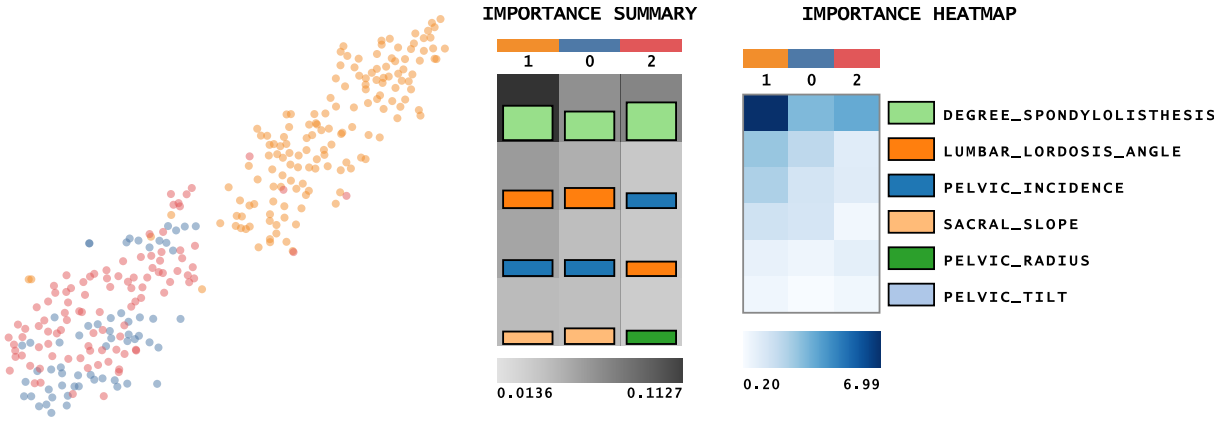


Figure 50 – Explanations for the *Vertebral Column* dataset. There is a clear separation among patients with complications (Spondylolisthesis or Hernia) and the healthy ones. The **Importance Summary** also supports visualization of such characteristics.

The **Importance Summary** in Fig. 50 shows that the degree of `spondylolisthesis` is a determinant factor for the presence of Spondylolisthesis—notice the feature legend next to the **Importance Heatmap**. Although it is important for characterizing all clusters, the color intensity and the heatmap values show that lower means of Shapley values can

characterize the absence of Spondylolisthesis. By coordinating the scatterplot and the dot plot for class 1 (see Fig. 51), its data points have higher values for the `degree of spondylolisthesis`. Further that, these higher feature values assume negative Shapley values and contribute to cluster formation. The other three most important features also contributed to clustering cohesion. The pelvic incidence angle measures the pelvic shape and determines the position of the sacral endplate (TEBET, 2014). According to (LABELLE et al., 2005), pelvic incidence, sacral hill, pelvic tilt, and lumbar lordosis are greater in patients with developmental Spondylolisthesis.

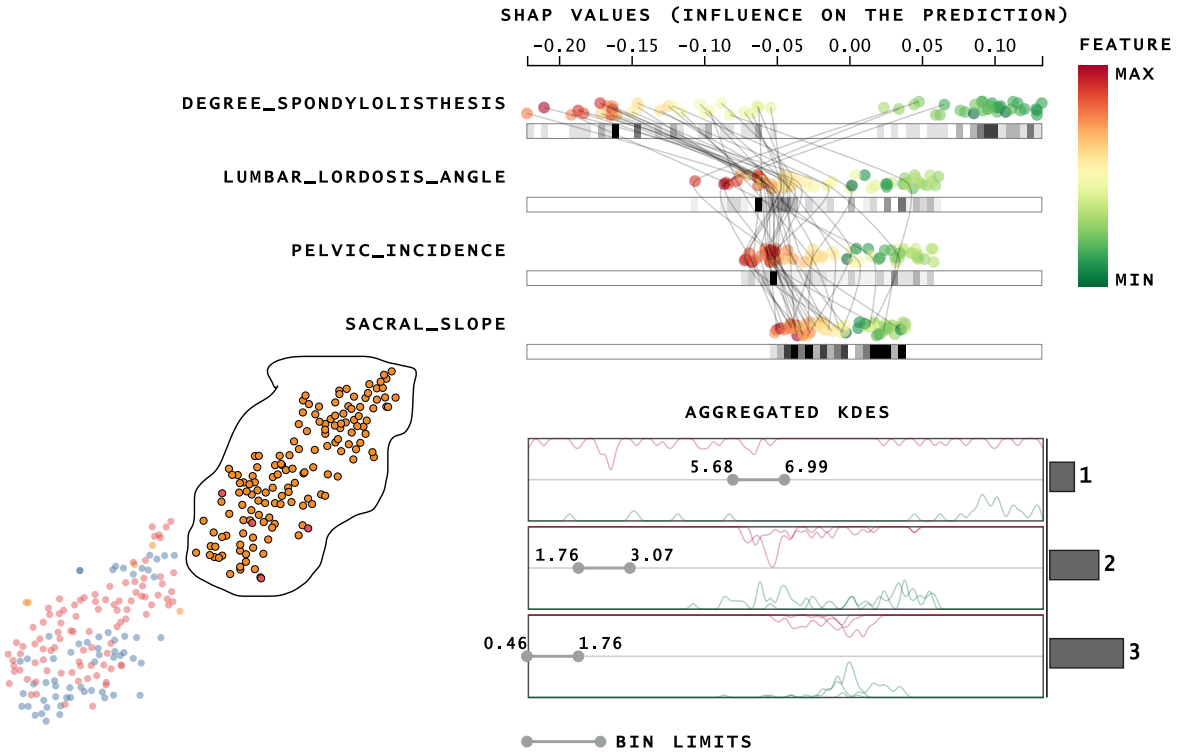


Figure 51 – Correlation between feature values and their importance for categorizing cluster 1. The four most important features contributed to differentiate patients with Spondylolistyhesis from the others.

Fig. 52 shows the dot plots for classes 0 and 2 . Given the influence of the `degree of spondylolisthesis` for both classes, Spondylolisthesis's disturbance is not likely to be present in those instances due to the low feature values. The DR technique did not impose a clear separation between these classes because the features are not distinctive. However, there is a slight separation due to the degree of feature contribution—as visualized in the aggregated KDEs. Finally, we see evidence for differentiating Hernia patients (B) from regular patients (A). That is, patients with Hernia have low values for sacral slope (see this pattern for patients with Hernia in Fig. 52(b)), which can indicate centralistic herniation (ROUSSOULY; PINHEIRO-FRACO, 2011).

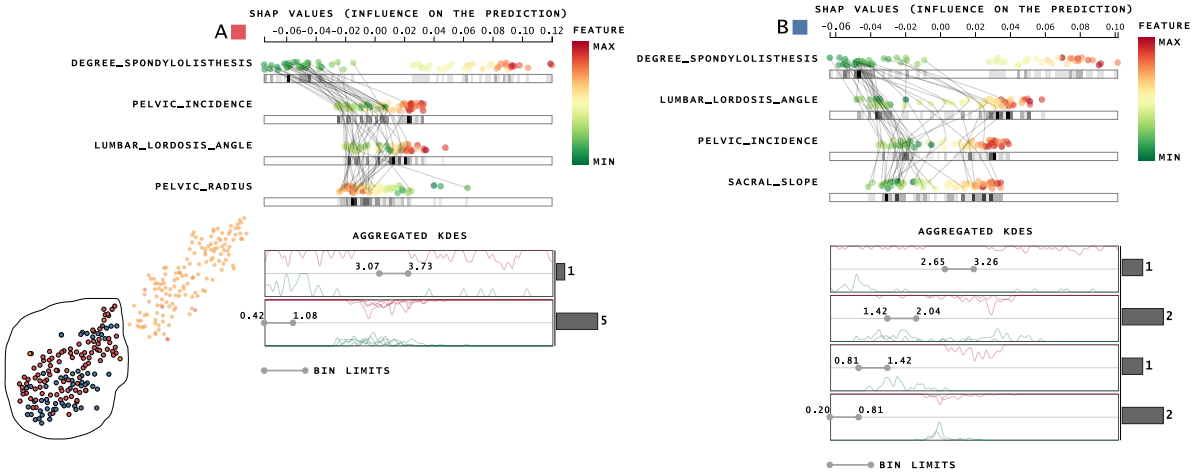


Figure 52 – Comparison between regular patients (A) and patients with Hernia (B). The difference in values for `sacral slope` was not good enough to impose a good separation on the dimensionality reduction result.

4.5.2 RED WINE QUALITY

For this case study, we understand the quality of red wines (DUA; GRAFF, 2017) by explaining manually defined clusters after dimensionality reduction. Fig. 53 shows a t-SNE projection with grayscale encoding wine quality. The dataset contains 1599 instances described by 11 features: `fixed acidity`, `volatile acidity`, `citric acid`, `residual sugar`, `chlorides`, `free sulfur dioxide`, `total sulfur dioxide`, `density`, `pH`, `sulphates`, and `alcohol`.

Unlike the previous case studies, ClusterShapley reveals a more heterogeneous set of feature importance for this dataset. However, certain similarities are noticed, such as clusters 6 and 8 and clusters 2 and 5. The most cohesive cluster (4)—more spatially distant from the others—has a few features that contributed to its distinction: `chlorides` and `sulphates`. There are other characteristics. For example, the `pH` index was determinant for characterizing cluster 0 , `density` in cluster 6 , and for cluster 2 , the features contributed equally.

Fig. 54 shows the line segments and Shapley values for class 4 to assess the contribution of chlorides and sulphates. Notice that while other clusters present lower `chloride` values and `sulphates`, cluster 4 presents higher values for these features. Compared with the other Shapley values, `chlorides` and `sulphates` contributed a lot to the cluster separation, i.e., Shapley values' negative contribution. While `chlorides` measures salt in the wine, `sulphates` contributes to sulfur dioxide gas levels and acts as an antimicrobial and antioxidant (MODELING. . . , 2009). Thus, such a cluster describes salty wines, being very distinguishable from the others.

Fig. 55 shows the dot plot of clusters 0 and 8 . Both of the clusters influence higher values of `pH` index. However, the two most influential features of cluster 0 are

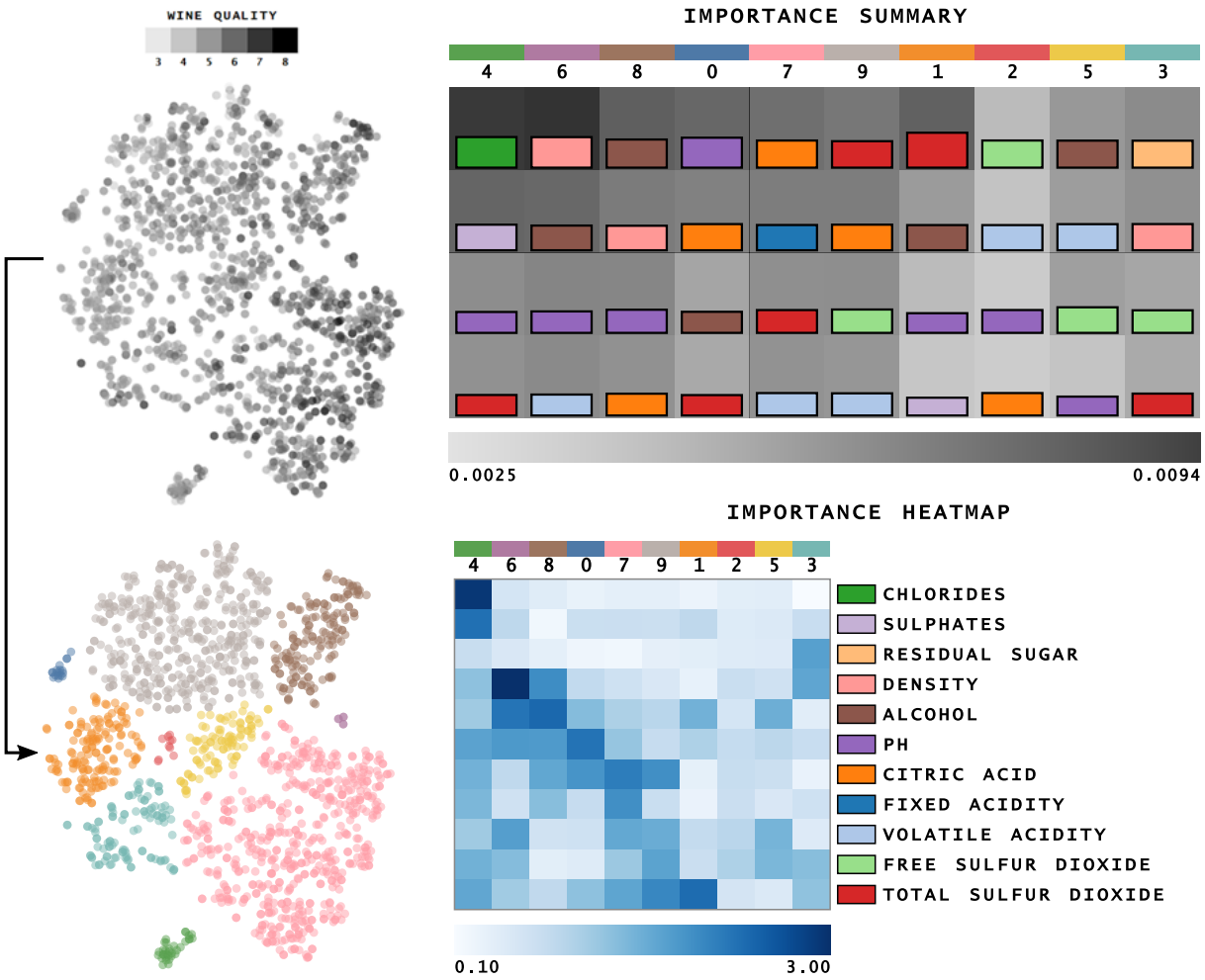


Figure 53 – Dimensionality reduction results for the *Red Wine Quality* dataset colored using intensity to encode quality index and using colors to depict clusters.

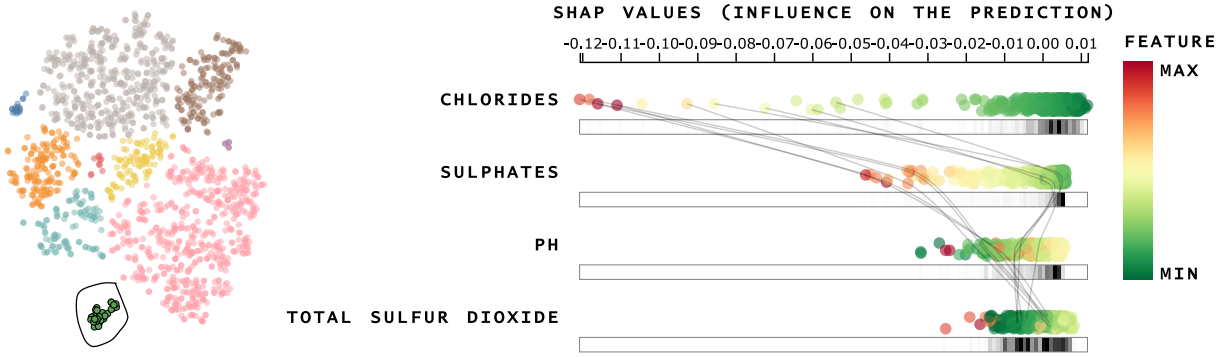
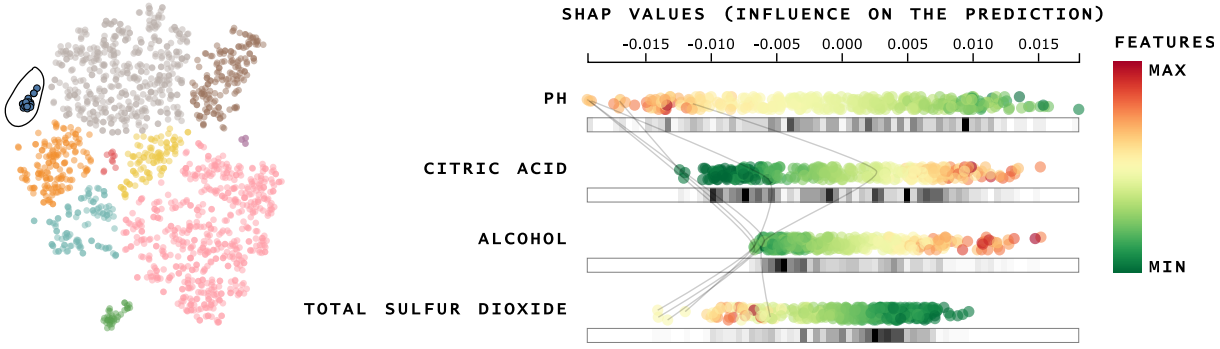
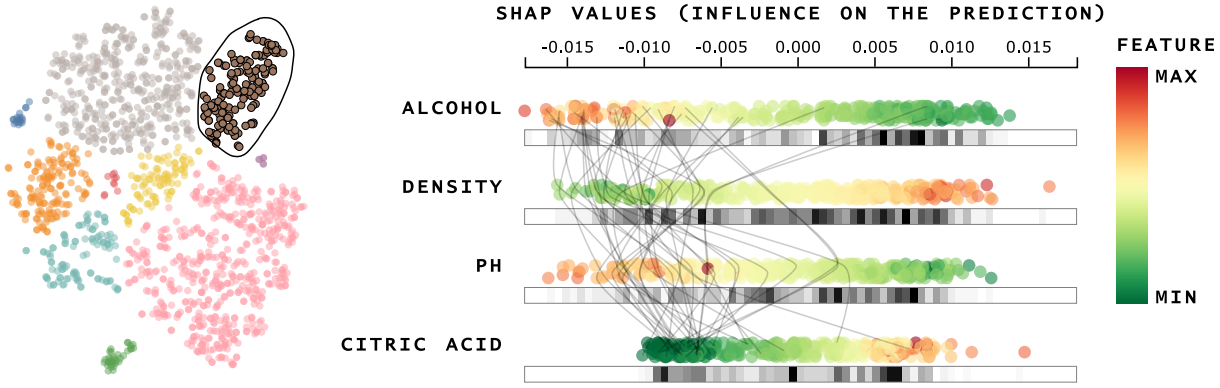


Figure 54 – Cluster 4 is well defined and distinguishable due to the salty flavor (high levels of `chrolides`) of the wines.

only the third and fourth for cluster 8 ■. Moreover, whilethe two groups of instances benefit from higher `pH` values and lower values of `critic ACID`; cluster 8 ■ presents higher values for `alcohol`, configuring among the more alcoholic wines and more tasteful according to the quality index.



(a) Lower levels of `Alcohol` were determinant for the lower quality of the wines in cluster 0.



(b) Higher levels of `Alcohol` determine higher quality for wines in cluster 8.

Figure 55 – Comparison between two clusters projected distant which share two important features. The difference between alcohol levels was determinant for differentiating the quality of the wine in these two clusters.

Recalling the embedding encoding the quality index, cluster 7 ■ represents higher quality wines (higher intensity in Fig. 53). Cluster 7 ■ presents `Fixed Acidity` as the second most important feature. A combination of higher values `Fixed Acidity` and *lower* `Volatile Acidity` values means higher quality for the wines. For instance, volatile acid in *higher* concentrations results in an unpleasant aroma and taste (DAVIS, 2004). Another characteristic that explains the concentration of higher quality is the higher levels of `Citric Acid`, which contributes to the wine's freshness. Besides that, lower values for `Total Sulfur Dioxide` feature also contribute to the higher quality—when in lower values, `Total Sulfur Dioxide` do not add flavor or smell to the wine (MODELING. . . , 2009).

Finally, focusing on cluster 1 ■, Fig. 57 shows that `Total Sulfur Dioxide` (SO2) is a determinant for the concentration of lower quality wines. While lower concentrations of S02 are mostly undetectable, higher concentrations of S02 becomes evident in the nose and taste. The other most important feature contributes to the quality as well. As we

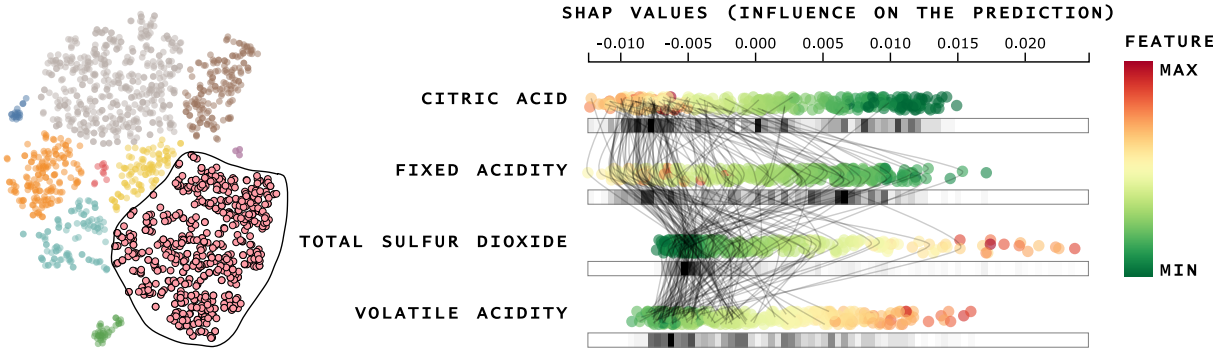


Figure 56 – A combination of higher values `Fixed Acidity` and lower values of `Volatile Acidity` can be unpleasant at higher levels and can explain higher quality for the data instances.

could see for clusters 4 and 8, lower values for `Alcohol` were also responsible for lower quality in cluster 4.

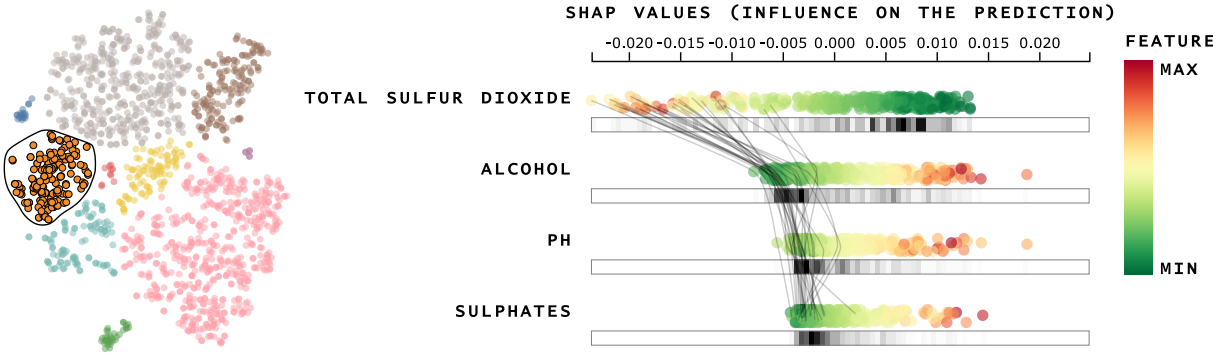


Figure 57 – At higher levels, `Total Sulfur Dioxide` tends to add flavor to the wine and be noticed in the nose. Such an important feature is the meaning of higher concentration of low-quality wines in cluster 1.

4.5.3 COMMUNITIES AND CRIME

In this final case study, we explore the Communities and Crime dataset (REDMOND; BAVEJA, 2002; DUA; GRAFF, 2017). We used the dataset preprocessed by (FUJIWARA; KWON; MA, 2019) to compare the patterns returned by our approach. The dataset contains 2215 instances described by 128 features. Fig. 58 shows the projection result with clusters manually selected together with the overview visualization of feature importance.

Apart from clusters 0 and 3, the most important features are very heterogeneous among the clusters. The first thing to notice is how `PctHousOccup` (percent of occupied houses) is important for characterizing cluster 3. Such cluster is different from the others by having a lower percentage of houses occupied—as shown in the dot plot of Fig. 59—indicating communities in safer areas.

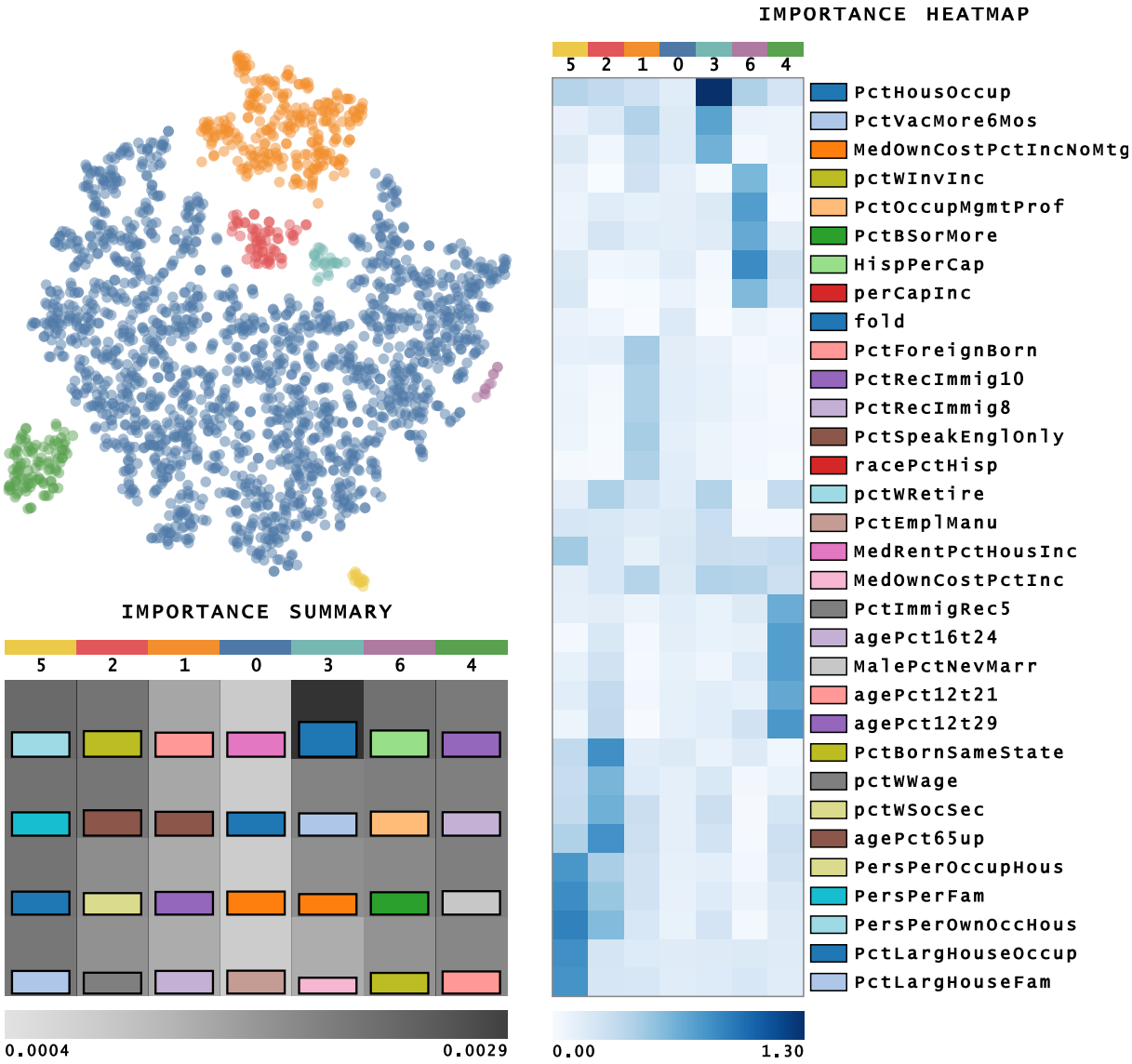


Figure 58 – Feature contributions and the dimensionality reduction result for the *Communities and Crime* dataset.

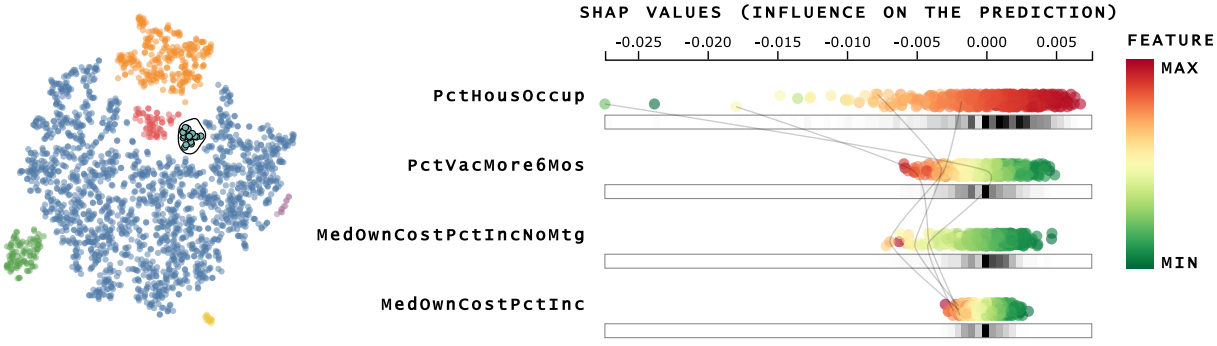
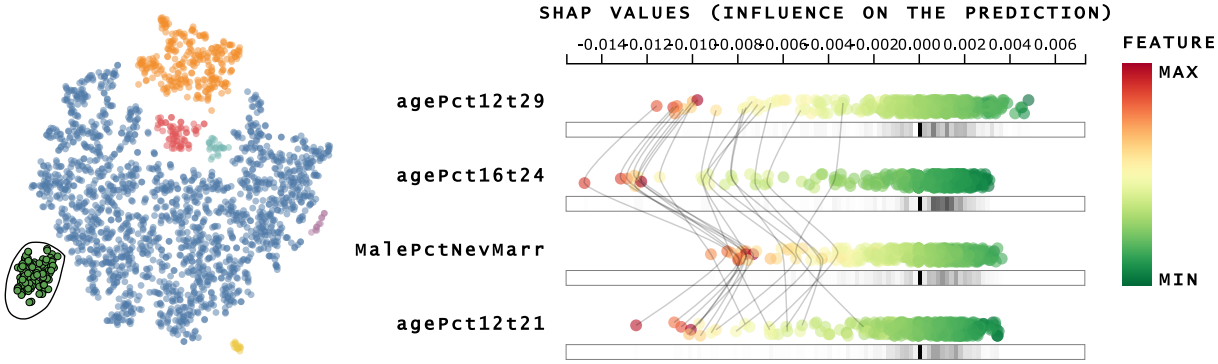
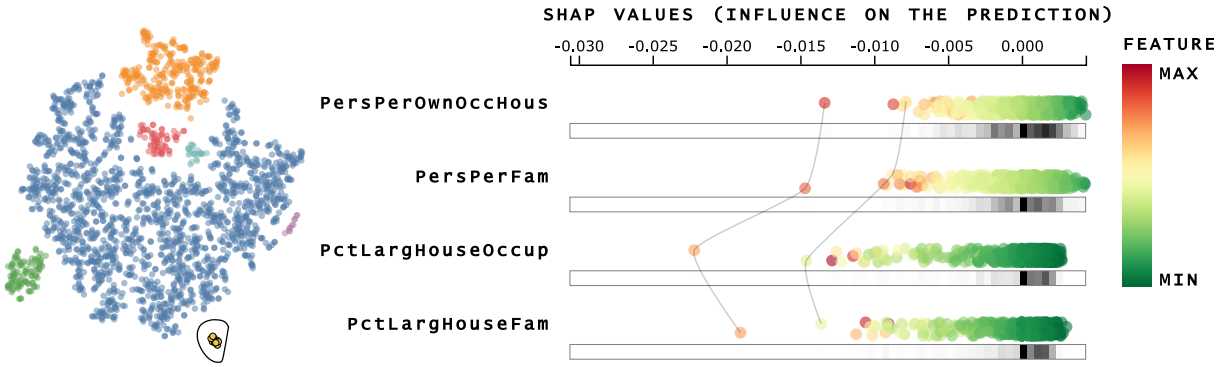


Figure 59 – Such community is very distinguishable from the other for having a lower percentage of houses occupied `PctHousOccup`.

Another interesting pattern is the importance of the features `PctLargHouseFam` (percent of large family households) and `PctLargHouseOccup` (percent of large house occupied) for cluster 5 (see Fig. 60b), as well as the features `AgePct12T29`, `AgePct16T24` (percent of population between 12 and 29, and percent of population between 16 and 24, respectively), and for the feature `MalePctNevMarr` (percent of Males who have never married) for cluster 4 (see Fig. 60a). For cluster 5, these features' higher values contribute to cohesion, representing bigger families and houses. The same pattern happens to cluster 4, the higher percentage of age explains the higher percentage of males that have never married. Since the age range in these features is somewhat low, the percentage of males who have never married tends to be high due to cultural aspects.



(a) A community that has higher percent of males who have never married (`MalePctNevMarr`) due to the high percent of young people (`AgePct12T29`, `AgePct16T24`, and `AgePct12T21`).



(b) A community that has bigger families/houses (`PctLargHouseFam`) and suffer from robbery (`PctLargHouseOccup`).

Figure 60 – Detailed analysis of clusters 4 and 5.

Finally, cluster 1 corresponds to a community of immigrants. The four most important features are `PctForeignBorn` (percent of people born in another country), `PctRecImmig8`, `PctRecImmig10` (percent of the population who have immigrated within the last 8 and 10 years), and `PctSpeakEngOnly` (percent of people who speak only English). While the first three features present high values, the latter shows low values. People who have immigrated are more likely to speak another language. Such analysis exemplifies the semantic power of our approach.

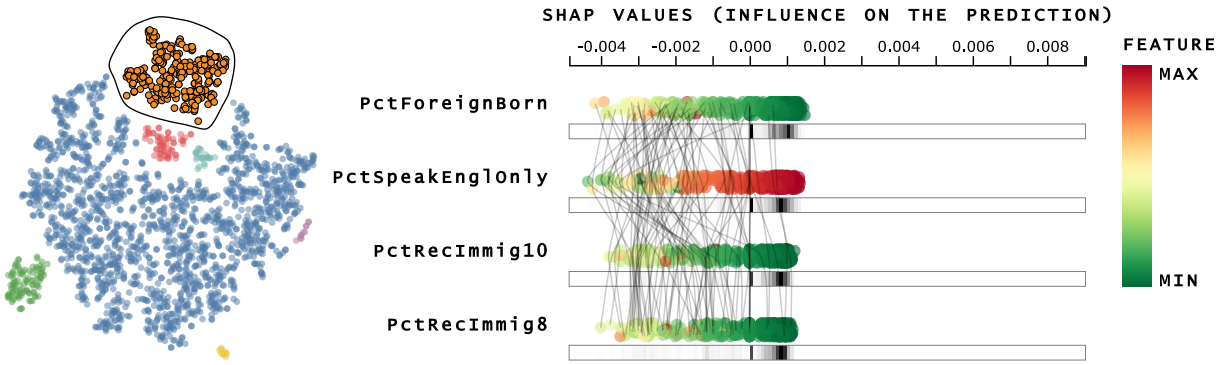


Figure 61 – A community of immigrants due to the high percent of immigrants (`PctForeignBorn`, `PctRecImmig8`, and `PctRecImmig10`) who speak more than one language besides English (`PctSpeakEngOnly`).

