A Five-Level Design Framework for Bicluster Visualizations

Maoyuan Sun, Chris North and Naren Ramakrishnan

Abstract— Analysts often need to explore and identify coordinated relationships (e.g., four people who visited the same five cities on the same set of days) within some large datasets for sensemaking. Biclusters provide a potential solution to ease this process, because each computed bicluster bundles individual relationships into coordinated sets. By understanding such computed, structural, relations within biclusters, analysts can leverage their domain knowledge and intuition to determine the importance and relevance of the extracted relationships for making hypotheses. However, due to the lack of systematic design guidelines, it is still a challenge to design effective and usable visualizations of biclusters to enhance their perceptablity and interactivity for exploring coordinated relationships. In this paper, we present a five-level design framework for bicluster visualizations, with a survey of the state-of-the-art design considerations and applications that are related or that can be applied to bicluster visualizations. We summarize pros and cons of these design options to support user tasks at each of the five-level relationships. Finally, we discuss future research challenges for bicluster visualizations and their incorporation into visual analytics tools.

Index Terms—Biclusters, interactive visual analytics, coordinated relationships, design framework.

1 Introduction

Meaningful coordinated relationships discovery is a common problem in visual analytics. Coordinated relationships are groups of shared relations between sets of entities of different types. For example, intelligence analysts often examine large unstructured textual datasets to identify coordinated relations between different entity types (e.g., people, locations, dates) that might be evidence for collusion [47]. Bioinformaticians explore coordinated relations from expression and interaction datasets to identify groups of genes and/or proteins that are commonly expressed or regulated conditions and species [1, 70]. Analysts in cyber security trace coordinated relations between processes, hosts and network domains to detect distributed coordinated attacks [91]. While coordinated relations are thus important in many areas, we use text analytics as an example throughout this paper.

With training, analysts can manually identify and explore coordinated relations in data, but with significant cognitive effort. This process usually involves three essential repetitive tasks: 1) identify and extract meaningful entities, 2) investigate entities to verify whether a set of entities are related to the same specific entity or entities, and 3) cluster or group entities based on their shared relationships. For example, to find four people who all visited the same five cities, analysts may read numerous documents, identify names and cities from the documents, compare many co-occurring people-city pairs among different scenarios, and test many possible combinatorial groupings of the pairs, to finally discover the four people who are all paired with the same five cities.

For example, Jigsaw [35] supports exploratory text analysis, and its List View provides a solution for exploring simple 1:1 relationships from textual datasets based on term co-occurrence. By selecting an entity (in a list) of interest (e.g., a person's name), users can easily find related entities (e.g., locations and dates) because Jigsaw highlights these related entities located in other lists. However, Jigsaw has limited capability to help users to identify coordinated relations. For instance, consider testing whether there is any set of four people who have visited the same five cities with Jigsaw. Users can simultaneously select multiple entities to view the relationships they have in common. However, Jigsaw does not explicitly guide users as to which four people to select, so users must iteratively select many possible combinations of four names to find potential overlap in their related cities, which is a time consuming task. The problem gets even more complicated when composing or chaining multiple such coordinated relationships (e.g. were these people traveling on the same dates?). Thus, tools like Jigsaw do not effectively assist users in exploring coordinated relations.

Computation can ease this combinatoric exploration through the use of effective data mining algorithms. Analysts seek the help of visual analytics to support sensemaking [68], for the benefit of both advanced computational power and human cognitive abilities [82]. Specifically, biclustering algorithms can provide an efficient solution to identify coordinated relations.

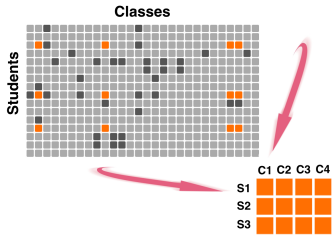

Fig. 1. An example of a bicluster extracted from a students-to-classes relation. Dark cells indicate existing relationships, and orange cells represent relationships part of this bicluster. Three students took the same four classes.

Biclustering is a data mining technique that has been extensively used in bioinformatics, especially for gene expression data analysis [5, 18, 60, 61, 62, 69, 72, 81], Biclusters, the computational outcome of biclustering algorithms, potentially provide a rich high-level abstraction that represents coordinated relationships between groups of entities of different types (e.g., a group of genes behave similarly under a group of conditions). Biclusters have been applied in intelligence analysis tasks to help analysts discover coordinated relations from textual datasets that may represent collusions [28, 80]. In general, a bicluster can be considered a complete bipartite graph where every vertex of one set is connected to all vertices of another set. Specifically, a bicluster in a relation can be viewed as a bundling of individual relationships into a pair of sets. For instance, as is shown in Figure 1, from a relation capturing attendance of students in specific classes, we might infer a bicluster involving a set of students [S1, S2, S3] who all attend the same set of classes [C1, C2, C3, C4]. In this case, no additional students or classes can be added to this bicluster; otherwise it will break the requirement that biclusters are maximal.

While biclusters provide a good mathematical foundation for identifying coordinated relationships, biclusters must be made usable for analysts through interactive visual representations. Several designs of bicluster visualizations, with the purpose of improving human percep-

• *Maoyuan Sun, Chris North and Naren Ramakrishnan are all with the Discovery Analytics Center, Department of Computer Science, Virginia Tech. E-mail: {smaoyuan | north | naren}@cs.vt.edu.*

tion of biclusters and using them to facilitate analysis, have been implemented (e.g., BicAT [6], BiVisu [17], Bixplorer [28], BiGGEsTS [34], BiCluster viewer [40] and BicOverlapper [74]) and reported promising results.

However, the challenge is a lack of systematic design guidelines to direct the design of efficient, human perceptible and usable visual representations of biclusters with necessary interactions to assist human sensemaking. Several key questions related to the design of bicluster visualizations still remain unanswered, such as what are the goals of bicluster visualizations, how users navigate within a list of many output biclusters to identify interesting biclusters, and how to design visual representations and interactions that leverage the highly abstracted information of biclusters along with the detailed contextual information from the original dataset to support human sensemaking.

In this paper, we present a five-level design framework for bicluster visualizations, with a survey of the relevant state-of-the-art design considerations and applications. We summarize pros and cons of these design options for supporting user tasks at each of the five-level relationships. Finally, we discuss the further research challenges in exploring the design space of bicluster visualizations and their possible incorporations into visual analytics tools.

2 Biclustering and Chaining Biclusters

Clustering is a well-established concept, which has been comprehensively explored over the past fifty years [4]. The basic idea of clustering is that we are given n points or entities in a given m-dimensional space and a distance or similarity function defined over that space. The goal is to identify subsets (clusters) of entities such that points within a cluster are more similar (or nearer) to each other than to points from other clusters.

2.1 Biclustering

Compared with the concept of clustering, biclustering is a relatively younger concept. The idea of biclustering (although not under this name) has existed since 1972 [38]. Biclustering generalizes the idea of clustering by simultaneously finding both subsets of entities and subsets of dimensions such that the selected entities are homogeneous (only) within the selected dimensions. Biclustering thus treats the notion of points and dimensions more uniformly, which is different from clustering. Also, while clusters form a partition of the dataset (i.e., they are mutually exclusive and collectively exhaustive), biclusters can overlap and may not collectively span the entire matrix of relationships. If these two conditions are imposed, biclustering is also referred as *co-clustering* [23].

Starting with relations between entity sets, we formalize the notion of biclusters that we use for this paper as follows:

Relations between two Entity Sets. An entity set is a set of objects from a specific domain (e.g., dates). We assume that entities have been extracted from datasets (e.g., documents) by using entity recognizers such as LingPipe [15] or similar tools. Given two entity sets E and F, a (binary) relationship R *(E, F)* between E and F is a subset of $E \times F$ (the Cartesian product of E and F). We say that E is connected to F. It is useful to view R as both a matrix and as a bipartite graph. In text analytics, R can be used variously to model document co-occurrence, associations, or specific relations extracted by natural language processing. For instance, person X can be related to organization Y if they are mentioned in the same sentence, or if a dependency parse followed by a semantic labeling infers a "works-for" relationship between X and Y.

Bicluster. We define a *bicluster* (E', F') on R *(E, F)* as a set $E' \subseteq E$ and a set $F' \subseteq F$ such that $E' \times F' \subseteq R$. That is, there is a relationship between every element of E' with every element of F'. A *bicluster* (E', F') is *thin* if there is only one entity in either E' or F'.

Closed bicluster. A *bicluster* (E', F') is closed if:

(i) For every entity $e \in E - E'$, there is some entity $f \in F'$ such that $(e, f) \notin R$, and

(ii) For every entity $f \in F - F'$, there is some entity $e \in E'$ such that $(e, f) \notin R$.

