Reorderable Matrices

Andrej Knaus and Esma Karic

26 May 2025

Abstract

Reorderable matrices are powerful tools for revealing structure in tabular data. Rearranging rows and columns can expose patterns, clusters, and outliers, and help with information interpretability. This survey explores the historical development of physical reorderable matrices, including Jacques Bertin's work and its modern reconstructions, as well as a range of algorithmic techniques, such as seriation, spectral ordering, hierarchical clustering, biclustering, and multidimensional scaling. Each method is discussed using a set of defined criteria. The survey also reviews key software tools for interactive matrix reordering.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reorderable matrices are data visualization tools that allow rows and columns of a matrix to be rearranged to reveal patterns and outliers, support clustering, and improve the interpretability of complex datasets. By changing the placement of cells without altering the underlying values, reorderable matrices help to uncover latent structures that may otherwise be hidden in a fixed arrangement. These matrices are especially useful in domains dealing with high-dimensional, relational, or noisy data, such as bioinformatics, finance, and network analysis.

The idea of reorderable matrices is rooted in earlier techniques such as sorting and seriation, which existed before the modern form of the reorderable matrix [Liiv 2010]. The physical origins of the reorderable matrix date back to the mid 1960s, when French cartographer Jacques Bertin developed a mechanical matrix reoordering tool for visually exploring tabular data.

Figures 1.1 and 1.2 show a clear example of how reordering can reveal hidden structures. The data comes from a dataset created by Jacques Bertin [Bertin 1981]. It shows which public services (such as schools, doctors, or police stations) are available in a number of rural French townships. The matrix shown in Figure 1.1 is scattered and unordered, which making it difficult to see meaningful patterns and groupings. After reordering the rows and columns, clear blocks begin to appear, as shown in Figure 1.2. These blocks show which townships have or do not have certain services. This simple example shows how changing the ordering without changing the data can make important patterns easier to see.

This survey paper provides a comprehensive overview of reorderable matrices, beginning with a discussion of the origin of the physical reorderable matrix, followed by a review of algorithmic techniques used to compute effective reorderings. Several use cases across different domains are then presented, illustrating how reorderable matrices can help extract valuable insights from large amounts of raw data. Finally, modern tools which support interactive matrix reordering are surveyed.

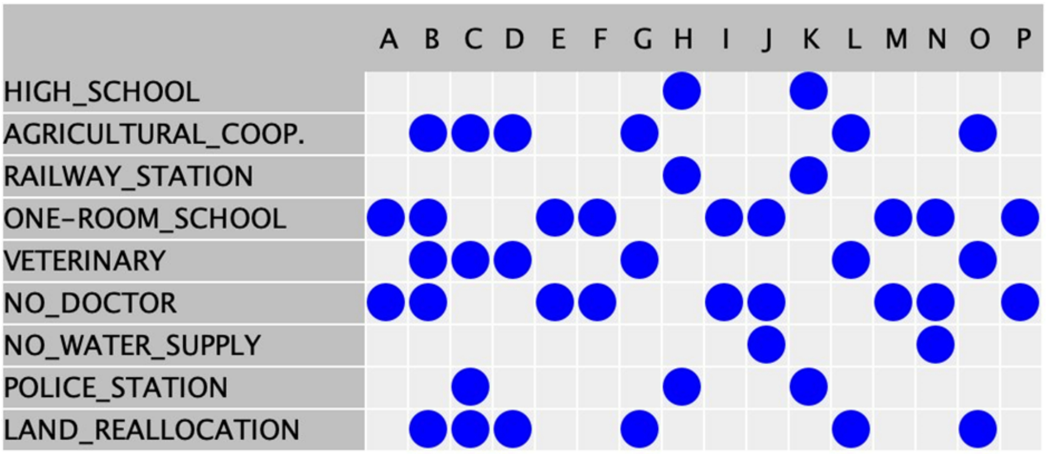

Figure 1.1: Bertin's township dataset before reordering. [Screenshot taken by Andrej Knaus.]

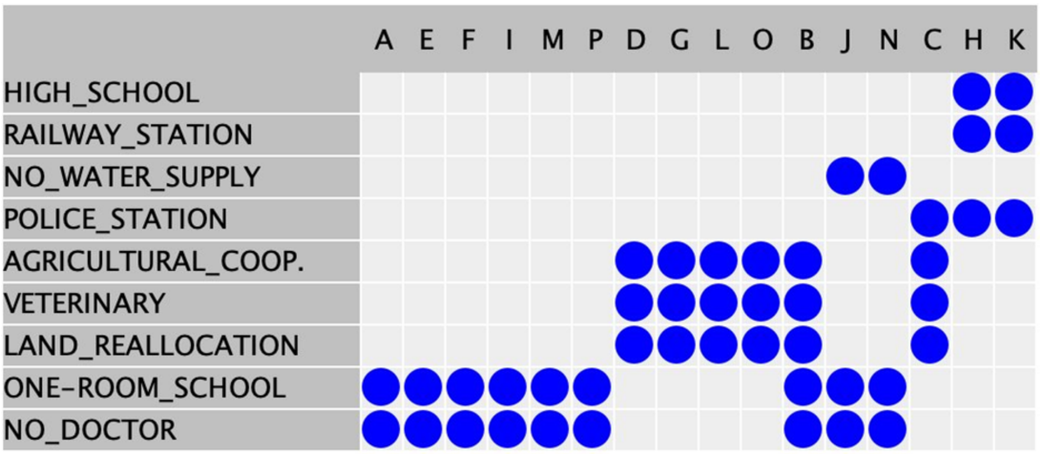

Figure 1.2: Bertin's township dataset after reordering. Groupings and patterns are now clearly visible. [Screenshot taken by Andrej Knaus.]

Chapter 2

Physical Reorderable Matrices

Before digital visualization tools, the idea of reordering matrices to reveal structure was explored through physical, tangible systems. These early tools allowed researchers to interact directly with data using movable elements, encouraging hands-on pattern discovery. Among the pioneers of this approach was Jacques Bertin, whose physical matrix design laid the groundwork for modern reorderable visualization techniques.