4.5.4 AUTOMATIC ANNOTATION WITH CLUSTERING TECHNIQUES

In this section, we aim to analyze ClusterShapley using clustering algorithms. To this end, we choose the CBR-ILP-IR (PAULOVICH et al., 2008) document collection, also aiming to evaluate our technique for thousands of dimensions. The CBR-ILP-IR dataset contains 574 documents representing papers of three fields: Case-based Reasoning (CBR), Inductive Logic Programming (ILP), and Information Retrieval (IR). We followed (ELER et al., 2018) to preprocess the dataset, using Porter Stemming, removing terms with a frequency below one and *tf-idf* transformation. The resulting dataset consists of 574 papers per 18694 terms. Fig. 62 shows the resulting clusters on a UMAP (MCINNES; HEALY; MELVILLE, 2018) projection and the contribution analysis for *Original labels*, *k-Means* with four clusters (*k*-Means 1), *k-Means* with eight clusters (*k*-Means 2), and *Agglomerative clustering* with nine clusters.

The *Original labels* show that this is a straightforward dataset in terms of class separation. The **Importance Heatmap** helps us to realize which cluster represents the main area of the dataset: Case-based Reasoning (CBR) in blue (●) (see important terms `reason` and `case-bas`); Intuitive Logic Programming (ILP) in orange (●) (see important terms `program`); and Information Retrieval (IR) in red (●) (see important term `inform`)—we highlight these terms in the heatmap. The clustering result with four clusters (*k-Means (1)*) can uncover the information that the smaller cluster is positioned distant from the others since it has no unique influence by specific terms, such as `case-bas`, `reason`, `inform`, or `program`. The clustering result with eight clusters (*k-Means (2)*) and its respective **Importance Heatmap** further adds information to the analysis. More terms about Intuitive Logic Program (ILP) appear for the cluster in red (●) (such as `logic` and `program`)—see these terms in the heatmap highlighted in red. Another important aspect of this clustering result is that the `learn`, `inform`, `retrieve`, `reason`, and `case-bas` seem to make cluster ● relate with cluster ●. This characteristic is because these terms have similar (with a more significant difference for the term `learn`) important terms for both

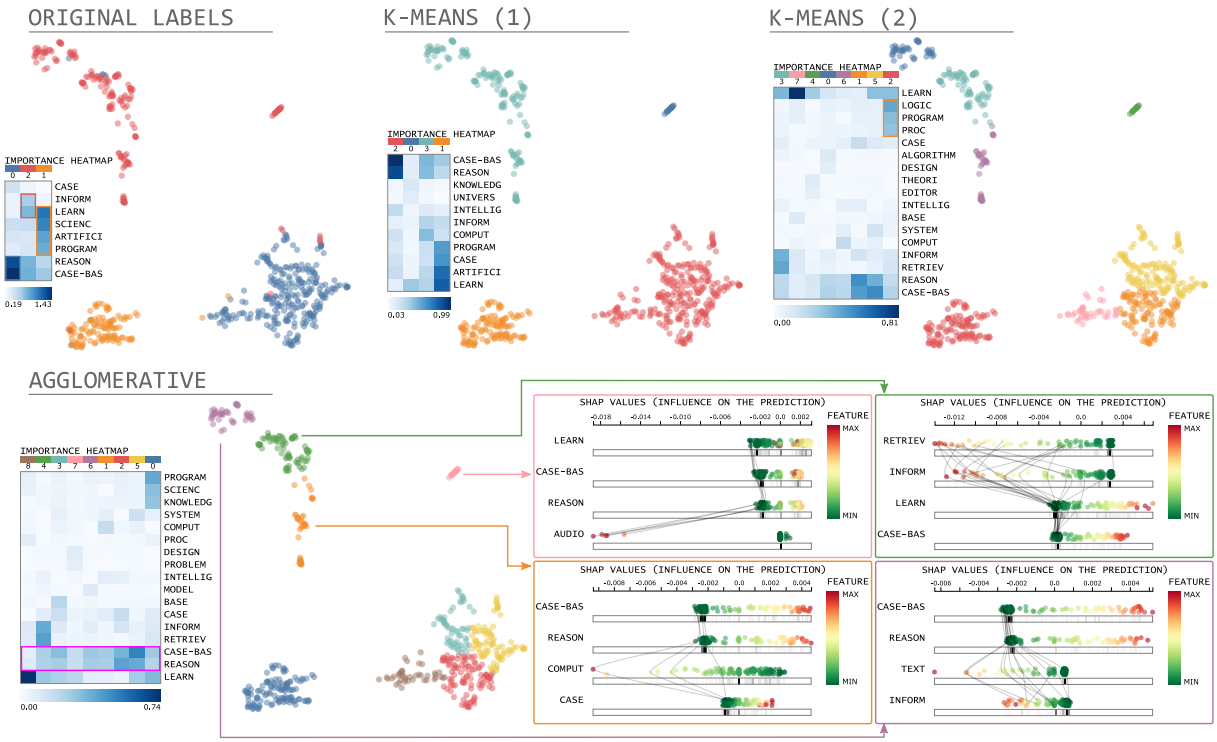


Figure 62 – Using clustering techniques to feed ClusterShapley with different clustering results. The original dataset (`original labels`) has three labels, and we use three clustering configurations: four clusters using *k*-Means (`k-means (1)`), eight clusters using *k*-Means (`k-means (2)`), and nine clusters using Agglomerative clustering (`agglomerative`).

clusters.

The third clustering result (*Agglomerative*) shows other interesting characteristics. Here, although only the clusters positioned on the right-bottom of the projections talk about Case-based Reasoning, the **Importance Heatmap** shows that all clusters contribute from terms of `case-base` and `reason`. While these terms contribute to the cluster formation after dimensionality reduction, they could have positive contribution—meaning that for the clusters not talking about Case-based Reasoning, these terms influence on the cluster spreading. The Shapley values representation explain this characteristic for the clusters positioned on top (●, ●, ●, and ●). Each cluster has terms that the presence in the document contributes to its formation. For example, the documents of cluster ● seem to be influenced by the term `audio`; the documents of cluster ● seem to talk about information retrieval (see terms `retrieve` and `inform`); and so on. However, the dimensionality reduction technique also considered the absence of terms `case-bas` and `reason` as an important characteristic for cluster definition.

4.6 FINAL REMARKS

This research effort's main contribution is to explain dimensionality reduction results using clusters in the projected space. This property is particularly important for non-linear dimensionality reduction techniques. While non-linearity is required for unfolding complex structures in high-dimensional spaces, there is currently no way to reverse the calculations and track the contributions of the features to the projection.

Our methodology establishes a link between feature values and their contributions to cluster outcomes. Analysts can use this correlation to answer questions like, "How will increasing or decreasing values for a given feature affect the classes/clusters in the dataset?" Finally, the coordination mechanism that exists between the scatterplot and the dot plot enables the identification of intra-cluster patterns.

4.6.1 SUPPORTING EXPLORATORY ANALYSIS

In exploratory data analysis, analysts want to confirm a hypothesis about a high-dimensional dataset or even discover previously unknown information, so dimensionality reduction techniques are a good choice. These features make ClusterShapley a valuable tool for interpreting dimensionality reduction techniques and assisting in high-dimensional data analysis. Analysts can understand various aspects of the data by using the explanations generated by ClusterShapley, such as why data samples belong to a cluster, why different clusters present a relationship, or why data samples of different classes appear very dissimilar in the projected space.

ClusterShapley has a wide range of applications, emphasizing the importance of our technique. ClusterShapley could be used by machine learning practitioners to investigate the quality of feature spaces and understand how different classes appear similar and cause confusion in classification tasks. ClusterShapley can also help with dataset annotation because it provides distinct characteristics of different clusters. ClusterShapley would be used by biologists to discover new cell types by analyzing the contribution of genes in each visible cluster in the projected space. Finally, ClusterShapley may be useful in applications that monitor the deep learning model training process. An exploratory method using ClusterShapley in this case consists of an early-stopping strategy when the model's learned features are investigated using a DR technique and ClusterShapley corresponds to the analyst's mental model.

4.6.2 RUN-TIME EXECUTION

Shapley values are difficult to compute, necessitating the use of estimation techniques. While model-specific strategies have a reasonable execution time, general model implementations like KernelSHAP (LUNDBERG; LEE, 2017) (the one we used in this

research) can take too long to produce explanations for larger datasets due to the use of weighted linear regression.

Table 2 shows how the execution time of KernelSHAP for estimating Shapley values varies with dataset size, number of clusters, and number of features. Such a characteristic opens future work for the development of model-specific explanations for explaining dimensionality reduction techniques.

Table 2 – Time for estimating Shapley values.

Dataset	Size	Dim.	Clusters	Time (s)
Iris	150	4	3	0.1698
Vertebral Column	310	6	3	0.2555
Indian Liver	167	10	6	38.22
Red Wine	1599	11	10	315.71
Com. and Crime	2215	128	7	462.93
CBR-ILP-IR	574	18694	3	1098.67
CBR-ILP-IR KM 1	574	18694	4	1259.23
CBR-ILP-IR KM 2	574	18694	8	1646.52
CBR-ILP-IR Aggl.	574	18694	9	1659.87

4.6.3 CLUSTER-ORIENTED ANALYSIS

We explain the contribution of features to the data organization in $\mathbb{R}^2$ imposed by dimensionality reduction techniques using a cluster-oriented analysis. That is, ClusterShapley returns the contribution of each feature to the formation of each cluster in the projected dataset. However, in some cases, it would be beneficial to understand how features contribute to different parts of the projection in comparison to the rest of the dataset. While we do not address this scenario, we intend to develop alternative strategies in the future. For example, we could understand local contributions by defining two parts of the embedding, the area of interest and the remaining part of the embedding.

Another limitation of our work is related to the quality of the clusters when used by clustering algorithms. In Fig. 62, the clusters returned by the k-means and agglomerative clustering defined the dataset labels as the first step before using ClusterShapley. Suppose the clusters returned by the clustering algorithm do not convey the real clusters present in the dataset. In that case, ClusterShapley will generate results in which explanations consider dissimilar data points (in different real clusters) as similar data points and consequently mislead analysis. Nevertheless, the visualizations provided in this work make it easy for analysts to be aware of these issues.

5 HUMAP: HIERARCHICAL UNIFORM MANIFOLD APPROXIMATION AND PROJECTION

Wilson E. Marcílio-Jr, Danilo M. Eler, Fernando V. Paulovich, Rafael M. Martins
HUMAP: Hierarchical Uniform Manifold Approximation and Projection, arXiv preprint arXiv:2106.07718, 2021.

In most knowledge domains, high-dimensional data analysis has become an essential task. However, as high-dimensional data becomes more widely available, new applications emerge. Visualization tools for navigation and analysis are becoming increasingly important in these scenarios, with dimensionality reduction (DR) being one of the most prominent examples.

DR techniques have traditionally operated at a single level of detail, yielding layouts that highlight general patterns that convey the overall distribution of points in the high-dimensional space. These layouts, however, may conceal important data characteristics, such as differences between samples within a cluster, which would provide an overview of the structures and benefit analysis when focusing on important information first. To address this issue, one can always generate a new projection using a cluster of interest or a selection of data instances; however, the resulting layout will not account for the contribution of the remaining points, and context will not be preserved. This limitation can be overcome by employing Hierarchical DR (HDR) techniques, which facilitate the exploration of DR results by employing the visual information-seeking mantra (Shneiderman, 1996): *Overview first, zoom and filter, then details-on-demand*.

The detailed structures seen in traditional representations are summarized in HDR, but groups within clusters are better encoded. Users expand clusters in HDR representations to investigate lower (and more detailed) hierarchy levels, discovering relationships between subclusters that are more visible in the top-level representation. This idea is well illustrated in Fig. 63 for the class in purple, which is also highlighted by a red box. Using HDR approaches, the separation of two dominant structures is clearly visible for top-level representations, which is not true for traditional approaches. Then, one aspect brought by our approach is mental-map preservation for maintaining context throughout hierarchical exploration. This is significant because of the various non-linear transformations that occur to the data after an embedding step, as shown in Fig. 63(B) for HSNE for the second and third embeddings.

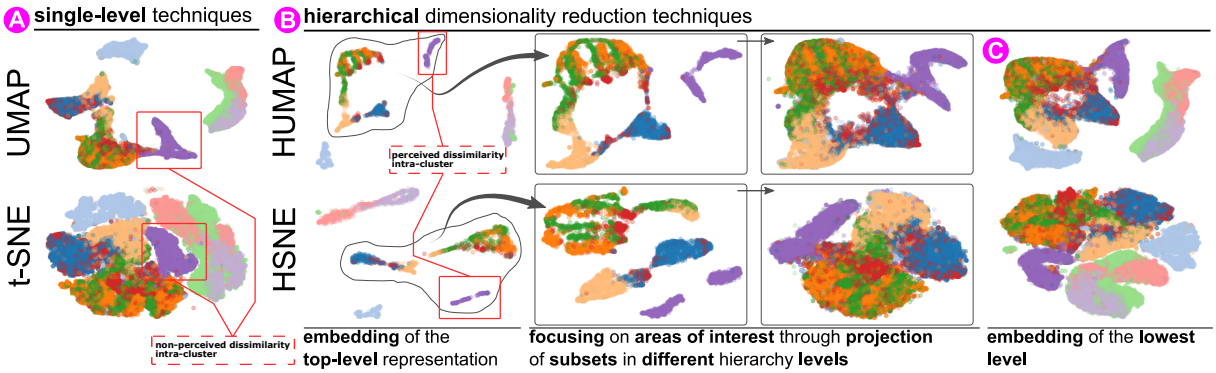


Figure 63 – Comparison of traditional dimensionality reduction (DR) techniques (A) vs. hierarchical DR techniques (B). Traditional techniques operate in a single level of detail and focus on delivering a big picture that depicts the dataset as a whole, ignoring fine-grained details of intra-cluster distributions and making it difficult to proceed with more detailed investigations. On the other hand, hierarchical DR (HDR) approaches summarize important intra-cluster information already on the top-level representations of the hierarchy imposed on the input data without affecting the layout's overall organization.

Although HDR research has been active in recent years (PEZZOTTI et al., 2016; HÖLLT et al., 2019; KUCHROO, 2020), current techniques have a few limitations. HiPP (PAULOVICH; MINGHIM, 2008) creates an overview of the dataset and guides users through further analysis by using nodes (encoded as circles with varying areas). It is based on Least Square Projection (LSP) (PAULOVICH et al., 2008), which does not scale well for datasets with thousands of points. Using a GPU implementation, HSNE (PEZZOTTI et al., 2016) can successfully depict manifolds and deal with thousands of data points/dimensions. However, as the analyst descends the hierarchy, HSNE fails to preserve the user's mental map. That is, the embedding generated for a hierarchical level does not take into account the preceding level, and the results usually become increasingly dissimilar to the previous level's embeddings as the hierarchy is expanded—Fig. 63 (B, bottom) depicts such an HSNE embedding property. Recently, a hierarchical technique known as Multiscale PHATE (KUCHROO, 2020) was proposed for the analysis of mass cytometry data. Although the technique is capable of handling massive datasets with continuous phenomena, it does not appear to work well with much higher dimensions or when the dataset does not have such a continuous characteristic. Furthermore, when drilling down into the hierarchy, Multiscale PHATE is very rigid in interaction because users can only choose from the clusters created during hierarchy definition.

These techniques cannot fully address two important HDR design considerations: maintaining a good trade-off between preserving global and local structures in the dataset's low-dimensional representation and preserving the mental map across hierarchy levels. From the standpoint of local and global preservation, some hierarchical approaches (e.g., HiPP) cannot fully uncover complex manifold structures or struggle with this task depend-

ing on the dataset type—for example, Multiscale PHATE works well for datasets with continuous/evolution characteristics but struggles when the goal is to separate clusters. Finally, there is a current requirement for techniques that uncover complex structures while preserving global and local relationships among data points. The preservation of global and local relationships, for example, is critical when analyzing deep learning models (RAUBER et al., 2017; MARCíLIO-JR et al., 2020) and gene expression data (LäHNEMANN; KöSTER; SZCZUREK, 2020). These requirements are presented by HSNE for higher hierarchy levels. However, it does not preserve the mental map as the hierarchy is expanded, making it difficult for users to switch between overview and details.

To address these issues, this chapter presents Hierarchical Uniform Manifold Approximation and Projection (HUMAP)[1]. Based on UMAP, our HDR technique creates a hierarchy on the dataset by using a two-step similarity definition that encodes global and local similarity information between data points. The technique maintains the mental map as the user drills down the hierarchy by using projected data points from higher levels to influence the low-dimensional representation of lower hierarchy levels while successfully uncovering complex structures present in the dataset's many granularities. Through visual and quantitative evaluation, we provide experimental evidence that HUMAP is superior to the techniques discussed above. We also present two qualitative case studies that investigate the features learned in a deep learning model and the hierarchical explanation of DR results.

As for the previous chapters, we first present the related works in Section 5.1. Then, we describe our technique in Section 5.2. Section 5.3 three cases studies using HUMAP, with three different data types. The quantitative evaluation is provided in Section 5.4. We finalize with final remarks in Section 5.5.

5.1 RELATED WORK

By reducing the number of dimensions while maintaining structural and similarity relations in the low-dimensional representation, dimension reduction techniques aid in the analysis of high-dimensional datasets(Nonato; Aupetit, 2019). When using a DR technique, users typically look for data patterns such as clusters, shapes, and outliers. Understanding the data and available DR techniques is critical for a successful data exploration process. Linear DR techniques (JOLLIFFE, 1986; PAULOVICH et al., 2008; JOIA et al., 2011), for example, are fast approaches that are well known for uncovering variability and other simple structures. Non-linear DR techniques(MAATEN; HINTON, 2008; PEZZOTTI et al., 2016; MCINNES; HEALY; MELVILLE, 2018; MOON; DIJK; WANG, 2019), on the other hand, can reveal more complex structures in high-dimensional space while being

[1] https://github.com/wilsonjr/humap

generally more complex and time-consuming.

The scatterplot metaphor is commonly used to represent DR results in various knowledge domains. Despite its popularity for exploratory data analysis (MICALLEF et al., 2017), the overlap among markers makes scatterplots less effective (SEDLMAIR; MUNZNER; TORY, 2013; SARIKAYA; GLEICHER; SZAFIR, 2018). When exploring large datasets as a whole, traditional DR techniques may hide important details within and between clusters—some datasets, for example, have intrinsically multilevel structures (LäHNEMANN; KöSTER; SZCZUREK, 2020) that necessitate more flexibility from a DR technique. In this sense, strategies that make exploration easier and guide users through the exploratory process aid in knowledge discovery. HDR techniques, for example, provide exploration mechanisms based on the *overview first & details-on-demand* (Shneiderman, 1996) mantra, focusing on important information based on user demand.

There are numerous traditional DR approaches (refer to (Nonato; Aupetit, 2019; ESPADOTO et al., 2019) for useful surveys), but only a few hierarchical DR (HDR) techniques (Nonato; Aupetit, 2019; HÖLLT et al., 2019). These HDR techniques have the advantage of first establishing a hierarchical structure and then allowing for multilevel exploration. The difficulty is conveying different levels while maintaining context and neighborhood structures (Nonato; Aupetit, 2019). As a result, current research focuses on defining the hierarchical structure while employing the projection engine of previous traditional DR approaches. MDSteer (WILLIAMS; MUNZNER, 2004) solves the MDS computational complexity problem by progressively computing a multidimensional scaling layout based on the user's demand. Glimmer (INGRAM; MUNZNER; OLANO, 2009) also addresses MDS complexity by performing projection using a multilevel scheme on the GPU. To describe it simply, the authors construct a hierarchy by interpolating high hierarchy levels. Nonetheless, MDS-based approaches cannot adequately represent complex structures (e.g., manifolds) found in most practical and real-world datasets (e.g., deep learning features, image collections, or biological data). The lack of strategies for conveying non-linear structures also hinders hierarchical PCA variants (WESTERHUIS; KOURTI; MACGREGOR, 1998; JANNé et al., 2001; AGARWAL et al., 2007).

HiPP (PAULOVICH; MINGHIM, 2008) organizes data using hierarchical clustering and landmarks. The coarser level landmarks represent and influence the position of data points in the finer levels. HiPP uses LSP landmarks for context preservation and conveying hierarchy levels, and it employs a force algorithm to resolve overlaps when heuristically positioning points. One of the most robust approaches is HSNE (PEZZOTTI et al., 2016). HSNE constructs a hierarchy by using random walks on a transition matrix to retain global and local relationships at high hierarchy levels. HSNE's embeddings, on the other hand, lose the structural relationship presented at higher levels during hierarchical exploration, hampered the analysis when mental map preservation is critical. Finally, in

order to provide interactive analysis in a reasonable amount of time, HSNE requires a GPU. Multiscale PHATE (KUCHROO, 2020) (M. PHATE) is a recent technique that uses diffusion condensation (BRUGNONE et al., 2019) to compute a manifold-intrinsic diffusion space on the input data and condenses data points towards centroids to generate groups with multiple granularities. M. PHATE can handle massive datasets, but only in a few dimensions. Furthermore, it presents difficulties to map structures when the input dataset does not contain continuous phenomena, which is a problem with its embedding engine (PHATE) (MOON, 2018). These methods share one feature: the hierarchy levels are made up of representative samples or landmarks. Surprisingly, any one-level-of-detail dimensionality reduction technique that uses landmarks for data point projection can have a hierarchical version (Nonato; Aupetit, 2019).

Finally, during hierarchical exploration, users essentially deal with embeddings of increasing size data subsets. To ensure that geometrical transformations do not deceive users during exploratory data analysis, the mental map between consecutive projections must be preserved. The preservation of the mental map in DR layouts has been receiving attention when exploring deep learning features (RAUBER; FALCÃO; TELEA, 2016; CANTAREIRA; PAULOVICH, 2020), which is usually called *Projection Alignment*. Dynamic t-SNE (RAUBER; FALCÃO; TELEA, 2016) generates projections of a time-varying dataset in order to reduce temporal variability that is unimportant to the final layout. This is accomplished by including a term in the cost equation of the t-SNE, which controls the trade-off between alignment and standard t-SNE optimization. In feature fusion tasks, VFF (HILASACA; PAULOVICH, 2020) also employs a trade-off parameter to control projection alignment. VFF, on the other hand, is not appropriate for preserving neighborhoods or local structures (CANTAREIRA; PAULOVICH, 2020). Finally, Cantareira and Paulovich (CANTAREIRA; PAULOVICH, 2020) proposed a generic model for projection alignment that incorporates the original cost function of the DR technique as well as an alignment penalty term. In contrast to these approaches, our strategy for preserving the mental map entails controlling the movement of data samples already projected at high hierarchy levels and influencing the placement of new data samples.

We present HUMAP to address the problems of previous techniques. Our approach is very flexible for preserving both local and global relations and provides a tunable parameter to preserve the mental map among hierarchy levels and during drill-down operations. These characteristics are apparent in Fig. 63. HUMAP maintains the structures of selected subsets, and the highest hierarchy level (top-level) resembles the lowest hierarchy level (B). Compared to single-level techniques (A), it summarizes the information while providing details about the relationships within and among clusters.

5.2 HIERARCHICAL DIMENSIONALITY REDUCTION

HUMAP is composed of two major components: hierarchy construction and projection (Fig. 64). In the former, we use a similarity measure among landmarks to impose a hierarchical structure on the high-dimensional dataset. In the latter, we embed the hierarchy levels based on the user's demand for more detailed information. All of the steps (A-G) of these components are depicted in Fig. 64.

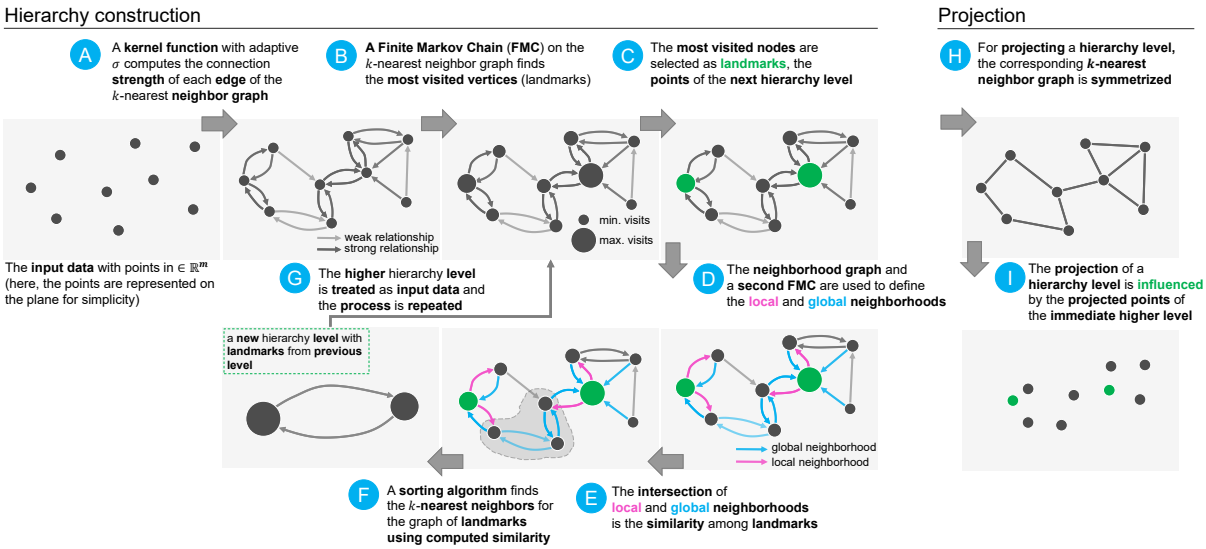


Figure 64 – Hierarchy construction and projection with HUMAP. The hierarchy is constructed from bottom to top. First, a k-nearest neighbor graph approximates the connection strength among data points in the high-dimensional space (A). Then, the graph structure and the connections' strengths are used to find the most visited nodes (landmarks) after several random walk steps (B), which correspond to points on superior levels (C). The representation neighborhood—local and global relationships (D)—is used for computing the similarity among landmarks (F). For projecting hierarchy levels (or subsets of them), HUMAP applies—as in UMAP—a symmetrization to the nearest neighbor graph (H) and use data points from higher levels to preserve the mental map (I).

The first step in building a hierarchy from bottom to top is to use a kernel function to determine the connection strengths (local affinities) of a k-nearest neighbor graph of data points in the high-dimensional space $\mathbb{R}^m$ (step A). Then, as in previous work (PEZZOTTI et al., 2016), we use the Finite Markov Chain (FMC) to find the most visited nodes (step B), which are the superior level landmarks (step C). The FMC procedure is also used to build a neighborhood structure for high hierarchy levels (level > 1) and to encode local and global neighborhood information for each landmark (steps (D) and (E)). Finally, using the computed similarity to define a new hierarchy level, a new neighborhood graph is created (step F) (step G).

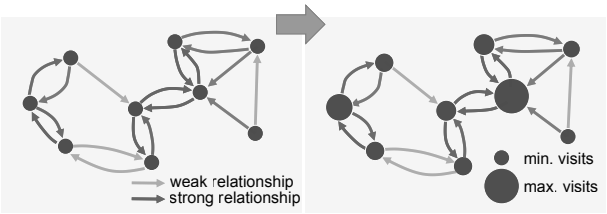
The neighborhood graph is first symmetrized (step H) before projecting hierarchy

levels, so that the strength of each edge can be used to find a suitable position in the low-dimensional representation. Except for the top hierarchy level, the low-level projection is influenced by the low-dimensional positions (I) of data points in higher levels for mental map preservation, as discussed in Section 5.2.3.2.

We use a few UMAP (MCINNES; HEALY; MELVILLE, 2018) strategies to build our approach, such as the kernel function to compute the connection strengths among data samples, the embedding strategy, and ideas from HSNE (PEZZOTTI et al., 2016) for landmark selection. The components for hierarchical dimensionality reduction with HUMAP are discussed in the following section.

5.2.1 LANDMARK SELECTION

To capture manifolds in high-dimensional spaces, DR techniques typically rely on k-nearest neighbor graphs. Such techniques approximate global distances through the sum of local relationships in this process (MOON; DIJK; WANG, 2019), defining the distance between two arbitrary points in high-dimensional space as the path formed by the vertices in the k-nearest neighbor graph. Finding a k-nearest neighbor graph is thus the first step in many DR techniques that can reveal manifolds in data (MAATEN; HINTON, 2008; MCINNES; HEALY; MELVILLE, 2018; MOON; DIJK; WANG, 2019).



After finding the k-nearest neighbor graph, a kernel function defines the strength (or probability) of each edge in the graph. For two data points $x_i, x_j \in X$, we used UMAP's (MCINNES; HEALY; MELVILLE, 2018) kernel function in the high-dimensional space:

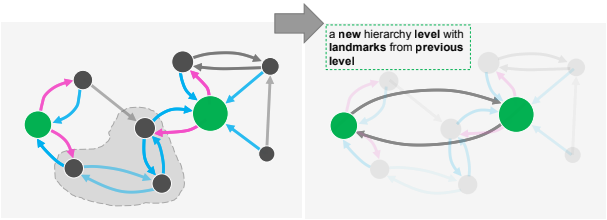
$$p_{i|j} = e^{-\frac{d(x_i,x_j)-\rho_i}{\sigma_i}} \tag{5.1}$$

where $d(x_i, x_j)$ is the euclidean distance between the data points x_i and x_j, ρ_i is the euclidean distance of x_i to its closest neighbor, and σ_i is a kernel value that depends on the number of neighbors k, according to the following equation (MCINNES; HEALY; MELVILLE, 2018),

$$k = 2^{\sum_i p_{i|j}}. \tag{5.2}$$

Notice that σ_i is unknown. We can, however, use Equation 5.2 to compute it using a binary search by fixing a value for k. That is, Equation 5.1 is plugged into Equation 5.2 for each data point x_i (matrix row), and a binary search is used to find the best σ_i value given the number of nearest neighbors (k). As a result, the σ_i value that produces the

closest k' to the actual k is chosen. This procedure ensures that each data point and its neighborhood have a unique kernel value (σ_i)—for a fixed k, higher σ_i values encode dense neighborhoods, while lower σ_i values encode sparse neighborhoods. In other words, σ_i encodes how close the data points in the neighborhood of x_i are. Finally, the parameter ρ_i provides a locally adaptive kernel for each data point and ensures that high-dimensional data is well-represented topologically.



Determining $p_{i|j}$ for each pair of data points produces the connection strength between each neighbor based on the neighborhood density (please see Equation 5.1). We choose good representative data points (or landmarks) based on the strengths of these connections to summarize the dataset on higher hierarchy levels. Finite Markov Chain (FMC) are employed for this task, as described by Pezzotti et al (PEZZOTTI et al., 2016). The path from an initial state to the final state after μ steps (or hops) on the neighborhood graph is defined as a random walk with length μ. As a result, the transition probabilities for the FMC from $p_{i|j}$ are computed using the following equation:

$$T_{i,j} = \frac{p_{i|j}}{\sum_k p_{i|k}} \ (j,k \in NH(i)), \tag{5.3}$$

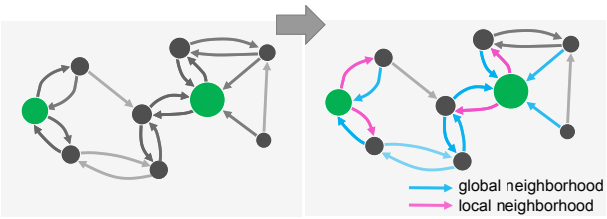
where data points x_j and x_k are in the neighborhood of x_i ($NH(i)$). The landmarks are then sampled from $T_{i,j}$ using a straightforward Markov Chain Monte Carlo technique. We begin n random walks of length μ for each data point i—empirically, $n = \mu = 10$ produce embeddings that faithfully represent the underlying data structure. The steps from (A) to (C) in Fig. 64 represent the entire process, from kernel smoothing to landmark selection.

Unlike previous approaches (PEZZOTTI et al., 2016; KUCHROO, 2020), users have control over the number of data points in each hierarchy level, making it easier to experiment with datasets with varying characteristics (e.g., cluster density)—although threshold-based approaches can be used (PEZZOTTI et al., 2016). The number of data points in a hierarchy level is set by users by specifying the number of landmarks on the level i ($|H^i|$). We store the random walk endpoints and choose the $|H^i|$ most visited data points as landmarks during the Monte Carlo execution.

5.2.2 DISSIMILARITY DEFINITION

The hierarchical level H^i encodes information from the level below (H^{i-1}) when building the hierarchy. This ensures that the top-level information accurately represents the information in the entire dataset. As a result, we transfer the manifold and structural relationship from bottom to top for each subsequent hierarchy level. The similarity between

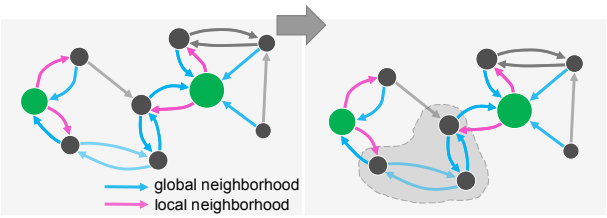
two data points x_i and x_j is computed for the first hierarchical level using Equation 5.1 based on euclidean distance ($d(x_i, x_j)$), as shown in Fig. 64 (A). The similarities among landmarks in H^i follow the data organization of H^{i-1} at higher levels. Such similarities are computed using the k-nearest neighbor graph of H^{i-1}.



The similarity of two landmarks, l_u^i and l_v^i on level i, is computed using the intersection of two components: global and local neighborhoods. Fig. 64 (D-F) depicts these two components for two landmarks colored in green. The local neighborhood (in pink) is composed of the k-nearest neighbors as usual, whereas the global neighborhood (computed only for $H^i, i \leq 1$) is composed of random walks (in blue). Two landmarks that share more local neighbors are very close in high-dimensional space because they are connected by a single data point. In Fig. 64, the green landmarks only have a global relationship, which is highlighted in gray between steps (E) and (F).