That is, adding an entity in $E - E'$ or $F - F'$ to the bicluster will violate the condition that defines a bicluster mentioned above. In other words, a closed bicluster is the bicluster to which we cannot add additional rows or columns if it is represented in the form of matrix. Hence, a closed bicluster can be regarded as maximal in height and width (although the term "maximal bicluster" is sometimes reserved for other interpretations in the data mining community). In this paper, our notation of *biclusters* refers to *closed biclusters*.

With closed itemset algorithms proposed in the data mining literature (e.g., LCM [83] and CHARM [89]), closed biclusters can be mined from the original dataset (e.g., documents) based on the pre-extracted entities. These algorithms work level-wise, such as, by finding biclusters with just one row (or column), and then aiming to grow them by adding more rows (or columns) and observing how many columns (or rows), if there are any, are affected.

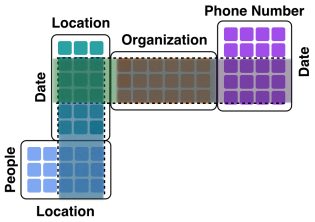

Fig. 2. Chaining four biclusters through multiple relations by approximately matching sets of entities across common domains.

2.2 Chaining Biclusters

Since every bicluster is discovered in a single relation, it is possible to compose separately identified biclusters across two relations by (approximately) matching biclusters with the shared domains. Jin et al. presented this approach to identify compositional patterns in multi-relational datasets [50]. As is shown in Figure 2, four biclusters that indicate four different relations can be chained together using the common interfaces (e.g., use location to connect the blue bicluster with the green one). By chaining biclusters across multiple relations, relationships from a diversity of domains can be bundled in a coherent manner. Results of such compositions can be read sequentially from one end to the other, which is similar to a story. For instance in the scenario from Figure 2, we might learn about 'a group of faculty from computer science, psychology and other departments', many of whom 'are planning a trip to Toronto and nearby places', the dates of which are approximately aligned with 'the last week of April 2014'; this might lead us to infer that they are likely HCI researchers planning to attend the CHI 2014 conference. Context information supporting these relations can then be inspected to gather evidence for this hypothesis.

Chaining biclusters can be achieved by using similarity search algorithms and data structures, (e.g., the cover tree, an efficient data structure for calculating nearest neighbors [8]). For each unique domain (e.g., people, locations, dates, etc.), one cover tree can be defined. For every bicluster discovered, the set of rows and the set of columns within the bicluster are indexed into two corresponding cover trees. After all biclusters are indexed, similarity searches can be readily conducted to find closest overlaps to all identified biclusters [28], which works as the basis for chaining biclusters.

With a clear notation of entity, biclustering, biclusters and chaining biclusters, we reach a common ground about these important concepts that are used in this paper. Each of them corresponds to one or several user tasks in intelligence analysis, which directs us to pursue systematic and comprehensive design guidelines for bicluster visualizations. This design space should cover design requirements for users to better perform these tasks as much as possible, such as efficient visual representations to illustrate relations corresponding to these concepts, and fluent navigations to direct users to different visual metaphors. Because bicluster visualizations have been primarily explored in bioinformatics, much of past research in this space falls in this domain.

3 Five Levels of Relationships

There are five levels of relationships (or connections) that underlie the notions of biclusters and chaining biclusters. These relations are closely related to the logic of the workflow that analysts may follow for sensemaking. To decompose the complexity of the discovery of relationships, we categorize these underlying relations into the following five levels (from low to high). Lower-level relations provide the critical basis that supports the exploration and identification of higher-level relations.

Entity* Level: Single Entity Relationships (*Entity-LR*).** This is the most basic relationship, in the mathematical form of ***1:1. All other higher levels of relations build on this logical unit. In this relationship, for an entity in a particular domain, there is a corresponding entity that comes from either the same domain or another domain that relates to this entity based on some data or rules. For example, person A is related to city B, because person A has visited city B. Two domains, people and location, are involved in this relationship. As another example, gene X is similar to gene Y because they behave similarly under condition Q. In this relationship, despite the fact that there are two domains, genes and conditions, the two related entities are actually from the same domain, genes. To form relations in the next four levels, we need entities from different domains, so our discussion in this paper about *Entity-LR* refers to those with entities from two different domains, rather than the same domain.

Group* Level: Entity Group Relationships (*Group-LR*).** This level of relationship is in the mathematical form of ***1:n or ***n:1***. In such a relationship, there are, in total, ***n+1*** entities from two different domains. The semantics of such a relationship is that for an entity in one domain, there is a corresponding related group of entities in the other domain. For example, 15 people are related to Amazon, because they all usually buy items from there or they all work for Amazon. A *Group-LR* relationship can result from the union of several *Entity-LR* relations that share connections with the same entity.

Bicluster* Level: Coordinated Relationships (*Bicluster-LR*).** This type of relationship is in the form of ***m:n. There are two domains with ***m+n*** entities involved in this relationship, indicating that for a group of entities in one domain, a corresponding group of entities from another domain are related to them. For example, six people are connected with five locations, because they have each visited all the five locations. This level is represented by biclusters. This type of relationship can be formed by combining a series of *Group-LR* where every single entity belongs to the same domain and the corresponding groups of entities in these different *Group-LR* relations are the same.

Chain* Level: Chained Coordinated Relationships (*Chain-LR*).** This is a more complex level of coordinated relations, in the mathematical form of ***m:n:...:z. *Chain-LR* can be considered an extension of *Bicluster-LR* because multiple individual coordinated-relations are connected together based on the shared entities between each pair. With intermediate groups of entities, at least three domains with ***m+n...+z*** entities are connected with each other in *Chain-LR*. For example, four students, five cities and seven dates could be connected because all the four students visited the same five cities during the same week. Since there are more than two domains involved in this relationship, compared with *Bicluster-LR*, it takes more effort to mine or recognize *Chain-LR*. It is also more difficult for humans to understand them, especially when the number of involved domains is large. However, *Chain-LR* contains more relations, which may provide analysts with meaningful story-like information (e.g., who plans to do what at which locations on what dates) for making hypotheses.

***Schema* Level: Schema Level Relationships (*Schema-LR*).** This type of relationship presents highly abstracted, database-like, patterns within a dataset. *Schema-LR* indicates connections among all domains within a given dataset, which reveals an overview of the dataset. For example, in an intelligence analysis task, *Schema-LR* may refer to relations across all potentially meaningful domains for this task, such as people, organizations, locations, dates, and so on. Relevant domains within *Schema-LR* are usually defined or identified by domain experts, although some software (e.g., Entity Workspace [9], Jigsaw [35] and NetLens [53]) allow users to choose domains (from those that can be identified) based on specific tasks. *Schema-LR* can also be potentially formed on the basis of the search for involved domains by traversing those in all discovered *Chain-LR*.

Table 1 briefly summarizes these five levels of relationships. These relations cover most meaningful relations that analysts may want to explore, which leads to two design concerns: 1) how to visually represent these relations, and 2) how to interact with visual metaphors that can assist analysts to pick or find meaningful ones. Also, to enable human-in-the-loop [20] analysis, users may need control some key parameters in the data mining algorithms (e.g., biclustering and chaining), so that meaningful visualizations can be generated based on expected mining results. For *Bicluster-LR*, the size of a bicluster (the number of rows and columns) and domains are two key parameters for users to control; and for *Chain-LR*, the size of an overlap between two biclusters and domains of this share region are two user customizable parameters. Domains are also a user controllable parameter for both *Entity-LR* and *Group-LR*. The size of a group is another parameter for users to choose in instances of *Group-LR*. There is no obvious user customizable parameters for *Schema-LR* because this relationship is usually determined by datasets. These parameters offer opportunities for interaction in bicluster visualizations.

4 The Five-level Design Framework

Exploration and identification of meaningful five-level relations are essential tasks for sensemaking, which needs support from bicluster visualizations. On the basis of these relations, our discussion about the design framework of bicluster visualizations focuses on visual representation design and interaction design. The former addresses visual design choices for the five levels of relations with the purpose of summarizing feasible visual representation techniques to improve the perceptibility of computational results, especially biclusters and chaining biclusters. The latter discusses interaction design options with a principal task-driven purpose: guiding users to explore potentially meaningful relations. Thus, the interaction design can reinforce the perceptibility of visual representations by making them usable. With the combination of both aspects, we present a five-level design framework for bicluster visualizations to provide systematic design guidelines that inform the design of future visual analytics tools with use biclusters.