2.1 Bertin's Dominos

The earliest and most recognizable iteration of the physical reorderable matrix was that of French cartographer Jacques Bertin as part of his broader exploration in his work on Graphical Semiology [Bertin 1983]. He built five different versions of his device, which he called Dominos, over a period of 10 years (1970–1980) [Perin et al. 2015]. They came in small, medium, and large sizes [Perin et al. 2014b] and featured individual plastic cells that were shaded or textured to represent varying data values, lighter shades for lower values, darker for higher ones. Each cell was inserted into the grid and could be repositioned using a rod mechanism. The mechanism was also used to lock and unlock each row and column for reordering, which allowed for rearranging the data in such a way that exposed meaningful patterns and groupings. One of the original Dominos is shown in Figure 2.1.

Bertin's process involved assembling a tabular dataset, transforming values into bins (0–10, 10–20, etc.), and constructing a physical matrix where each cell represented a binned value range. Reordering was then performed to expose trends, outliers, or meaningful groupings within the data. This interactive, hands-on process focused on discovering patterns by visually rearranging the data, rather than relying on formal statistical methods. When a meaningful configuration was achieved, the matrix was photographed or photocopied to preserve the results for interpretation or presentation.

2.2 The DIY Matrix

A notable modern reconstruction of Bertin's matrix was exhibited at the IEEE VIS 2014 conference in Paris [Team 2015]. Called the DIY Matrix, it was built using laser-cut plywood and engraved domino tiles, as can be seen in Figure 2.2. The matrix was constructed with over 4400 individual parts. Magnetic caps were used to switch out visual encoding schemes.

Physical reorderable matrices remain valuable educational tools in modern data visualization. They highlight the importance of exploratory analysis, offering a way to directly engage with data, help reveal patterns and insights that might otherwise go unnoticed.

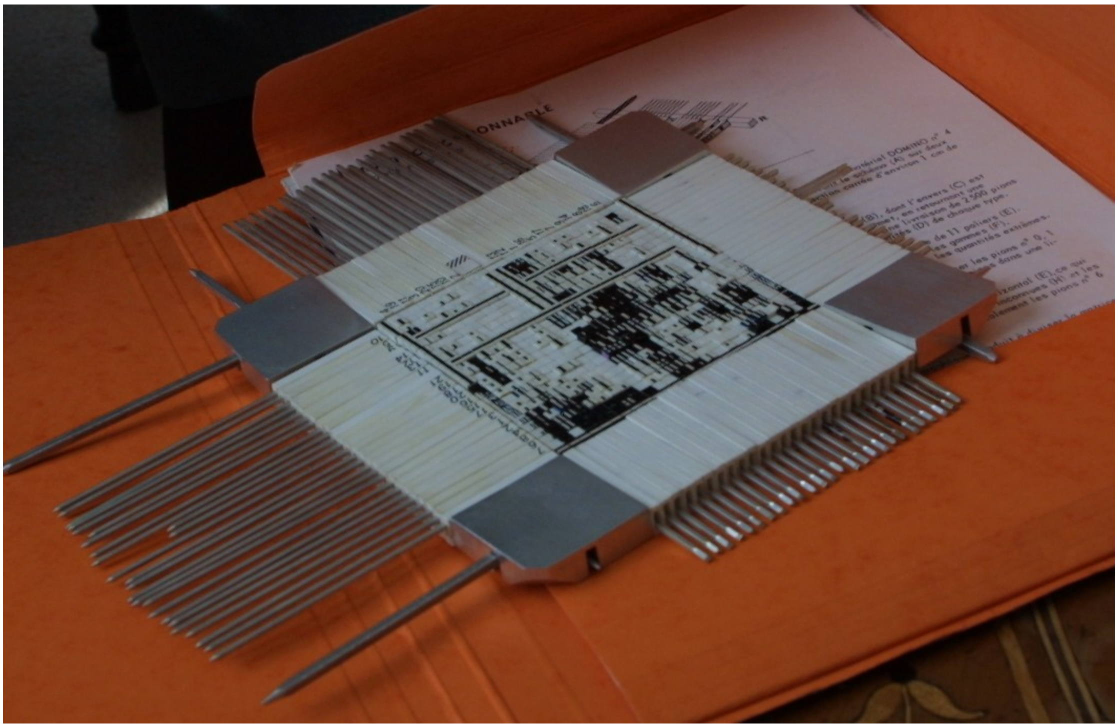

Figure 2.1: One of Bertin's original Domino matrices. [Photo taken by Jean-Baptiste Labrune [Labrune 2004]. Used under the terms of a CC BY-SA 2.0 licence.]