Similarly to the HSNE (PEZZOTTI et al., 2016) approach, we compute the global neighborhood using random walks (the blue relationship in Fig. 64). In this case, we begin ω random walks of length υ from each non-landmark (x_k^{i-1}) on the k-nearest neighbor graph of level H^{i-1}. When a random walk reaches at a landmark (l_u^i), we add the non-landmark (x_k^{i-1}) to the "representation neighborhood" of the landmark l_u^i. It should be noted that such a representation neighborhood (RNH) is only created for similarity computation.

The local neighborhood component adjusts our approach in order to preserve more of local structures—illustrated in Fig. 64 by the pink relationship. To that end, we augment the representation neighborhoods of each landmark l_u^i with the $\beta \times |NH(l_u^i)|$ first nearest neighbors of l_u^i, where β is a threshold between 0 and 1. Thus, the greater the value of β, the more important the k-nearest neighbors are, and the similarity measure will encode more local relationships, allowing HUMAP to support parameter tuning to improve neighborhood preservation. The local neighborhood of a landmark is in the set of its nearest neighbors, and it can be increased by adjusting the parameter β in $[0, 1]$, which adds $\beta|NH(i)|$ neighbors to the local neighborhood (by default, β is set to 0). We tested HUMAP with various β values and discovered a clear relationship between local neighborhoods and neighborhood preservation.



Finding these two components results in the representation neighborhood (RNH) for each landmark l_u^i on level H^i. Such a neighborhood consists of a sparse matrix RNH of dimensions $|H^i| \times |H^{i-1}|$ where $RNH_{u,v}$ is 1 if $x_v^{i-1} \in \text{RNH}(l_u^i)$ or 0, otherwise. The similarity among landmarks is

computed based on the intersection of representation neighborhoods, that is, the shaded area of Fig. 64 (E-F). We compute the intersection with the RNH matrices as follows. The representation neighborhood (RNH) for each landmark l_u^i on level H^i is obtained by locating these two components. A neighborhood of this type consists of a sparse matrix RNH with dimensions $|H^i| \times |H^{i-1}|$, where $RNH_{u,v}$ is 1 if $x_v^{i-1} \in \text{RNH}(l_u^i)$ or 0 otherwise. The intersection of representation neighborhoods, that is, the shaded area of Fig. 64, is used to compute the similarity among landmarks (E-F). The intersection with the RNH matrices is computed as follows:

$$\begin{aligned} d_{RNH}(l_u^i, l_v^i) &= \frac{(RNH_u)^T RNH_v}{M}, \\ M &= max_m(\sum_{n=1}^{|L^{i-1}|} RNH_{m,n}) \end{aligned} \tag{5.4}$$

where M is a normalization factor that ensures 0 and 1 are similar—notice that because RNH is sparse, computing the above matrix multiplication is fast in practice. It is worth noting that the intersection of neighborhoods is also used in HSNE to calculate the similarity of two landmarks. Unlike HSNE, however, we do not directly use such a similarity measure to compute the embeddings. HUMAP, on the other hand, extends the neighborhood to encode global information and uses UMAP's kernel to keep its properties consistent across the hierarchy.

The above operation produces a similarity measure, which means that a value of 1 corresponds to landmarks in the same representation neighborhood—as illustrated by operation (F) in Fig. 64. As a result, we subtract the similarity value from 1 to get a dissimilarity measure. To take full advantage of UMAP's kernel function (Equation 5.1), it is applied to each hierarchical level. Finally, the process of creating the neighborhood with Markov Chain also eliminates the need for another kNN step.

5.2.3 GENERATING THE EMBEDDING

Equation 5.1 uses the dissimilarity values $(1 - D_{RNH})$, where D_{RNH} is a matrix with d_{RNH} as rows, to generate the matrix p_{ij} for each hierarchical level. Then, as with the UMAP technique, HUMAP employs the following matrix symmetrization to generate an undirected weighted graph:

$$p_{ij} = p_{i|j} + p_{p|j} - p_{i|j}p_{j|i}. \tag{5.5}$$

The matrix p_{ij} is used to generate an embedding that converges to low-dimensional positions. While McInnes et al. (MCINNES; HEALY; MELVILLE, 2018) provide a detailed description of the UMAP algorithm, it is important to note that Spectral Embedding (NG;

JORDAN; WEISS, 2001) is used to create an initial low-dimensional representation of the dataset. Then, through Stochastic Gradient Descent (RUDER, 2016) optimization, p_{ij} is used to position the data points using a force-directed graph layout whose convergence is performed by decreasing attractive and repulsive forces.

5.2.3.1 Subset projection

Each landmark represents a set of data points on the level below during the definition of hierarchical levels. It is critical to define the functionality of interaction with hierarchical levels and request more detail (more data points) of a data subset in order to keep track of such representation.

We assign each non-landmark data point on level H^{i-1} to a landmark on level H^i in two steps. The first step is to iterate through the landmarks and assign them to neighbors who will be influenced by them. A landmark will eventually attempt to "represent" an already represented data point. As a result, we assign the data point in the neighborhood of the nearest landmark. This procedure assigns the majority of data points to a landmark; however, it is necessary to look for cases where data points are not in the neighborhood of any landmark. In this case, we iterate over each data point on H^{i-1} to see if it is not associated with any landmark, in case we look for one in its neighborhood. This procedure is described in detail in Algorithm 3. We explore the neighborhood of each non-landmark data point (Lines 2 and 3). (Line 5). If a neighbor v is a landmark (Line 6), we assign v to the current data point (Lines 7 and 8). If v is not a landmark but is associated to one (Line 9), the landmark that represents v will also represent the current data point (Lines 10 and 11). Finally, if neither of these cases applies to all neighborhoods, we conduct a depth-first search in the k-nearest neighbor graph and associate the first landmark found with the current data point.

Algorithm 3 Associate Landmarks.

```
1:  procedure ASSOCIATE_LANDMARK(X)
2:      for u ∈ X do
3:          if not landmark(u) then
4:              found ← false
5:              for v ∈ NH(u) do
6:                  if landmark(v) then
7:                      found ← true
8:                      associate[u] ← v
9:                  else if associate[v] ≠ NIL then
10:                     found ← true
11:                     associate[u] ← associate[v]
12:             if not found then
13:                 landmark ← depth-first search on graph X starting in u
14:                 associate[u] ← landmark
```

Users can choose from a list of indices on level H^i to embed the associated data points on H^{i-1} using the data points assigned to each landmark. Assuming the list of indices in H^i is denoted by I, we only need to construct the matrix $p_{mn}^{i-1} = \{p_{mn} \mid A[m] \in I \wedge A[n] \in I\}$, where $A[u]$ returns the landmark that represents the data point x_u.

5.2.3.2 Mental map preservation

The idea of mental map preservation consists of keeping the same overall layout shape and positioning of the data points when changes are made to the initial visualization through updates to the current dataset at time t (ARCHAMBAULT; PURCHASE; PINAUD, 2011; FUJIWARA; KWON; MA, 2019). These updates can be related to additional data points or changes in the features of data points—in the case of hierarchical dimensionality reduction, the changes include additional points from lower hierarchy levels.

The subsets of data chosen for fine-grained exploration on level l-1 must resemble the organization of the level l during hierarchical exploration, especially for unlabeled datasets. While the majority of work on mental map preservation has focused on dynamic graph drawing(XU; KLIGER; HERO, 2012; LEYDESDORFF; SCHANK, 2008), Rauber et al. (RAUBER; FALCãO; TELEA, 2016) and Fujiwara et al. (FUJIWARA et al., 2020) present strategies for dealing with mental map preservation in the context of dimensionality reduction. In both cases, the goal is to reduce the analysis burden during exploration and make changes to the embedding layout more smoothly. Fig. 65 depicts consecutive hierarchical projections using HUMAP, regardless of whether the mental map—the visual similarity between consecutive layouts—is used. It should be noted that without mental map preservation, keeping track of the data transformations that result in the final embedding of level 0 is difficult.

HUMAP guides the positioning of level l-1 by using the coordinates of higher—and already projected—data points on level l. The coordinates for the projected data points in l are used as starting points when initializing the low-dimensional representation before Stochastic Gradient Descent (SGD) optimization, along with the remaining data points initialized with Spectral Embedding—these coordinates only move a fraction during SGD optimization. Thus, we provide this fraction hyperparameter *theta* to be tuned according to one's needs, although $theta = 0.01$ generates embeddings that best preserve the mental map throughout the hierarchy expansion.

5.3 CASE STUDIES

We present two case studies in this section. In the first section, we use a dimensionality reduction explanation approach to demonstrate how HUMAP benefits the analysis in terms of the various facets that a hierarchical dimensionality reduction can provide. Then,

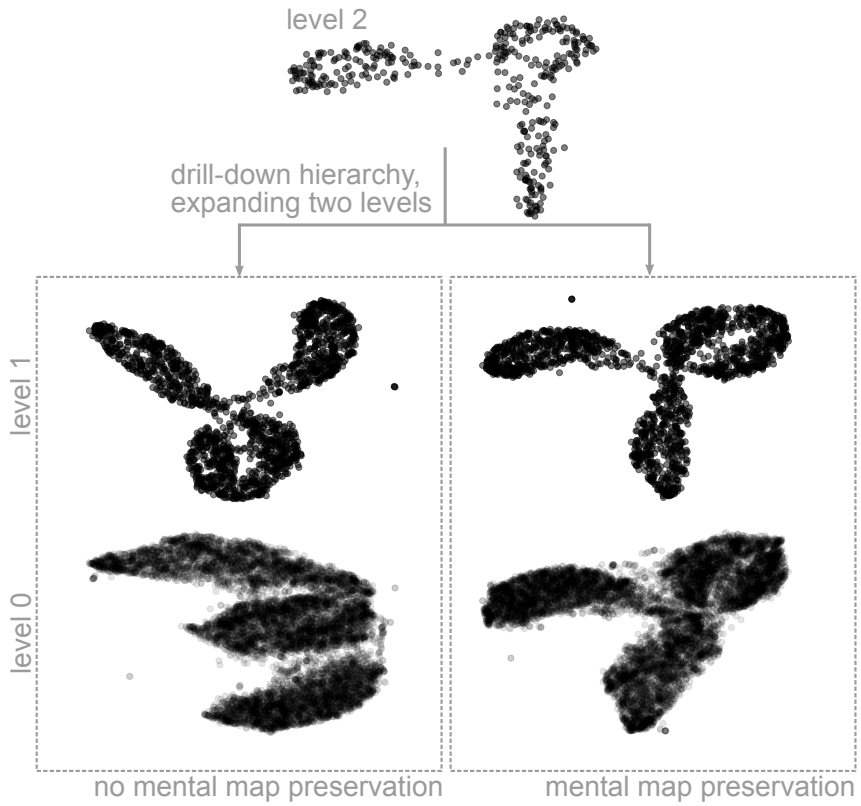


Figure 65 – Mental map preservation across hierarchical levels.

we evaluate HUMAP's performance in terms of mental map preservation when dealing with large datasets. We set the number of random walks $\omega = 20$ and the length of each random walk $v = 30$ for both case studies.

5.3.1 SUPPORTING HIERARCHICAL EXPLANATION OF DIMENSIONALITY REDUCTION RESULTS

The explanation of dimensionality reduction algorithms has been receiving a lot of attention in the recent years (FUJIWARA; KWON; MA, 2019; MARCíLIO-JR; ELER, 2021; CONTRASTIVE. . . , 2021; SOHNS et al., 2021; TIAN et al., 2021). The ability to explain DR results enhances exploratory analysis and can generate previously unknown facts (MARCíLIO-JR; ELER, 2021). In this use case, we demonstrate how HUMAP can be beneficial and offer more detailed explanations when applied with a DR explanation approach called ClusterShapley (MARCíLIO-JR; ELER, 2021).

ClusterShapley leverages a model-agnostic approach, called SHAP (LUNDBERG; LEE, 2017), to explain DR results. Despite the runtime limitations of SHAP, the drill-down operation provided by HUMAP provides faster results by helping users to focus on important information, cut corners, and generate explanations for only a subset of samples. Throughout the hierarchical exploration using HUMAP and explanations supported by ClusterShapley, we focus on understanding the differences among lower-level clusters by

first investigating the dominant structures embedded by HUMAP. So, the procedure for cluster selection consists of defining clusters on level l and drilling down by selecting points from at least two clusters.

Fig. 66 illustrates the clusters—manually annotated—and their explanations, which represent the contribution of each feature to cluster formation. In this case, negative SHAP values (x-axis) encode contributions to cluster cohesion, while positive SHAP values encode contributions for spreading the cluster. In the visual representation, the authors encode feature values using a green-to-red color scale and the contributions on positions on the x-axis. When investigating the explanations for a cluster, we can highlight its data points on the visualization by drawing poly-lines connecting their feature contributions—as seen for the green cluster.

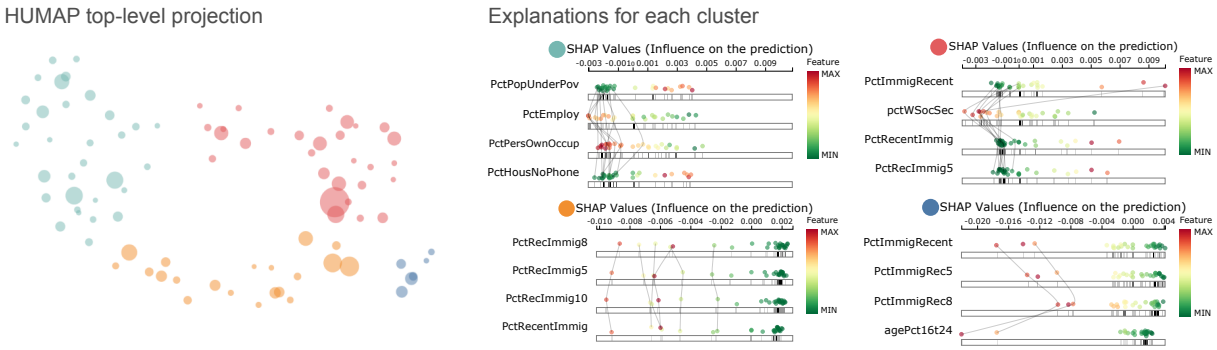


Figure 66 – Explaining dimensionality reduction results. (A) Explanation of the whole dataset provides one story about the clusters defined by the user. (B) Explanation of the top-level HUMAP representation gives an overview of the explanations that can be refined according to user demands.

Using HUMAP, we can understand various facets of the data and focus only on what is important for a given analysis. In addition, it reduces the computing time to generate explanations to help analysts on their exploratory tasks. For instance, based on the clustering result of HUMAP's top-level in Fig. 66, we can identify that the green cluster (left most cluster) corresponds to communities with higher acquisitive power due to negative SHAP values for lower values of `PctPopUnderPov`, `PctHousNoPhone`, and higher values of `PctEmploy` and `PctPersOwnOccuP`; the orange and blue clusters represent communities with higher number of immigrants while the red cluster represents communities with higher number of elderly given by negative SHAP values for higher values of `pctWSocSec` and lower values of `PctImmigRecent`, `PctRecentImmig`, and `PctRecImmig5`.

To explore more of the dataset, Fig. 67 shows how HUMAP is beneficial for understanding various characteristics hidden in data. From (A) to (B), we select a subset of data points together with the projection of the lower hierarchy level. It is important to emphasize that we manually defined the clusters for each hierarchy level, which explains the change of colors when looking at the projections of (A) and (B)—after `project`, we manually define the clusters using `clustering`. After defining the cluster and computing

explanations (C), there is a division between communities composed of high numbers of black people (orange cluster) and white people (blue cluster)—check higher values of `racepctblack` for the orange cluster and `racePctWhite` for the blue cluster associated with negative SHAP values. Finally, the red cluster is associated with a higher number of crimes (rapes and burglaries). It is also associated with a higher number of `HousVacant`. The number of vacant households.

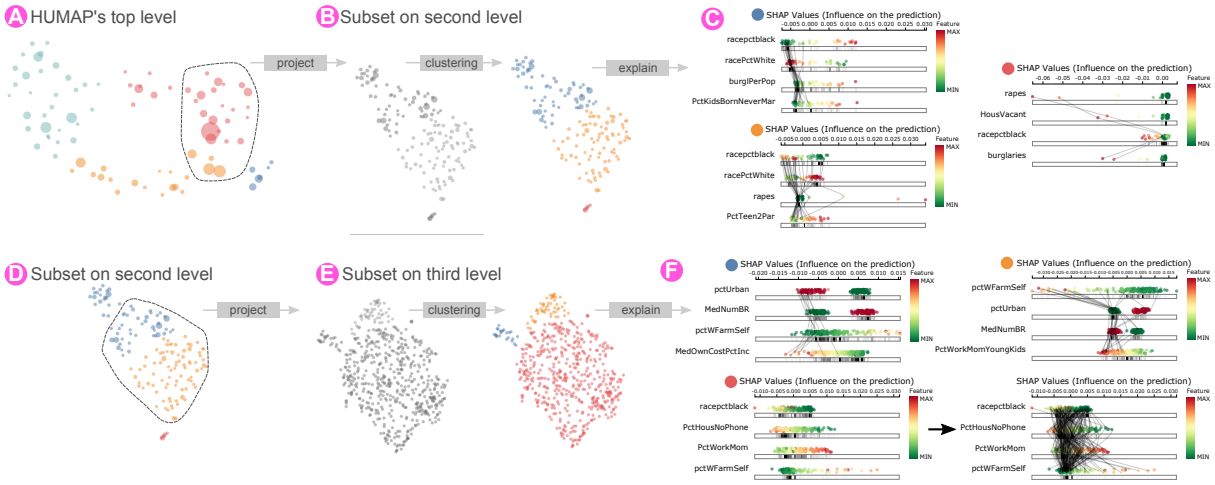


Figure 67 – User interaction with HUMAP to get more detailed information about the data. At each level explanation (C and F), users can generate diverse insights by focusing on distinct subsets of data to investigate using HUMAP's drill-down operation.

Further exploring the projection in (B), we select data points from two clusters to investigate how they differentiate (E) with a distinct clustering assignment. The blue cluster consists of communities of people living in urban areas (higher `pctUrban`) in smaller apartments or houses (lower `MedNumBR`). Further, such a cluster is also characterized by a lower percentage of households with farms or self-employment income (lower `pctWFarmSelf`). Interestingly, the orange cluster presents opposite characteristics from the blue cluster, in which there are more households with farms or self-employment income (higher `pctWFarmSelf`), a lower percentage of people living in urban areas (lower `pctUrban`), and bigger houses (higher `MedNumBR`). Finally, there is an interesting aspect of the red cluster in which lower values for `racepctblack` contribute to the cluster cohesion (negative SHAP values), and higher values contribute to the confusion of this cluster to the two others (positive SHAP values). A higher percentage of houses with no phone (higher `PctHousNoPhone`), a lower percentage of working mothers (lower `PctWorkMom`), and a lower percentage of households with farms and self-employment are three features that contribute to the cluster formation—besides being a common aspect for the white and black communities in this cluster.

5.3.2 EXPLORING LATENT FEATURES

In this case study, we explore the learned features of a convolutional neural network (CNN). Here, we aim to emphasize mental map preservation across hierarchy levels, which is beneficial when exploring a large number of data points. We used the Fashion MNIST dataset (XIAO; RASUL; VOLLGRAF, 2017), composed of 70,000 of 28×28 grayscale images of fashion items (clothing, footwear, and bags). Our architecture consists of 32 3×3 convolutional filters, 64 3×3 convolutional filters, a dense layer of 128 neurons, and a softmax layer of ten neurons. We used the dense layer as features for each image after training the dataset, resulting in a $70,000 \times 128$ matrix. Fig. 68(A) shows the top level of HUMAP created using three hierarchy levels. Each data point is color-coded using its ground truth labels, where the gray border encodes misclassification by the CNN. Notice the four main clusters, where three of them we further explore by selecting data points for drilling down the hierarchy, as indicated by the arrows.

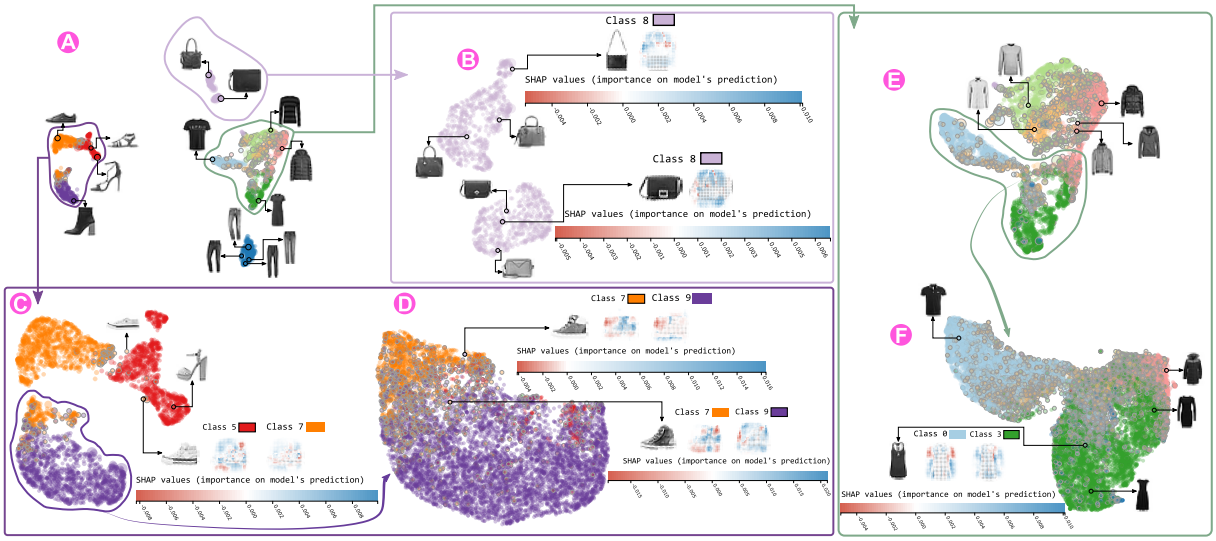


Figure 68 – Hierarchical exploration of the learning weights and explanations of a CNN trained on Fashion MNIST dataset. The top hierarchy level (A) encodes the overview of the dataset. Exploring lower levels by drilling-down the hierarchy (B-F) reveals more detailed information while preserving the mental map.

We start by analyzing the separation of cluster ● in the projection. The subsequent level of the hierarchy (see Fig. 68(B)) and the images representing a few data samples show that the features learned by the CNN can differentiate purses with and without visible handles. The explanation computed using SHAP (LUNDBERG; LEE, 2017) shows how each part of the image contributes to augmenting (in blue) or decreasing (in red) the prediction probability for each class. Based on SHAP analysis, having a visible shoulder strap is very determinant in assigning a data sample in the bag group. If a bag does not have this item, its rectangular format contributes to the correct classification.

Then, for the cluster highlighted in purple in Fig. 68(A), we investigate the misclassification of data samples by projecting another hierarchy level (C). The selection

of data samples from and in Fig. 68(C) show that the CNN confusion of classifying sneakers as high-heels was due to image intensity. For the instance explained using SHAP, the CNN assigned too much importance to the middle part of the sneakers, augmenting the probability for class . The sneaker has darker colors, a characteristic that CNN learned to recognize high-heels. Then, we project the last hierarchy level (Fig. 68(D)) to investigate the group of boots ( ). Inspecting the explanations for two data samples, the sample of class was apparently misclassified by the fact that it does not present protection (or darker color in the image) on the ankle area—see this aspect on SHAP values for classes and .

Finally, we drill down to the second level using the data samples from the cluster highlighted in green in Fig. 68(A), resulting in Fig. 68(E). Then, we proceed to the lowest level of the hierarchy by selecting another cluster (Fig. 68(F)). The cluster correspond to dresses , t-shirts , and a smaller number of samples from coats . Focusing on CNN's misclassifications, we see several samples from class that do not have sleeves (tank tops and blouses), making the CNN misclassify them with dresses. The SHAP explanations show that the highlighted tank top was misclassified as a dress (class) since CNN focused on the absence of sleeves.

5.4 NUMERICAL EVALUATION

In this section, we numerically evaluate HUMAP[2] and compare it against existing HDR techniques. We compare our approach with Hierarchical Stochastic Neighbor Embedding (HSNE)[3] (PEZZOTTI et al., 2016), Multiscale PHATE[4] (KUCHROO, 2020), and UMAP[5] (MCINNES; HEALY; MELVILLE, 2018) by using their official implementation. We cannot control the number of data points in each hierarchy level in HSNE and Multiscale PHATE. Thus, we generate HSNE with three levels and a HUMAP hierarchy with three levels based on the number of data points in each HSNE level. Multiscale PHATE does not accept a parameter to specify the number of levels. After fitting the Multiscale PHATE hierarchy, we search for a level with size similar to the top-level produced by HUMAP and HSNE techniques. Finally, we evaluate UMAP to demonstrate its differences from HUMAP regarding mental map and structure preservation. Thus, to simulate hierarchy levels using UMAP, we project the same data points generated in the HUMAP hierarchy.

We use the following datasets for evaluation. MNIST (LECUN; CORTES, 2010) is a dataset composed of $70,000$ 28×28 pixel grayscale images of handwritten digits classified in ten classes (from 0 to 9), where each flattened image results in a 784 dimensional vector.

[2] https://github.com/wilsonjr/humap
[3] https://github.com/biovault/nptsne
[4] https://github.com/KrishnaswamyLab/Multiscale_PHATE
[5] https://github.com/lmcinnes/umap

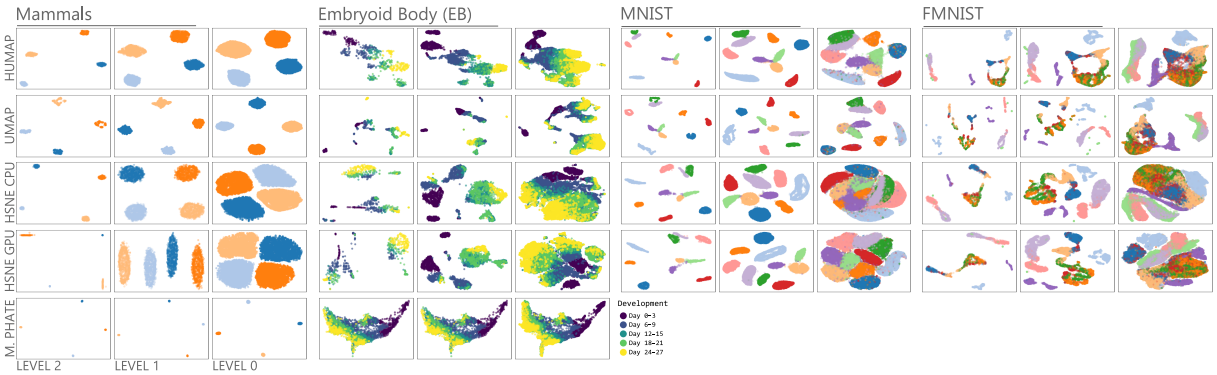


Figure 69 – Visual analysis of the embeddings generated for top and lowest hierarchical levels using a three-level hierarchy. For each dataset, top-level embedding appears on the left, and the lowest level (whole dataset) appears on the right.

Fashion MNIST (XIAO; RASUL; VOLLGRAF, 2017) (FMNIST) is a dataset composed of $70,000$ 28×28 pixel grayscale images of fashion items (clothing, footwear, and bags) divided into ten classes; each flattened image results in a 784 dimensional vector. Mammals is a synthetic dataset designed to have four well-separated classes, and it consists of $20,000$ data points described by 72 dimensions. Embryoid Body (EB) is a single-cell RNA sequencing dataset for embryoid body data generated over 27 days (MOON, 2018) divided into five periods. For this dataset, we aim to visualize the development of the $31,000$ cells after preprocessing them and using the first 50 principal components to generate the embeddings. The experiments were performed in a computed with the following configuration: Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz, 32 GB RAM, Ubuntu 64 bits, NVIDIA GeForce GTX 1660 Ti 22 GB.

Dataset	Size	Dimensions
Mammals	20000	72
Embryoid Body (EB)	31000	17845 $\rightarrow$ PCA(50)
MNIST	70000	784
FMNIST	70000	784

Table 3 – Datasets used for experimentation.

Fig. 69 shows the embeddings for the hierarchy levels. HSNE technique does not maintain the structure presented on the top level for all datasets. While it presents a good relationship among clusters on the top-level embedding, it embeds clusters too close to each other for the projection of the whole datasets—for example, the clusters for the mammals dataset on level 0. For the EB dataset, it does not present the continuous nature of the biological data. Finally, one major issue is that it cannot preserve the mental map throughout hierarchy levels.

Multiscale PHATE could not generate embeddings for FMNIST and MNIST datasets due to their high dimensionality. Although it uncovered the EB dataset's continu-

ous nature, it produce embeddings for the mammals dataset that difficult analysis since the data points for each cluster are too close to each other. On the other hand, HUMAP preserves the mental map across hierarchy levels, encoding the information present in lower levels on top hierarchy levels. For Multiscale PHATE, the main difficulty seems to reduce high-dimensional datasets without reducing the dimensionality to a tractable number (e.g., by using PCA). This is mainly related to computational resources rather than the ability of Multiscale PHATE to identify structures and similarity relationships correctly.

Finally, UMAP shows on level 2 the correct overview only for the *easier* datasets (e.g., *mammals* and *MNIST*). For *EB* and *FMNIST* datasets, UMAP cannot uncover the structures shown in the lowest level. This was an expected result since the relationship among data points needs to be transferred to the sampled using for projection—which is not the goal for UMAP. On the other hand, HUMAP projects higher levels based on the relationship present on lower levels, which results in the ability to encode moderate-sized datasets.

For the numerical evaluation, we compare the techniques using well-known and traditional quality metrics for embedding evaluation (e.g., *Trustworthiness* and *Continuity*) as well as the metrics that appear in the related works, such as *DeMaP* (MOON; DIJK; WANG, 2019) and *Neighborhood Preservation* (PAULOVICH et al., 2008) that aim to evaluate how the low-dimensional embeddings represent the structures in higher dimensions.

5.4.1 RUNNING-TIME

Since these techniques have stochastic components, we run the experiments twenty times for each evaluation metric following the procedure discussed at the beginning of this section. Fig. 70 shows boxen-plots in log scale for runtime execution (seconds) to fit the hierarchy and to embed each hierarchy level. Except for level 0, HUMAP presents similar runtime execution to HSNE in GPU, making it a promising approach for users with low resources when reasonable runtime executions are needed. Although HUMAP shows slightly bigger runtime execution than HSNE on CPU for fitting the hierarchy, it is faster when embedding the hierarchy levels. Multiscale PHATE seems unreasonable for interactive applications when embedding many data points (for example, level 0 of mammals and EB datasets). Finally, the difference between HUMAP and UMAP decreases as the number of data points increase.

5.4.2 NEIGHBORHOOD PRESERVATION

Fig. 71 shows the neighborhood preservation (NP) (PAULOVICH et al., 2008) for varying number of neighbors k ($k \in [1, 30]$). Such a metric computes the average ratio

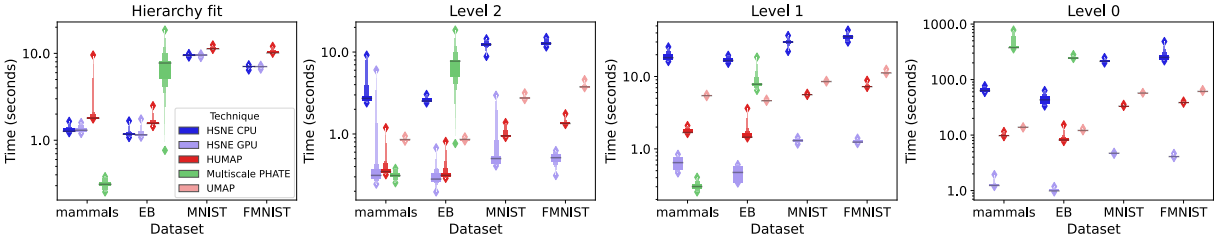


Figure 70 – Run-time execution in seconds using a $\log_{10}$ scale. The boxen-plots show the run-time for 20 runs of each method under the same condition – we measure the run-time to fit the hierarchy (Hierarchy fit), to embed the top level (Level 2), and to embed the whole dataset (Level 0).

of neighbors preserved in the projection. HUMAP technique shows lower neighborhood preservation for datasets that are meant for other analyses, that is, when the main goal is to identify global relationship among clusters rather than only focusing on local structures.

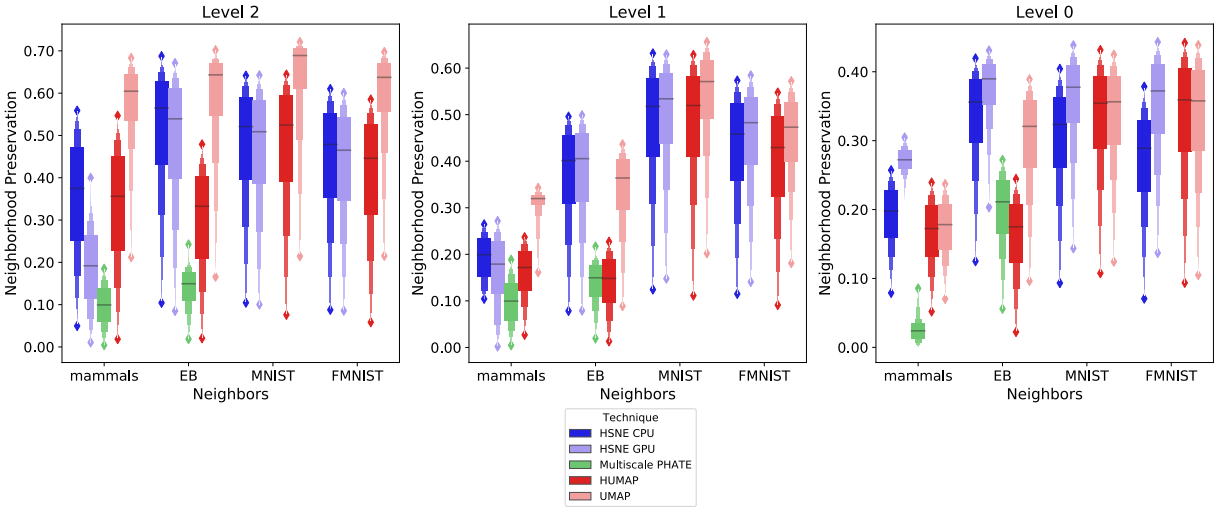


Figure 71 – Neighborhood Preservation after embedding on $\mathbb{R}^2$. HSNE outperforms HUMAP by a great margin only for *EB* dataset.