4.1 Design Choices for *Entity* and *Group* Levels

Several visual representations for graph layouts and interaction techniques have been discussed in [84], many of which can potentially be applied to present and explore *Entity-LR* and *Group-LR*. *Entity-LR* are easy for humans to interpret. Based on *Entity-LR*, *Group-LR* can also be easily formed given our previous discussion. In this section, we focus on the node-link diagram, since other visual representations (e.g., matrix) are more powerful to present higher levels of relations.

The node-link diagram is an intuitive way to visually represent relations between entities for relatively small datasets [41], although the shape of nodes or the type of links may be different (e.g., use circles or squares for nodes and use straight lines or curve lines for links). A single instance of *Entity-LR* has just two entities, and whatever shapes of nodes or types of links are used, it is easy for people to understand. By visually following an edge, regardless of its line types, people can easily understand that two nodes are related with each other. However, the situation becomes different when many *Entity-LR* instances, which may form *Group-LR*, are to be visualized, because there may be too many lines crossing with each other that obscures relationships among entities. The study from Ghoniem et al. [32] shows that there is significant difference in the node-link graph readability between the graph with straight lines and that with curved lines because curved lines are more efficient to reduce edge-crossing than straight lines. In addition, edge aggregation techniques, such as edge bundling [45], provide solutions to avoid clutter caused by too many lines. Selecting and highlighting (e.g., via changing shapes, colors, or size) are two important interactions usually applied in node-link diagrams to help discriminate some relations from others, because highlighted nodes or links become visually prominent for humans to perceive. For example, a node with bigger size is easily differentiated from those with normal

Table 1. A Brief Summary of the Five-Level Design Framework.

Level of Relationships	Format	Number of Domains	Number of Entities	User-Controllable Parameters
Entity Level	1 : 1	2	2	Domains
Group Level	$1:n$ or $n:1$	2	$n+1$	The size of a group; domains
Bicluster Level	$m:n$	2	$m+n$	The size of a bicluster; domains
Chain Level	$m:n:\ldots:z$	At least 3	$m+n+\ldots+z$	The size of overlap between two biclusters; domains
Schema Level	$1:\ldots:1$	Multiple	NA	NA

size. However, node-link diagrams can hardly represent *Bicluster-LR* and *Chain-LR* in an easily perceptible way due to the following three limitations:

***L1*: Random locations**. In a node-link diagram, entities are randomly placed in the space by connecting one another with links. Without clearly visual, spatial structures, users have to manually reorganize the location of entities to form a new visual structure of the identified *Bicluster-LR* and *Chain-LR* so that they can easily understand.

***L2*: Number of links does not scale**. Visually following links is the only way to explore relations between entities. The difficulty of doing so depends much on the number of edges in the graph.

***L3*: Difficult to incorporate domain information to spatially aggregated entities**. Color coding is usually applied in node-link diagrams to indicate entities' domain (or categorical) information. As a result, entities with the same domain information are not spatially aggregated, and users have to track both colors and links to find out two specific groups of entities that are related.

Using better layout techniques, matrix-based visualizations and parallel coordinates [48] are solutions that can help to overcome the above three limitations. Matrix-based visualizations and parallel coordinates show greater suitability for exploring *Bicluster-LR* and *Chain-LR* than *Entity-LR* and *Group-LR*, so we discuss them later in *Section 4.2.1* and *Section 4.2.2*, respectively.

Tree visualizations and layouts incorporating spatial distance (e.g. a force-directed layout [30]) are two common layouts for node-link diagrams. They can improve readability of the node-link diagram by overcoming *L1* because the location of nodes is determined based on certain rules. For example, in a force-directed layout, two nodes are placed near each other because they are considered as similar. If instances of *Entity-LR* and *Group-LR* are in hierarchical relationships, tree visualizations are good choices. However, tree visualizations cannot be applied to explore *Bicluster-LR* and *Chain-LR*. The definition of a tree violates that of *Bicluster-LR* and *Chain-LR* discussed in *Section 3*, since all nodes in a tree belong to the same domain, rather than different ones.

When using spatial distance to enhance node-link diagrams, interactions that support spatially organizing information (e.g., dragging entities and spatially grouping entities) are key design concerns, which enable users to navigate and/or create spatializations for spatial reasoning [13]. Tools such as IN-SPIRE [87] and ForceSPIRE [24] implemented these design choices to support users to spatially organize visual metaphor of documents in the workspace. Vizster [39] applied the force-directed layout in the node-link diagram for social network analysis, and it used "blobs" (transparent coloring regions) surrounding entities to represent community structures. Noack's LinLog energy model [63] applies energy-based and force-directed methods to layout clusters. Clusters in LinLog are defined as a group of nodes that have many internal edges but few external edges to nodes outside this group. Similar to Vizster, LinLog uses spatial separation to show different clusters, but it does not show edges between entities. Because of missing edges, it is impossible to explore *Entity-LR* and *Group-LR* from LinLog's visual representation without any necessary interaction (e.g., clicking nodes to show its edges). Spatial distance is readily perceived by humans, which can be used to indicate structures of a dataset. However, to find meaningful relations between specific entities, visual metaphors that work as scaffolding are still indispensable. Therefore, the number of links for entities still is a key constraint for the application of visually using spatial distance to explore *Bicluster-LR* and *Chain-LR*.

4.2 Design Choices for *Bicluster Level*

There are two major design concerns for *Bicluster-LR*: how to visualize a single bicluster; and how to visualize all possible biclusters identified from a dataset and navigate users to find meaningful ones. The first concern may lead to a simple visual metaphor of a single bicluster that is easy for humans to perceive, and the second concern may result in specific visualization techniques (e.g., focus+context [43, 57]) that allows users to explore meaningful biclusters based on the context. Matrix-based visualizations and parallel coordinates provide possible visual solutions to meet these two design concerns, and the former has been studied in the bioinformatics domain [67, 73].

4.2.1 Matrix-Based Visualizations

Matrix-based visualizations represent *Entity-LR*, *Group-LR* and *Bicluster-LR*, where a relationship is indicated by a cell in the matrix and the two corresponding entities are respectively listed as a row name and a column name of the matrix. For *Entity-LR* and *Group-LR*, compared with node-link diagrams, matrix-based visualizations are less intuitive for humans to perceive [33]. For *Bicluster-LR*, matrix-based visualizations are superior to node-link diagrams by overcoming the three constrains mentioned in *Section 4.1*. In a matrix, entities are listed as names of rows or columns, rather than randomly located in the space. By using cells to indicate relations, matrix-based visualizations effectively avoid visual clutter [71] caused by edges crossing and/or overlapping. Besides, domain information can be easily incorporated into matrix-based visualizations, and entities fitting in the same domain can be spatially listed near each other. For example, columns and rows of a matrix respectively belong to two different domains. This offers a clear visual representation for a single bicluster.

Bixplorer applied this idea to visualize individual biclusters mined from textual datasets, and reported that users could perform text analysis using these visual biclusters [28, 80]. However, for an overview of all mined biclusters from the text, Bixplorer simply listed all mined biclusters, requiring users to select biclusters from the list and view them in a detailed preview panel to determine whether each bicluster might be useful. Bixplorer emphasized a bottom-up approach, enabling users to discover relevant biclusters based on the documents and entities in their focus of investigation. The relevant biclusters were visually embedded directly into a user's spatial document workspace, thus placing them in context. Methods are needed to enable top-down overview of textual datasets from the perspective of biclusters that help direct users to meaningful biclusters and then to supporting document details.

Similar to Bixplorer, matrix-based visualizations enhanced with heatmaps are widespread in the bioinformatics domain for gene expression data analysis (e.g., BicAT [6], BiCluster viewer [40], BicOverlapper 2.0 [75], BiGGEsTS [34], BiVoc [37], Expression Profiler [54] and GAP [88]). To perform gene expression analysis, the collected raw microarray data are transformed into gene-expression matrices, where rows usually represent genes and columns stand for conditions [12]. Matrix-based visualizations are a good fit for this task. By simultaneously reordering rows and columns in the matrix, biclusters can be formed from the gene-expression matrices [61, 81], which helps to identify co-expressed genes under a shared set of conditions. Les

Misérables Co-occurrence[1] developed with D3 [10] is a good example that visually shows this process. Compared with static visualizations, presenting the dynamic reordering process helps users to understand how biclustering works and how biclusters are formed from a matrix.

A typical matrix-based visualization that shows the result of two biclusters identified from gene-expression matrices is shown in Figure 3[2]. The big matrix represents a gene-expression matrix and two small matrices indicate two identified biclusters. Comparing the two biclusters, we find that all genes are the same except two, and there are six conditions shared across the two biclusters. Parts of relations within the two biclusters overlap, so it is impossible to visually separate these two biclusters by just reordering rows and columns of the big matrix. Therefore, although the big matrix contains all relations to form biclusters, extra techniques are required to layout all possible biclusters in a human perceptible way and navigate for exploratory analysis. This is conceptually a double Euler diagram problem on two domains simultaneously.