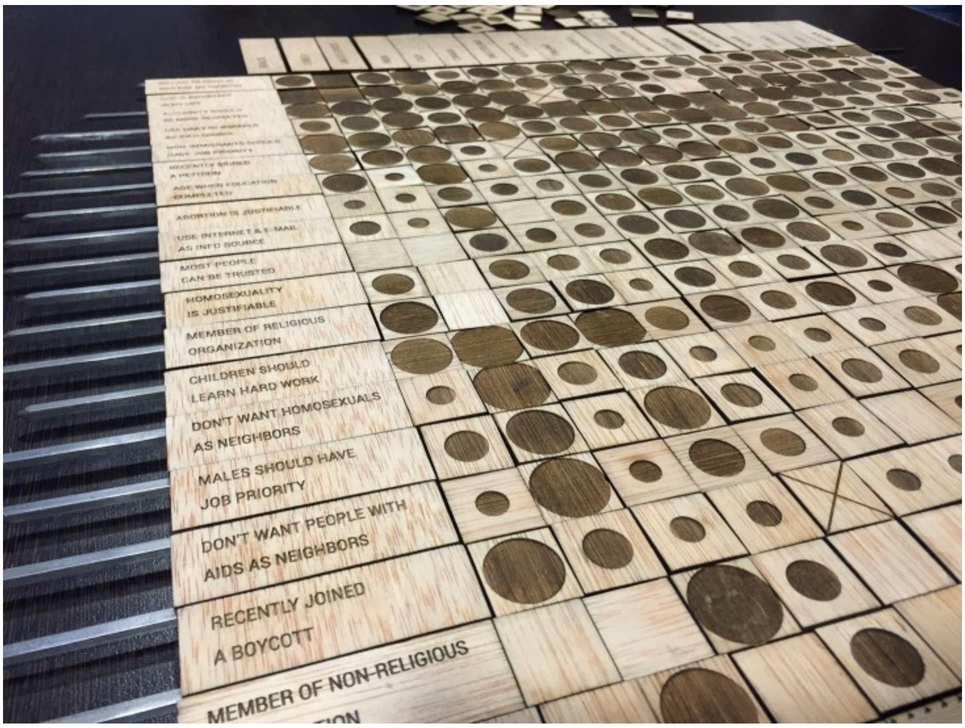

Figure 2.2: The DIY Matrix reconstruction on display at IEEE VIS 2014. [Photo taken by Robert Kosara [Kosara 2014]. Used under the terms of a CC BY-SA 4.0 licence.]

Chapter 3

Algorithmic Solutions

Reorderable matrices use various forms of algorithmic techniques in order to make patterns, clusters, or gradients more noticeable. These techniques aim to reduce disorder and enhance interpretability by optimizing the arrangement of rows and/or columns. Although no single method is universally best, each provides different advantages depending on the dataset, the symmetry of the matrix, and the user's goals.

3.1 Seriation

Seriation is one of the most traditional approaches for matrix reordering. It consists of rearranging the rows or columns based on similarity or continuity in order to reveal gradients or progression in the data. Figure 3.1 illustrates the general workflow of seriation. The process begins with a similarity or dissimilarity matrix and then applies an optimization criterion to define a target ordering. A suitable algorithm is selected to find a permutation that best satisfies the objective. Finally, this new order is applied to the matrix, revealing an interpretable structure that may uncover trends or relationships not visible in the original layout.

Some popular seriation methods include:

- Optimal Leaf Ordering (OLO): Starts with a hierarchical clustering dendrogram and rearranges its leaves (rows or columns) to find an ordering that minimizes the distance between neighboring objects [Hahsler et al. 2008].
- 2-Sum Optimization : Calculates the sum by multiplying the similarity between objects with the square of the difference in their ranks.
- Traveling Salesperson Problem-based methods (TSP): Treats the ordering task as a TSP, with rows or columns as nodes and their differences representing the distances. Solving the TSP approximates the best linear ordering [Hahsler et al. 2008].

These techniques are especially effective in highlighting continuous changes or gradual transitions in the data, such as temperature profiles, gene expression over time, or user preferences in recommendation systems. They are often used in exploratory data analysis to reveal latent structure without imposing rigid assumptions.

3.2 Spectral Methods

Spectral methods rely on the construction of a graph Laplacian derived from a similarity or adjacency matrix, where the structure of the graph is encoded in matrix form. By analyzing the eigenvectors of this Laplacian, particularly the Fiedler vector (the eigenvector corresponding to the second smallest eigenvalue), a meaningful ordering can be derived.

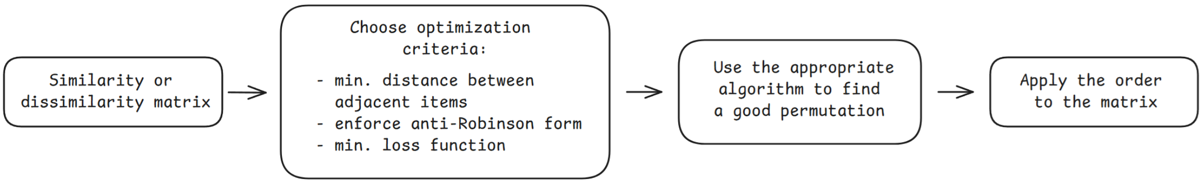

Figure 3.1: Seriation methods typically proceed in four steps. [Diagram drawn by Andrej Knaus]

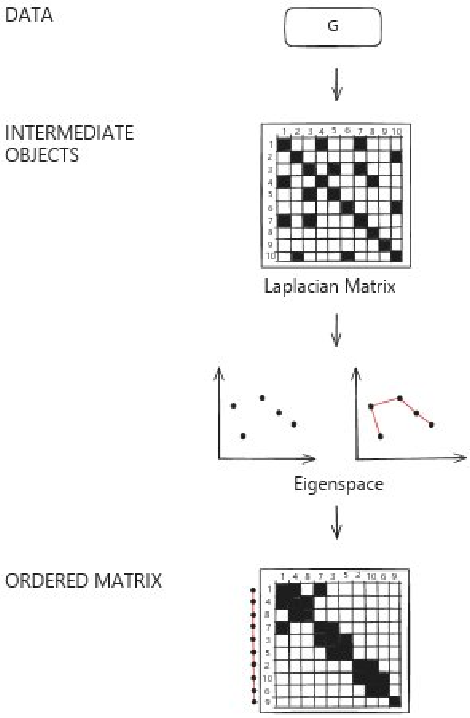

Figure 3.2: The spectral reordering process. [Redrawn by Andrej Knaus from the original by Behrisch et al. [2016].]

The Fiedler vector projects high-dimensional data onto a one-dimensional axis, revealing linear or sequential patterns in the data. Spectral methods are especially useful for preserving global relationships and are well-suited for datasets with inherent ordering or gradual transitions.

Figure 3.2 provides a visual summary of this process. A matrix is first transformed into a Laplacian matrix, then mapped into an eigenspace through its eigenvectors. The resulting Fiedler vector is used to reorder the original matrix so that similar nodes or features are placed close together, often revealing clustered or block-like structures that were not obvious in the original configuration.

Spectral ordering is commonly used in fields such as network analysis, computational biology, and dimensionality reduction, where uncovering structure in large, interconnected datasets is essential.

3.3 Agglomerative Hierarchical Clustering (AHC)

Agglomerative Hierarchical Clustering (AHC) is another widely used technique. It treats each data point as its own cluster and then merges the two most similar clusters at each step until all points are grouped into a single cluster. The result is a tree-like structure known as a dendrogram, which represents the nested grouping of the data.