The *EB* dataset is meant for continuity analysis, and the *mammals* was designed to have well-defined classes. Thus, HUMAP receives lower NP when projecting such datasets by not focusing only on local structures. HSNE, which inherits the SNE characteristics, presents higher NP for focusing on separating the clusters—see the EB projections, for example.

We evaluated HUMAP with different values of β (which controls the contribution of local neighbors to the similarity among landmarks) and empirically show a relationship between local neighborhood and neighborhood preservation. Besides that, adding only 3% of the current neighborhood (100 neighbors) makes HUMAP surpass HSNE by a great margin for MNIST dataset on top hierarchy levels. Visually, the β parameter makes HUMAP cluster together similar data points—these results can be further investigated in the Appendix C (Figs. 3 and 4).

5.4.3 DEMAP

To analyze how well the techniques convey manifolds, clusters, and other high-dimensional space structures, we use the DEMaP metric (MOON; DIJK; WANG, 2019)—it computes the Spearman correlation between geodesic distances on high-dimensional and euclidean distances on low-dimensions. Fig. 72 shows that HUMAP presented better results than HSNE and UMAP on level 2 and HSNE on GPU on level 0 for the MNIST and FMNIST datasets. From the images, the pair of distributions of (HUMAP, UMAP) for the EB, FMNIST datasets on level 2 and (HUMAP, UMAP) for the EB dataset on level 0 are not statistically different after a t-test—the others are statistically different with a p-value of at least < 0.00001 (see Appendix C for the details). For the mammals dataset, HUMAP shows higher values when embedding the whole dataset, providing evidence of our technique's stability across hierarchical levels. While HSNE and Multiscale PHATE show better results for the first level, they lose essential information when drilling down the hierarchy. On the HSNE side, the relationship between clusters is lost. Finally, HUMAP conveys the continuous structures present in the EB dataset even for the top-level embedding, reflected by its DEMaP values. At level 0, however, HSNE on CPU and Multiscale PHATE present higher DEMaP values.

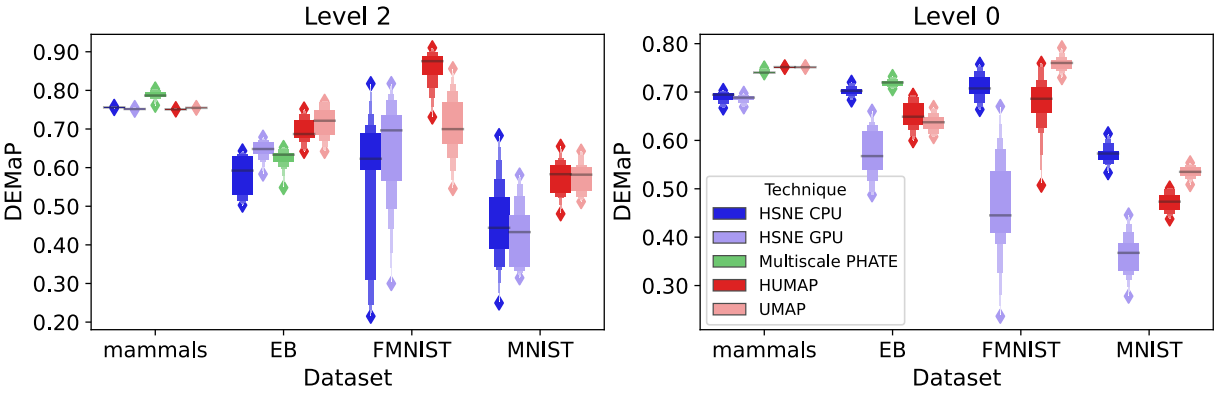


Figure 72 – Evaluation of techniques' ability to represent complex structures such as clusters, manifolds, and other relationships.

Although HUMAP presents lower DEMaP values for mammals on the top-level and EB on level 0, Fig. 69 shows the mental map preservation across hierarchical levels. The preservation of the mental map is important to not mislead users and add cognitive overload during exploration, especially for unlabelled datasets—this characteristic also holds for subsets of data points, which we address in the following section. Finally, the fixing term for preserving the mental map reduces the quality of the whole projection (level 0) according to metric DEMaP. When setting the embedding of lower hierarchy levels to follow the pattern of higher hierarchy levels, we limit the optimization algorithm's freedom when positioning the data points. Appendix C presents the same analysis by setting free the optimization algorithm ($\theta = 1$), which results in higher DEMaP values.

5.4.4 MENTAL MAP PRESERVATION

To quantify the mental map preservation, we use Procrustes Analysis on subsequent hierarchy levels. That is, for the hierarchy levels i and $i+1$, we retrieved the points present in both i and $i+1$ and used them to compute disparity between the sets of points. Fig. 73 shows that HUMAP is way superior over the other techniques at preserving the mental map (the lower disparity the better). This result confirms the visual representations of Fig. 69, where HUMAP clearly reduces the amount of rotations and variations between hierarchy levels.

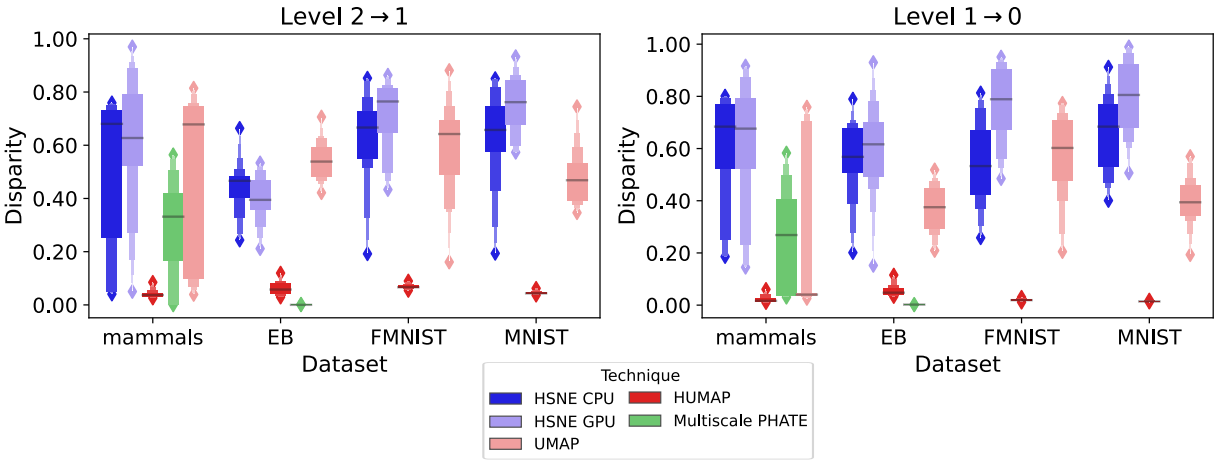


Figure 73 – Mental map preservation measured using Procrustes Analysis. HUMAP decreases the variation when positioning points across hierarchy levels.

5.4.5 CONTINUITY

Fig. 74 shows the continuity values for varying number of neighbors, which measures the extent to which original clusters are preserved. Except for Multiscale PHATE on *mammals* for level 2, all of the techniques performed really well with values greater than 0.9. Here, we can see the expected effect that HUMAP preserves the global structure more than focusing only on cluster formation for the EB dataset—the Fig. 69 illustrates this effect.

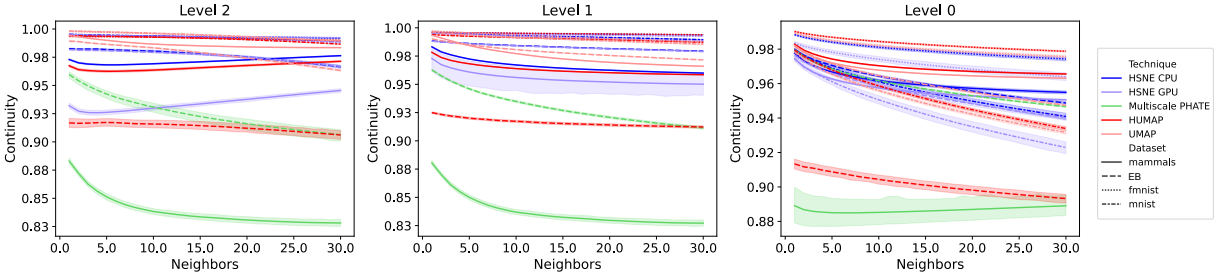


Figure 74 – Continuity values for varying neighborhoods. All of the techniques performed well at preserving clustering structures.

5.4.6 TRUSTWORTHINESS

The results from neighborhood preservation are comparable with Trustworthiness since it measures to what extent the local structure is retained, shown in Fig. 75. As well as for continuity, all of the techniques performed well with values greater than 0.9, with exception for Multiscale PHATE which shines when dealing for continuous datasets and conveying global information. Finally, as commented in previous sections, HUMAP balances the local and global structures and provides comparable results with techniques that do not worry about this trade-off.

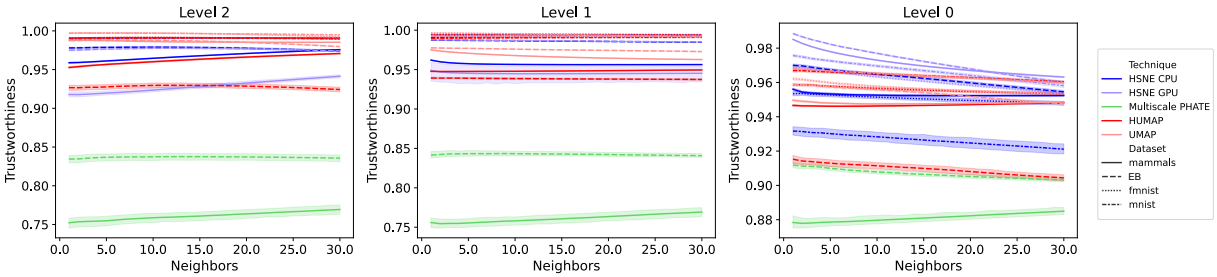


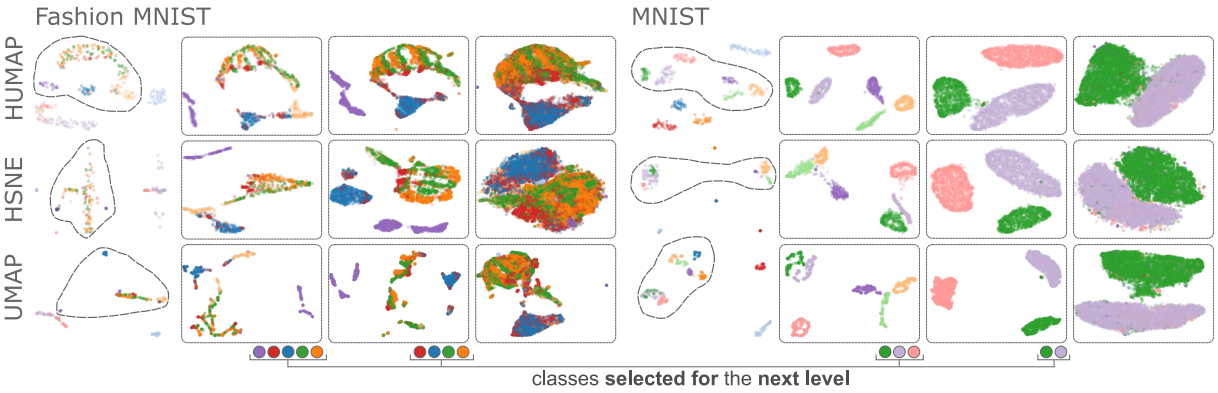
Figure 75 – Trustworthiness values showing similar information to Neighborhood Preservation values.

5.4.7 PROJECTING SUBSETS OF DATA

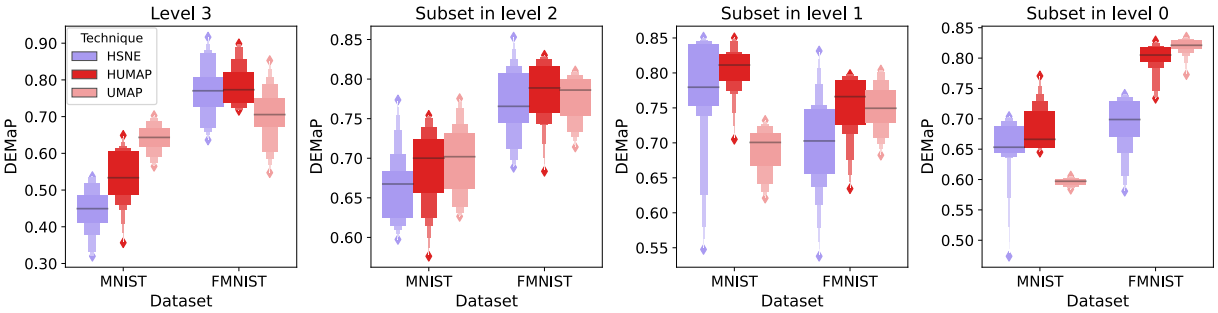
This section focuses on comparing the techniques on their ability to project subsets of data points. We do not analyze Multiscale PHATE since there is no way to drill-down a hierarchy based on classes of data points—besides that Multiscale PHATE does not handle MNIST and FMNIST's dimensionality. Fig. 76a shows the same pattern of Fig. 69, in which HUMAP conveys the structures on top-level and preserves the mental map across hierarchical levels. In this case, we project a hierarchy with four levels, adding more complexity to the analysis. In Fig. 76a, the subset on first level was lasso-selected and the subsequent subsets were selected based on specific classes.

Fig. 77 shows the disparity between subsequent hierarchy levels when projecting subsets of data points. Unlike UMAP and HSNE, HUMAP successfully preserves the mental map even for smaller portions of data, which confirms the visual representation of Fig. 76a.

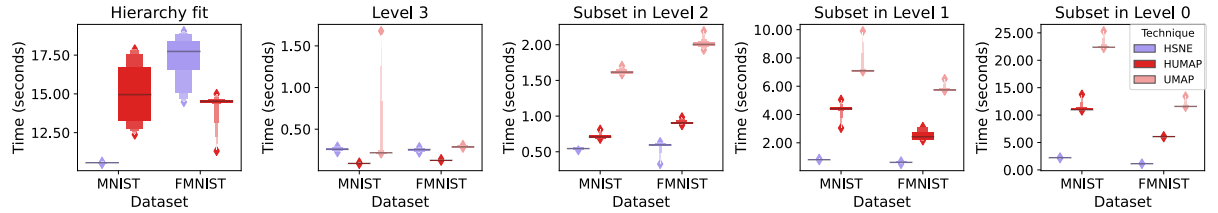
The results presented here show that HUMAP is visually superior to HSNE and UMAP at maintaining the relationship among clusters and data points across hierarchy levels, supported by the quantitative metrics. This characteristic is essential to maintain a mental map and not mislead analysts during interaction. For this analysis, except to UMAP, the techniques take nearly the same time to embed data points on the four levels, and HSNE has advantage on embedding the subset for level 0 due to the GPU implementation (please, see the results for all metrics in Appendix C).



(a) Top-level and selection of subsets for detailed analysis.



(b) DEMaP values to evaluate the quality of embeddings.



(c) Runtime execution for each projection.

Figure 76 – Using HUMAP, UMAP, and HSNE to project subset of data.

5.5 FINAL REMARKS

The experiments show that HUMAP outperforms current approaches in terms of the ability to represent structures in high-dimensional spaces while remaining competitive with GPU-based techniques in terms of run-time execution. Another important aspect of our approach is that it preserves the relationship between data clusters and other structures at various hierarchy levels. Because it preserves the overall structures of higher hierarchy levels in lower hierarchy levels, this critical feature makes HUMAP suitable for progressive analytics.

5.5.1 EMBEDDING INITIALIZATION

Recent research indicates that the initialization of the low-dimensional representation has a significant impact on the quality of these two algorithms, with PCA preferable to random initialization for t-SNE (KOBAK; LINDERMAN, 2021). HSNE employs t-

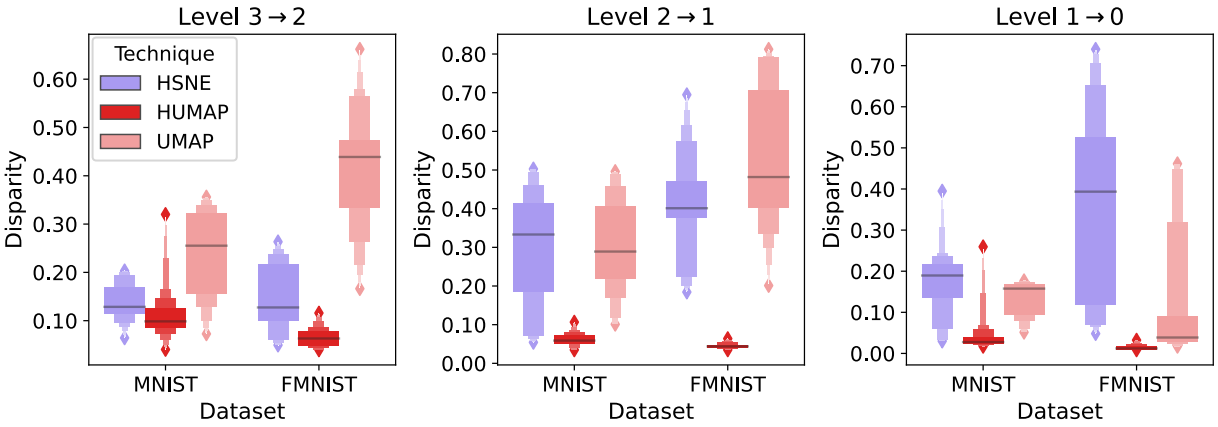


Figure 77 – Mental map preservation when projecting subsets of datapoints. HUMAP shows stability across subsequent hierarchy levels.

SNE with random seeding. This may explain the lower HSNE results for the lowest level (whole dataset) based on the DEMaP metric. However, for higher hierarchy levels, HSNE (and HUMAP) encodes the global neighborhood well, eliminating the need for PCA initialization.

Another intriguing aspect of the numerical comparison is that HSNE on the CPU consistently outperforms HSNE on the GPU. But why is there such a disparity? To begin, HSNE on GPU computes the embeddings on GPU using GPGPU approximation. Although such approximation provides satisfactory embeddings, it sacrifices accuracy for runtime. Finally, HSNE on the CPU computes the embeddings using the Barnes-hut implementation with $\theta = 0$, which is the speed/accuracy trade-off parameter that ranges from 0 to 1, and $\theta = 0$ corresponds to the exact t-SNE.

5.5.2 GENERALIZABILITY AND SCALABILITY

Throughout the experiments, we compared HUMAP to various hierarchical dimensionality reduction techniques, as well as UMAP, using various metrics to demonstrate HUMAP's effectiveness for hierarchical exploration. We also presented case studies in which we used HUMAP to investigate datasets with varying characteristics. In this case, we provide the analysis using the techniques' default parameters. While a more detailed study would be an interesting topic for future research, we attempted to track different applications of a dimensionality reduction technique—from understanding clusters in multidimensional datasets to visualizing single-cell sequencing data. Appendix C1 also contains a detailed analysis of HUMAP's scalability and generalizability.

6 CONCLUSION

This chapter summarizes the main research efforts conducted during the period of doctorate. It also includes additional discussions and open questions to help clarify intriguing research opportunities. Finally, a section is devoted to presenting all of this doctorate's scientific output, whether or not it is related to this thesis.

6.1 FURTHER DISCUSSION AND OPEN QUESTIONS

6.1.1 SADIRE

SADIRE is a new sampling technique for scatterplot-based dimensionality reduction representations. When class separation and outlier visualization are taken into account, it better represents the overall visual space ($\mathbb{R}^2$).

One significant limitation of SADIRE is its assumption of the layout generated by dimensionality reduction techniques. That is, the quality of the dimensionality reduction technique used to uncover the structures (e.g., class separation) present in the dataset is directly related to the result of SADIRE sampling. Finally, because of the neighborhood computation inside each cell, SADIRE's running time is affected when too dense regions are present in the layout formed by the dimensionality reduction technique, though this can be easily alleviated with approximated k-NN algorithms and the parameters for spreading the layout generated that are common in state-of-the-art dimensionality reduction techniques.

SADIRE should be evaluated on different dimensionality reduction techniques in the future to provide a guide based on their different characteristics. Furthermore, because SADIRE is robust in terms of class boundary preservation, applications in machine learning, such as defining training sets for classification, can be investigated. Another idea is to perform the sampling process with a mosaic of hexagons to preserve even more of the structures imposed by the dimensionality reduction technique.

6.1.2 CEXPRESSION

When analyzing dimensionality reduction results represented by scatterplots, understanding the influence of features on the formation of clusters and sub-clusters is a promising approach. cExpression allows for cluster formation differentiation by emphasizing which features distinguish each cluster. While accounting for simple and quick statistical tests to derive separation measures, cExpression also provides focus+context visualization to aid users in information discovery.

The main drawbacks of cExpression are related to visual scalability, which can be addressed with a few analysis strategies. For instance, as the number of dataset features increases (e.g., for textual datasets), users have a wide range of information to visualize and comprehend feature distributions. cExpression, on the other hand, provides an ordering based on feature importance, allowing users to focus their analysis on the first 20 or 30 features. Second, the number of interactions required to investigate each cluster increases linearly with the number of clusters. This is a problem that the biology community is already dealing with (LINDERMAN et al., 2019), but it can be alleviated with heatmap visualization and hierarchical exploration dimensionality reduction.

There are numerous applications for cExpression that can be investigated for future work, such as annotating cell types and textual datasets. Another interesting improvement to the technique would be to change the cluster definition to a more flexible definition of regions for statistical test computation. As a result, the analysis would be carried out locally, as opposed to a global analysis based on clusters.

6.1.3 CLUSTERSHAPLEY

ClusterShapley is a novel technique for interpreting dimensionality reduction results, which is related to the thesis's third goal. ClusterShapley enables users to comprehend the meaning of cluster formation through addictive explanations that are not solely based on attribute similarity among data points. Understanding which factors influenced cluster formation can aid in the discovery of patterns in data and the disclosure of ubiquitous information. This is particularly useful for non-linear dimensionality reduction techniques that make tracking the features that contributed to the resulting clusters difficult.

ClusterShapley's main limitation is its run-time execution for datasets with a large number of features. This is because we use KernelSHAP (LUNDBERG; LEE, 2017) as our background strategy to approximate Shapley values—KernelSHAP, as an agnostic model, cannot assume any model structure. Finally, like cExpression, ClusterShapley bases its analysis on the visual space clusters.

For future work, one can think of an explanatory model that leverages the internal common structures of non-linear dimensionality reduction techniques to quickly generate explanations. This could also be useful for obtaining local explanations rather than being restricted to global explanatory information.

6.1.4 HUMAP

The HUMAP technique is a new hierarchical dimensionality reduction technique. It addresses problems that other approaches do not adequately address, making it a viable option for high-dimensional data analysis. HUMAP, in addition to preserving the mental

map and global and local relationships, provides tunable parameters that allow users to focus on global or local neighborhood preservation. Finally, HUMAP has competitive run-time execution, even when compared to techniques that use GPUs.

One critical aspect of HUMAP is its strategy for preserving the mental map across hierarchical levels. To accomplish this, it provides a fixing parameter that limits the movement of already project data points in the next level's layout optimization. This limitation can reduce metrics related to layout quality and the preservation of high-dimensional structures (even by a small margin). We intend to investigate this parameter further in the future, as well as consider different strategies for mental map preservation.

In terms of future work, one interesting idea is to use HUMAP hierarchical processes for topic extraction. This is due to the fact that top-level representations from HUMAP represent dominant structures from datasets, and textual data could result in the main collection documents. Finally, we intend to investigate novel strategies for assessing hierarchical DR techniques, as well as conduct user experiments on the importance of mental map preservation for hierarchical approaches, particularly when exploring unlabeled datasets.

6.2 PRACTICAL IMPLICATIONS

This section highlights the practical implications of the research efforts conducted during this doctorate.

- **cExpression**: in Chapter 3, we discussed a method for differentiating cluster formation by emphasizing the characteristics that most distinguish them. While this concept is broadly applicable to cell-type annotation (LUECKEN; THEIS, 2019), the cExpression visualization component enhances the analysis' visual scalability. Annotation and class definition for textual datasets is another important practical application of cExpression.

- **ClusterShapley**: in Chapter 4, we detailed a technique that provides addictive explanations for the organization of the embeded layout after dimensionality reduction. ClusterShapley augments the exploratory analysis power, assisting analysts in discovering important information and relationships among data points, as demonstrated by the case studies. ClusterShapley can also be used to understand the separation ability of deep learning models after projecting bottleneck layers—this is especially important because deep learning models are well-known for being interpretability (MOLNAR, 2019) black boxes.

- **HUMAP**: in Chapter 5, we presented a novel hierarchical dimensionality reduction technique. The practical applications of dimension reduction are numerous and

would require more than one paragraph to cover, so we will concentrate on the HUMAP-specific aspects. HUMAP provides a high-level overview of the dataset, making analysis easier and emphasizing intracluster characteristics that are difficult to see in a one-level projection. The discovery of cell types and subtypes is a direct application of this characteristic (LäHNEMANN; KöSTER; SZCZUREK, 2020).

6.3 SCIENTIFIC PRODUCTION

This section covers all of the scientific output created during the thesis development period, from 2019 to 2022. All of the scientific efforts discussed in this thesis, as well as the research activities that resulted in papers worthy of publication, are included in the first-authored publications. The co-authored publications then detail all of the collaborations that occurred during the time period.

6.3.1 FIRST-AUTHORED PUBLICATIONS

6.3.1.1 Journal articles

1. *SADIRE: a context-preserving sampling technique for dimensionality reduction visualizations*. **WE Marcilio-Jr**, DM Eler. Journal of Visualization 23 (6), 999-1013.

2. *Visual analytics of COVID-19 dissemination in São Paulo state, Brazil*. **WE Marcílio-Jr**, DM Eler, RE Garcia, RCM Correia, RMB Rodrigues. Journal of Biomedical Informatics 117, 103753.

3. *Explaining dimensionality reduction results using Shapley values*. **WE Marcílio-Jr**, DM Eler. Expert Systems with Applications 178, 115020.

4. *Contrastive analysis for scatterplot-based representations of dimensionality reduction*. **WE Marcílio-Jr**, DM Eler, RE Garcia. Computers & Graphics 101, 46-58.

6.3.1.2 Conference papers

5. *A hybrid visualization approach to perform analysis of feature spaces*. **WE Marcílio-Jr**, DM Eler, RE Garcia, RCM Correia, LF Silva. 17th International Conference on Information Technology–New Generations.

6. *From explanations to feature selection: assessing shap values as feature selection mechanism*. **WE Marcílio-Jr**, DM Eler. 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).

6.3.1.3 Preprint

7. *Model-agnostic interpretation by visualization of feature perturbations.* **WE Marcílio-Jr**, DM Eler, F Breve. arXiv preprint arXiv:2101.10502. Submitted to Expert Systems with Applications.

8. *HUMAP: Hierarchical Uniform Manifold Approximation and Projection.* **WE Marcílio-Jr**, DM Eler, FV Paulovich, RM Martins. arXiv preprint arXiv:2106.07718. Submitted to IEEE Transactions on Visualization and Computer Graphics.

6.3.2 CO-AUTHORED PUBLICATIONS

6.3.2.1 Conference papers

9. *Visual approach to support analysis of optimum-path forest classifier.* DM Eler, MP Batista, RE Garcia, DR Pereira, **WE Marcílio-Jr**, 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), 777-782.

10. *A class-based evaluation approach to assess multidimensional projections.* J Teixeira, **WE Marcílio-Jr**, DM Eler, A Artero, B Brandoli. 2020 24th International Conference Information Visualisation (IV), 174-181.

11. *Entropy-Based Filter Selection in CNNs Applied to Text Classification.* R Bezerra de Menezes Rodrigues, **WE Marcílio-Jr**, DM Eler. Brazilian Conference on Intelligent Systems, 497-510.

12. *PlaceProfile: Employing Visual and Cluster Analysis to Profile Regions based on Points of Interest.* R Christófano, **WE Marcílio-Jr**, DM Eler.

6.3.2.2 Preprint

13. *Overlap Removal of Dimensionality Reduction Scatterplot Layouts.* GM Hilasaca, **WE Marcílio-Jr**, DM Eler, RM Martins, FV Paulovich. arXiv preprint arXiv:1903.06262. Submitted to IEEE Transactions on Visualization and Computer Graphics.

BIBLIOGRAPHY

ABDI, H.; VALENTIN, D. Multiple correspondence analysis. *Encyclopedia of Measurement and Statistics*, p. 651–657, 2007. Citado na página 47.

ABID, A. et al. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature communications*, v. 9, n. 1, p. 2134, 2018. Citado 2 vezes nas páginas 47 and 77.

ACHEN, C. *Interpreting and Using Regression*. Thousand Oaks, California: [s.n.], 1982. Citado na página 79.

AGARWAL, A. et al. Efficient hierarchical-pca dimension reduction for hyperspectral imagery. In: *2007 IEEE International Symposium on Signal Processing and Information Technology*. [S.l.: s.n.], 2007. p. 353–356. Citado na página 104.

ARCHAMBAULT, D. W.; PURCHASE, H. C.; PINAUD, B. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, p. 539–552, 2011. Citado na página 112.

BECHT, E. et al. Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology*, v. 37, p. 38–44, 2018. Citado 2 vezes nas páginas xiv and 166.

BENATO, B. C.; TELEA, A. C.; FALCãO, A. X. Semi-supervised learning with interactive label propagation guided by feature space projections. In: *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2018. p. 392–399. Citado 2 vezes nas páginas 24 and 25.

BERTIN, J. *Semiology of Graphics*. [S.l.]: University of Wisconsin Press, 1983. ISBN 0299090604. Citado na página 81.

BONI, R. B. D. et al. Depression, anxiety, and lifestyle among essential workers: A web survey from brazil and spain during the covid-19 pandemic. *J Med Internet Res*, v. 22, n. 10, Oct 2020. ISSN 1438-8871. Citado na página 175.

BRUGNONE, N. et al. Coarse graining of data via inhomogeneous diffusion condensation. *2019 IEEE International Conference on Big Data (Big Data)*, p. 2624–2633, 2019. Citado na página 105.

CANTAREIRA, G. D.; PAULOVICH, F. V. A Generic Model for Projection Alignment Applied to Neural Network Visualization. In: TURKAY, C.; VROTSOU, K. (Ed.). *EuroVis Workshop on Visual Analytics (EuroVA)*. [S.l.]: The Eurographics Association, 2020. ISBN 978-3-03868-116-8. ISSN 2664-4487. Citado na página 105.

CICHOCKI, A.; PHAN, A.-H. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions*, p. 708–721, 2009. Citado 2 vezes nas páginas 72 and 150.

COIMBRA, D. B. et al. Explaining three-dimensional dimensionality reduction plots. *Information Visualization*, v. 15, n. 2, p. 154–172, 2016. Citado 5 vezes nas páginas 25, 47, 49, 76, and 78.

CONTRASTIVE analysis for scatterplot-based representations of dimensionality reduction. *Computers & Graphics*, v. 101, p. 46–58, 2021. ISSN 0097-8493. Citado na página 113.

DAVIS, E. N. U. *Wine spoilage is legally defined by volatile acidity, largely composed of acetic acid*. 2004. https://waterhouse.ucdavis.edu/whats-in-wine/volatile-acidity. [Online; accessed 01-29-2020]. Citado na página 93.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, [Royal Statistical Society, Wiley], v. 39, n. 1, p. 1–38, 1977. ISSN 00359246. Citado na página 39.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018. Citado na página 64.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. http://archive.ics.uci.edu/ml. Citado 5 vezes nas páginas 69, 83, 89, 91, and 94.

ELER, D. et al. Analysis of document pre-processing effects in text and opinion mining. *Information*, 2018. Citado 2 vezes nas páginas 49 and 97.

ELER, D. M. et al. Visual analysis of image collections. *The Visual Computer*, v. 25, n. 10, p. 923–937, Oct 2009. ISSN 1432-2315. Citado na página 24.

ESPADOTO, M. et al. Towards a quatitative survey of dimension reduction techniques. *IEEE Transactions on Visualization & Computer Graphics*, 2019. Citado na página 104.

FELIX, C.; DASGUPTA, A.; BERTINI, E. The exploratory labeling assistant: Mixed-initiative label curation with large document collections. In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2018. (UIST '18), p. 153–164. ISBN 9781450359481. Citado na página 49.

FREY, B. J.; DUECK, D. Clustering by passing messages between data points. *Science*, v. 315, p. 2007, 2007. Citado na página 144.

FUJIWARA, T. et al. An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE Transactions on Visualization and Computer Graphics*, v. 26, p. 418–428, 2020. Citado 2 vezes nas páginas 26 and 112.

FUJIWARA, T.; KWON, O.-H.; MA, K.-L. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Trans. Vis. and Comp. Graph.*, v. 26, p. 45–55, 2019. Citado 14 vezes nas páginas 24, 25, 47, 49, 56, 71, 76, 77, 78, 86, 94, 112, 113, and 152.

HELLERSTEIN, J. M. et al. Interactive data analysis: The control project. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 32, n. 8, p. 51–59, ago. 1999. ISSN 0018-9162. Citado na página 30.

HILASACA, G. M.; PAULOVICH, F. V. Visual feature fusion and its application to support unsupervised clustering tasks. *Information Visualization*, v. 19, n. 2, p. 163–179, 2020. Citado na página 105.

HOFFMAN, M.; BACH, F. R.; BLEI, D. M. Online learning for latent dirichlet allocation. In: LAFFERTY, J. D. et al. (Ed.). *Advances in Neural Information Processing Systems 23*. [S.l.]: Curran Associates, Inc., 2010. p. 856–864. Citado 2 vezes nas páginas 72 and 150.

HÖLLT, T. et al. Cyteguide: Visual guidance for hierarchical single-cell analysis. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE InfoVis 2017)*, v. 24, n. 1, p. 739 – 748, 2018. Citado 2 vezes nas páginas 24 and 49.