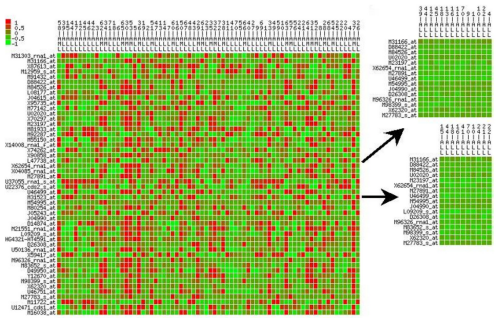

Fig. 3. An example of the matrix-based visualization to illustrate two biclusters mined from a gene-expression matrix[2].

Grothaus et al. [37] proposed an automatic layout algorithm that allows for replicating rows or columns to optimize the layout of matrix-based bicluster visualizations. The optimization refers to two aspects for the big matrix: *1)* to form contiguous subregions and each of them contains as many overlapping biclusters as possible; and *2)* to keep the size of the big matrix as small as possible. Bicluster Viewer [40] applied this algorithm to a matrix-based visualization with five key interactions to help navigate and explore biclusters from gene-expression matrices. Users can zoom in/out of the matrix, and highlight selected biclusters and their corresponding rows and columns. Users can choose to show all biclusters within the big matrix, with replicated rows and columns, and a rectangle with a colored frame is used to indicate the region of each bicluster. In addition, Bicluster Viewer can show biclusters without replicating any rows or columns. In this case, each bicluster may be split into different subregions within the big matrix, which are visually indicated by rectangles with the same colored dashed lines. To help navigate among biclusters, Bicluster Viewer maintains a list of identified biclusters, where the selected biclusters are highlighted with yellow and biclusters formed by replicating rows and colums are colored with red. However, similar to the problem with Bixplorer, the bicluster navigation list in Bicluster Viewer displays all biclusters in a simple list without user-defined names or labels. The arbitrary bicluster identifier names do not provide users with semantic information. Semantic meanings influence human interpretation [16, 26], which requires appropriate information scent to enable users to understand relations within a bicluster in a brief manner.

A matrix-based visualization provides an efficient visual representation for a single bicluster, which is easy for human to perceive. By replicating rows or columns, it is possible to layout all biclusters within a big matrix, and interactions applied in Bicluster Viewer provide a feasible instance to help users navigate among these biclusters. However, these replicated rows or columns may cause confusion, particularly when they are repeated several times and these repetitions may appear spatially near or far from each each. Given the combinatorial nature of biclusters, this process may require a large number of row and column replications.

4.2.2 Reduced Parallel Coordinates with Two Domains

Parallel coordinates is a well explored visual technique to present high-dimensional or multivariate data [22, 44, 86]. Since *Bicluster-LR* just has two involved domains, our discussion of parallel coordinates in this section refers to the reduced version with two domains. Instead of randomly placing entities in a two-dimensional space, parallel coordinates spatially sorts entities in a list based on their domains. Compared with the node-link diagram discussed in *Section 4.1*, parallel coordinates uses locations to separate one group of entities from another, and display them in an easily perceivable way. For example, in parallel coordinates, *Entity-LR* is represented as two entities from two lists with a line between them; *Group-LR* is displayed as one entity from a list that has several lines connecting with several entities from another list, and *Bicluster-LR* is more complex, which is represented as many entities from a list and each of them has the same number of links to the same entities in another list.

Jigsaw applied parallel coordinates in its List View [35], where entities are organized in different lists based on their domains. Similarly to [52] and [27], Jigsaw allows users to select domains (e.g., people, dates, locations, etc.) to be displayed in the List View. With interactions such as selecting, highlighting and ordering entities, users can easily explore *Entity-LR* and *Group-LR* of interest in Jigsaw. With these interactions, analysts can find biclusters in Jigsaw, and an example is shown in Figure 4. In this example, *"The Sign of the Crescent"* dataset [47] is imported in Jigsaw[3] and it takes three steps (from A to C) to find a bicluster that indicates the three key persons involved in the Atlanta event and a group of locations that they all visited. Jigsaw uses highlighting, particularly highlighting relevant entities by orange based on word co-occurrence, to guide users to perform exploratory analysis. However, there is little guidance for users to find biclusters. After step A in Figure 4, how do users know which entity in the list to consider adding into the bicluster in the next step? Users have to apply trial-end-error to finally reach the 3x6 bicluster shown in Figure 4. Thus, visually discriminating entities in the same list may better help analysts to find entities for their next analysis step. Coloring relevant entities in the same list based on the number of entities (in another list) shared with the selected one is a possible solution.

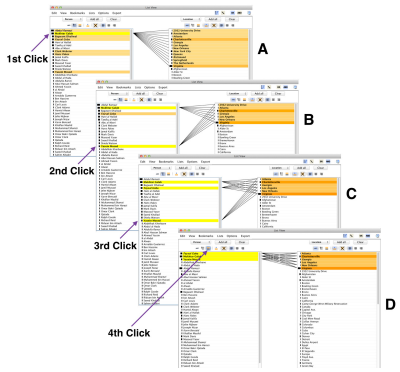

Fig. 4. An example of finding a bicluster in Jigsaw's List View. Yellow indicates entities that a user selected and orange indicates relevant entities corresponding to the selected one(s) based on word co-occurrence.

Parallel coordinates is also applied in bioinformatics to display biclusters (e.g., BicAT [6], BiCluster viewer [40], BicOverlapper 2.0 [75] and BiVisu [17]), where each vertical axis indicates a condition and a polyline represents a gene. However, compared with matrix-based visualizations, parallel coordinates is less used in this domain

[1]This visualization can be found at http://bost.ocks.org/mike/miserables/

[2]Taken from http://genomics10.bu.edu/terrence/gems/help.html

[3]Version 0.53, from http://www.cc.gatech.edu/gvu/ii/jigsaw/

[73] and few interactions are available in these tools. Parallel coordinates may do a better job to show variation in trends of genes under different conditions than explore biclusters. Although these tools show results of possible biclusters, none of them perform user studies to evaluate whether these biclusters in parallel coordinates can be perceived or not. Johansson et al. [51] tested readability of parallel coordinates with five stimulus patterns, and their study shows that difficulty to discriminate these five patterns in parallel coordinates increases when the noise level goes above 13%. This suggests that if lines indicating relations of different biclusters can be visually well organized, users may identify biclusters in parallel coordinates.

Four Design Choices. There are four different design choices discussed in relevant literature that can be applied to improve the display of biclusters in parallel coordinates. The most basic one is to move entities belonging to a bicluster together and use different color to highlight them, such as step D in Figure 4. This aggregates entities spatially close with each other, which helps to separate these entities from others. Another way is to replace straight polylines with curved lines [36] and add force, similar to force directed layout, to these curved lines to aggregate or bundle them based on certain rules [93]. In this way, possible biclusters can be explored by starting analysis from the aggregated curved lines. The third way is to aggregate entities and polylines respectively and use colored ribbons, similar to the *Bubble Sets* technique [19], to wrap the aggregated entities and polylines from the first vertical axis to the final one [3, 31, 55, 58, 65, 92], which can be used to indicate a bicluster that is determined by the smallest set of the shared entities across all axes in this region. Finally, tile-based parallel coordinates [2] provides an efficient way to avoid visual clutter, since it divides the plotting space into rectangular tiles and colors these tiles based on the sum of polylines that intersect with the tile. This can be applied to show biclusters with a modification of color coding rules for tiles. For example, based on the selected entities, a set of polylines (denoted *SetA*) can be formed by a union operation of all polylines starting from these entities. Then for each tile, a set of shared polylines (denoted *SetB*) can be found by an intersection operation between the polylines passing through this tile and those in *SetA*. Finally, each tile is colored based on the total number of polylines in *SetB*. By following the colored tiles, it is possible to identify whether biclusters exist or not for the selected entities.