By traversing the leaf order of the dendrogram, we can derive a sequence that arranges the matrix

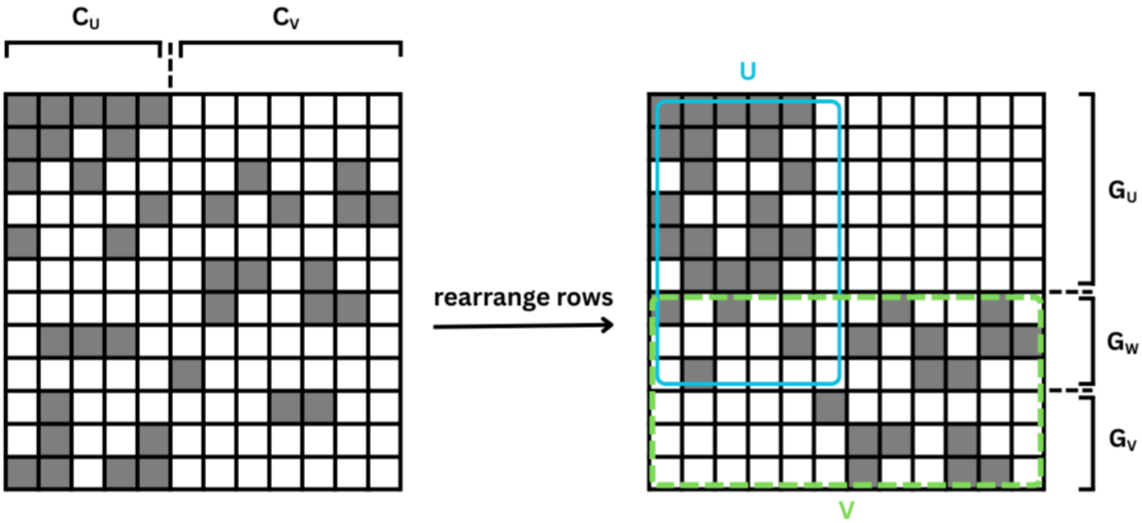

Figure 3.3: The BiMax algorithm. [Redrawn by Andrej Knaus from the original by Prelić et al. [2006].]

such that similar rows or columns are grouped together. This reordering often reveals block structures, modular patterns, or hierarchical relationships in the data that may be hidden in the original layout.

Different linkage strategies define how the distance between clusters is calculated during the merging process:

- *Single linkage (nearest neighbor):* Uses the smallest distance between elements of two clusters.
- *Complete linkage (furthest neighbor):* Uses the largest distance between elements of two clusters.
- *Average linkage:* Uses the average pairwise distance between all elements of two clusters.

Each linkage method leads to a different hierarchical structure, which in turn affects the resulting matrix reordering. The choice of linkage can significantly influence how clusters are formed and should be selected based on the nature of the data and the intended analysis.

3.4 Bi-Clustering

Bi-clustering simultaneously groups both rows and columns to uncover submatrices that exhibit consistent internal patterns. Unlike traditional clustering, which operates along one dimension at a time, bi-clustering identifies local structures that span subsets of both rows and columns. This approach is particularly effective for detecting localized patterns and has been implemented in algorithms such as BiMax, spectral co-clustering, and the Plaid model.

Bi-clustering often produces checkerboard-like patterns in heatmaps, where blocks of high or low values align across both rows and columns. These blocks reveal areas of local correlation or similarity, making bi-clustering a powerful tool for applications such as gene expression analysis, market segmentation, and collaborative filtering.

Figure 3.3 illustrates the concept. The matrix on the left shows an initial random ordering. The BiMax algorithm begins by splitting the columns into two groups based on a reference pattern. It then reorders the rows to group those that match only one column subset, both subsets, or the other. This results in three distinct row groups, which intersect with the two column groups to form smaller, coherent submatrices.

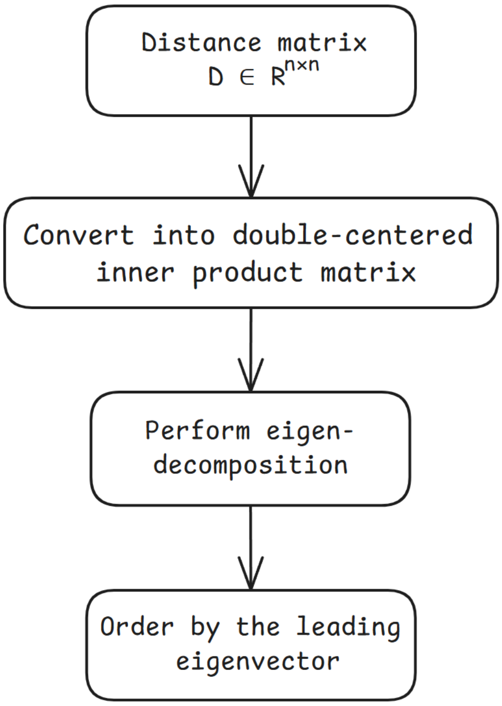

Figure 3.4: The classical MDS process. [Diagram drawn by Andrej Knaus]

3.5 Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a dimensionality reduction technique that projects high-dimensional data into a lower-dimensional space while preserving the pairwise distances between data points as closely as possible [Wikipedia 2025]. In the context of reorderable matrices, MDS is often used to reduce the data to a single dimension, with the resulting 1D coordinates providing an ordering for the rows or columns.

There are two main types of MDS:

- *Classical MDS* uses eigen decomposition to preserve Euclidean distances between points.
- *Non-metric MDS* focuses on preserving the rank order of distances rather than their exact values.

In the classical MDS process (see Figure 3.4) the distance matrix is first converted into a double-centered inner product matrix. Then, an eigen-decomposition is performed and the matrix is reordered based on the leading eigenvector.

Compared to techniques like Principal Component Analysis (PCA), which aim to maximize variance along orthogonal axes, MDS is more flexible in terms of the type of input it accepts. It is particularly effective when working with precomputed distance or similarity matrices, making it a suitable choice for many matrix reordering applications [StackExchange 2011].