HÖLLT, T. et al. Focus+context exploration of hierarchical embeddings. *Computer Graphics Forum (Proceedings of EuroVis 2019)*, v. 38, n. 3, 2019. Citado 2 vezes nas páginas 102 and 104.

INGRAM, S.; MUNZNER, T.; OLANO, M. Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics*, v. 15, p. 249–261, 2009. Citado na página 104.

IZENMAN, A. J. Linear discriminant analysis. In: _____. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. New York, NY: Springer New York, 2008. p. 237–280. Citado 2 vezes nas páginas 49 and 78.

JANNé, K. et al. Hierarchical principal component analysis (pca) and projection to latent structure (pls) technique on spectroscopic data as a data pretreatment for calibration. *Journal of Chemometrics*, v. 15, n. 4, p. 203–213, 2001. Citado na página 104.

JEANMOUGI, M. et al. Should we abandon the t-test in the analysis of gene expression microarray data: a comparison of variance modeling strategies. *PLoS One*, 2010. Citado na página 75.

JOIA, P. et al. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 17, n. 12, p. 2563–2571, dez. 2011. ISSN 1077-2626. Citado 3 vezes nas páginas 31, 36, and 103.

JOIA, P.; PETRONETTO, F.; NONATO, L. G. Uncovering representative groups in multidimensional projections. *Computer Graphics Forum*, v. 34, p. 281–290, 2015. Citado 14 vezes nas páginas 24, 25, 30, 31, 35, 36, 39, 47, 49, 76, 77, 78, 141, and 142.

JOLLIFFE, I. *Principal Component Analysis*. [S.l.]: Springer Verlag, 1986. https://www.bibsonomy.org/bibtex/21b71c6fb66b439679c9e0974777a8677/marymcglo. Citado 3 vezes nas páginas 24, 47, and 103.

KAUFMAN, L.; ROUSSEUW, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. Principles and Practice: Wiley-Interscience, 2005. Citado na página 81.

KNUTH, D. E. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN 0-201-89684-2. Citado 3 vezes nas páginas 24, 30, and 35.

KOBAK, D.; LINDERMAN, G. C. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature biotechnology*, 2021. Citado na página 124.

KRUEGER, R. et al. Facetto: Combining unsupervised and supervised learning for hierarchical phenotype analysis in multi-channel image data. *IEEE Transactions on Visualization and Computer Graphics*, v. 26, n. 1, p. 227–237, 2020. Citado 2 vezes nas páginas 24 and 25.

KRUSKAL, J.; WISH, M. *Multidimensional Scaling*. [S.l.]: Sage Publications, 1978. https://www.bibsonomy.org/bibtex/24e10a135eb8c3cb07dfa456979d95a25/tmalsburg. Citado 2 vezes nas páginas 49 and 78.

KUCHROO, M. e. a. Multiscale phate exploration of sars-cov-2 data reveals multimodal signatures of disease. *bioRxiv*, Cold Spring Harbor Laboratory, 2020. Citado 5 vezes nas páginas 26, 102, 105, 108, and 117.

KWON, B. et al. Clustervision: Visual supervision of unsupervised clustering. *IEEE Trans. Vis. Comput. Graph.*, IEEE Computer Society, Los Alamitos, CA, USA, v. 24, n. 01, p. 142–151, jan 2018. ISSN 1941-0506. Citado 2 vezes nas páginas 49 and 78.

KWON, B. C. et al. Sampling for scalable visual analytics. *IEEE Comput. Graph. Appl.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 37, n. 1, p. 100–108, jan. 2017. ISSN 0272-1716. Citado na página 30.

LABELLE, H. et al. The importance of spino-pelvic balance in l5-s1 developmental spondylolisthesis: A review of pertinent radiologic measurements. *Spine*, v. 30, p. 27–34, 2005. Citado 2 vezes nas páginas 71 and 90.

LE, T.; AKOGLU, L. Contravis: Contrastive and visual topic modeling for comparing document collections. In: *The World Wide Web Conference*. New York, NY, USA: Association for Computing Machinery, 2019. (WWW '19), p. 928–938. ISBN 9781450366748. Citado 4 vezes nas páginas 24, 25, 47, and 50.

LECUN, Y.; CORTES, C. MNIST handwritten digit database. 2010. Citado 2 vezes nas páginas 41 and 117.

LEYDESDORFF, L.; SCHANK, T. Dynamic animations of journal maps: Indicators of structural changes and interdisciplinary developments. John Wiley and Sons, Inc., USA, v. 59, n. 11, p. 1810–1818, sep 2008. ISSN 1532-2882. Citado na página 112.

LINDERMAN, G. C. et al. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, v. 16, p. 243 – 245, 2019. Citado na página 127.

LOPES, A. et al. Visual text mining using association rules. *Computers & Graphics*, v. 31, n. 3, p. 316 – 326, 2007. ISSN 0097-8493. Citado na página 49.

LUECKEN, M. D.; THEIS, F. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, v. 15, 2019. Citado na página 128.

LUNDBERG, S. M. et al. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, Nature Publishing Group, v. 2, n. 1, p. 2522–5839, 2020. Citado 2 vezes nas páginas 24 and 79.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems 30*. [S.l.]: Curran Associates, Inc., 2017. p. 4765–4774. Citado 8 vezes nas páginas 27, 79, 80, 82, 99, 113, 116, and 127.

LäHNEMANN, D.; KöSTER, J.; SZCZUREK, E. e. a. Eleven grand challenges in single-cell data science. *Genome Biol*, v. 31, 2020. Citado 6 vezes nas páginas 25, 26, 76, 103, 104, and 129.

MA, B.; WEI, Q.; CHEN, G. A combined measure for representative information retrieval in enterprise information systems. *Journal of Enterprise Information Management*, v. 24, p. 310–321, nov 2011. Citado na página 36.

MAATEN, L. J. P.; HINTON, G. E. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, v. 9, p. 2579—2605, 2008. Citado 9 vezes nas páginas 25, 36, 41, 76, 83, 89, 103, 107, and 153.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: CAM, L. M. L.; NEYMAN, J. (Ed.). *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.]: University of California Press, 1967. v. 1, p. 281–297. Citado na página 39.

MARCILIO-JR, W. et al. A hybrid visualization approach to perform analysis of feature spaces. *International Conference on Information Technology–New Generations*, v. 1134, 2020. Citado 3 vezes nas páginas 23, 25, and 76.

MARCILIO, W. E.; ELER, D. M.; GARCIA, R. E. An approach to perform local analysis on multidimensional projection. *30th SIBGRAPI Conf. on Graph., Patterns and Images (SIBGRAPI)*, p. 351–358, 2017. Citado 2 vezes nas páginas 49 and 78.

MARCíLIO-JR, W. E.; ELER, D. M. Sadire: a context-preserving sampling technique for dimensionality reduction visualizations. *Journal of Visualization*, v. 23, p. 999–1013, 2020. Citado na página 57.

MARCíLIO-JR, W. E.; ELER, D. M. Explaining dimensionality reduction results using shapley values. *Expert Systems with Applications*, v. 178, p. 115020, 2021. ISSN 0957-4174. Citado na página 113.

MARCíLIO-JR, W. E. et al. A hybrid visualization approach to perform analysis of feature spaces. In: LATIFI, S. (Ed.). *17th International Conference on Information Technology–New Generations (ITNG 2020)*. Cham: Springer International Publishing, 2020. p. 241–247. ISBN 978-3-030-43020-7. Citado 4 vezes nas páginas 26, 49, 103, and 141.

MARCíLIO-JR, W. E. et al. Explorertree: A focus+context exploration approach for 2d embeddings. *Big Data Research*, v. 25, p. 100239, 2021. ISSN 2214-5796. Citado 3 vezes nas páginas 28, 45, and 75.

MARCíLIO-JR, W. E. et al. Humap: Hierarchical uniform manifold approximation and projection. *ArXiv e-prints*, fev. 2021. Citado na página 75.

MARTINS, R. et al. Visual analysis of dimensionality reduction quality for parameterized projections. *Comput. Graph.*, v. 41, p. 26–42, 2014. Citado na página 49.

MCINNES, L.; HEALY, J.; MELVILLE, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. Citado 11 vezes nas páginas 25, 41, 61, 76, 83, 97, 103, 107, 110, 117, and 168.

MICALLEF, L. et al. Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, v. 23, n. 6, p. 1588–1599, 2017. Citado 2 vezes nas páginas 54 and 104.

MINGHIM, R.; PAULOVICH, F. V.; LOPES, A. A. Content-based text mapping using multidimensional projections for exploration of document collections. *SPIE Proceedings: Visualization and Data Analysis*, v. 6060, 2006. Citado na página 143.

MODELING wine preferences by data mining from physicochemical properties. *Decision Support Systems*, v. 47, n. 4, p. 547 – 553, 2009. ISSN 0167-9236. Smart Business Networks: Concepts and Empirical Evidence. Citado 2 vezes nas páginas 91 and 93.

MOLNAR, C. *Interpretable Machine Learning*: A guide for making black box models explainable. [S.l.: s.n.], 2019. https://christophm.github.io/interpretable-ml-book/. Citado 2 vezes nas páginas 80 and 128.

MOON, K. *Embryoid Body data for PHATE*. 2018. https://data.mendeley.com/datasets/v6n743h5ng/1. Online; accessed 02 February 2021. Citado 2 vezes nas páginas 105 and 118.

MOON, K.; DIJK, D. van; WANG, Z. e. a. Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnol*, p. 1482–1492, 2019. Citado 5 vezes nas páginas 103, 107, 119, 121, and 153.

MUNZNER, T. *Visualization Analysis and Design*. [S.l.]: CRC Press, 2015. https://www.bibsonomy.org/bibtex/29643da4f95f0ec7783796c8120323731/dirk-rathje. (AK Peters Visualization Series). ISBN 9781498759717. Citado na página 60.

MURIE, C. et al. Comparison of small n statistical tests of differential expression applied to microarrays. *BMC Bioinformatics*, v. 10, p. 45 – 45, 2008. Citado 2 vezes nas páginas 49 and 75.

NG, A. Y.; JORDAN, M. I.; WEISS, Y. On spectral clustering: Analysis and an algorithm. In: *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. [S.l.]: MIT Press, 2001. p. 849–856. Citado na página 111.

NGUYEN, T. T.; SONG, I. Centrality clustering-based sampling for big data visualization. *International Joint Conference on Neural Networks (IJCNN)*, IEEE, p. 24–29, 2016. ISSN 21614407. Citado na página 31.

NONATO, L. G.; AUPETIT, M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, p. 1, 2018. ISSN 1077-2626. Citado 2 vezes nas páginas 24 and 53.

Nonato, L. G.; Aupetit, M. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, v. 25, n. 8, p. 2650–2673, 2019. Citado 3 vezes nas páginas 103, 104, and 105.

PAGLIOSA, L. de C.; PAGLIOSA, P. A.; NONATO, L. G. Understanding attribute variability in multidimensional projections. In: *29th Conf. Graphics, Patterns and Images, SIBGRAPI, 2016, Sao Paulo, Brazil, October 4-7, 2016*. [S.l.: s.n.], 2016. p. 297–304. Citado 5 vezes nas páginas 25, 47, 49, 76, and 78.

PARZEN, E. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 33, n. 3, p. pp. 1065–1076, 1962. ISSN 00034851. Citado 2 vezes nas páginas 77 and 84.

PAULOVICH, F. V. et al. Piecewise laplacian-based projection for interactive data exploration and organization. In: *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*. Chichester, UK: The Eurographs Association & John Wiley & Sons, Ltd., 2011. (EuroVis'11), p. 1091–1100. Citado na página 31.

PAULOVICH, F. V.; MINGHIM, R. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visulization and Computer Graphics*, v. 14, n. 6, p. 1229–1236, 2008. Citado 4 vezes nas páginas 102, 104, 141, and 143.

PAULOVICH, F. V. et al. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visulization and Computer Graphics*, v. 3, p. 564–575, 2008. Citado 11 vezes nas páginas xv, 26, 31, 36, 45, 83, 97, 102, 103, 119, and 142.

PAULOVICH, F. V.; SILVA, C. T.; NONATO, L. G. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 16, n. 6, p. 1281–1290, nov. 2010. ISSN 1077-2626. Citado 2 vezes nas páginas xv and 142.

PAULOVICH, F. V.; SILVA, C. T.; NONATO, L. G. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 16, n. 6, p. 1281–1290, nov. 2010. ISSN 1077-2626. Citado 2 vezes nas páginas 31 and 36.

PEKALSKA, E. et al. A new method of generalizing sammon mapping with application to algorithm speed-up. In: *Proc. 5th Annual Conference of the Advanced School for Computing and Imaging (ASCI1999)*. [S.l.: s.n.], 1999. Citado na página 31.

PEZZOTTI, N. et al. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, p. 98–108, 2018. Citado 2 vezes nas páginas 23 and 76.

PEZZOTTI, N. et al. Hierarchical stochastic neighbor embedding. *Comput. Graph. Forum*, The Eurographs Association & John Wiley & Sons, Ltd., Chichester, UK, v. 35, n. 3, p. 21–30, jun. 2016. ISSN 0167-7055. Citado 11 vezes nas páginas 26, 31, 75, 102, 103, 104, 106, 107, 108, 109, and 117.

RAUBER, P. E. et al. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, v. 23, p. 101–110, 2017. Citado 5 vezes nas páginas 24, 25, 26, 36, and 103.

RAUBER, P. E.; FALCãO, A. X.; TELEA, A. C. Visualizing time-dependent data using dynamic t-sne. In: . Goslar, DEU: Eurographics Association, 2016. (EuroVis '16), p. 73–77. Citado 2 vezes nas páginas 105 and 112.

RAUBER, P. E. et al. Interactive image feature selection aided by dimensionality reduction. In: *EuroVA@EuroVis*. [S.l.: s.n.], 2015. Citado na página 54.

REDMOND, M.; BAVEJA, A. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, v. 141, n. 3, p. 660–678, 2002. Citado na página 94.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "why should I trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. [S.l.: s.n.], 2016. p. 1135–1144. Citado na página 24.

RITCHIE, M. E. et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, v. 43, n. 7, p. e47–e47, 01 2015. ISSN 0305-1048. Citado na página 49.

ROBINSON, M. D.; MCCARTHY, D. J.; SMYTH, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, v. 26, n. 1, p. 139–140, 11 2009. ISSN 1367-4803. Citado na página 49.

RöDER, M.; BOTH, A.; HINNEBURG, A. Exploring the space of topic coherence measures. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015. (WSDM '15), p. 399–408. ISBN 9781450333177. Citado 2 vezes nas páginas 72 and 150.

ROSENBLATT, M. Remarks on some nonparametric estimates of a density function. v. 27, n. 3, p. 832–837, set. 1956. ISSN 0003-4851 (print), 2168-8990 (electronic). Citado 2 vezes nas páginas 77 and 84.

ROUSSOULY, P.; PINHEIRO-FRACO, J. Biomechanical analysis of the spino-pelvic organization and adaptation in pathology. *Eur Spine Journal*, 2011. Citado na página 90.

RUDER, S. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747, 2016. Citado na página 111.

SALAMON, J. et al. Network visualization and analysis of spatially aware gene expression data with insitunet. *Cell systems*, v. 6 5, p. 626–630.e3, 2018. Citado na página 24.

SARIKAYA, A.; GLEICHER, M. Scatterplots: Tasks, data, and designs. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, n. 1, p. 402–412, 2018. Citado na página 54.

SARIKAYA, A.; GLEICHER, M.; SZAFIR, D. A. Design factors for summary visualization in visual analytics. *Computer Graphics Forum*, v. 37, n. 3, p. 145–156, 2018. Citado 3 vezes nas páginas 30, 56, and 104.

SEDLMAIR, M.; MUNZNER, T.; TORY, M. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, v. 19, n. 12, p. 2634–2643, 2013. Citado 2 vezes nas páginas 54 and 104.

SHAPLEY, L. A value for n-person games, vol ii of contributions to the theory of games. University Press, Princeton, 1953. Citado 2 vezes nas páginas 77 and 79.

Shneiderman, B. The eyes have it: a task by data type taxonomy for information visualizations. In: *Proceedings 1996 IEEE Symposium on Visual Languages*. [S.l.: s.n.], 1996. p. 336–343. Citado 2 vezes nas páginas 101 and 104.

SILVA, R. R. O. d. et al. Attribute-based Visual Explanation of Multidimensional Projections. In: BERTINI, E.; ROBERTS, J. C. (Ed.). *EuroVis Workshop on Visual Analytics (EuroVA)*. [S.l.]: The Eurographics Association, 2015. Citado 3 vezes nas páginas 49, 51, and 78.

SMIRNI, P.; LAVANCO, G.; SMIRNI, D. Anxiety in older adolescents at the time of covid-19. *Journal of Clinical Medicine*, v. 9, n. 10, 2020. ISSN 2077-0383. Citado na página 175.

SOHNS, J. et al. Attribute-based explanations of non-linear embeddings of high-dimensional data. *CoRR*, abs/2108.08706, 2021. Citado na página 113.

SOMARAKIS, A. et al. Imacyte: Visual exploration of cellular microenvironments for imaging mass cytometry data. *IEEE transactions on visualization and computer graphics*, 2019. Citado na página 24.

STAHNKE, J. et al. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. on Vis. and Comp. Graph.*, v. 22, p. 629–638, 2016. Citado 3 vezes nas páginas 47, 49, and 78.

STEINBACH, M.; KARYPIS, G.; KUMAR, V. A comparison of document clustering techniques. In: *In KDD Workshop on Text Mining*. [S.l.: s.n.], 2000. Citado na página 141.

STRUMBELJ, E.; KONONENKO, I. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, Springer-Verlag, Berlin, Heidelberg, v. 41, n. 3, p. 647–665, dez. 2014. ISSN 0219-1377. Citado na página 79.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. [S.l.]: Pearson Education, 2006. https://www.bibsonomy.org/bibtex/21312364d45c358ebe9a7d1575d6af725/dhruvbansal. Citado na página 39.

TEBET, M. Current concepts on the sagittal balance and classification of spondylolysis and spondylolisthesis. *Rev Bras Ortop*, p. 3–12, 2014. Citado 2 vezes nas páginas 71 and 90.

TIAN, Z. et al. Using multiple attribute-based explanations of multidimensional projections to explore high-dimensional data. *Computers & Graphics*, v. 98, p. 93–104, 2021. ISSN 0097-8493. Citado 3 vezes nas páginas 49, 51, and 113.

TILLÉ, Y. Sampling algorithms. In: _____. *International Encyclopedia of Statistical Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 1273–1274. ISBN 978-3-642-04898-2. Citado na página 31.

TRAAG, V.; WALTMAN, L.; ECK, N. J. van. From louvain to leiden: guaranteeing well-connected communities. *arXiv preprint arXiv:1810.08473*, 2018. Citado na página 61.

TUKEY, J. W. The future of data analysis. *Annals of Mathematical Statistics*, v. 33, p. 1–67, 1962. Citado na página 23.

TURKAY, C. et al. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, v. 18, n. 12, p. 2621–2630, 2012. Citado 5 vezes nas páginas 25, 47, 49, 76, and 78.

TUSHER, V. G.; TIBSHIRANI, R.; CHU, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 98, n. 9, p. 5116–5121, 2001. ISSN 0027-8424. Citado na página 75.

UNEN, V. v. et al. Visual analysis of mass cytometry data by hierarchical stochastic neighbor embedding reveals rare cell types. *Nature Communications*, v. 8, n. 1740, p. 1 – 10, 2017. Citado 2 vezes nas páginas 24 and 25.

UNEN, V. van; HöLLT, T.; PEZZOTTI, N. e. a. Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types. *Nat Commun*, n. 1740, 2018. Citado 2 vezes nas páginas 23 and 76.

VITTER, J. S. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, ACM, New York, NY, USA, v. 11, n. 1, p. 37–57, mar. 1985. ISSN 0098-3500. Citado 4 vezes nas páginas 24, 30, 35, and 141.

WANG, G. et al. Topic hypergraph: hierarchical visualization of thematic structures in long documents. *Science China Information Sciences*, v. 56, n. 5, p. 1–14, May 2013. Citado na página 49.

WANG, Y. et al. Linear discriminative star coordinates for exploring class and cluster separation of high dimensional data. *Computer Graphics Forum*, v. 36, n. 3, p. 401–410, 2017. Citado 2 vezes nas páginas 49 and 78.

WARD, M. O. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, Palgrave Macmillan, v. 1, n. 3/4, p. 194–210, dez. 2002. ISSN 1473-8716. Citado na página 60.

WESTERHUIS, J. A.; KOURTI, T.; MACGREGOR, J. F. Analysis of multiblock and hierarchical pca and pls models. *Journal of Chemometrics*, v. 12, n. 5, p. 301–321, 1998. Citado na página 104.

WILLIAMS, M.; MUNZNER, T. Steerable, progressive multidimensional scaling. In: *Proceedings of the IEEE Symposium on Information Visualization*. Washington, DC, USA: IEEE Computer Society, 2004. (INFOVIS '04), p. 57–64. ISBN 0-7803-8779-3. Citado na página 104.

XIAO, H.; RASUL, K.; VOLLGRAF, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017. Citado 2 vezes nas páginas 116 and 118.

XU, K. S.; KLIGER, M.; HERO, A. O. A regularized graph layout framework for dynamic network visualization. *Data Mining and Knowledge Discovery*, v. 27, p. 84–116, 2012. Citado na página 112.

A SADIRE - EMPLOYING SADIRE INTO EXPLORERTREE & ADDITIONAL USER EXPERIMENTS

In this Appendix, we provide all the projections and graphs generated from the comparison performed for the sampling selection technique. Also, we provide an overview of the tool developed for ExplorerTree.

A.1 EXECUTION TIME, COVERAGE AND REDUNDANCY

To evaluate representative data points, we look at their Coverage and Redundancy features. Coverage accounts for representing the projected dataset structures, while Redundancy accounts for the similarity among the representative data points and the projected space. The *Coverage* (σ_C) consists of the lowest distances from non-representative data points to representative data points. The *Redundancy* (σ_R), on the other hand, consists of the mean of the lowest distances of each representative to other representative points. Using these metrics, as defined in the following, a good representative selection technique shows higher values for σ_R and lower values for σ_C. Notice that N stands for the number of data observations in the dataset X, M stands for the size of representative data points in the representative set X'.

$$\begin{aligned} \sigma_C &= \frac{1}{N-M} \sum_{x \in X} \min_{x' \in X'} (\parallel x - x' \parallel) \\ \sigma_R &= \frac{1}{M} \sum_{x'_i \in X'} \min_{x'_j \in X'} (\parallel x'_i - x'_j) \parallel) \end{aligned} \tag{A.1}$$

Using these metrics, defined in (MARCíLIO-JR et al., 2020), we tested the SADIRE sampling selection strategy against selecting representatives using Bisecting K-Means (STEINBACH; KARYPIS; KUMAR, 2000) (used in HiPP (PAULOVICH; MINGHIM, 2008)), Reservoir (VITTER, 1985) (a random sampling algorithm), and CSM (JOIA; PETRONETTO; NONATO, 2015) (that selects representative data points based on PCA loadings). To select the number of representatives for each technique, we first ran SADIRE and set the other techniques to retrieve the same number. Each algorithm was executed 20 times due to its stochastic nature. Table 4 shows the datasets used in the experiments. We used for $\alpha = 1$ and $k = 5$.

Table 4 – The datasets used in the experiments. All the projections were performed by LSP (PAULOVICH et al., 2008), except for Shuttle, in which we used PLMP (PAULOVICH; SILVA; NONATO, 2010a).

Dataset	Size	Dimensions
CBR-ILP-IR	574	35
Shuttle	43500	9
Corel-1k	100	150
Corel-5k	5000	4096
Diabetes	768	8

The robustness of the SADIRE sampling selection technique regarding redundancy presented the best results for all of the datasets. In general, Bisecting K-Means presented the second-best results due to its strategy of partitioning the projected dataset into two subsets in each iteration, facilitating the return of separated (less redundant) data points. However, some redundancy may remain since it considers only the partitioning scheme, not how much separated are the data points in the visual space. Figure 78 shows this pattern in the resulting representative data points. CSM and Reservoir returned the worse results. In CSM, the representatives are selected based on their ability to represent the dataset according to PCA loadings. Higher variation influences a lot on the resulting representative set (JOIA; PETRONETTO; NONATO, 2015). On the other hand, the Reservoir is influenced by denser areas due to the random strategy, leading to redundancy.

When analyzing *Coverage*, we must take care of the lowest distances among representative data points and non-representative data points. Besides returning lower mean distances (σ_C), a technique must maintain a lower standard deviation to represent outliers or even clusters with lower density. Figure 78 illustrates this idea. While Bisecting K-Means and Reservoir uncover denser areas of Shuttle projection, much of the projection information is lost. To understand such a characteristic, instead of plotting the means and standard deviations of each technique evaluation based on *Coverage*, we use Boxenplot to visualize all of the distances computed in the process, as shown in Figure 79. The plots show that SADIRE returns more stable results (with only a few outlier distances) while uncovering representative data points that need lower distances to represent a non-representative data point – see how the majority of the distances are located in the bottom of the graphs. Other techniques, however, show many outliers since they are only able to represent denser areas.



Figure 78 – A sampling technique's ability to uncover the projection structures must take into account the outliers and less dense clusters. The results show that SADIRE is the most stable method among the analyzed, covering the projected dataset being taking into account all of its structures.

A.2 EXPLORERTREE TOOL

A.2.1 OVERVIEW

Figure 80 shows the three components of the ExplorerTree tool. In **a.**, users can specify datasets, choose different dimensionality reduction techniques and different sampling mechanisms. In **b.**, users can hierarchically explore multidimensional embeddings using the focus+context as explained in the manuscript. In **c.**, users can select among different visual summarization strategies to visualize the tool-tips.

A.2.2 USING DIFFERENT DIMENSIONALITY REDUCTION TECHNIQUES

Using ExplorerTree, users can choose between different dimensionality reduction techniques, as shown in Figur 81 for the CBR-ILP-IR dataset. For exemplification, IDMAP (MINGHIM; PAULOVICH; LOPES, 2006) (in **a.**) and LSP (PAULOVICH; MINGHIM, 2008) (in *b.*) were used.

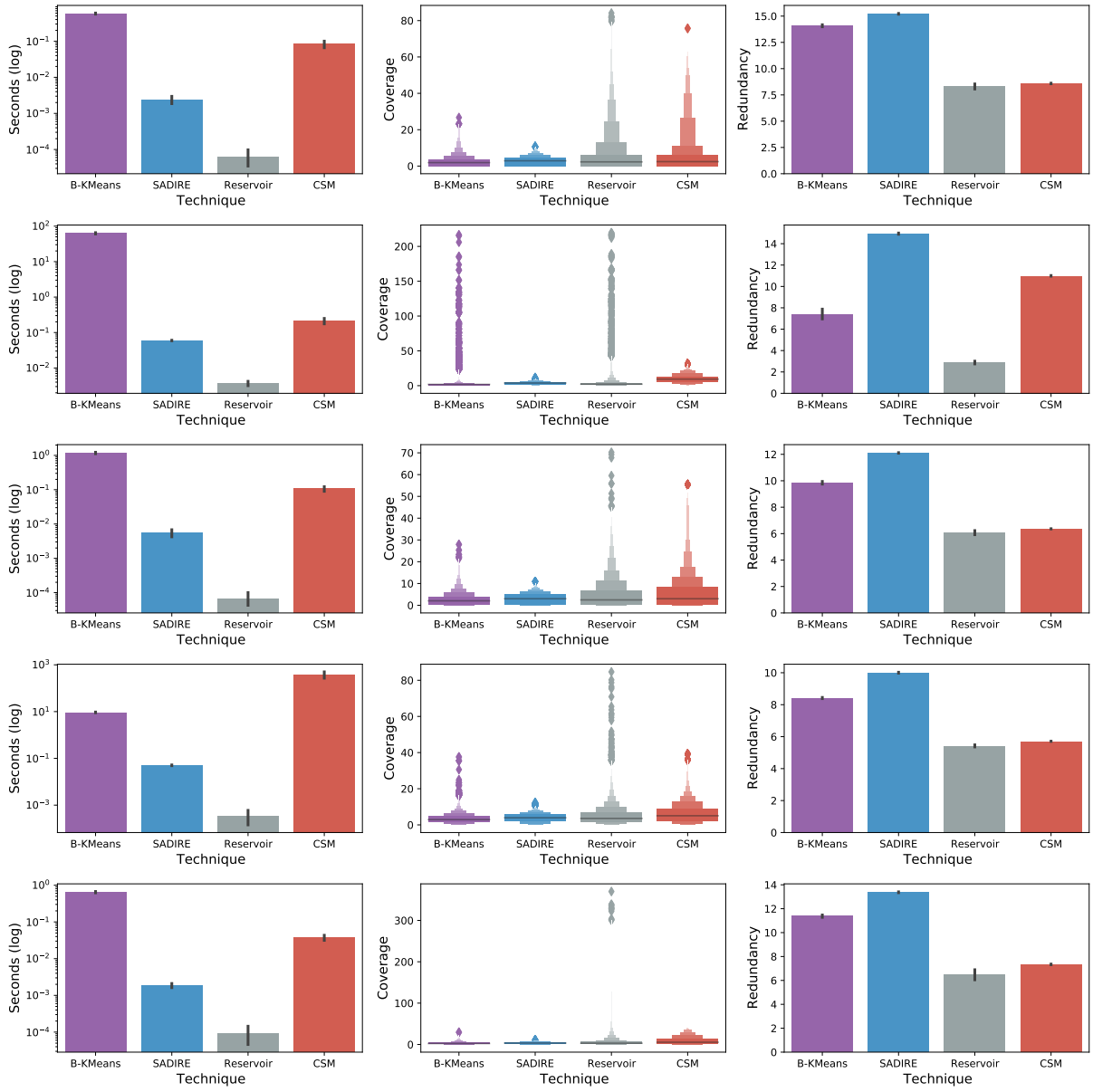


Figure 79 – Results for in *Execution time (s)*, *Coverage*, and *Redundancy* for CBR-ILP-IR, Shuttle, Corel-1k, Corel-5k, and Diabetes (each line) .

A.2.3 USING DIFFERENT SAMPLING SELECTION MECHANISMS

Different sampling techniques can also be employed in ExplorerTree. In Figure 82, two sampling techniques were used for constructing the hierarchical structure: **a.** SADIRE (ours); **b.** Affinity Propagation (FREY; DUECK, 2007).

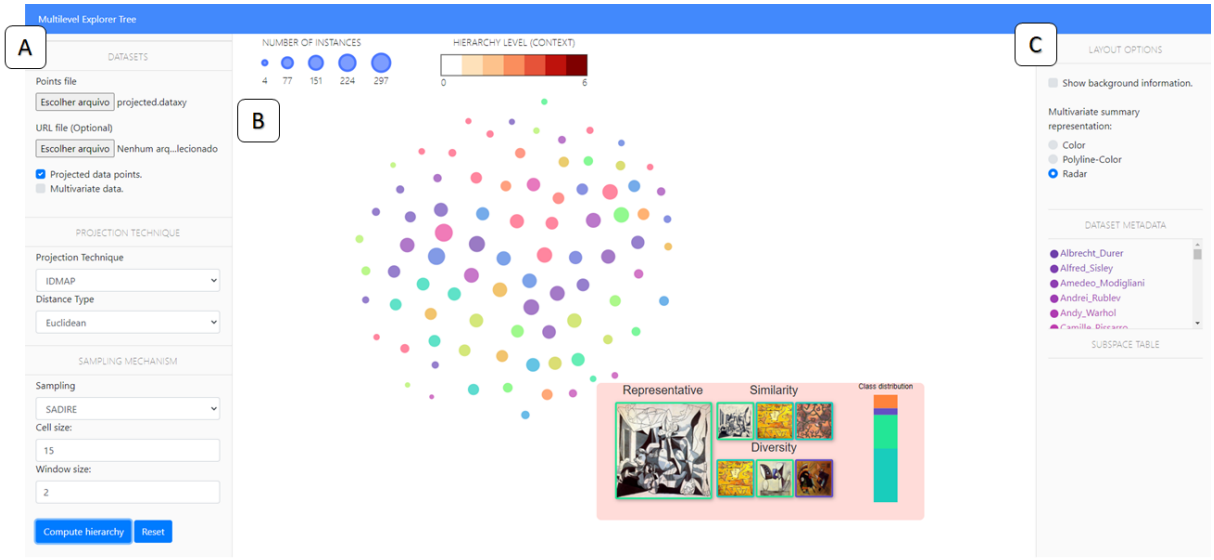


Figure 80 – The overview of ExplorerTree tool.

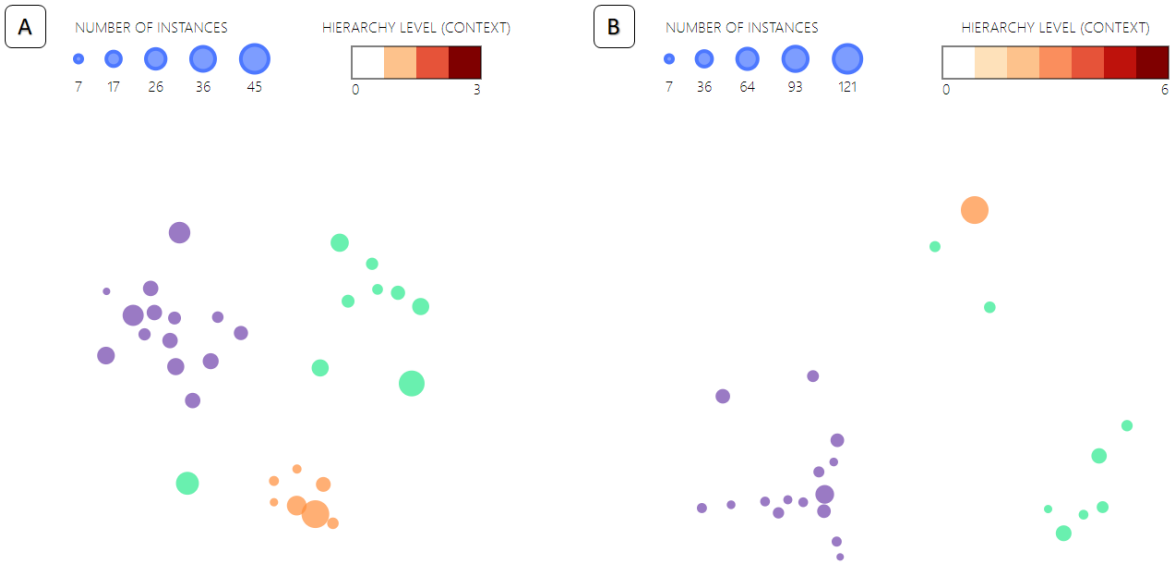


Figure 81 – Choosing different dimensionality reduction techniques: **a.** IDMAP; **b.** LSP.