These four design concerns provide possible solutions for presenting a relatively small number of biclusters in parallel coordinates. If there are overlaps between biclusters, the first design choice will not completely display all biclusters unless some entities are replicated. Applying the third design choice with replicated entities, biclusters in parallel coordinates are still difficult to identify, because many regions may overlap with each other. These overlaps may lead to misunderstanding and obscure the exact number of clusters [21]. The second design choice avoids aggregating entities together, but visual clutter may still appear due to many curved lines, especially when there are many bicluters. For the last design choice, if many biclusters exist, various tiles may have the same color. In this case, it is difficult to differentiate biclusters. However, the 1-dimensional sorting of parallel coordinates should reduce the replication problem in comparison to the 2-dimensional sorting of the matrix-based approach. Although there may be some interesting optimizations that attempt to sort two vertical axes in parallel coordinates so that the bicluster links are as horizontal as possible. Ultimately though, this solution devolves into a linear list of biclusters as used in Bixplorer.

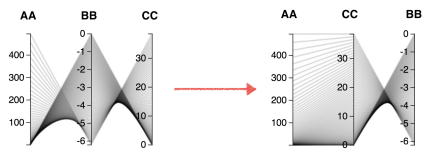

Fig. 5. An example of rearranging axes by switching axes BB and CC[4].

To overcome these drawbacks, interactions are a key requisite to identify meaningful biclusters. In addition to the basic interactions mentioned in the Jigsaw example (e.g., select and highlight entities), brushing and rearranging axes are two important interactions to explore parallel coordinates [78]. Brushing allows users to create a customized region (e.g., a focus area) in an axis and move it to select a set of polylines [79]. By following these polylines, analysts can determine whether biclusters exists or not for entities enclosed in the bin. For example, all three entities shown in step D in Figure 5 have six lines connecting to the same six locations, so that the three people and the six locations form a bicluster. Axes rearrangement assists to explore relations between two specific axes, which may reduce polylines crossing and help to find relations between entities in two nonadjacent axes. An example of axes rearrangement in parallel coordinates is show in Figure 5[4]. By switching two axes BB and CC, relations between entities in AA and CC are clearly revealed. If axes AA, BB and CC are different domains, axes rearrangement also provides a feasible way to explore *Chain-LR*.

4.2.3 Zoned Node-Link Diagram

Node-link diagrams can also be used to explore *Bicluster-LR*. BicOverlap [74] applied modified node-link diagrams to show biclusters and overlaps among biclusters. In BicOverlap, each node represents an entity, and the layout of nodes are determined based on a force-directed layout algorithm. Nodes in different domains are indicated with different visual marks. All nodes in each bicluster are wrapped in a "zone". The boundary of this "zone" is determined by the outermost nodes. To avoid visual clutter, edges between each pair of nodes are hidden. It seems that this design works for both single and all biclusters cases, but at least three drawbacks exist. Visual marks help to discriminate one domain from another, but they are hard to remember without a legend. By hiding edges, the visual representation in BicOverlap implicitly emphasizes entities rather than relations, so relations may be obscured by a large number of entities. Overlaps among "zones" indicate overlaps among biclusters, but the perceptibility of this depends on the number of biclusters overlapping with each other. Small-size biclusters, those with a small number of entities, within a heavily overlapping region may be ignored. However, the advantage is that this design is able to convey an overview of biclusters within a dataset. Furthermore, enhanced with interactions (e.g., filter biclusters by domains and popup a bicluster of interest), this design can help explore meaningful biclusters within the overall context.

4.3 Design Choices for *Chain Level*

How to visually represent a bicluster-chain or all bicluster-chains and help users navigate to meaningful chains is a crucial task for *Chain-LR* visualizations. Hybrid matrix diagrams provide a feasible solution to fulfill these demands. Hybrid matrix diagrams combine node-link diagrams or parallel coordinates with matrix based visualizations, which substitutes nodes in the node-link diagram or entities in axes of the parallel coordinates with matrices. Parallel coordinates discussed in this section are those with multiple domains and design concerns discussed in previous section can also be applied here. Each matrix in the hybrid diagram indicates a bicluster, and the node-link diagram or parallel coordinates illustrates how several biclusters are connected together, which also specifies the structure of bicluster-chains.

4.3.1 Node-Link Diagrams + Matrices

NodeTrix [42] and Bixplorer [28] are two systems that apply hybrid matrix diagrams, and a similar design is also mentioned in [7]. An example of hybrid matrix diagrams generated with Bixplorer[5] is shown in Figure 6. In this example, there are three biclusters that (from left to right) respectively represent relations between people and location, phone number and people, and date and phone number. Curved lines indicate shared entities between two biclusters. Bixplorer's bottom-up approach allows users to interactively expand chains from a given bicluster.

[4]Modified based on the example of reordering from the following website http://syntagmatic.github.io/parallel-coordinates/

[5]The tool can be found at http://recsys.cs.vt.edu/mineviz

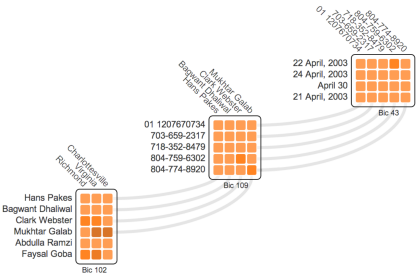

Fig. 6. An example of bicluster chain with three biclusters in Bixplorer.

In a more top-down design that attempts to support Ben Shneiderman's visual information seeking mantra [77] of overview first, NodeTrix uses a node-link diagram to show many connected matrices. NodeTrix provides three types of links: "underlying links" (simple curved lines, same as those in Bixplorer), "underlying links with full size" (curved lines, the thickness of which equals to the width of a matrix cell's edge) and "underlying links with attributes" (curved lines highlighted with different colors). The first type of link shows detailed connections, and the last type of link visually differentiates some links from others. Users can also extract a node from a matrix and merge two matrices together in NodeTrix. The design of NodeTrix is able to show several bicluster chains, actually a graph of related biclusters. The interaction design of NodeTrix concentrates on assisting the analysis of the connected parts of the graph by splitting and merging biclusters to explore alternate configurations of chains. Still, there is a need for designs that can guide users in exploring these chains.

Visually dynamic path extraction [46] seems a promising way to enhance NodeTrix's design for chain exploration. enRoute [66] implemented dynamic path extraction for biological pathway analysis, and used *Bubble Sets* [19] based techniques (using isocontours to create a colored region to wrap a set of entities) to visualize a selected path and its alternatives. In enRoute, all possible paths between the user-selected start and end node are visually presented, and users can add nodes to extend selected paths. Incorporating this into NodeTrix's design, users can begin chain exploration by choosing a start and end bicluster, or the system shows all computed chains with heatmap styled bubble sets. The color of these bubble sets can be encoded based on the two domains of a bicluster. For biclusters with the same two domains, the more cells in a bicluster, the darker its color will be. Moreover, user-selected biclusters or chains can rise up to the front layer. Thus, based on this design, users can perform chain analysis by starting with the overview and seeking guided visual metaphors on demands.

4.3.2 Parallel Coordinates + Matrices

Some design considerations discussed in *Section 4.2.2* (e.g., wrap entities and polylines with colored ribbons, brush and rearrange axes) also apply for *Chain-LR* exploration. The design of hybrid diagrams that combine parallel coordinates and matrices provides a better solution for chain exploration, and it was applied for comparing results of different clusters. HCE [76] applied parallel coordinates with matrices to compare results of two hierarchical clustering algorithms of genomic microarray data, and its biology users showed positive feedbacks about this design. The Caleydo Matchmaker technique [59] applied this design to conduct visual comparison among multiple groups of clusters. In this design, matrices in each axes can represent biclusters with two specific domains, and matrices connected among multiple axes can indicate chains. Compared with the previous hybrid diagram design, biclusters in this design are better organized. Since entities are aggregated into different biclusters, compared with parallel coordinates with entities on axes, this design reduces the number of links between each pair of adjacent axes. Moreover, using colored ribbons or the *Bubble Sets* technique to wrap matrices and links works as a salient visual representation of bicluster chains.

4.4 Design Choices for *Schema Level*

Schema-LR indicates relations among domains. The number of domains is much smaller than that of entities, so visual representations of *Entity-LR* can also be applied to *Schema-LR*. For example, the normal node-link diagram (e.g., database schema diagrams) is an obvious visualization that can clearly convey *Schema-LR* in a dataset. In this representation, each node stands for a domain in the dataset, and the thickness of links can indicate the strength that two domains are connected. The connection strength can be calculated based on the number of connections between entities in the two domains or the number of biclusters formed with entities of the two domains, or the number of chains participated in.