3.6 Discussion

Each reordering method discussed in this chapter offers distinct advantages depending on the type of data and the user's analytical goals:

- *Seriation*: works well when the goal is to highlight gradual changes or sequential trends, such as temporal progression. It tends to place similar items close together, making patterns more visible, though it can become computationally intensive on large datasets.
- *Spectral Methods*: derive orderings that reflect the overall structure of the data. They are effective

	Seriation	*Spectral*	*AHC*	*Bi-Clustering*	*MDS*
Goal	Minimize adjacent distances	Reveal structure via eigenvectors	Group similar items in nested clusters	Find local clusters	Preserve pairwise distances
Ordering	Distance matrix	Eigenvector	Leaf order	Row and column subsets	Distance preservation
Use Cases	Gradients, time-like structure	Continuous patterns	Clusters, groups	Gene expression, recommender systems	Spatial data, gradual patterns
Methods	TSP, Anti-Robinson, OLO	Spectral sorting (Fiedler vector)	Single, complete, average linkage	BiMax, spectral co-clustering	Classical, non-metric MDS
Strengths	Captures progression	Structure unseen by clustering	Reveals global clusters, groups	Finds local patterns	Continuous ordering
Weaknesses	Expensive for large N	Noise sensitive	May hide local clusters	Lacks global context	May distort structure, limited interpretability

Table 3.1: Comparison of different matrix reordering and clustering techniques.

at revealing broad, global patterns but can be sensitive to noise, which may distort the results.

- *Agglomerative Hierarchical Clustering (AHC)*: constructs a hierarchy by iteratively merging the most similar clusters. It is well-suited for uncovering block-like or modular structures, though it may overlook smaller, localized groupings.
- *Bi-Clustering*: is ideal when patterns emerge across both rows and columns. By clustering them simultaneously, it can highlight compact, highly correlated submatrices that other methods may miss. However, it may not effectively capture overarching trends in the data.
- *Multidimensional Scaling (MDS)*: projects data into a low-dimensional space while preserving pairwise distances. It is useful for detecting smooth, continuous structures, but its results can be difficult to interpret when the underlying distance information is noisy or imprecise.

Choosing the appropriate method depends on the nature of the data, the use case, and desired focus, be that on global structure, local clusters, or smooth transitions (see Table 3.1).

Chapter 4

Use Cases

Reorderable matrices are used in many fields to transform chaotic or dense data representations into meaningful, interpretable structures.

In the field of bioinformatics, particularly genomics, datasets often contain thousands of genes across hundreds of samples, resulting in visually overwhelming raw matrices. Applying techniques such as hierarchical clustering or bi-clustering allows researchers to identify co-expressed gene clusters, helping to uncover meaningful biological pathways [Kong et al. 2016].

In network science, large graphs are often difficult to interpret using traditional node-link diagrams due to visual clutter and complexity. Representing them instead as reordered adjacency matrices, organized by community structure or node centrality, reveals tightly knit clusters, bridges, and isolated components with much greater clarity. For example, in social networks, this approach can uncover communities of individuals who interact frequently, making the underlying structure of the network more interpretable.

In the field of finance, correlation matrices of hundreds of asset returns are often dominated by noise, which can obscure underlying patterns. Reordering these matrices using hierarchical or spectral methods helps highlight distinct block structures, aiding in the identification of asset classes or groups of co-moving instruments, such as technology stocks or government bonds. This reorganization supports better portfolio analysis and risk assessment.

Beyond these primary domains, reorderable matrices have also been applied in areas such as education, survey analysis, sports analytics, and human-computer interaction research, where visual clarity is essential for interpreting relationships. Their strength lies in the ability to expose structures that are not immediately apparent, using visual reordering to reveal patterns. This technique offers a valuable balance between algorithmic structure and human interpretability, enabling more intuitive data exploration.

Chapter 5

Tools for Matrix Reordering

Four existing matrix reordering tools were reviewed for this survey: Matrix Explorer, Bertifier, Clustergrammer, and PermutMatrix, each offering distinct features for matrix visualization and reordering. These tools were assessed using the following criteria:

- *Reordering Methods*: What algorithms or techniques are available (e.g., hierarchical clustering, spectral ordering, seriation)?
- *Interactivity*: What interactive features support matrix exploration?
- *Customization*: To what extent can users adjust visual encodings, labels, color schemes, or clustering parameters?
- *License*: Is the tool open-source, free for academic use, or commercially restricted?
- *Hosting Options*: Can the tool be run locally, in a browser, or does it require server-based deployment?
- *Technology Stack*: What programming languages and frameworks are used (e.g., JavaScript, Python, Java)?

5.1 Matrix Explorer

Matrix Explorer was developed by students Marco Garzia, Weinan Huang, Christin Seifert, and Wolfgang Wallisch [Garcia et al. 2010]. It is a simple Java implementation of Bertin's Reorderable Matrix, supporting both manual and algorithmic reordering (the 2D Sort algorithm [Mäkinen and Siirtola 2000]), and offering basic data encodings. However, it uses a custom format for importing and exporting data, which makes it difficult to test the same dataset across different tools. Matrix Explorer is shown in Figure 5.1 with Bertin's hotel dataset.

5.2 Bertifier

Bertifier [Perin et al. 2014a] is a web application designed for user-driven structuring and exploration of tabular data. Users can input their own data via a CSV file or choose from a set of provided sample datasets. The tool supports the reordering of rows and columns through drag-and-drop, and it also allows users to "glue" rows or columns together to move them simultaneously. Bertifier offers a variety of visual encodings for data values and includes options for customization. Visualizations can be exported as either SVG or PDF files. The tool is released under a BSD 3-Clause License. Figure 5.2 shows Bertifier with Bertin's hotel dataset.