A.3 EXPLORERTREE USING DIFFERENT SAMPLING STRATEGIES

Figure 83 shows the first level of ExplorerTree when using different sampling strategies. Although users can select any sampling mechanism, we argue that SADIRE provides a better understanding of the projected space due to its high coverage and low redundancy characteristic.

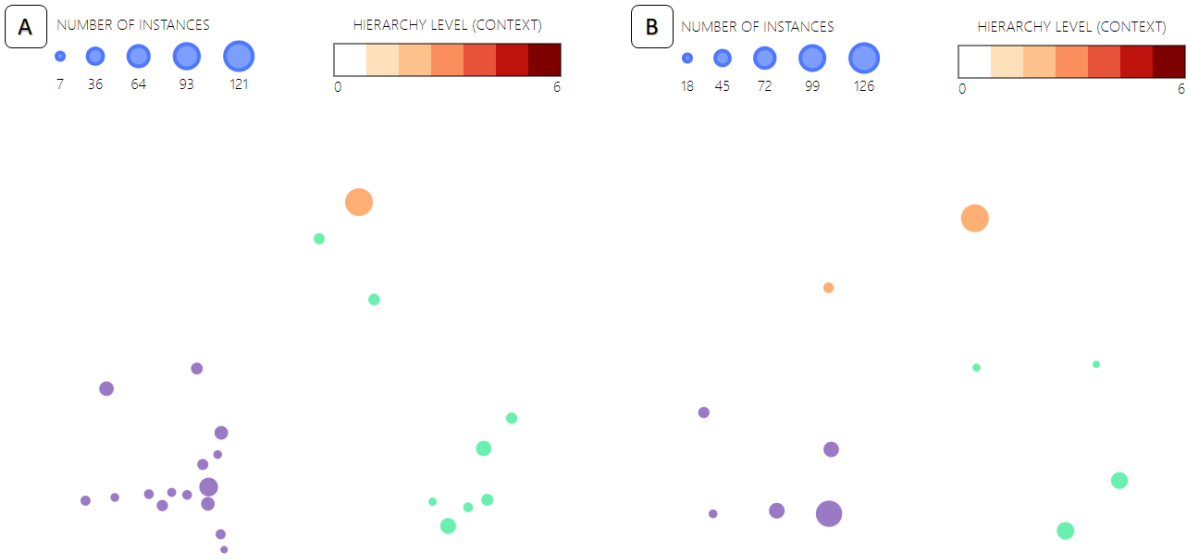


Figure 82 – Choosing different sampling techniques: **a.** SADIRE (ours); **b.** Affinity Propagation.

A.4 USER EXPERIMENT - EVALUATING SAMPLING STRATEGIES FOR EXPLORERTREE

To evaluate the quality of ExplorerTree on representing the projected dataset using different sampling mechanisms, we performed a second user experiment. We generated an ExplorerTree structure for four datasets using Bisecting k-Means, CSM, Reservoir, and SADIRE. We presented the traditional scatter plot representation of the dataset next to the ExplorerTree's first level and asked the participants to score on how much the ExplorerTree's level represents the dataset (see Figure 83). When giving the scores, we asked the participants to analyze if the clusters are visible, if the class separation remains, and if relevant information about the projection is lost.

Table 3 shows the participants' responses when evaluating how ExplorerTree encodes the whole dataset (see Figure 83). For each dataset, we asked the participants one question ("How well does the strategy represent the whole dataset?") using ExplorerTree using four different sampling techniques ($Q1$ - Bisecting k-means, $Q2$ - CSM, $Q3$ - Reservoir, and $Q4$ - SADIRE).

The experiment shows (see Figure 84) that the participants attributed higher scores for ExplorerTree using SADIRE and Bisecting k-Means, although these two strategies receive lower results CBR-ILP-IR dataset. Comparing the two best strategies (SADIRE and Bisecting k-Means) shows a slight advantage for Bisecting k-Means for the Corel1k and Diabetes datasets, from which we believe the participants tend to focus on the density information rather than the uniformity throughout the representative data samples. However, since ExplorerTree needs rapid run-time execution, we believe SADIRE consists

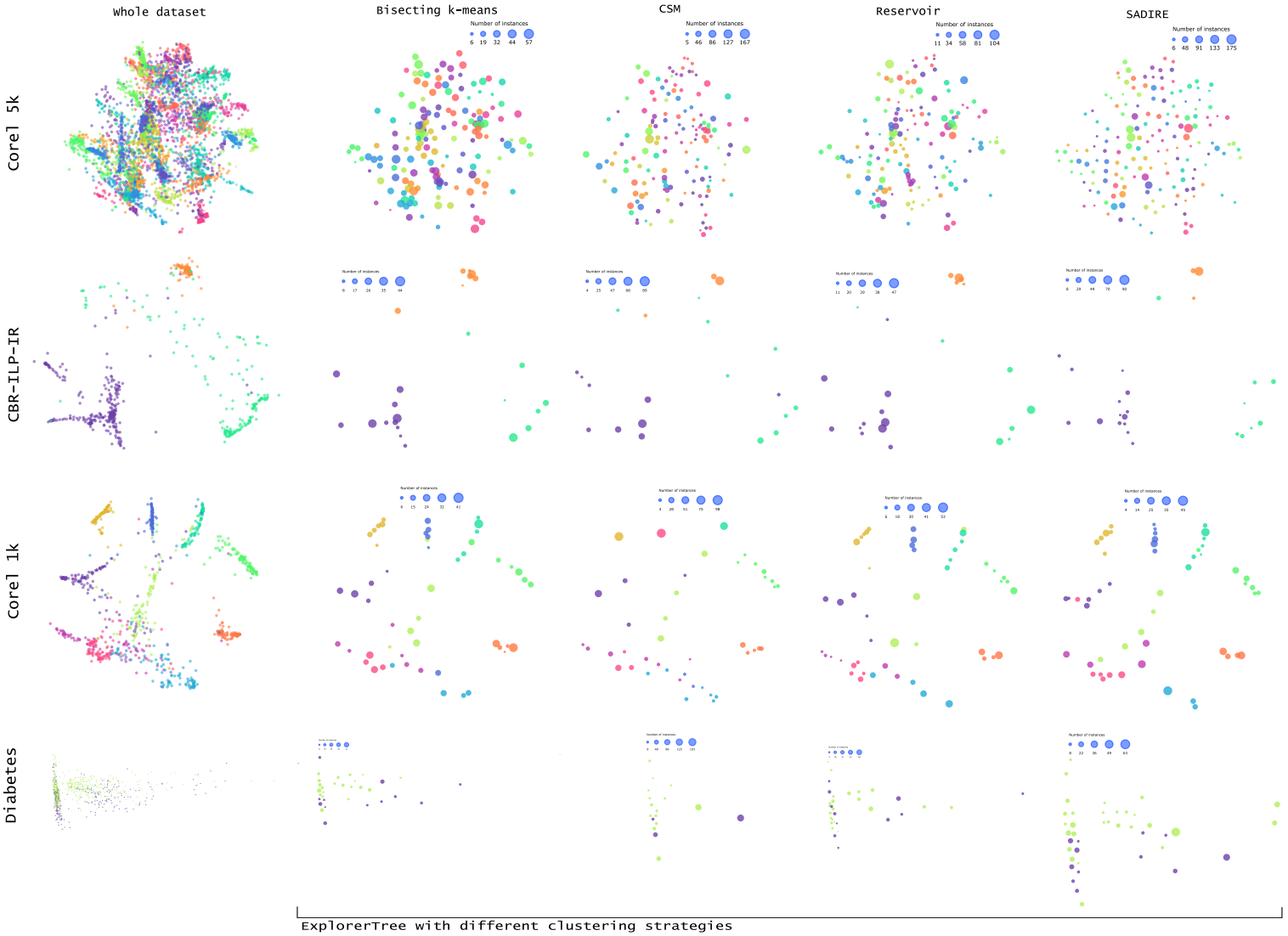


Figure 83 – Traditional scatter plot representations of dimensionality reduction results next to the first level of ExplorerTree using different sampling strategies.

Participant information		**CBR-ILP-R**				**COREL 1k**				**COREL 5k**				**DIABETES**			
Date	Occupation	$Q1$	$Q2$	$Q3$	$Q4$	$Q1$	$Q2$	$Q3$	$Q4$	$Q1$	$Q2$	$Q3$	$Q4$	$Q1$	$Q2$	$Q3$	$Q4$
2021/02/20	Professor	7	9	9	7	10	7	10	10	8	7	7	8	6	3	6	4
2021/02/20	Student	4	5	6	7	7	8	8	8	8	7	8	9	7	6	6	6
2021/02/20	Student	4	7	5	2	10	1	7	9	8	6	5	7	8	4	6	6
2021/02/20	Student	4	3	4	4	6	3	4	5	6	5	7	5	5	4	6	4
2021/02/20	Student	2	7	5	4	10	1	8	7	10	5	8	6	7	3	6	5
2021/02/20	Front-end developer	8	6	10	9	10	5	9	9	9	7	10	8	10	9	10	9
2021/02/20	Student	5	7	7	4	8	8	7	9	8	8	7	6	7	4	6	4
2021/02/20	Student	7	7	4	4	9	5	7	7	5	3	4	7	6	5	7	6
2021/02/20	Visualization	6	7	7	6	9	6	7	7	7	6	7	6	7	5	8	6
2021/02/21	Technology entrepreneur	6	9	8	4	6	7	8	9	8	7	7	10	7	5	6	7
2021/02/21	Assistant manager	7	5	5	6	7	5	5	5	7	5	5	5	6	6	6	6
2021/02/21	assistant manager	5	10	8	9	9	10	8	8	8	7	8	10	10	9	7	8
2021/02/21	Physiotherapist	9	10	10	9	10	10	10	10	9	10	10	10	10	9	9	9
2021/02/21	Student	8	8	8	7	8	7	9	9	9	8	7	9	9	8	7	8
2021/02/21	Developer	6	7	6	5	6	5	5	6	7	5	5	5	6	4	5	4
2021/02/21	Professor	8	6	7	6	9	7	8	9	9	7	8	6	9	6	7	8
2021/02/22	Developer	6	8	7	8	9	6	9	10	8	9	8	8	8	7	9	7
2021/02/22	Student	6	4	3	3	7	5	5	7	6	4	3	6	3	2	2	2

Table 3 – Scores given by the participants

of the best option due to the compromise between quality and run-time execution.

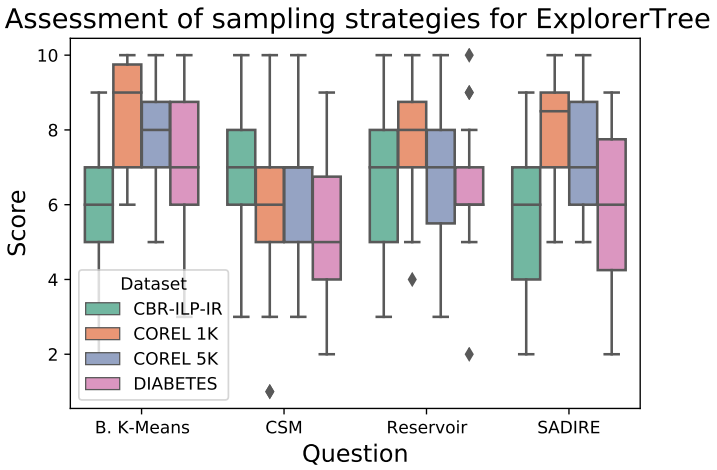


Figure 84 – SADIRE shows slightly lower results compared to Bisecting k-means. However, SADIRE is superior over Bisecting k-means in run-time execution, an important characteristic for our scenario.

B CEXPRESSION

B.1 TOPIC EVALUATION

The *cExpression* technique was evaluated at topic extraction tasks, where we compare it against two well-known and commonly used topic extraction techniques: LDA (HOFFMAN; BACH; BLEI, 2010) and NMF (CICHOCKI; PHAN, 2009). We used *Topic Coherence* (RöDER; BOTH; HINNEBURG, 2015) as an evaluation metric. The *Topic Coherence* metric, which is applied to the top m words from the topic, is defined as the average/median of the pairwise word-similarity scores of the topic's words. A good model will generate coherent topics, i.e., topics with high topic coherence scores. The evaluation was performed using three datasets: 495 news articles (205 terms) and 40794 tweets related to COVID-19 symptoms (295 terms) – please, see the original manuscript –, and the *20newsgroups*[1] dataset. For the *20newsgroups*, we use a subset comprising the classes `alt.atheism`, `comp.graphics`, `comp.windows.x`, `rec.motorcycles`, `sci.electronics`, `sci.med`, `talk.politics.guns`, `talk.politics.misc`, `talk.religion.misc`. After pre-processing the dataset – removal of English stop-words and terms appearing in a proportion of documents below 0.5 –, the dataset resulted in a bag-of-words representation of 4828 documents by 3347 terms.

To analyze the `20newsgroups` dataset, besides varying the number of terms, we also varied the number of topics returned from the algorithm. Figure 85 shows that our technique could surpass LDA and NMF for all number of topics (expect two topics).

[1] http://qwone.com/ jason/20Newsgroups/

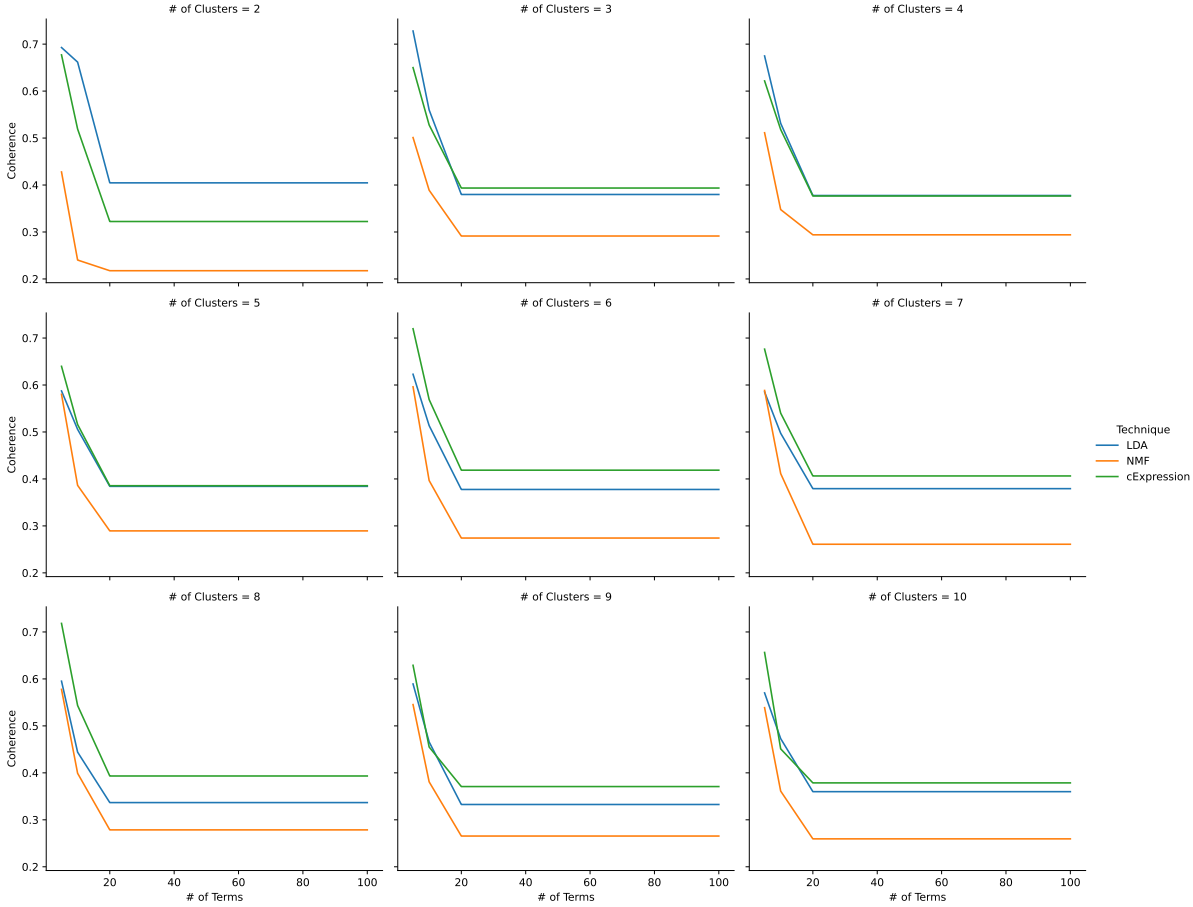


Figure 85 – Coherence values for different number of topics.

B.2 RUN-TIME EVALUATION

Figure 86 shows additional analyzes for the run-time execution of ccPCA (FUJIWARA; KWON; MA, 2019) and cExpression. cExpression shows scalability for all of the scenarios.

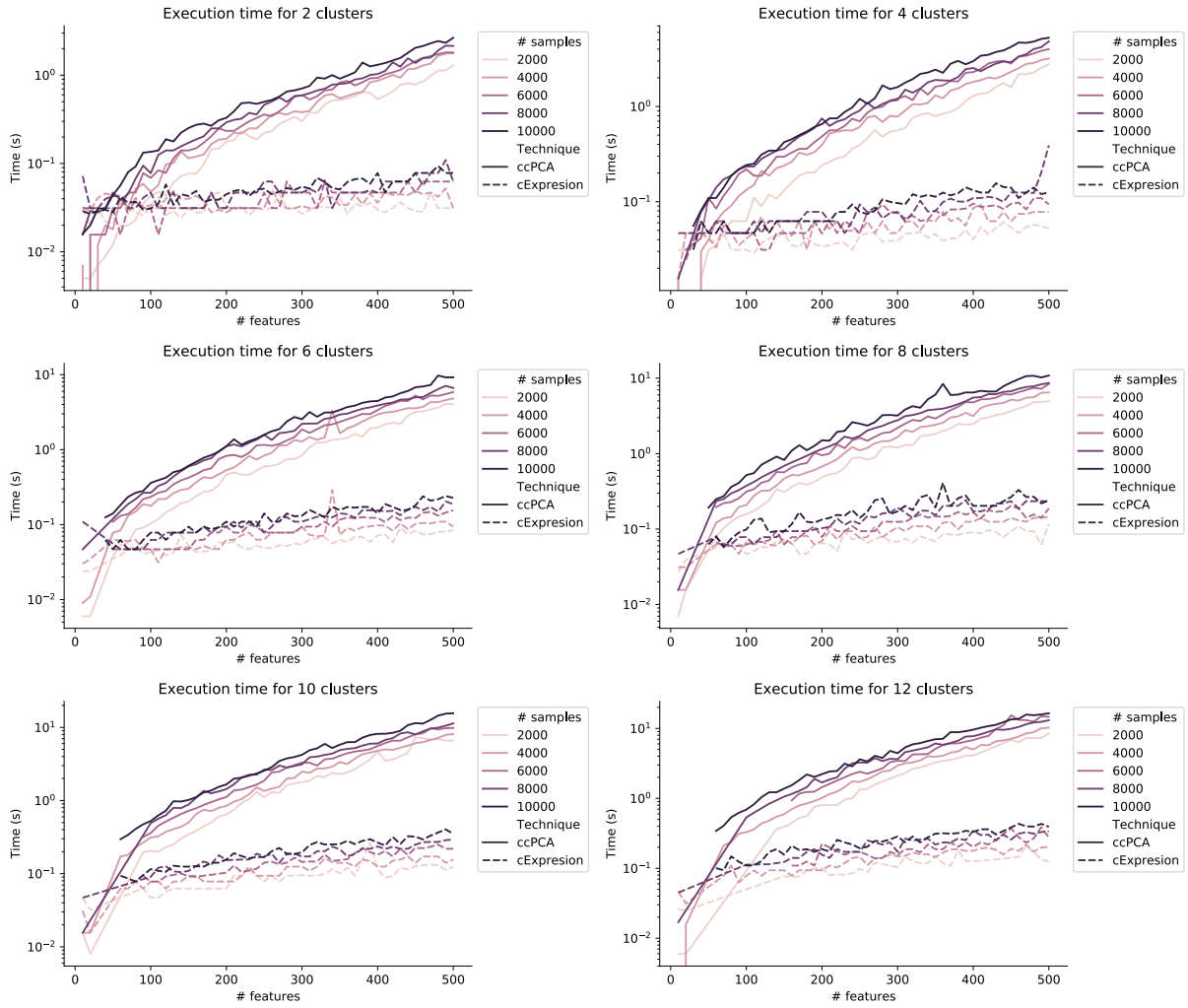


Figure 86 – Run-time execution for cExpression and ccPCA when varying the number of data points, dimensionality, and a number of clusters.

B.3 DIFFERENT DIMENSIONALITY REDUCTION TECHNIQUES

In this section, we use different dimensionality reduction techniques to understand the effect of applying cExpression. For instance, Figure 87 shows the projection and the four most distinctive terms for three clusters after dimensionality reduction with t-SNE (MAATEN; HINTON, 2008) and PHATE (MOON; DIJK; WANG, 2019). The dataset (consisting of news documents) is the same as the one used in the first use case of the manuscript.

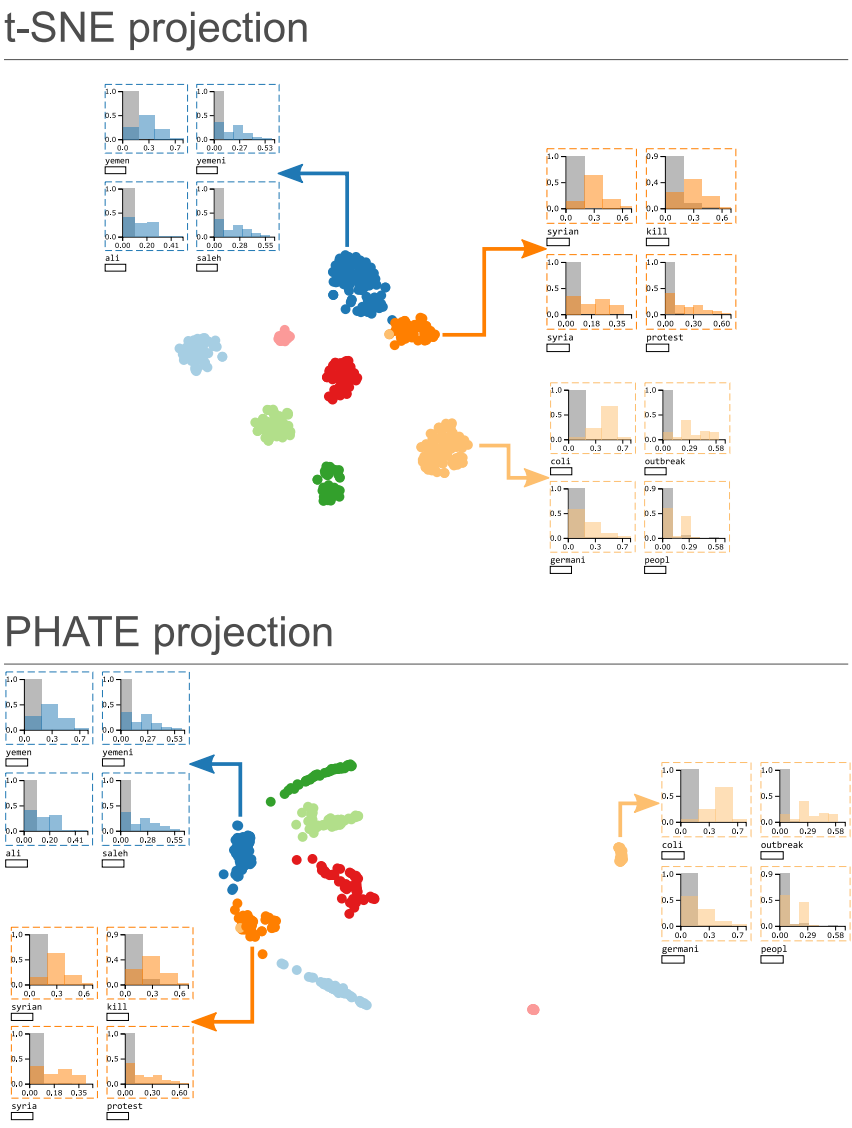


Figure 87 – Interpretation of two dimensionality reduction techniques using cExpression. Since the sets are used to compare the feature distributions, the result remains the same.

Figure 87 shows that although the organization of the visual space after dimensionality reduction is different, cExpression returns the same results for the three highlighted clusters—see the arrows indicating user interactions. This happens because we are treating fixed sets of data too feed cExpression, i.e., the clusters encoded by colors. In further

works, we plan to use k nearest neighbors to compute the statistics, which will result in a more flexible approach.

Another interesting aspect to analyze is how different number of clusters after the result. Figure 88 shows the result if we merge two clusters from the t-SNE projection of Figure 87—please, note that difference in the color code. After requesting details about the light blue cluster (1), we see that the most distinctive terms are the same returned for the dark-blue and dark-orange clusters in Figure 87 (2). Lastly, the inspection of the seven most distinctive terms (2) visualized using the cell-based encoding (3) shows the majority of the unique characteristics lies in the subcluster of left of Figure 88 (1), which corresponds to the dark-blue cluster in Figure 87 of t-SNE projection.

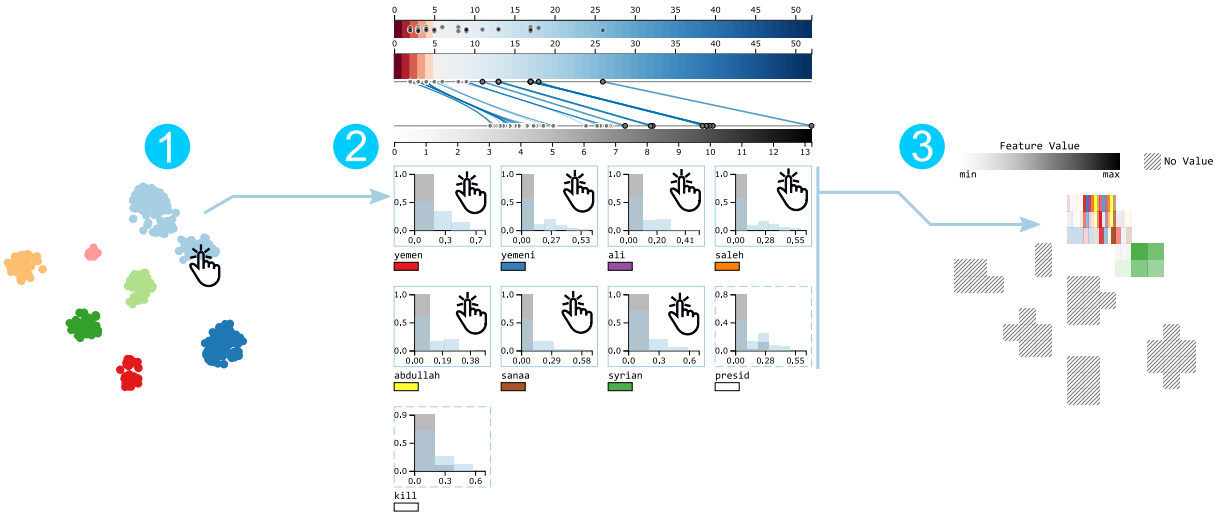


Figure 88 – Analyzing the same dataset with different clustering results. cExpression merges the unique features from the two clusters (dark-blue and dark-orange) from Figure 87.

C HUMAP

In this appendix, we provide all the additional embeddings and analyses of the HUMAP paper. For metrics in which we compare the distributions using boxen-plots, we also use t-test to measure the statistical significance of their differences. Thus, we always show the statistical significance of the difference between HUMAP and the best performing technique other than HUMAP. Finally, we use the following code strategy to accommodate all of the results in tables:

- HUMAP → T1;
- UMAP → T2;
- HSNE GPU → T3;
- HSNE CPU → T4;
- MPHATE → T5.

C.1 HIERARCHICAL DIMENSIONALITY REDUCTION COMPARISON

Figures 89 and 90 show the all of the results for embedding hierarchy levels demonstrated in the paper. Tables 4, 5, and 6 show the the p-values after running t-test using HUMAP and the better performing technique other than UMAP.

	Hierarchy fit		Level 2		Level 1		Level 0	
Dataset	Tech.	p-value	Tech.	p-value	Tech.	p-value	Tech.	p-value
mammals	T1, T5	1.71-e5	T1, T5	0.048	T1, T5	2.46e-33	T1, T3	2.45e-39
EB	T1, T3	7.46e-9	T1, T3	0.247	T1, T3	1.49e-12	T1, T3	9.07e-22
MNIST	T1, T3	2.79e-25	T1, T3	0.069	T1, T3	1.39-e56	T1, T3	3.10e-59
FMNIST	T1, T3	9.16e-21	T1, T3	4.94e-28	T1, T3	8.07e-41	T1, T3	3.62e-58

Table 4 – p-values for the running time metric.

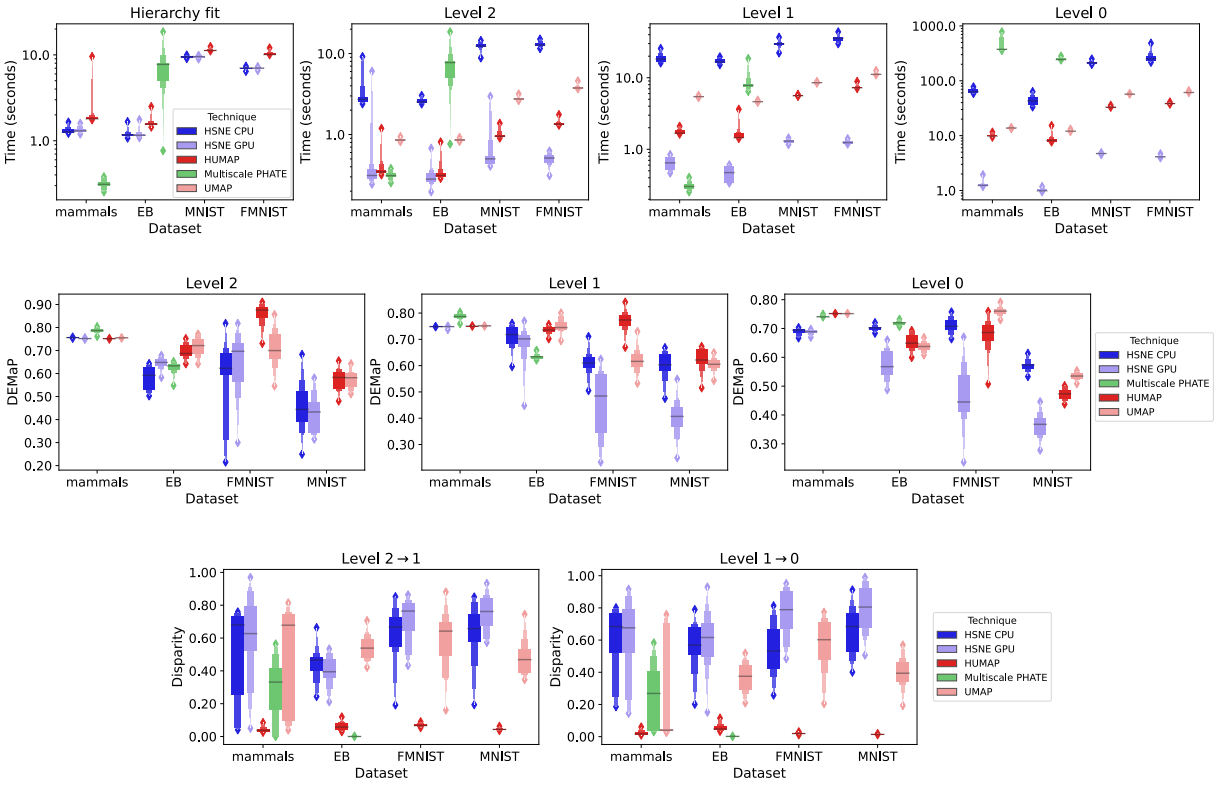


Figure 89 – Results for Time, DEMaP, and Disparity for twenty executions of each HDR technique.

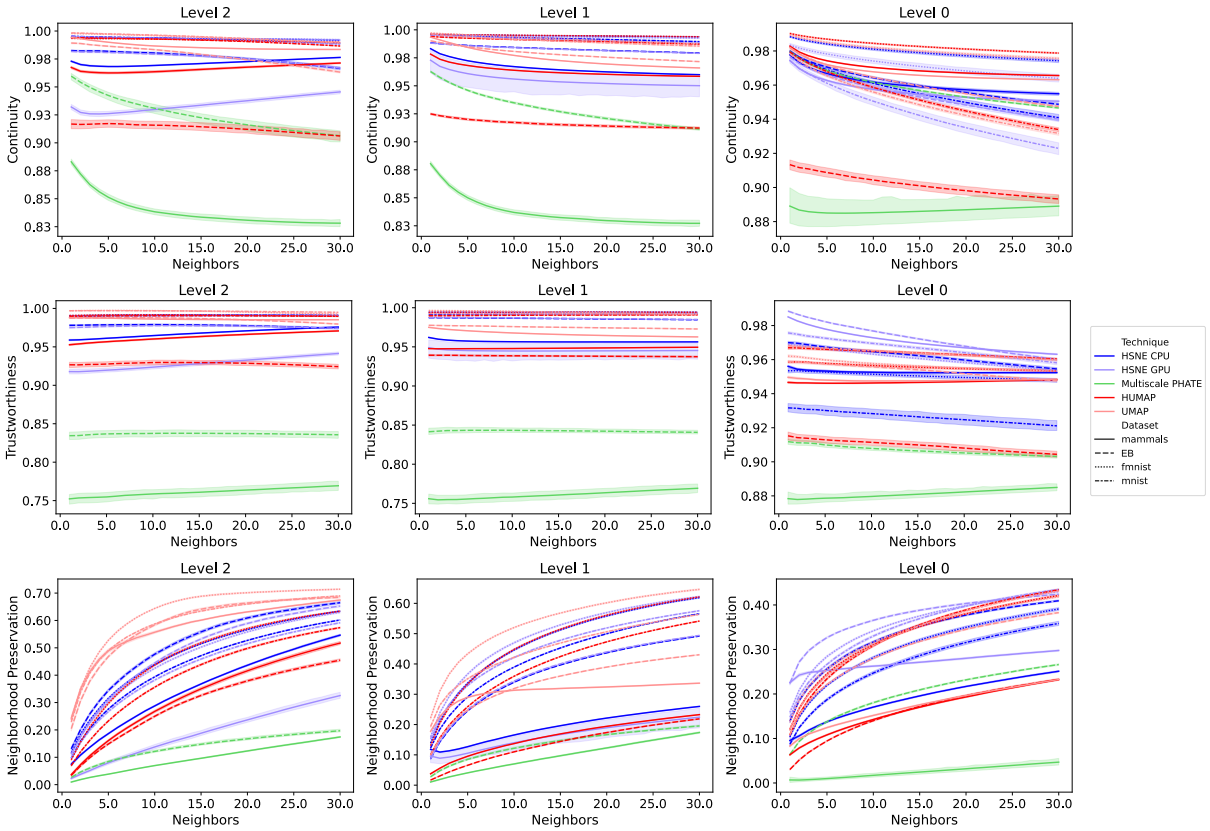


Figure 90 – Results for Neighborhood Preservation, Continuity, and Trustworthiness for twenty executions of each HDR technique.