This design gives a clear overview of the dataset, and an alternative design is the clutter map proposed in [29]. A clutter map is similar to a node-link diagram with more detailed information. In the clutter map, the size of nodes is determined based on the number of entities belonging to this class (or domain) and edges are balloon-shaped that are visually merged with connected nodes. The size of the balloon depends on the shared entities between two domains, which can be modified to determine ballon size based on the number of biclusters relevant to the two domains. Another similarly applicable design is that from PivotGraph [85], which lays out aggregated nodes in a grid, similar to a chessboard. Node positions in the grid are determined by an algorithm that minimizes the number of edge-crossings. The size of nodes and the thickness of edges can be applied to encode the number of entities and biclusters respectively. Curved lines with arrowheads are used in PivotGraph to demonstrate directed edges, if there is any, which can also be applied to direct possible analytical paths for users to explore. For example, the path from *Domain A* to *Domain B*, then to *Domain C* is thicker than the path from *Domain A* to *Domain B*, then to *Domain D*, which indicates that starting analysis with *Domain A* to *Domain B* and *Domain C* may be more reasonable.

A fourth design is the chord diagram[6], which is inspired by Circos [56]. In the chord diagram, each chord can represent a domain in the dataset, and the length of a chord depends on the number of entities in this domain. Ribbons connecting two chords can indicate biclusters relevant to the two domains, and the thickness of ribbons may be determined by the number of the shared biclusters. Compared with the previous three design options, the chord diagram may not work well for a dataset with too many domains, because all chords are aligned in a circle. If the number of domains is large, the length of each chord will become small, and the number of ribbons will grow, which may lead to visual clutter inside the circle.

An important task of *Schema-LR* visualizations is to direct users to drill down to one or several domain(s) to explore more information for further analysis. On the basis of understanding visual representations of *Schema-LR*, users still need interactions to find domains that are meaningful for them. Dynamic path extraction, as discussed in *Section 4.3.1*, presents patterns among different domains based on extracted paths. This may guide users to further explorations of bicluster-chains related to some specific domains. It may also be useful to consider a chain-centric approach to Schema visualization, that overlays many chain paths onto the schema diagram.

5 Four-Level of Interaction Design

From the perspective of intent, there are four-level of interactions that potentially can be applied to bicluster visualizations: ***Readability Level*** (*Readability-LI*), ***Navigation Level*** (*Navigation-LI*), ***Parameter Level*** (*Parameter-LI*) and ***Object Level*** (*Object-LI*) [25]. *Readability-LI* aims at improving the readability of visualizations, so most interactions discussed in *Section 4* (e.g., select or highlight a node) belong to this category. *Navigation-LI* enables users to navigate between the five relationship levels and their visual representations. *Parameter-LI* enables users to control key parameters of algorithms. *Object-LI* helps users focus on their analytics process. *Navigation-LI* enhanced with *Parameter-LI* provides promising solutions to navigate visualizations of one relationship level to another.

[6] An interactive example is at http://bl.ocks.org/mbostock/4062006

For *Parameter-LI*, users are exposed to the underlying algorithm parameters. Using sliders to control the parameters values of an algorithm is a common example. iPCA [49] implemented this, where users can control how much a dimension will contribute to the PCA calculation by manipulating the sliders and choosing dimensions by check boxes. Bixplorer applied this to enable users to control the minimum size of biclusters to be identified from a textual dataset [80]. PivotSlice [90] allows users to drag parameters to specify domains for relationship discovery.

By controlling the parameters in the last column of Table 1, it is possible for users to navigate in either a top-down manner or a bottom-up one. For a top-down example, by choosing domains in *Schema-LR*, users can get specific *Chain-LR* or *Bicluster-LR* that they then can drill down to the *Group-LR* and *Entity-LR*. Bixplorer applied this by enabling users to extract, from a bicluster, a row or a column as a thin bicluster, and further extracting a cell from the thin bicluster. The row or column belongs to *Group-LR* and the specific cell is an instance of *Entity-LR*. Conversely, in a bottom-up fashion, Bixplorer users can also merge cells together to form a new row, a new column or a new bicluster in the form of a matrix, and then users can link their customized biclusters with each other to form a new chain that is meaningful for them. Thus, *Parameter-LI* of the higher level relations determines the computational results of the lower level, while *Navigation-LI* directs users to actually transform from one level of visualizations to another level. Visualizations at each of the five levels can be either linked or visually integrated to enable drill-down and roll-up through the levels. Bixplorer visually integrates the lower levels, but also provides separate linked views (lists) of the higher levels.

Similar to *Parameter-LI*, *Object-LI* can be applied to enhance *Navigation-LI* between and to direct the mining of biclusters and chains on various domains. *Object-LI* is an implicit way to control algorithms compared with *Parameter-LI*, since not all users realize that some of their interactions with visual metaphors are used as parameters of algorithms to control the future output. ForceSPIRE [24] is an example of such interactions (e.g., moving, annotating and highlighting) that re-weight the term dimensions in the distance metric and recalculate similarities among documents. EvoGraphDice [11, 14] employed a similar idea to dynamically change a scatterplot matrix. *Object-LI* enable users to focus on the data and perform exploratory analysis (for *Bicluster-LR* and *Chain-LR*) by implicitly tuning the algorithm parameters. For example, in Analyst's Workspace [46], the system shows bicluster chains to users that connect pairs of documents that the user interacts with, and then updates the chains by adding more relevant biclusters into the path of the chain based on the specific biclusters and documents that the users either keeps or eliminates. This implicitly incorporates analysts' judgement about members of a bicluster chain into the computation or visualization of results.

6 Conclusion and Future Challenge

All specific design choices discussed above are summarized in Table 2. Node-link diagrams support tasks relevant to almost all five levels, although variation exists for each specific level. Matrix based visualizations and parallel coordinates are valuable for the exploration of *Bicluster-LR* and *Chain-LR*. The hybrid visualization that combines the node-link diagram or parallel coordinates with matrices can do a better job, because it can display the overview of a dataset with the structure from the node-link diagram or parallel coordinates, and detailed relations buried in matrices. Several tools implement these design choices and have been evaluated with user studies. Positive user feedback from them indicate that these visual representations are good directions to pursue. However, at least three challenges still exist when applying this design framework to the design of future visual analytics tools for biclusters.

***C1*: Integration Challenge.** How to pick design options from this framework and snap them together into an integrated whole is a basic problem. Users likely need multiple coordinated visualizations for exploratory analysis across all five levels [64]. We have identified design options for each level but still lack examples that successfully combine them together to fulfill tasks across all five levels for sensemaking.

***C2*: Traversal Challenge.** How visual analytics tools should guide users' traversal through this five level framework is still not clear. Although Shneiderman's visual information seeking mantra has much impact on visualization design, the analytical process may not always work the same way. Biclusters offer a bridge connecting the overview and details in a dataset. However, at which end should bicluster visualizations start (i.e., *Schema-LR* first, or *Entity-LR* first)? How can we enable rapid bi-directional navigation among these levels, as suggested by Pirolli's sensemaking model [68]?

***C3*: Layout Challenge.** How to effectively layout all biclusers and bicluster chains in an overview still needs further research. Although possible solutions are discussed in this design framework, the replicated information may still cause confusion. Also, it may be difficult to enable users to customize the layouts generated by automatic layout algorithms.

These three challenges direct future research paths for the further exploration of a design space of bicluster visualizations. ***C1*** brings the question of how these design choices can be combined together. Results from ***C2*** may give clues to the answer to ***C1***, because the design of bicluster visualizations should follow users' analytical processes. Together with novel layouts identified in ***C3***, this framework and agenda can make biclusters usable for efficiently discovering coordinated relationships in visual analytics.

Acknowledgments

This work was supported in part by a grant from L-3 Communications, and NSF grants CCF-0937133 and IIS-1218346.

References

[1] D. B. Allison, X. Cui, G. P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, 7(1):55–65, 2006.

[2] J. Alsakran, Y. Zhao, and X. Zhao. Tile-based parallel coordinates and its application in financial visualization. *IS&T/SPIE Electro Imaging*, 2010.

[3] G. Andrienko and N. Andrienko. Blending Aggregation and Selection: Adapting Parallel Coordinates for the Visualization of Large Datasets. *The Cartographic Journal*, 42(1):49–60, June 2005.

[4] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, June 2010.

[5] M. H. Asyali, D. Colak, and O. Demirkaya. Gene expression profile classification: a review. *Current Bioinformatics*, 1(1):55–73, 2006.

[6] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. BicAT: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.

[7] V. Batagelj, F. J. Brandenburg, W. Didimo, G. Liotta, P. Palladino, and M. Patrignani. Visual Analysis of Large Graphs Using (X,Y)-Clustering and Hybrid Visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1587–1598, 2011.

[8] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*. ACM, June 2006.

[9] E. A. Bier, E. W. Ishak, and E. Chi. Entity workspace: an evidence file that aids memory, inference, and reading. In *ISI'06: Proceedings of the 4th IEEE international conference on Intelligence and Security Informatics*, pages 466–472, Berlin, Heidelberg, May 2006. Springer-Verlag.