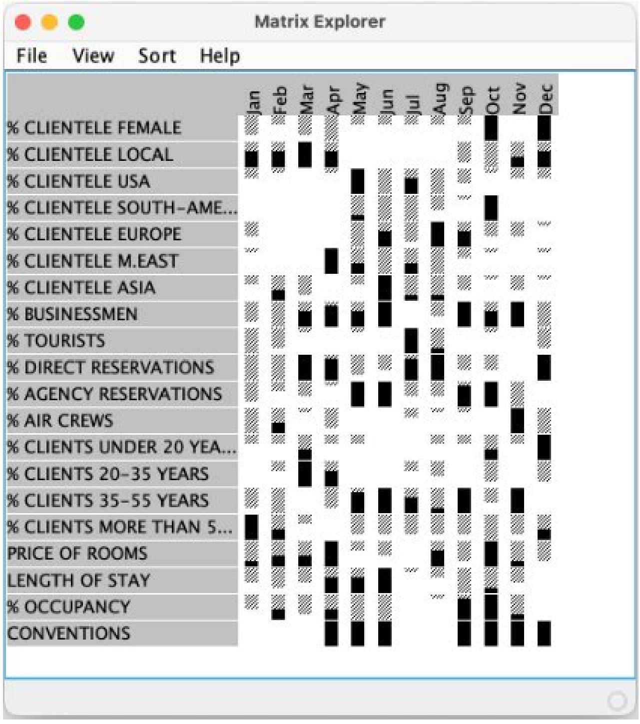

Figure 5.1: Matrix Explorer with Bertin's hotel dataset. [Screenshot taken by Andrej Knaus.]

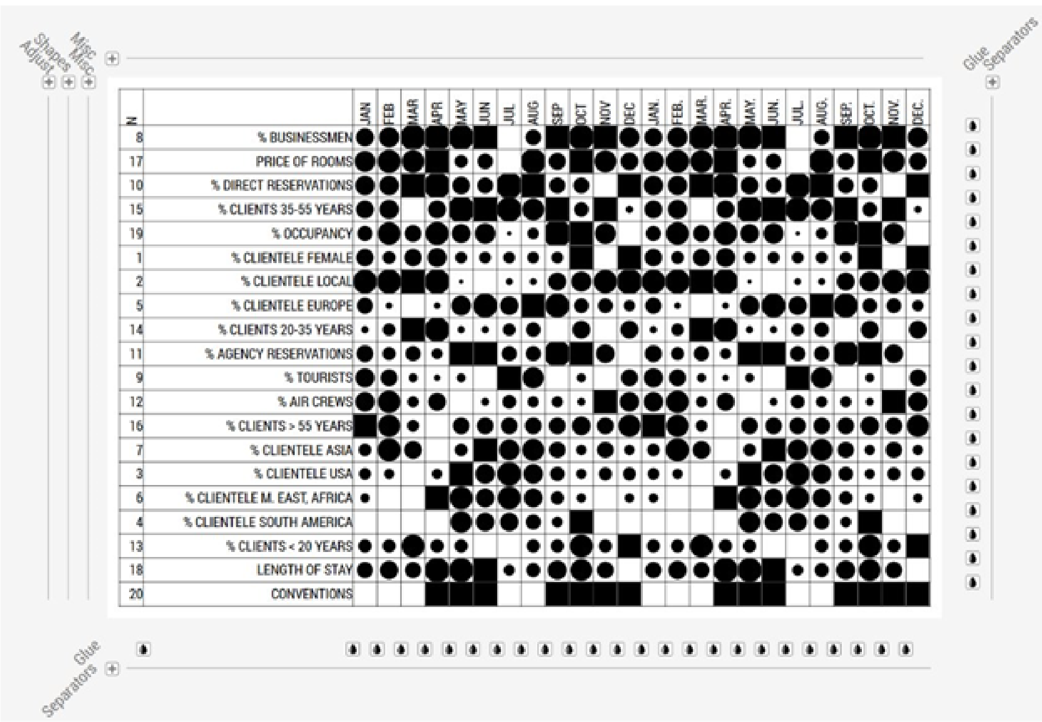

Figure 5.2: Bertifier with Bertin's hotel dataset. [Screenshot taken by Andrej Knaus.]

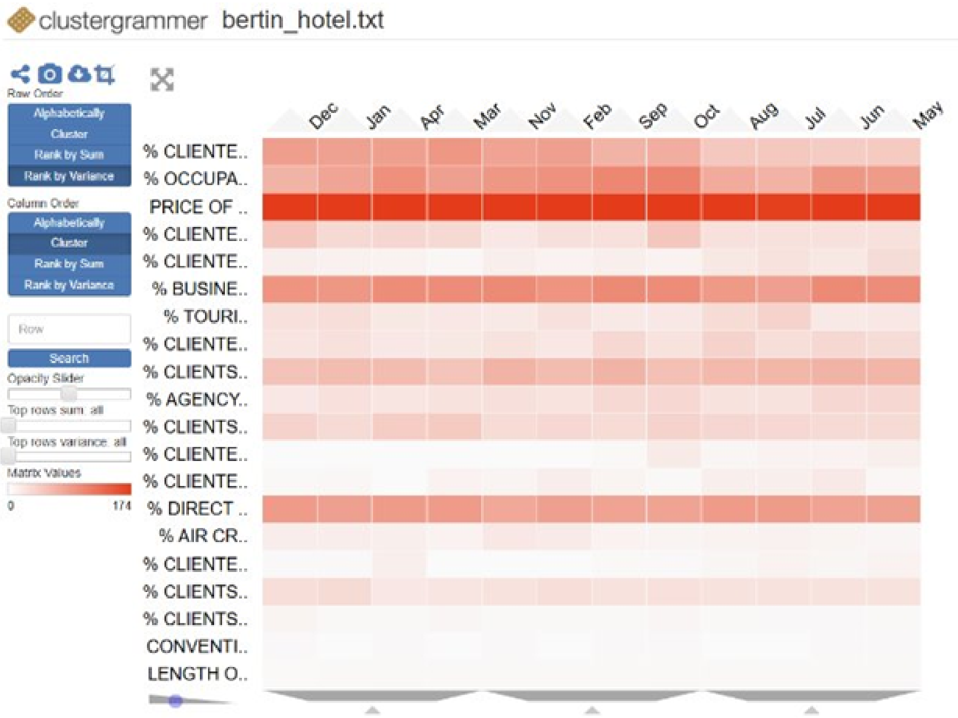

Figure 5.3: Clustergrammer with Bertin's hotel dataset. [Screenshot taken by Andrej Knaus.]