	Level 2		Level 1		Level 0	
Dataset	Tech.	p-value	Tech.	p-value	Tech.	p-value
mammals	T1, T5	9.73e-19	T1, T5	1.52e-18	T1, T2	0.71
EB	T1, T2	0.09	T1, T2	0.016	T1, T5	3.08e-12
MNIST	T1, T2	6.70e-9	T1, T2	7.24e-14	T1, T2	3.05e-7
FMNIST	T1, T2	0.99	T1, T2	0.15	T1, T4	2.83e-18

Table 5 – p-values for the DEMaP metric.

	Level 2 $\rightarrow$ 1		Level 1 $\rightarrow$ 0	
Dataset	Tech.	p-value	Tech.	p-value
mammals	T1, T5	1.44e-6	T1, T2	0.21
EB	T1, T5	1.02e-13	T1, T5	4.07e-13
MNIST	T1, T2	1.48e-15	T1, T4	8.82e-17
FMNIST	T1, T2	8.56e-21	T1, T2	1.50e-20

Table 6 – p-values for the disparity metric.

C.2 DRILL-DOWN ANALYSIS

Figures 91 and 92 show the all of the results for the drill-down operation demonstrated in the paper. Tables 7, 8, 9, and 10 show the the p-values after running t-test using HUMAP and the better performing technique other than UMAP.

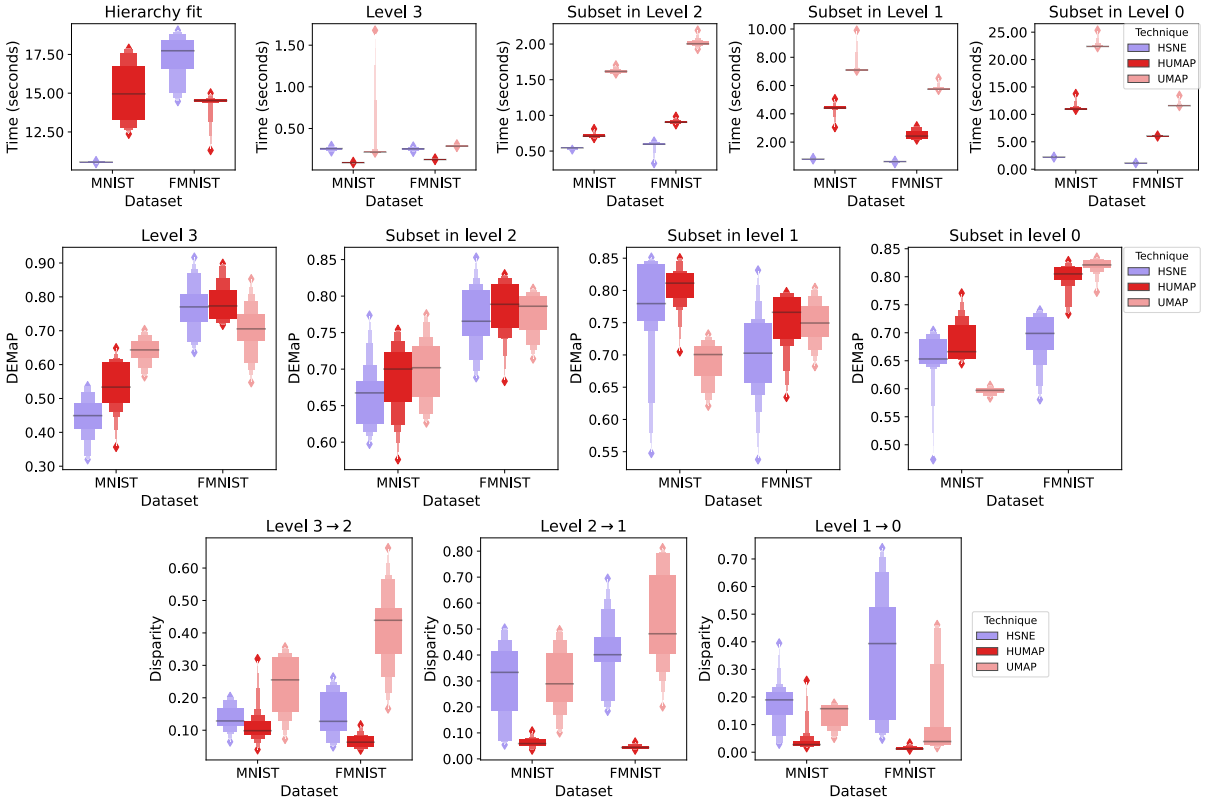


Figure 91 – Results for Time, DEMaP, and Disparity for twenty executions of each HDR technique.

	Hierarchy Fit		Level 3		Subset in Level 2	
Dataset	Tech.	p-value	Tech.	p-value	Tech.	p-value
MNIST	T1, T4	3.19e-9	T1, T2	2.33e-46	T1, T4	1.32e-34
FMNIST	T1, T4	1.12e-12	T1, T4	1.59e-22	T1, T4	1.46e-24

Table 7 – p-values for the running time metric.

	Subset in Level 1		Subset in Level 0	
Dataset	Tech.	p-value	Tech.	p-value
MNIST	T1, T4	0.05	T1, T4	0.05
FMNIST	T1, T4	2.81e-11	T1, T4	2.81e-11

Table 8 – p-values for the running time metric.

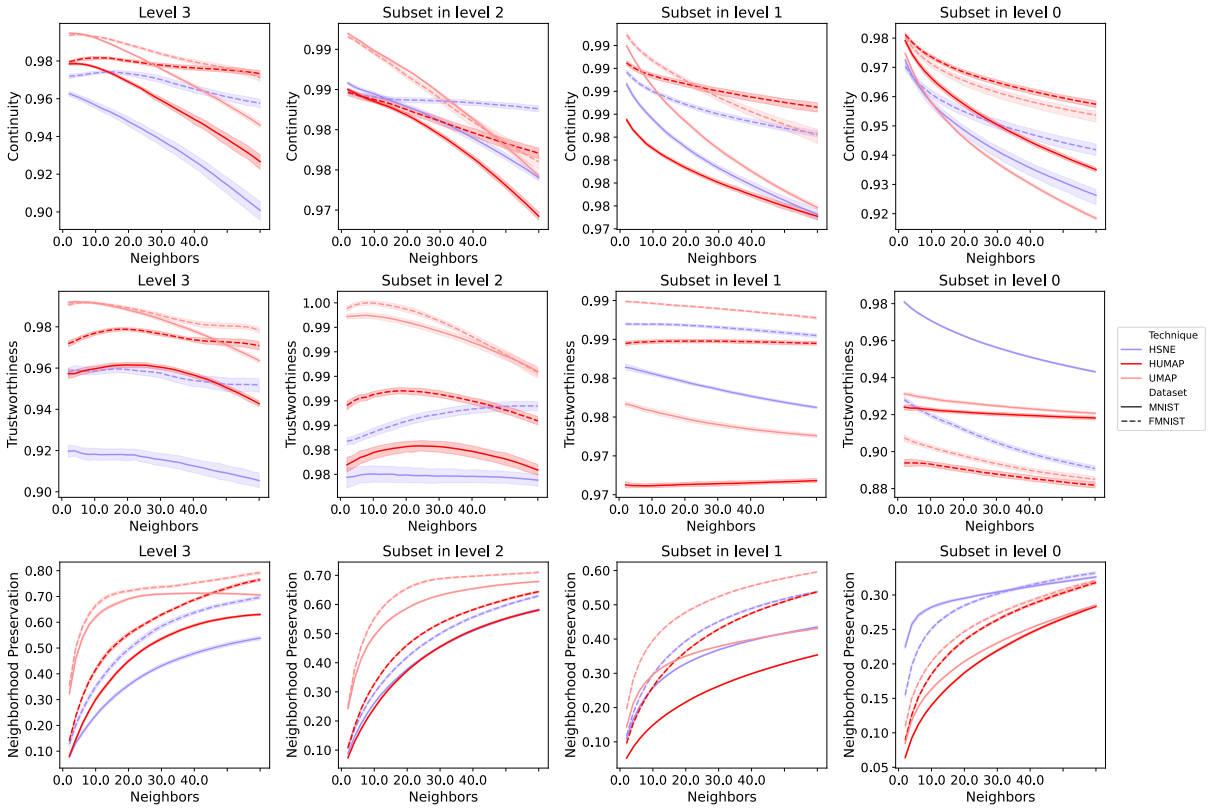


Figure 92 – Results for Neighborhood Preservation, Continuity, and Trustworthiness for twenty executions of each HDR technique.

	Level 3		Subset in Level 2		Subset in Level 1		Subset in Level 0	
Dataset	Tech.	p-value	Tech.	p-value	Tech.	p-value	Tech.	p-value
MNIST	T1, T2	2.17e-6	T1, T2	0.46	T1, T4	0.10	T1, T4	0.05
FMNIST	T1, T4	0.57	T1, T2	0.43	T1, T2	0.96	T1, T4	2.81e-11

Table 9 – p-values for DEMaP metric.

	Level 3 → 2		Level 2 → 1		Level 1 → 0	
Dataset	Tech.	p-value	Tech.	p-value	Tech.	p-value
MNIST	T1, T4	1.28e-5	T1, T2	6.93e-14	T1, T2	0.006
FMNIST	T1, T4	0.12	T1, T4	6.77e-8	T1, T2	4.06e-7

Table 10 – p-values for the disparity metric.

C.3 HIERARCHICAL DIMENSIONALITY REDUCTION WITHOUT PRESERVING THE MENTAL MAP

In the paper and Section C.1, we show that HUMAP loses quality regarding DEMaP metric when projecting whole hierarchy levels. The following results show the same comparison of the paper and Section C.1 by setting the fixing term (θ) to 1. As shown in the figures, by setting free the projection we get higher DEMaP values at the cost of losing the mental map.

Figures 93 and 94 show the projections of all hierarchy levels for the datasets used in the study.

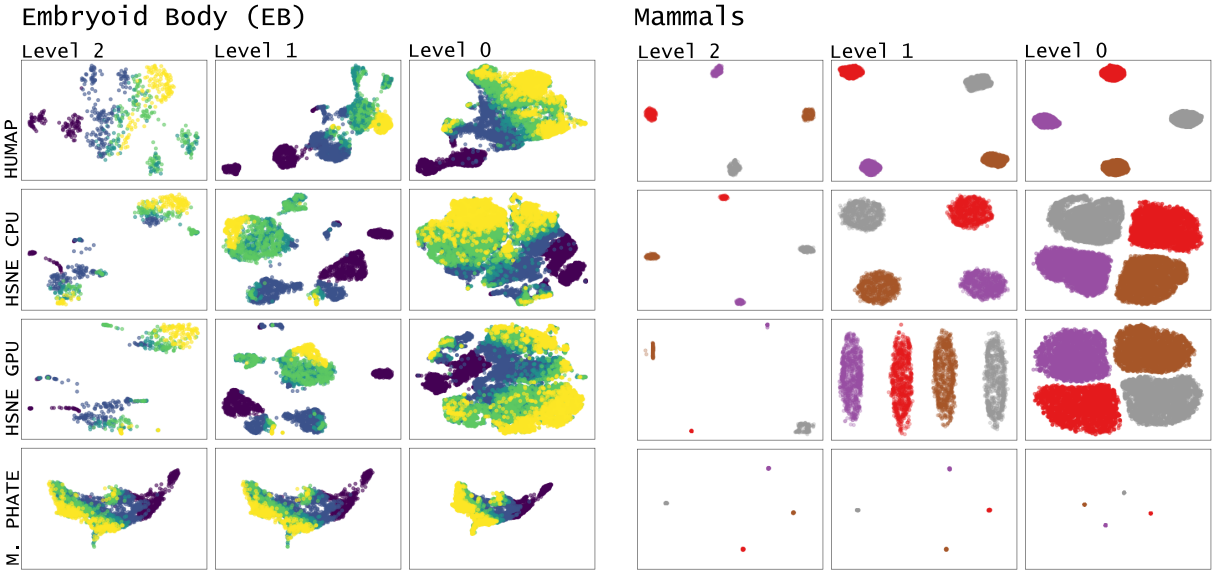


Figure 93 – Projections for Embryoid Body and Mammals datasets.

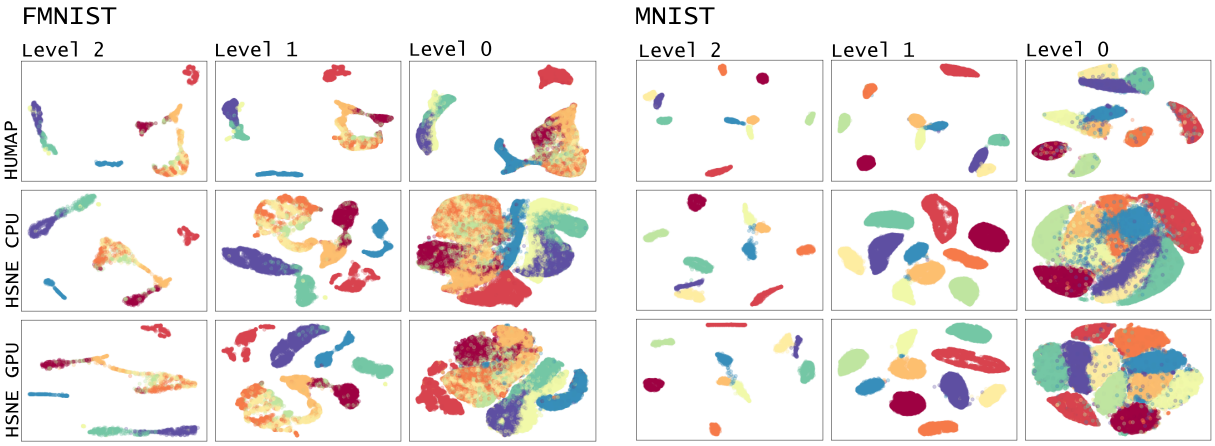


Figure 94 – Projections for FMNIST and MNIST datasets.

Figure 95 shows the results after twenty runs for each hierarchical dimensionality reduction technique,

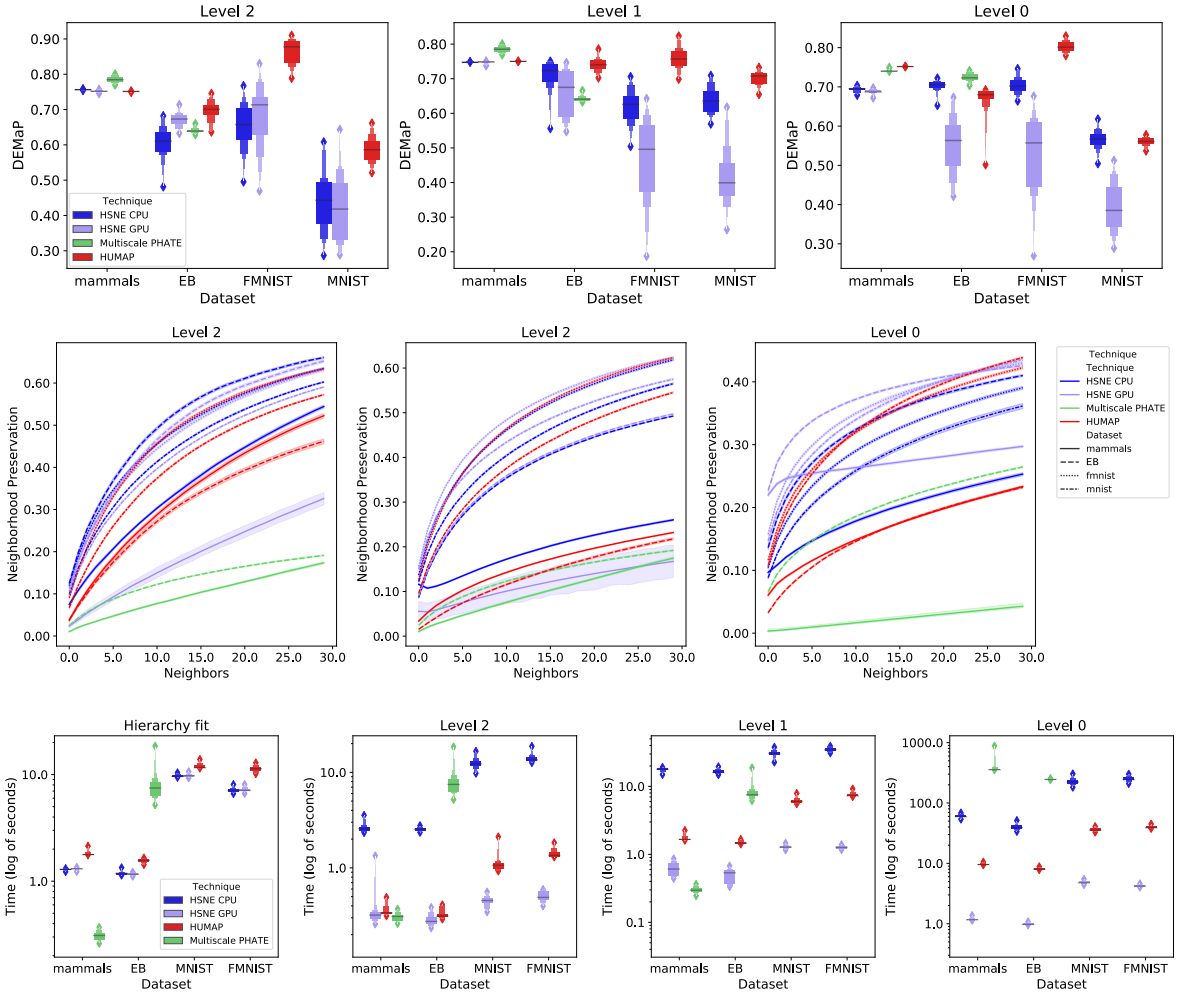


Figure 95 – Numerical analysis for twenty executions of each HDR technique.

C.4 ADDITIONAL LOCAL/GLOBAL ANALYSIS

C.4.1 MENTAL MAP PRESERVATION.

To quantify the mental map preservation, we use Procrustes Analysis on subsequent hierarchy levels. That is, for the hierarchy levels i and $i+1$, we retrieved the points present in both i and $i+1$ and used them to compute disparity between the sets of points. Fig. 96 shows that HUMAP is superior over the other techniques at preserving the mental map (the lower disparity the better). This result confirms the visual representations of **Fig. 7 in Manuscript**, where HUMAP reduces the amount of rotations and variations between hierarchy levels.

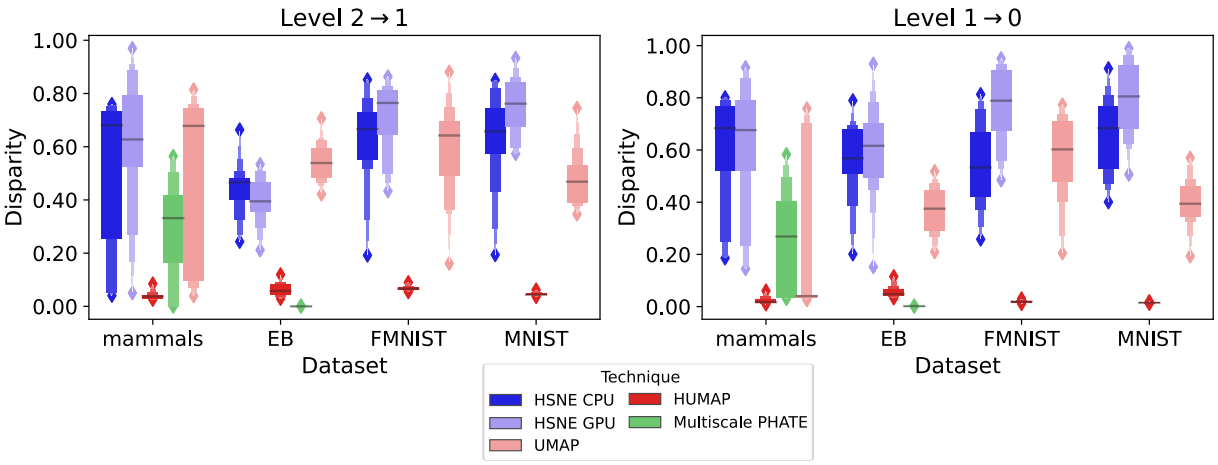


Figure 96 – Mental map preservation measured using Procrustes Analysis. HUMAP decreases the variation when positioning points across hierarchy levels.

C.4.2 CONTINUITY.

Fig. 97 shows the continuity values for varying number of neighbors, which measures the extent to which original clusters are preserved. Except for Multiscale PHATE on *mammals* for level 2, all of the techniques performed well with values greater than 0.9. Here, we can see the expected effect that HUMAP preserves the global structure more than focusing only on cluster formation for the EB dataset—the Fig. 7 illustrates this effect.

C.4.3 TRUSTWORTHINESS.

The results from neighborhood preservation are comparable with Trustworthiness since it measures to what extent the local structure is retained, shown in Fig. 98. As well as for continuity, all of the techniques performed well with values greater than 0.9, with exception for Multiscale PHATE which shines when dealing for continuous datasets and conveying global information. Finally, as commented earlier, HUMAP balances the local

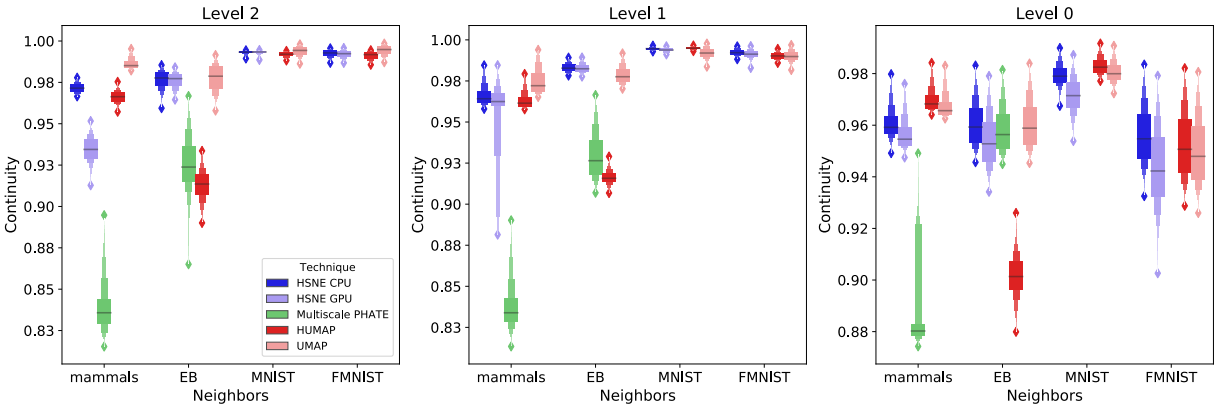


Figure 97 – Continuity values for varying neighborhoods. All of the techniques performed well at preserving clustering structures.

and global structures and provides comparable results with techniques that do not worry about this trade-off.

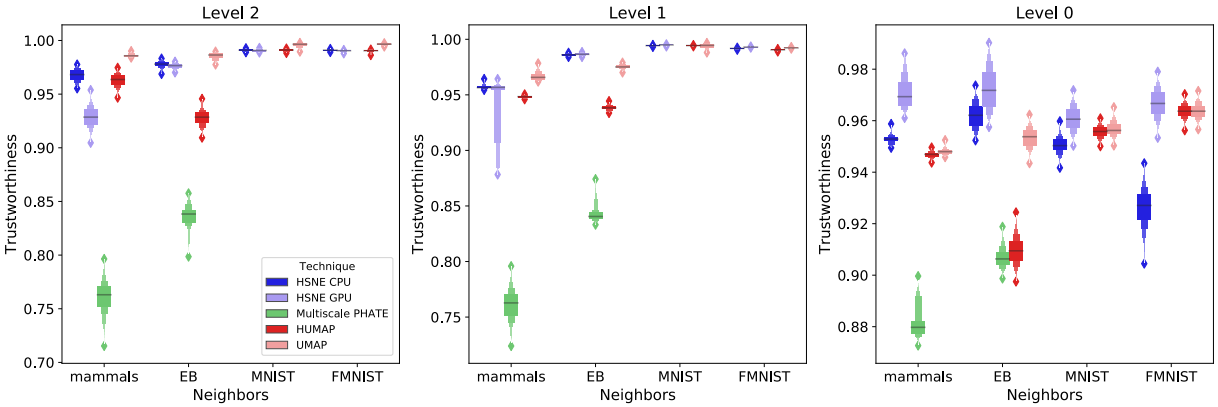


Figure 98 – Trustworthiness values showing similar information to Neighborhood Preservation values.

C.5 RUN TIME ANALYSIS - HUMAP STEPS AND SCALABILITY

In the following figure, we show the runtime execution for fitting a hierarchy using HUMAP for three levels. To measure the time for each step, we vary the size and dimensionality of the corpus dataset containing 100 thousand text embeddings with 768 dimensions.

Note that the "Fit" step consists of the UMAP component for KNN construction and edge strength definition. Clearly, the number of data points influences the running time of the "Fit" and "Neighborhood construction" steps, which can be explained by the KNN operations. Finally, the sampling step using Markov Chain is one of the least demanding steps for the HUMAP algorithm.

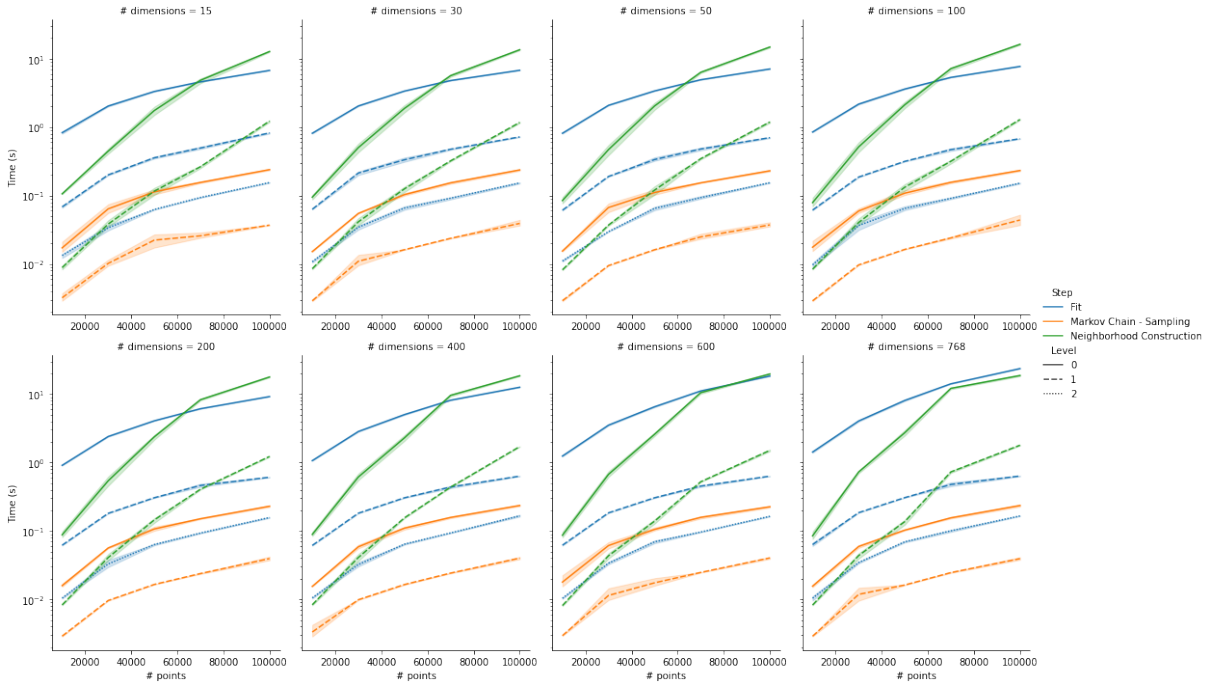


Figure 99 – Scalability analysis for the main steps in HUMAP algorithm.

C.5.1 SCALABILITY

Fig. 100 shows the run-time execution for the hierarchical dimensionality reduction techniques compared in the paper. MPHATE requires a lot of memory to run big datasets, which makes it difficult to analyze high-dimension datasets. This is mainly due to the exact KNN computation used in PHATE for the projection of the datasets in the official implementation[1].

Nevertheless, we experiment with a dataset containing 100k document embeddings and varying dimensionality to understand how it performs compared to HUMAP and HSNE on fitting the hierarchy and transforming the lowest level.

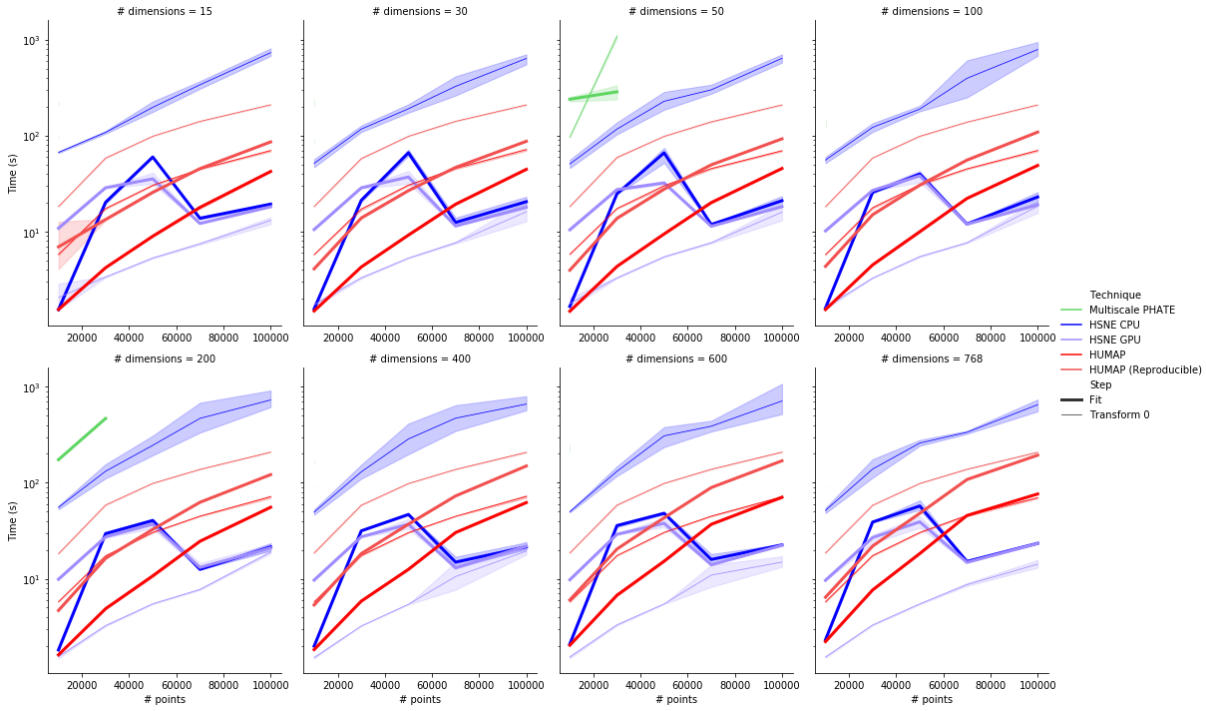


Figure 100 – Scalability analysis for the main hierarchical dimensionality reduction techniques.

Unfortunately, we could not execute Multiscale PHATE for most of the configurations on a 32GB RAM machine.

[1] https://github.com/KrishnaswamyLab/Multiscale_PHATE

C.5.2 REPRODUCIBILITY

The embeddings for generated by HUMAP can also be reproducible at the cost of bigger run-time execution. This is because we cannot use parallelism in a few parts of the Hierarchical (MCMC) and Projection (Cross-Entropy Minimization) steps. Fig. 101 shows the reproducibility of HUMAP compared to HSNE and UMAP. We followed the procedures from Becht et al. (BECHT et al., 2018) to compute reproducibility. For each dataset (MNIST, FMNIST, and Corpus) and algorithm, we store the two vectors of embedded coordinates (x, y). Then, for the same combination of dataset and algorithm, we obtain the embeddings coordinates (x^i, y^i)—we run this second step five times. For each (x^i, y^i), we then compute $(|cor(x, x^i)| + |cor(y, y^i)|)/2$, where $|.|$ denotes the absolute value and $cor(.,.)$ the Pearson correlation coefficient. All the analysis is performed for the top level in the hierarchy.