[10] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data-Driven Documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, Dec. 2011.

[11] N. Boukhelifa, W. Cancino, A. Bezerianos, and E. Lutton. Evolutionary visual exploration: evaluation with expert users. In *Computer Graphics Forum*, volume 32, pages 31–40. Wiley Online Library, 2013.

[12] A. Brazma and J. Vilo. Gene expression data analysis. *FEBS letters*, 480(1):17–24, Aug. 2000.

[13] R. Byrne and P. N. Johnson-Laird. Spatial reasoning. *Journal of memory and language*, 28(5):564–575, 1989.

[14] W. Cancino, N. Boukhelifa, A. Bezerianos, and E. Lutton. *Evolutionary visual exploration: experimental analysis of algorithm behaviour*. experimental analysis of algorithm behaviour. ACM, New York, USA, 2013.

[15] B. Carpenter. Phrasal Queries with LingPipe and Lucene: Ad Hoc Genomics Text Retrieval. *TREC*, 2004.

[16] R. M. Chapman, J. W. McCrary, and J. A. Chapman. Brain responses related to semantic meaning. *Brain and Language*, 5(2):195–205, 1978.

Table 2. Summary of the Five-level Design Framework for Bicluster Visualizations

Relations	Major Tasks	Design Choices			Pros	Cons
		Visual Representation	Supplementary Visual Technique	Interaction Design		
Entity Level	1. Show an entity 2. Show a group of entities 3. Show entity level relations (a single case or multiple cases) 4. Show single entity vs. groups of entities level relations (a single case or multiple cases) 5. Find relevant entities for a specific entity 6. Verify relations between some entities 7. Discriminate some entities from others 8. Mark important entities or relations	The Node-Link Diagram	1. Edge bundling 2. Use spatial distance (e.g. the force-directed layout) 3. Use spatial distance + hiding links 4. Color coding to separate nodes of different domains or selected and unselected nodes or links 5. Visual marks (e.g., shapes) to separate nodes and/or links	1. Select nodes/links 2. Highlight nodes/links 3. Drag nodes/links	1. Intuitive way to show either an entity or multiple entities and relations between entities 2. Customizable spatial layout for users 3. Links clearly show specific relations between entities	1. Entities are randomly placed in the space, so it may be difficult to find an entity if there are many entities 2. The number of links exerts much impact on the readability of the diagram 3. Without links, relations between entities cannot be identified easily 4. Color coding and visual marks are not efficient to visually separate domains
		A Simple Matrix	1. A single cell to represent Entity 2. A row or a column to represent Group 3. Use a heatmap	1. Select cells 2. Highlight cells 3. Extract a cell 4. Merge cells	Avoid visual clutters caused by too many links	1. Not as easy as node-link diagrams to perceive 2. Columns or rows rearrangement is the only way to change the layout
		Parallel Coordinates with Two Domains	1. Edge bundling 2. Using curved lines to indicate links	1. Select entities 2. Highlight polylines/entities 3. Brushing 4. Axes rearrangement 5. Entities reposition in axes	1. Place entities of the same group together 2. Relatively easy to find entities 3. Efficiently select multiple entities/polylines	1. The number of links exerts much impact on the readability of the diagram 2. Without links, relations between entities cannot be identified easily 3. Sometimes entity reposition (e.g., moving relevant entities to the top) is necessary to understand Group
Group Level		Tree Visualizations	1. Icicle 2. Bubble trees 3. Treemaps	1. Select nodes/links 2. Highlight nodes/links 3. Path extraction	Clearly represent hierarchical relations	1. Not all Groups are hierarchical relations 2. Cannot represent biclusters and bicluster-chains
Bicluster Level	1. Show a bicluster 2. Show all biclusters 3. Find biclusters of interest 4. Mark biclusters of interest	Matrices	1. Use a heatmap 2. Reorder rows or columns 3. Repeat rows or columns 4. Color coding the region of a bicluster	1. Reorder rows/columns 2. Select biclusters 3. Highlight biclusters 4. Replicate rows/columns	1. A visual representation that is easy to understand biclusters 2. Efficiently reduce visual clutters caused by many links	1. Difficult to display all biclusters without replicating rows and/or columns 2. Replicated rows or columns may cause confusion 3. Overlaps may obscure biclusters with less entities
		Parallel Coordinates with Two Domains	1. Edge bundling 2. Use curved lines 3. Wrap entities with polylines 4. Tile-based parallel coordinates	1. Select entities 2. Brushing 3. Highlight polylines/entities/ribbons 4. Axes rearrangement 5. Entities reposition in axes	1. Place entities of the same group together 2. Relatively easy to find entities 3. Efficiently select multiple entities/polylines	1. The number of links exerts much impact on the readability of the diagram 2. Without links, relations between entities cannot be easily identified 3. Sometimes entity reposition (e.g., moving relevant entities to the top) is necessary to understand the relation
		Zoned Node-Link Diagram	1. Wrap nodes of a bicluster in a colored region 2. Use force-directed layout 3. Hide links between nodes	1. Select nodes/links 2. Highlight nodes/links 3. Drag nodes/links	1. Customizable spatial layout for users 2. Links clearly show relations between specific entities 3. Easily find entities that are shared between biclusters	1. Entities are randomly placed in the space, so it may be difficult to find an entity if there are many entities 2. Without links, relations between entities cannot be identified easily 3. Biclusters with less entities may be obscured in the overlapping region
Chain Level	1. Show a chain 2. Show all chains 3. Find chains of interest 4. Mark chains of interest	Node-link Diagram + Matrices	Combine all supplementary visual techniques that the node-link diagram and matrix based visualizations can use and the Bubble Sets technique	Combine all interactions that the node-link diagram and matrix based visualizations can use and path extraction	1. Efficiently reduce the number of links 2. A customizable spatial layout for users 3. Show the overview of the data based on bicluster-chains 4. By following links, users can find out how a bicluster-chain is formed	1. Entities may replicate many times in multiple matrices 2. Not a trivial visualization for users to understand connections across several biclusters 3. Which bicluster to choose to start a bicluster-chain is a problem
		Parallel Coordinates + Matrices	Combine all supplementary visual techniques that parallel coordinates and matrix based visualizations can use and the Bubble Set technique	Combine all interactions that parallel coordinates and matrix based visualizations can use and path extraction		
Schema Level	1. Show the overview of a dataset 2. Guide the exploration of chains or biclusters	The Node-Link Diagram	1. Clutter Map 2. The PivotGraph technique 3. Color coding to indicate different domains 4. Visual marks (e.g., shapes) to separate nodes and/or links 5. Use spatial distance (e.g. force-directed layout) 6. Use spatial distance + hiding links	1. Select nodes/links 2. Highlight nodes/links 3. Dynamic path extraction	1. An intuitive way to show relations between domains 2. The size of nodes and the thickness of links can be used to encode the information of biclusters and/or chains	1. The layout of PivotGraph cannot be easily changed by users 2. Depend on links to perceive relations across several specific domains
		The Chord Diagram	1. Color coding of chords to indicate different domains 2. Use ribbons between chords to indicate connections	1. Select chords/ribbons 2. Highlight chords/ribbons	1. An intuitive way to show relations between domains 2. The length of chords and the thickness of ribbons can be used to encode the information of bicluster and/or chains	1. Not efficient for a dataset with many domains 2. Ribbons inside the diagram may form visual clutter 3. Paths inside the diagram may be obscured by too many crossing ribbons

[17] K. O. Cheng, N. F. Law, W. C. Siu, and T. H. Lau. BiVisu: software tool for bicluster detection and visualization. *Bioinformatics*, 23(17):2342–2344, Sept. 2007.

[18] Y. Cheng and G. M. Church. Biclustering of expression data. *Ismb*, 2000.

[19] C. Collins, G. Penn, and S. Carpendale. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1009–1016, 2009.

[20] L. F. Cranor. A framework for reasoning about the human in the loop. *UPSEC*, 8:1–15, 2008.

[21] A. Dasgupta, M. Chen, and R. Kosara. Conceptualizing visual uncertainty in parallel coordinates. *Computer Graphics Forum*, 31:1015–1024, 2012.

[22] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394, 2003.

[23] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD*, pages 269–274. ACM, 2001.

[24] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482, New York, USA, 2012.

[25] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.

[26] T. D. Erickson and M. E. Mattson. From words to meaning: A semantic illusion. *Jour of Verb Learning and Verb Behavior*, 20(5):540–551, 1981.

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

[28] P. Fiaux, M. Sun, L. Bradel, C. North, N. Ramakrishnan, and A. Endert. Bixplorer: Visual Analytics with Biclusters. *Computer*, 46(8):90–94, 2013.