5.3 Clustergrammer

Clustergrammer [Eddy et al. 2018] is an interactive web-based tool for generating hierarchically clustered heatmaps. It supports algorithmic reordering via Agglomerative Hierarchical Clustering (AHC) and allows users to specify both the distance metric and linkage method. Users can control the ordering of rows and columns using options such as alphabetical, clustered, ranked by sum, or ranked by variance. Figure 5.3 shows Clustergrammer with Bertin's hotel dataset.

Clustergrammer also provides additional features such as search functionality, adjustable cluster sizes, zoomed-in views, and similarity matrices for both rows and columns. Visualizations can be exported as PNG or SVG files. The tool is built using JavaScript and Python, and it can be deployed in various ways—including web deployment, self-hosting, or integration as a Jupyter widget. Clustergrammer is released under the MIT License.

5.4 PermutMatrix

PermutMatrix [Caraux and Pinloche 2005] is a Windows desktop application focused on the visual exploration and clustering of numerical matrices. It supports both Agglomerative Hierarchical Clustering (AHC) and seriation methods, and is implemented in Delphi. The software is provided as freeware for scientific purposes. Figure 5.4 shows PermutMatrix with Bertin's hotel dataset.

PermutMatrix accepts input in TSV (Tab-Separated Values) format. Users can reorder rows and columns, flip or swap them, and visualize dendrograms, which update dynamically when changing the linkage method. It also offers the ability to visualize similarity matrices, select subsets of the dataset, and export them as new datasets. Visualizations can be exported in JPG, PS (PostScript), or BMP formats.

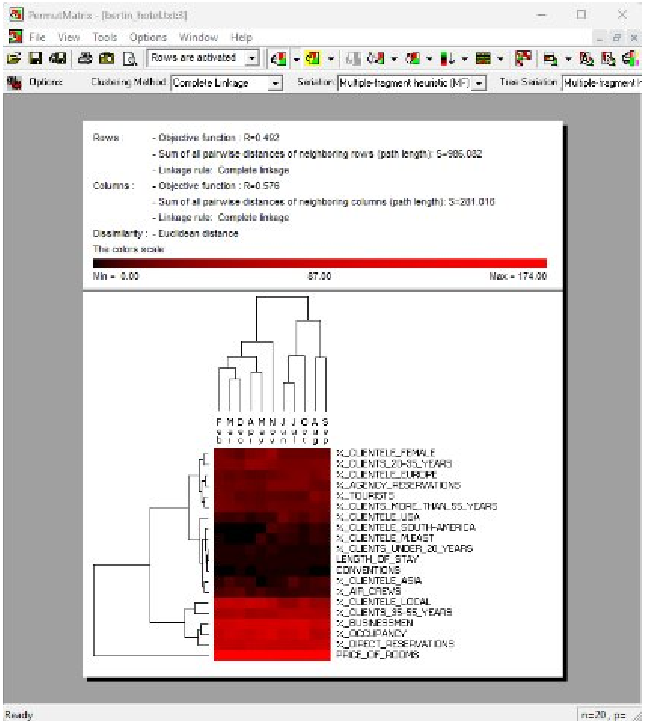

Figure 5.4: PermutMatrix with Bertin's hotel dataset. [Screenshot taken by Andrej Knaus.]

5.5 Discussion

The four tools differ significantly in both technical implementation and usability, as can be seen in Table 5.1. In particular:

- *Matrix Explorer*: A Java-based standalone application, supporting both manual and algorithmic reordering with moderate interactivity via drag-and-drop. While suitable for offline use, it lacks support for common data formats such as CSV or TSV, which limits its flexibility.

- *Bertifier*: A browser-based tool that emphasizes manual reordering through a user-friendly interface. It excels in visual refinement and export options (SVG, PDF), though it offers no algorithmic solver for reordering matrices.

- *Clustergrammer*: A powerful, analysis-oriented tool that supports AHC, offers rich interactivity (including zooming, hovering, and filtering), and is highly customizable through its API. It integrates seamlessly with modern workflows via Jupyter notebooks and web deployment, making it ideal for users familiar with JavaScript or Python.

- *PermutMatrix*: A Windows-only desktop application, which supports both AHC and seriation with moderate interactivity. While it provides basic styling and labeling features, it is limited by its platform restriction (Windows-only) and narrow data format support, accepting only TSV files instead of more common formats like CSV or JSON.

Licensing and hosting models also vary: some tools are open-source, while others are freeware or more commercially restricted. These differences influence their adoption in academic versus commercial contexts. Ultimately, the choice of tool depends on the user's technical skills, preferred workflow, and the complexity and scale of the dataset being analyzed.

	Matrix Explorer	*Bertifier*	*Clustergrammer*	*PermutMatrix*
Methods	Manual reordering, Seriation	Manual reordering	AHC	AHC and Seriation
Interactivity	Medium (drag-and-drop)	High (drag-and-drop, visual refinement)	High (hover, zoom, filtering)	Medium (swap, highlight)
Customization	Medium (visual style)	Medium (visual style)	High (API and various styling options)	Medium (color schema, label rows/columns)
License	–	BSD 3-Clause	MIT	Freeware for academic use
Hosting	Java Standalone App	Web App	Web App, self-hosting, Jupyter Widget available	Standalone desktop app for Windows only
Tech Stack	Java	JavaScript	JavaScript, Python	Delphi
Import	Custom format	CSV	TSV	TSV
Export	Custom format	SVG, PDF	SVG, PNG	JPG, PS, BMP

Table 5.1: Comparison of software tools for matrix reordering and visualization.

Chapter 6

Concluding Remarks

Reorderable matrices are a practical and intuitive way to uncover structure in complex datasets. This paper explored their historical development from early physical tools like Jacques Bertin's Domino to modern algorithmic methods that reorder data to highlight patterns, trends, and clusters. Whether the goal is to capture gradual transitions, distinct clusters, or distance-based relationships, each method offers unique insights into the underlying structure of the data.