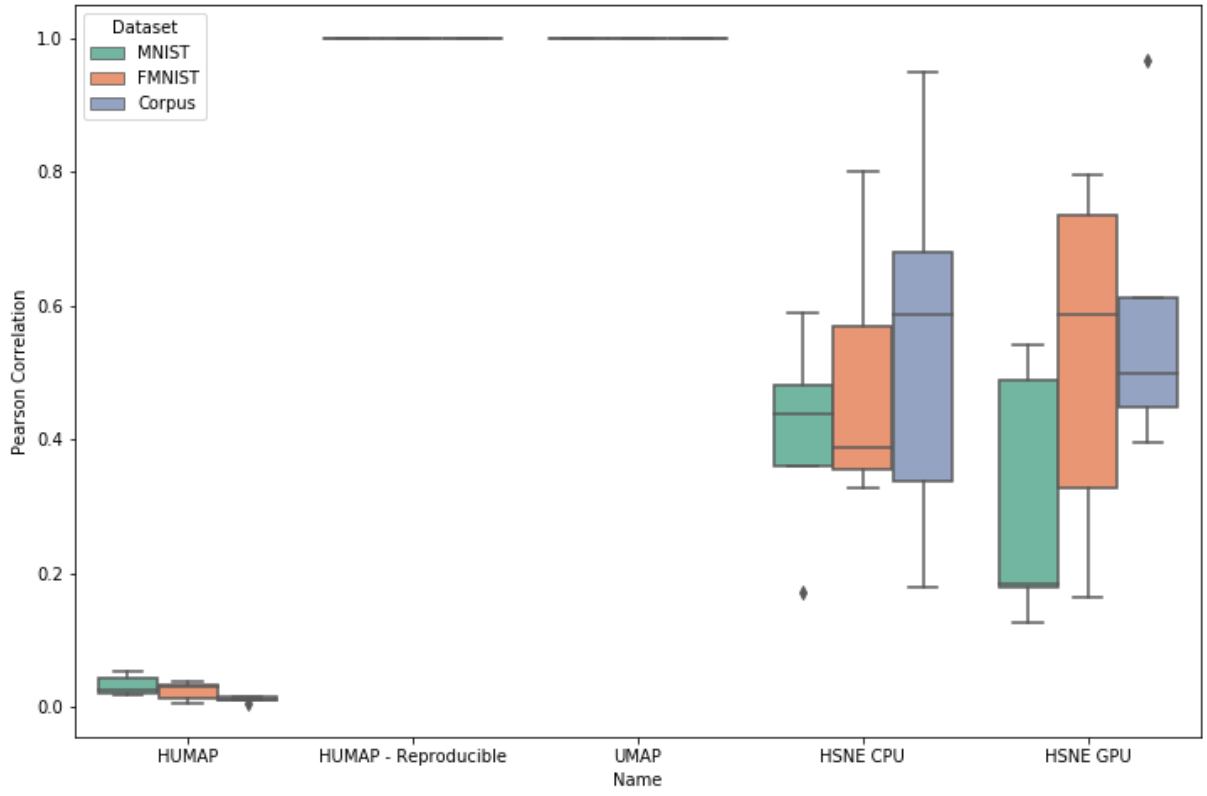


Figure 101 – Results for reproducibility following the procedures of Becht et al. (BECHT et al., 2018).

Regarding time difference, the main drawback consists when transforming data. The following figure shows the runtime distributions (in seconds) for the projection of different hierarchy levels when varying the dataset size from 10k to 100k and dimensionality from 15 to 768. Note that, while our implementation is in C++, the minimization process implemented in UMAP uses parallelism, which explains better performance when compared to HUMAP Reproducible. HUMAP relies on OpenMP for parallelism, which cannot produce

reproducible results when dealing with pseudo-random numbers (used in the minimization process by the Stochastic Gradient Descent algorithm).

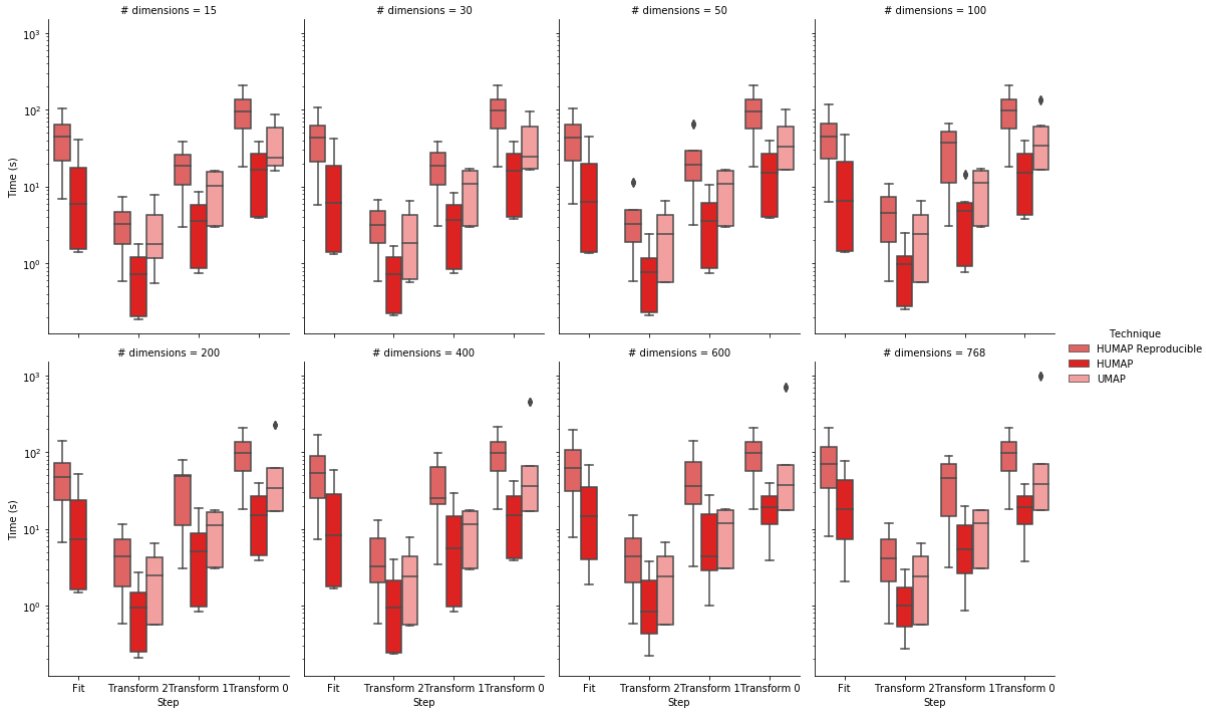


Figure 102 – Run-time distributions for HUMAP when generating reproducible results.

C.6 GENERALIZABILITY

In order to understand more of the HUMAP's generalizability—besides the few case studies we expose in the paper, we conducted a parameter analysis.

C.6.1 *MIN_DIST* AND *N_NEIGHBORS*

The first thing to understand is how HUMAP behaves when varying the "min_dist" and "n_neighbors" parameters that were inherited from UMAP technique.

Figs. 103 and 104 show the same pattern that is present for UMAP (MCINNES; HEALY; MELVILLE, 2018), where augmenting the "min_dist" or "n_neighbors" makes HUMAP focus on global information—yet, the local structures (clusters) are still visible for these datasets.

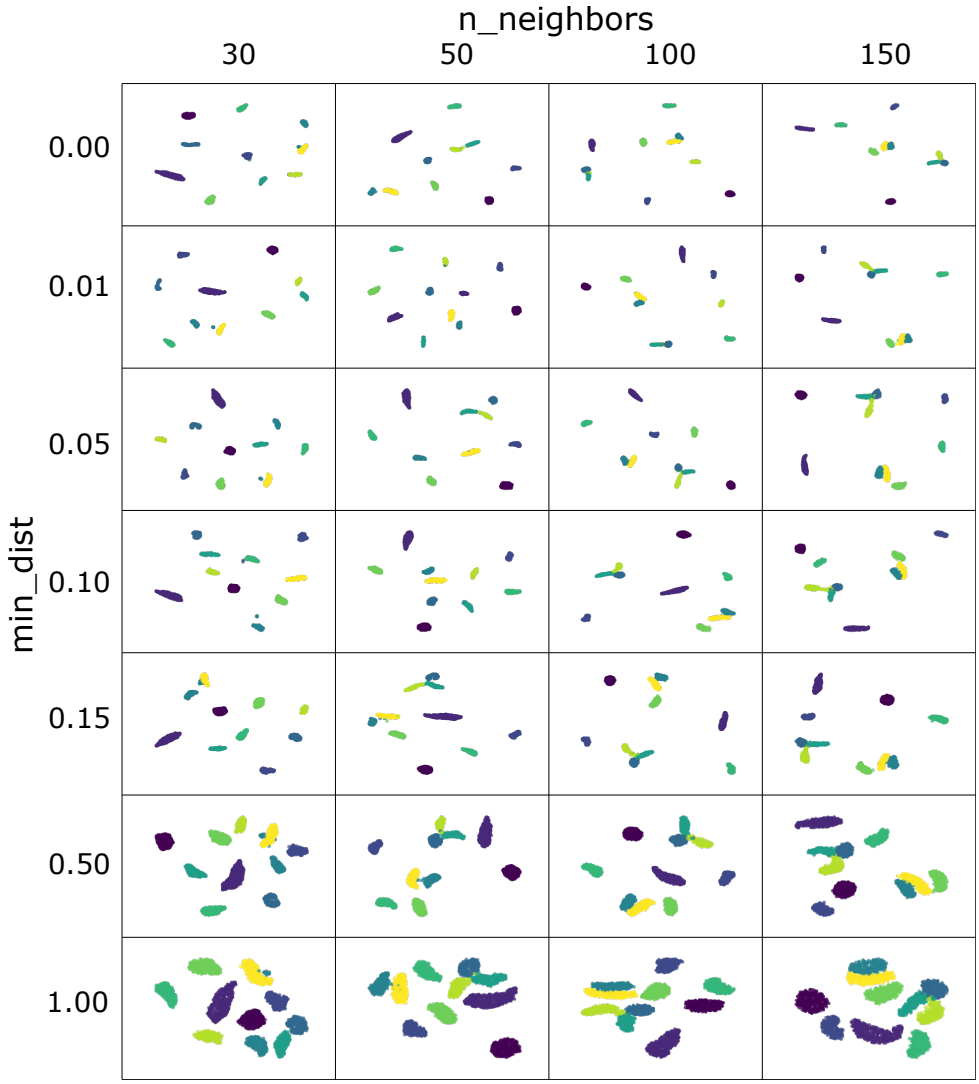


Figure 103 – Embeddings of the MNIST dataset for different values for "min_dist" and "n_neighbors".

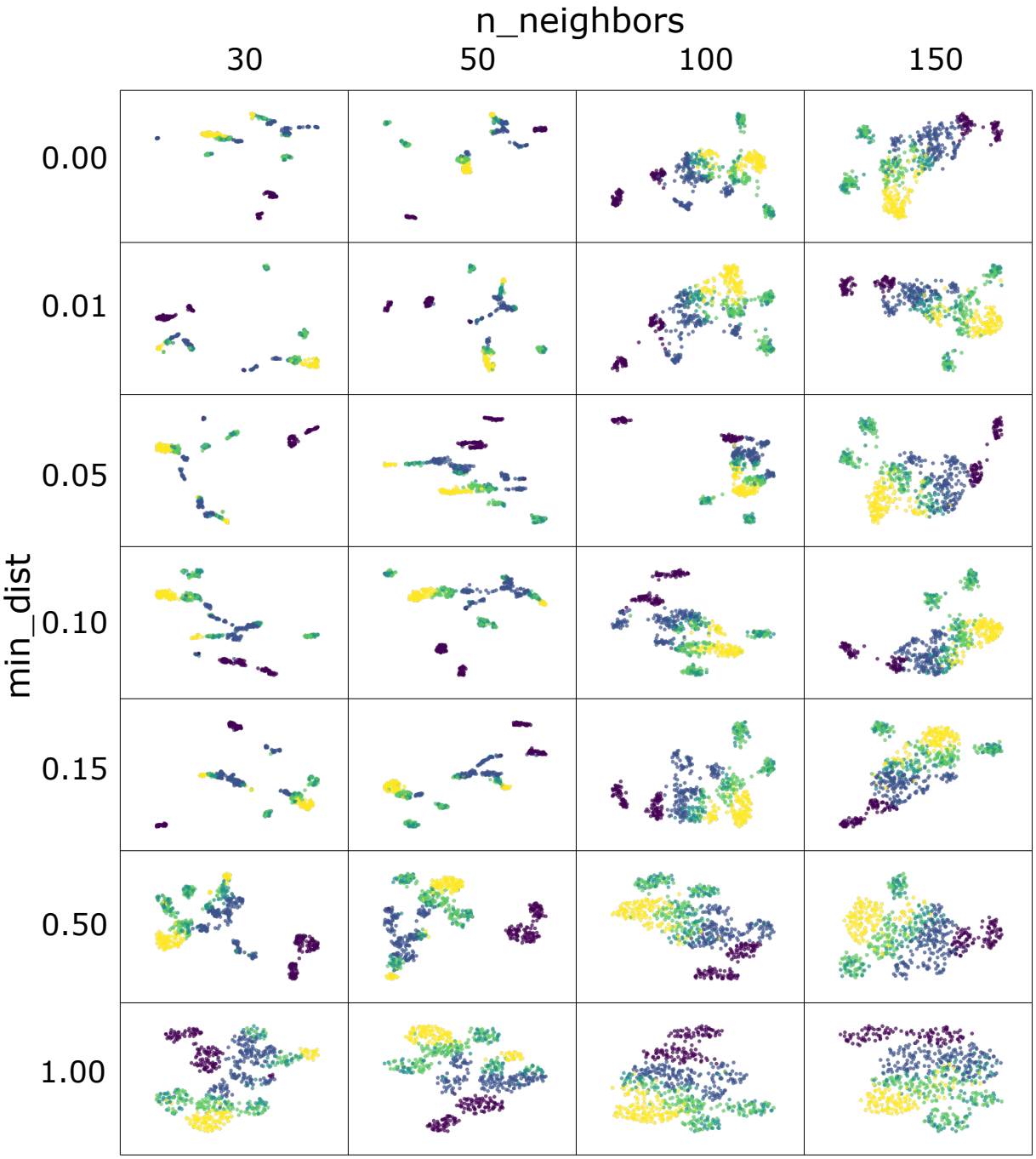


Figure 104 – Embeddings of the Embroyd Body dataset for different values for “min_dist” and “n_neighbors”.

C.6.2 β FOR LOCAL CONTRIBUTION

Another important parameter for HUMAP controls how much local contribution one wants to generate the embeddings. When increasing the value for β (from 0 to 1), the representation neighborhood receives more contribution from local neighbors, which makes HUMAP better regarding metrics for local structures (e.g., Neighborhood Preservation) and worse for metrics that evaluate global structures (e.g., DEMaP). When increasing β up to 1 makes HUMAP preserve more of local neighborhoods in higher hierarchy levels.

Figure 105 shows the behavior of HUMAP as we change β parameter according to the metrics used in the paper. We also compare it against HSNE on GPU.

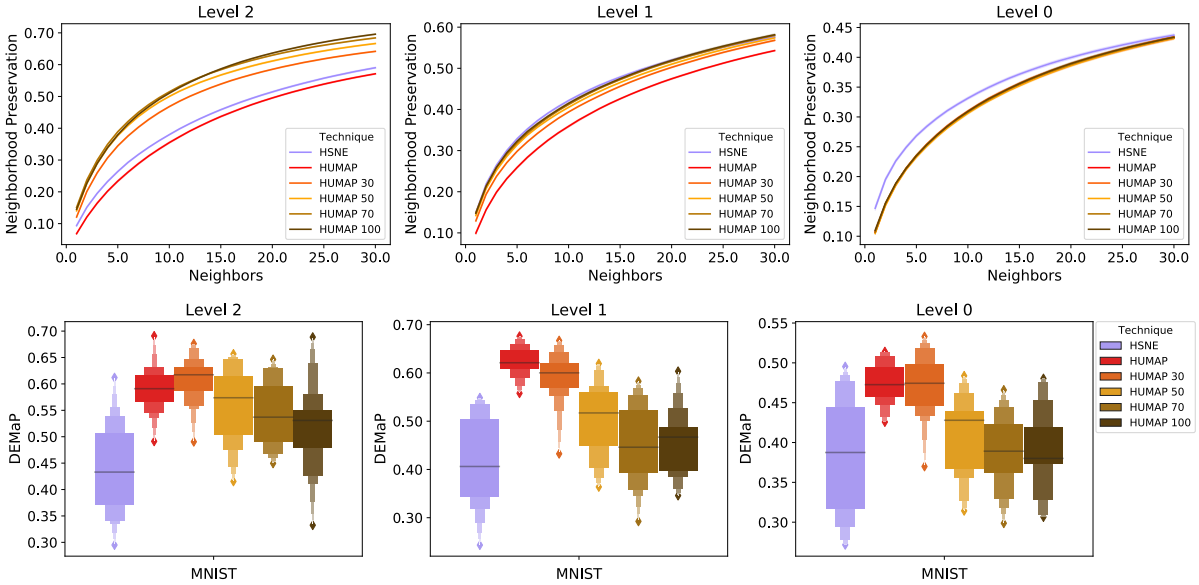


Figure 105 – Metrics for HUMAP with different β values.

Figure 106 shows the projections with best neighborhood preservation for different HUMAP configurations and HSNE on GPU.

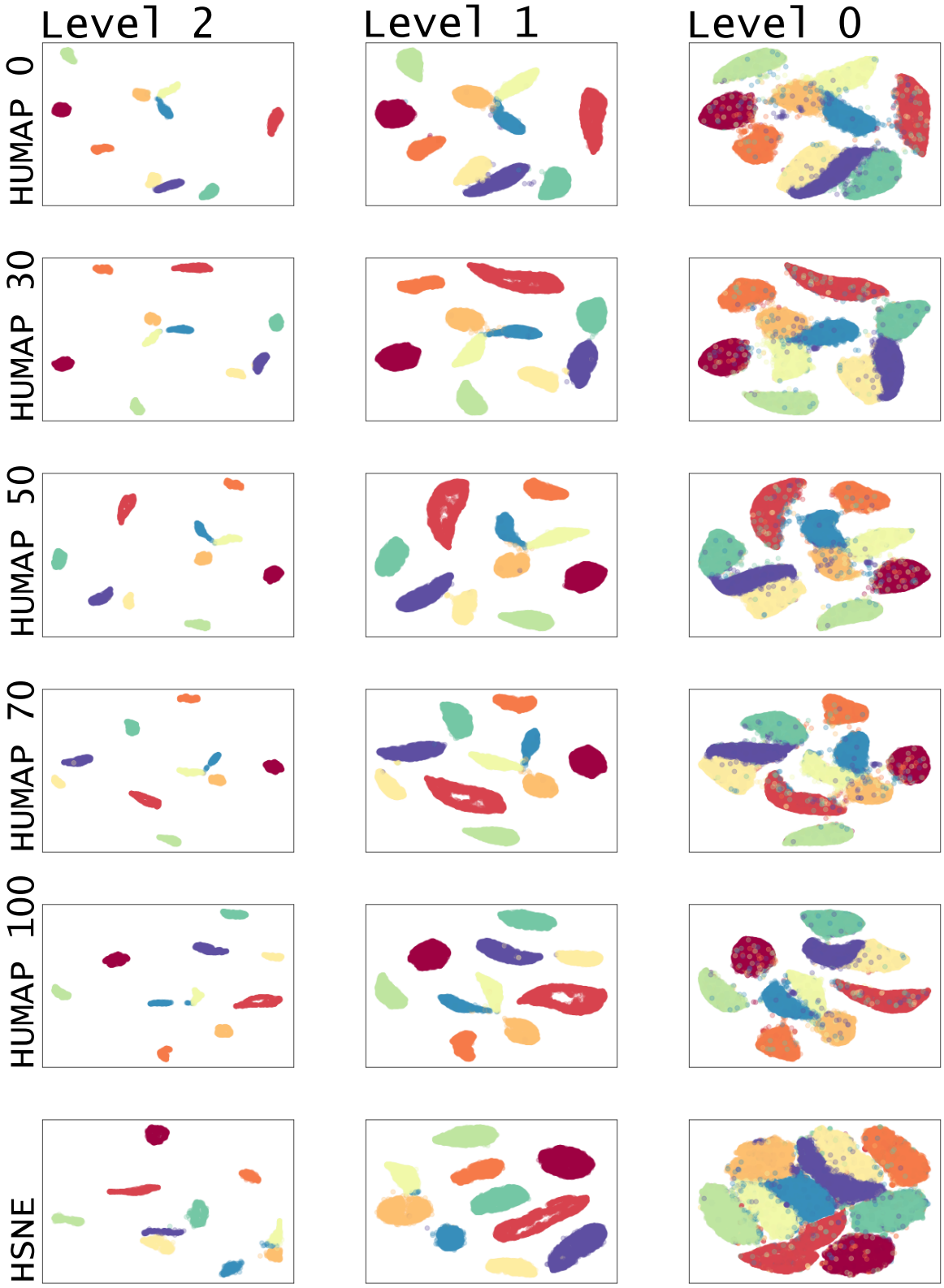


Figure 106 – Projections for HUMAP with different β values.

C.6.3 NUMBER OF DATA POINTS IN EACH LEVEL

Finally, we also analyze the HUMAP's behavior when creating the hierarchy for three levels and varying the percentage of data points on each level. Figs. 107 and 108 show the top-level projections.

The first thing to notice is the number of data points "passed" through higher levels affect the quality of the embedding w.r.t. cluster relationship. That is, when selecting too many data point for higher levels, the representation neighborhood starts to receive much influence from neighboring clusters, which makes them cluster together—see the embedding for LEVEL 1 = (0.1, 0.2, 0.3) and LEVEL 2 = 0.5 in both figures.

Lastly, we advise for a combination of LEVEL 1 $\in$ (0.2, 0.3) and LEVEL 2 $\in$ (0.2, 0.3) as a good starting point. We defined LEVEL 1 = LEVEL 2 = 0.2 as default for HUMAP.

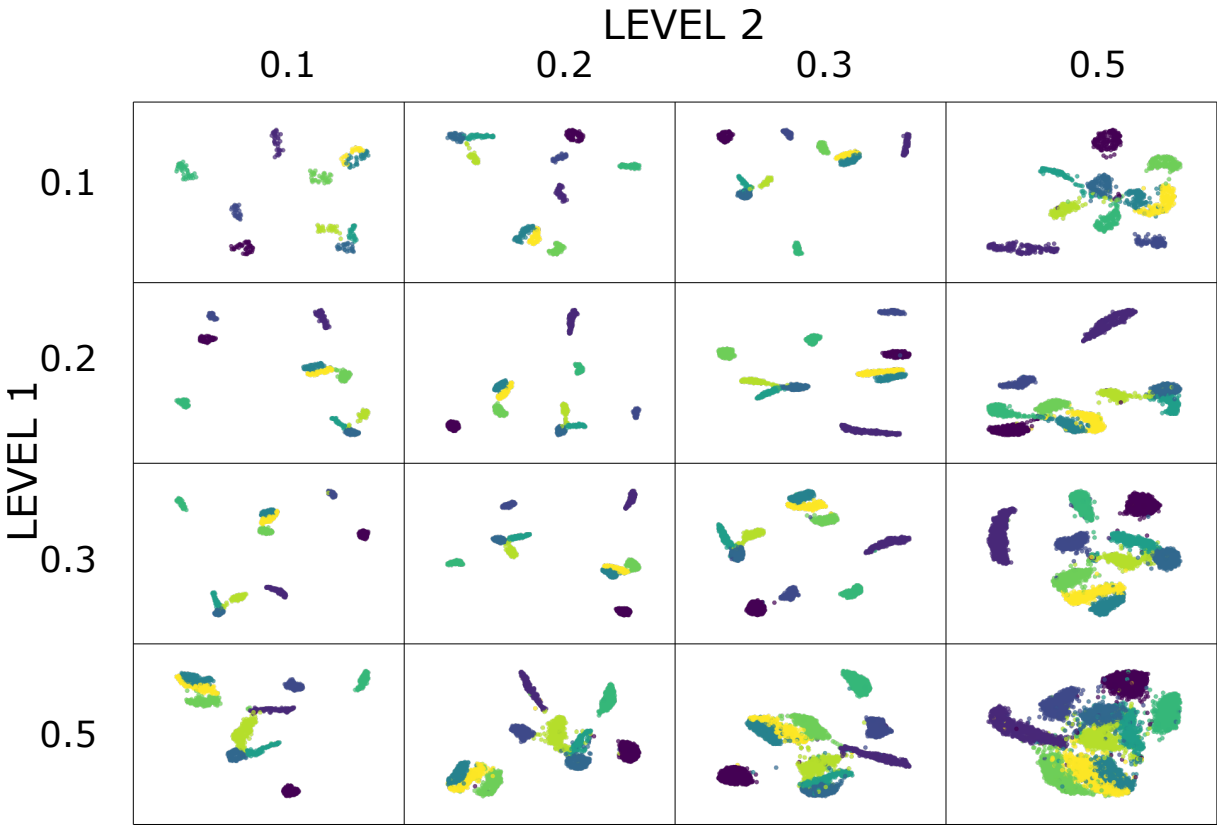


Figure 107 – Embeddings of the MNIST dataset for different values for .

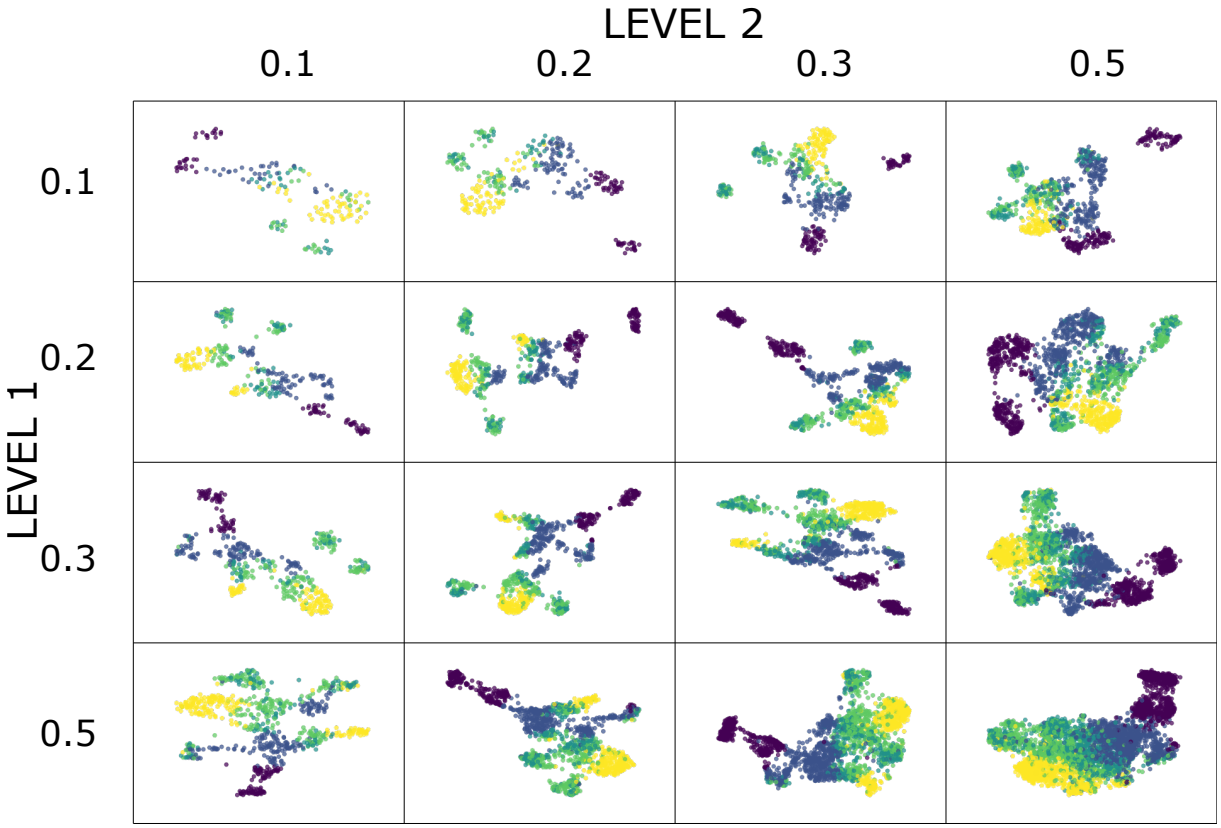


Figure 108 – Embeddings of the Embroyd Body dataset for different values for .

C.7 ADDITIONAL USE CASE - EXPLORATION OF COVID-19 TWEETS

This section aims to investigate the relationship of *tweets* about COVID-19 symptoms in the São Paulo state, Brazil. This dataset first appears in (**??**), where the authors investigate the whole dataset using contrastive analysis. Here, we use HUMAP to explore it using two hierarchical levels, aiming at discovering dominant structures (using the top-level) and detailed information about these dominant structures through interaction. The dataset was acquired from Twitter[ő] by querying COVID-19 symptoms (fever, high fever, cough, dry cough, difficulty breathing, and shortness of breath) retrieved from São Paulo state (Brazil) from March 2020 to August 2020. The authors classified the *tweets* according to their relevancy (relevant or not relevant) to COVID-19 infection. For this case study, we set HUMAP to freely find the embeddings as we drill-down the hierarchy by not projecting the low-levels bases on the higher levels (i.e., setting the fixing term $\theta = 1.0$).

To explore the dataset, we manually defined clusters in the visual space using lasso selection, as shown in Fig. 109(A-B). After associating each data point to a cluster using this procedure, we compute topics (F) and proceed to the second and lowest level of the hierarchy, choosing the desired cluster (e.g., cluster ▭). Then, we also compute the topics for the manually defined cluster of the new hierarchical level.

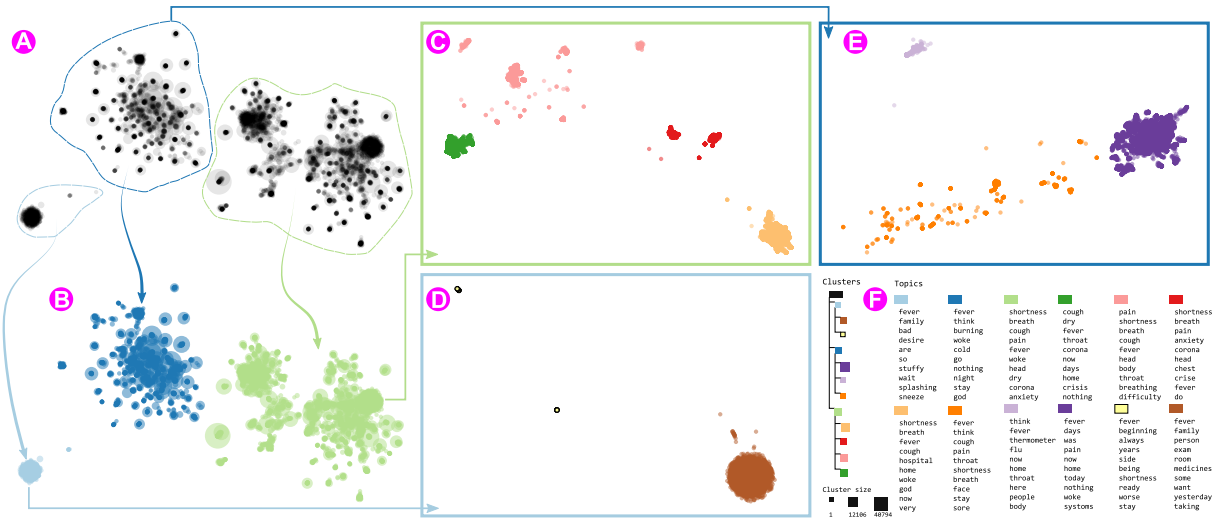


Figure 109 – HUMAP exploration and annotation of a document collection of COVID-19 *tweets*. The top-level hierarchy level shows unlabeled data points and three major structures (A). We annotate these three clusters (B) and compute their topics computed (F). For each cluster in (B), we also project their corresponding level (and final) hierarchy to look for other patterns, annotating the dataset (C, D, E) and computing their topics.

Fig. 109(B) shows three clusters with different characteristics. First, cluster ▭ is very cohesive and dissimilar from the other two dominant structures. Second, due to

visual proximity, cluster and share some information. Finally, these clusters (mainly and) contain various substructures that need further investigation. The topics for these three major clusters further explain their organization in the visual space (F). That is, cluster presents important terms related to respiratory problems that COVID-19 infection may cause, such as `stuffy` nose, `sneeze`, or `splashing`. Other important terms in this cluster can indicate *tweets* about individuals waiting (`wait`) for COVID-19 `exams` or are associated with `desire` to `sneeze`, which supports a hypothesis that this cluster corresponds to individuals worrying about symptoms that are not well-associated with the disease. Clusters and are associated with well-known COVID-19 symptoms. Cluster shows three terms in the topic associated with fever: `fever`, `burning`, and `cold`. These words describe the way individuals feel while worrying about COVID-19 infection or even describing the symptoms after a (possibly) positive test. Other terms such as `think`, `woke`, and `night` might be associated to phrases describing experiences with COVID-19 symptoms, such as: "I **think** I have **fever**", "Today I **woke-up** in the middle of the **night burning** in **fever**". Lastly, cluster is associated with dry cough and shortness of breath. The terms for this cluster in Fig. 109(F) show that the *tweets* talk about these symptoms while the term `anxiety` could be the cause of shortness of breath (SMIRNI; LAVANCO; SMIRNI, 2020).

We proceed to cluster to investigate its substructures and retrieve more information about the *tweets* associated with it. Then, we define the major structures and compute the topics for the clusters, as shown Fig. 109 (C) and (F). Here, there are a few interesting patterns. The first one is that the previous hierarchy level sufficiently gives an overview of the data organization since the topic for cluster encodes most of the information expressed in these sub-clusters. The second and most interesting aspect is the global relationship among these structures apparent in the embedding. The topics retrieved from each local structure explains this aspect. The leftmost cluster () shows terms related to `dry cough`, `throat` pain, and a few other important terms. Cluster has a relationship with cluster , which also adds `difficult breathing`, `body pain`, and headache ("pain in the **head**", using a direct translation from Portuguese). In cluster , `shortness of breath`, `anxiety`, `crisis`, and breast become more important. These symptoms might be easily confused with anxiety crisis, a common problem during COVID-19 pandemic (BONI et al., 2020). The last cluster () seems to be the most related to COVID-19 symptoms, showing the majority of term: `shortness of breath`, `fever`, and `cough`.

Analyzing the substructures of cluster , we define three clusters as shown in Fig. 109 (E). As well-defined by the higher-level cluster, all data points refer mainly to the COVID-19 symptom of fever. However, there are a few characteristics that might explain the differentiation of these clusters. For example, cluster present terms related to `cough`, `shortness` of `breath`, which do not appear in the other clusters. Lastly, cluster has an interesting characteristic since its topic suggest that individuals are commenting about

the period in which they experience fever: `fever`, `days`, `was`, `today`, and `home`.

The most cohesive cluster of top-level (▭) and its two subclusters (▭ and ▭)revealed by drilling-down the hierarchy show tweets that seem to talk more about daily aspects than truly COVID-19 symptoms.

In this case study, our technique reveals with little effort (looking at top-level structure) aspects of the classified COVID-19 dataset that a considerable number of *tweets* may not indicate relevant information. Further that, we can also augment the trust in this analysis by expanding the cluster of interest.