[29] C. Fluit, M. Sabou, and F. Van Harmelen. Ontology-based information visualization: toward semantic web applications. In *Visualizing the semantic web*, pages 45–58. Springer, 2006.

[30] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[31] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Visualization '99. Proceedings*, pages 43–508. IEEE, 1999.

[32] M. Ghoniem, J. Fekete, and P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24, 2004.

[33] M. Ghoniem, J. D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *InfoVis*, 4(2):114–135, 2005.

[34] J. P. Gonçalves, S. C. Madeira, and A. L. Oliveira. BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data. *BMC Research Notes*, 2(1):124, 2009.

[35] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko. Combining Computational Analyses and Interactive Visualization for Document Exploration and Sensemaking in Jigsaw. *Visualization and Computer Graphics, IEEE Transactions on*, 19(10):1646–1663, 2013.

[36] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Information Visualization, 2003. Proceedings. Seventh International Conference on*, pages 10–16. IEEE Comput. Soc, 2003.

[37] G. A. Grothaus, A. Mufti, and T. M. Murali. Automatic layout and visualization of biclusters. *Algorithms for Molecular Biology*, 2006.

[38] J. A. Hartigan. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, Mar. 1972.

[39] J. Heer and D. Boyd. Vizster: visualizing online social networks. *INFOVIS 2005. IEEE Symposium on*, pages 32–39, 2005.

[40] J. Heinrich, R. Seifert, M. Burch, and D. Weiskopf. BiCluster viewer: a visualization tool for analyzing gene expression data. In *ISVC'11: Proceedings of the 7th international conference on Advances in visual computing*. Springer-Verlag, Sept. 2011.

[41] N. Henry and J. Fekete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):677–684, 2006.

[42] N. Henry, J. Fekete, and M. J. McGuffin. NodeTrix: a Hybrid Visualization of Social Networks. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1302–1309, 2007.

[43] I. Herman, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[44] P. E. Hoffman and G. G. Grinstein. A survey of visualizations for high-dimensional data mining. *Information visualization in data mining and knowledge discovery*, pages 47–82, 2002.

[45] D. Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[46] M. S. Hossain, C. Andrews, N. Ramakrishnan, and C. North. Helping intelligence analysts make connections. In *Scalable Integration of Analytics and Visualization*, 2011.

[47] F. Hughes and D. Schum. Discovery-proof-choice, the art and science of the process of intelligence analysis-preparing for the future of intelligence analysis. *Washington, DC: Joint Military Intelligence College*, 2003.

[48] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization '90*, pages 361–378. IEEE Computer Society Press, 1990.

[49] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. iPCA: an interactive system for PCA-based visual analytics. In *EuroVis'09: Proceedings of the 11th Eurographics / IEEE - VGTC conference on Visualization*, pages 767–774. Eurographics Association, June 2009.

[50] Y. Jin, T. M. Murali, and N. Ramakrishnan. Compositional mining of multirelational biological datasets. *ACM Transactions on Knowledge Discovery from Data*, 2(1):1–35, Mar. 2008.

[51] J. Johansson, C. Forsell, M. Lind, and M. Cooper. Perceiving Patterns in Parallel Coordinates: Determining Thresholds for Identification of Relationships. *Information Visualization*, 7(2):152–162, June 2008.

[52] S. Johansson and J. Johansson. Interactive Dimensionality Reduction Through User-defined Combinations of Quality Metrics. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):993–1000, 2009.

[53] H. Kang, C. Plaisant, B. Lee, and B. B. Bederson. NetLens: Iterative Exploration of Content-Actor Network Data. *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pages 91–98, 2006.

[54] M. Kapushesky, P. Kemmeren, A. C. Culhane, S. Durinck, J. Ihmels, C. Korner, M. Kull, A. Torrente, U. Sarkans, J. Vilo, and A. Brazma. Expression Profiler: next generation–an online platform for analysis of microarray data. *Nucleic Acids Research*, 32:W465–W470, July 2004.

[55] R. Kosara, F. Bendix, and H. Hauser. Parallel Sets: interactive exploration and visual analysis of categorical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):558–568, 2006.

[56] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, Sept. 2009.

[57] J. Lamping, R. Rao, and P. Pirolli. *A focus+context technique based on hyperbolic geometry for visualizing large hierarchies*. ACM Press/Addison-Wesley Publishing Co., New York, USA, May 1995.

[58] A. Lex, H. Schulz, M. Streit, C. Partl, and D. Schmalstieg. VisBricks: Multiform Visualization of Large, Inhomogeneous Data. *TVCG*, 17(12):2291–2300, 2011.

[59] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative Analysis of Multidimensional, Quantitative Data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1027–1035, 2010.

[60] J. Liu, J. Yang, and W. Wang. Biclustering in gene expression data by tendency. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 182–193. IEEE, 2004.

[61] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 1(1):24–45, 2004.

[62] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477, Dec. 2006.

[63] A. Noack. An energy model for visual graph clustering. *Graph Drawing*, 2004.

[64] C. North and B. Shneiderman. *Snap-together visualization: a user interface for coordinating visualizations via relational schemata*. ACM, New York, New York, USA, May 2000.

[65] M. Novotny and H. Hauser. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):893–900, 2006.

[66] C. Partl, A. Lex, M. Streit, D. Kalkofen, K. Kashofer, and D. Schmalstieg. enRoute: Dynamic path extraction from biological pathway maps for in-depth experimental data analysis. *Biological Data Visualization (BioVis), 2012 IEEE Symposium on*, pages 107–114, 2012.

[67] G. A. Pavlopoulos, A.-L. Wegener, and R. Schn. A survey of visualization tools for biological network analysis. *BioData Mining*, 1(1):12, 2008.

[68] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis*, 2005.

[69] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, Apr. 2006.

[70] J. Quackenbush. Computational analysis of microarray data. *Nature reviews genetics*, 2(6):418–427, 2001.

[71] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *Journal of Vision*, 7(2):17–17, Aug. 2007.

[72] R. Santamaría, R. Therón, and L. Quintales. A Framework to Analyze Biclustering Results on Microarray Experiments. *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, 4881:770–779, 2007.

[73] R. Santamaría, R. Therón, and L. Quintales. A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, 9(1):247, 2008.

[74] R. Santamaria, R. Theron, and L. Quintales. BicOverlapper: A tool for bicluster visualization. *Bioinformatics*, 24(9):1212–1213, Apr. 2008.

[75] R. Santamaria, R. Theron, and L. Quintales. BicOverlapper 2.0: visual analysis for gene expression. *Bioinformatics*, 2014.

[76] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, 2002.

[77] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.

[78] H. Siirtola and K.-J. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, Dec. 2006.

[79] C. A. Steed, P. J. Fitzpatrick, T. J. Jankun-Kelly, A. N. Yancey, and J. E. Swan II. ARTICLE IN PRESS. *Computers and Geosciences*, 35(7):1529–1539, May 2009.

[80] M. Sun, L. Bradel, C. L. North, and N. Ramakrishnan. The role of interactive biclusters in sensemaking. In *Proceedings of the 32nd annual ACM CHI*, pages 1559–1562, 2014.

[81] A. Tanay, R. Sharan, and R. Shamir. Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 2005.

[82] J. J. Thomas and K. A. Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Comp Society Press, 2005.

[83] T. Uno, T. Asai, Y. Uchida, and H. Arimura. LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets. *FIMI*, 2003.

[84] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J. D. Fekete, and D. W. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6):1719–1749, Apr. 2011.

[85] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI*, pages 811–819. ACM, 2006.

[86] E. J. Wegman. Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of the American Statistical Assoc*, 85(411):664–675, 1990.

[87] P. C. Wong, B. Hetzler, C. Posse, M. Whiting, S. Havre, N. Cramer, A. Shah, M. Singhal, A. Turner, and J. Thomas. IN-SPIRE InfoVis 2004 Contest Entry. *INFOVIS 2004.*, pages 51–52, 2004.

[88] H.-M. Wu, Y.-J. Tien, and C.-h. Chen. Computational Statistics and Data Analysis. *Computational Stat and Data Analysis*, 54(3):767–778, 2010.

[89] M. J. Zaki and C. J. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining. *SDM*, pages 457–473, 2002.

[90] J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan. Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE TVCG*, 19(12):2080–2089, 2013.

[91] C. V. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.

[92] H. Zhou, W. Cui, H. Qu, and et al. Splatting the Lines in Parallel Coordinates. *Computer Graphics Forum*, 28(3):759–766, 2009.

[93] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual Clustering in Parallel Coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.