As datasets grow in size and complexity, visual techniques like reorderable matrices become increasingly important for interpretation and communication. These matrices support exploratory analysis, reveal hidden relationships, and make data more accessible — not only to experts and scientists, but also to broader audiences. With the integration of interactive interfaces and machine learning techniques, reorderable matrices are poised to remain essential tools in data analysis and visualization.

Bibliography

Behrisch, Michael, Benjamin Bach, Matthias Hund, Marius Delz, Tobias Schreck and Daniel Weiskopf [2016]. *Matrix Reordering Methods for Table and Network Visualization*. Computer Graphics Forum 35.3 (2016), pages 693–716. ISSN 1467-8659. doi:10.1111/cgf.12935 (cited on page 6).

Bertin, Jacques [1981]. *Graphics and Graphic Information-Processing*. Originally published in French as La graphique et le traitement graphique de l'information in 1977. De Gruyter, 01 Dec 1981. 280 pages. ISBN 3110088681 (cited on page 1).

Bertin, Jacques [1983]. *Semiology of Graphics: Diagrams, Networks, Maps*. Originally published in French as Sémiologie Graphique in 1967. University of Wisconsin Press, 31 Dec 1983. ISBN 0299090604 (cited on page 3).

Caraux, Gilles and Sylvie Pinloche [2005]. *PermutMatrix: A Graphical Environment to Arrange Gene Expression Profiles in Optimal Linear Order*. Bioinformatics 21.7 (2005), pages 1280–1281. doi:10.1093/bioinformatics/bti141 (cited on page 15).

Eddy, Nicholas L., Arunachalam Vinayagam and Avi Ma'ayan [2018]. *Clustergrammer: Web-based Heatmap Visualization and Analysis Tool*. 2018. https://maayanlab.cloud/clustergrammer/ (cited on page 15).

Garcia, Marco, Weinan Huang, Christin Seifert and Wolfgang Wallisch [2010]. *Literature Survey: The Reorderable Matrix*. Graz University of Technology, Information Visualisation Course, 05 May 2010. https://courses.isds.tugraz.at/ivis/surveys/ss2010/g5-survey-reord-matrix.pdf (cited on page 13).

Hahsler, Michael, Kurt Hornik and Christian Buchta [2008]. *Getting Things in Order: An Introduction to the R Package Seriation*. Journal of Statistical Software 25.3 (Mar 2008), pages 1–34. ISSN 1548-7660. doi:10.18637/jss.v025.i03 (cited on page 5).

Kong, Doo-Sik, Jinkuk Kim, In-Hee Lee, Sung Tae Kim, Ho Jun Seol, Jung-Il Lee, Woong-Yang Park, Gyuha Ryu, Zichen Wang, Avi Ma'ayan and Do-Hyun Nam [2016]. *Integrative Radiogenomic Analysis for Multicentric Radiophenotype in Glioblastoma*. Oncotarget 7.10 (01 Feb 2016), pages 11526–11538. doi:10.18632/oncotarget.7115 (cited on page 11).

Kosara, Robert [2014]. *VIS 2014: Wednesday Highlights*. 2014. https://eagereyes.org/blog/2014/vis-2014-wednesday (cited on page 4).

Labrune, Jean-Baptiste [2004]. *Bertin's Domino Matrix*. Apr 2004. https://flickr.com/photos/jeanbaptisteparis/3407305484/ (cited on page 4).

Liiv, Jarmo [2010]. *Seriation and Matrix Reordering Methods: An Historical Overview*. Statistical Analysis and Data Mining 3.2 (2010), pages 70–91. doi:10.1002/sam.10071 (cited on page 1).

Mäkinen, Erkki and Harri Siirtola [2000]. *Reordering the Reorderable Matrix as an Algorithmic Problem*. Proc. 1st International Conference on Theory and Application of Diagrams (Diagrams 2000) (Edinburgh, UK). Springer, 2000, pages 453–468. doi:10.1007/3-540-44590-0_37. https://mat.ucsb.edu/g.legrady/academic/courses/15w259/d/re_orderableMatrix.pdf (cited on page 13).

Perin, Charles, Pierre Dragicevic and Jean-Daniel Fekete [2014a]. *Bertifier: An Interactive Reorderable Matrix Visualization Tool*. 2014. https://aviz.fr/bertifier_app (cited on page 13).

Perin, Charles, Pierre Dragicevic and Jean-Daniel Fekete [2014b]. *Revisiting Bertin Matrices: New Interactions for Crafting Tabular Visualizations*. Proc. IEEE Conference on Information Visualization (InfoVis 2014) (Paris, France). 09 Nov 2014. doi:10.1109/TVCG.2014.2346279. https://aviz.fr/wiki/uploads/Bertifier/bertifier-authorversion.pdf (cited on page 3).

Perin, Charles, Mathieu Le Goc, Romain Di Vozzo, Jean-Daniel Fekete and Pierre Dragicevic [2015]. *DIY Bertin Matrix*. Proc. CHI 2015 Workshop on Exploring the Challenges of Making Data Physical (Seoul, Korea). 18 Apr 2015. http://jasonalexander.kiwi/workshops/physicaldata2015/papers/Perin.pdf (cited on page 3).

Prelić, Amela, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele and Eckhart Zitzler [2006]. *A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data*. Bioinformatics 22.9 (May 2006), pages 1122–1129. doi:10.1093/bioinformatics/btl060 (cited on page 7).

StackExchange [2011]. *What's the Difference Between Principal Component Analysis and Multidimensional Scaling?* 08 Aug 2011. https://stats.stackexchange.com/questions/14002/whats-the-difference-between-principal-component-analysis-and-multidimensional (cited on page 8).

Team, Aviz [2015]. *The DIY Matrix Project*. INRIA, 2015. https://aviz.fr/diyMatrix/ (cited on page 3).

Wikipedia [2025]. *Multidimensional Scaling*. Wikimedia Foundation, 17 Apr 2025. https://wikipedia.org/wiki/Multidimensional_scaling (cited on page 8).