

# AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations

M. Adil Yalçın, *Student Member, IEEE*, Niklas Elmqvist, *Senior Member, IEEE*, and Benjamin B. Bederson

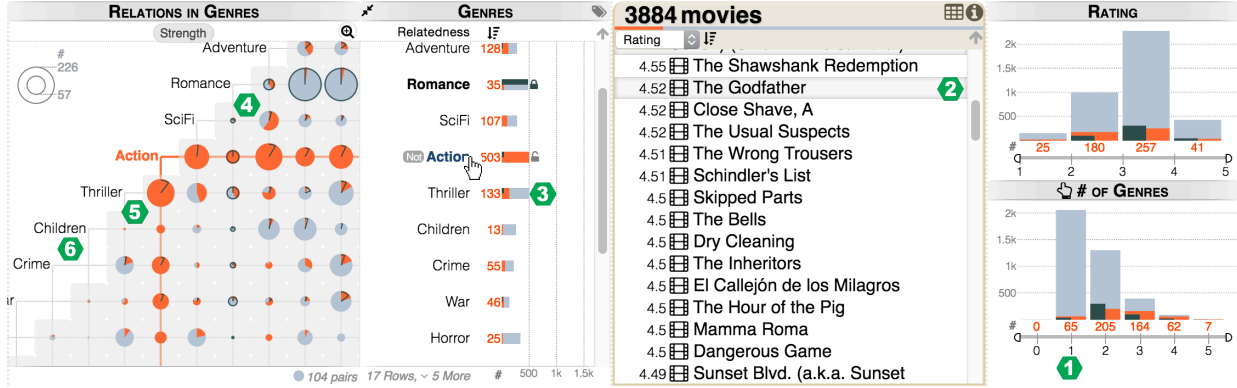


Fig. 1. Exploration of a movie dataset with multiple genres (sets) and ratings using AggreSet. Aggregate histograms are used for set-list and set-degrees, whereas the aggregate matrix (left) is used for set-pair intersections. The gray distributions visualize the number of elements per aggregate. The *Action* genre is selected by mouse-over . Mouse click will filter. We compare *Romance* (black lines) to *Action* (orange areas). ① Most movies (+2k) have one genre. 7 movies have maximum (5) genres. ② *The Godfather* is the only *Action* movie in the rating-sorted movie list. ③ Of *Thrillers*, 133 have *Action* (orange bar) and few have *Romance* (black line). ④ More than 50% of *SciFi* and *Adventure* movies have *Action*, while very few have *Romance*. ⑤ *Thriller* is more common with *Action* than with *Children* movies (circle size). ⑥ There is no *Children* movie with *Crime* (empty intersection).

**Abstract**—Datasets commonly include multi-value (set-typed) attributes that describe set memberships over elements, such as genres per movie or courses taken per student. Set-typed attributes describe rich relations across elements, sets, and the set intersections. Increasing the number of sets results in a combinatorial growth of relations and creates scalability challenges. Exploratory tasks (e.g. selection, comparison) have commonly been designed in separation for set-typed attributes, which reduces interface consistency. To improve on scalability and to support rich, contextual exploration of set-typed data, we present AggreSet. AggreSet creates aggregations for each data dimension: sets, set-degrees, set-pair intersections, and other attributes. It visualizes the element count per aggregate using a matrix plot for set-pair intersections, and histograms for set lists, set-degrees and other attributes. Its non-overlapping visual design is scalable to numerous and large sets. AggreSet supports selection, filtering, and comparison as core exploratory tasks. It allows analysis of set relations including subsets, disjoint sets and set intersection strength, and also features perceptual set ordering for detecting patterns in set matrices. Its interaction is designed for rich and rapid data exploration. We demonstrate results on a wide range of datasets from different domains with varying characteristics, and report on expert reviews and a case study using student enrollment and degree data with assistant deans at a major public university.

**Index Terms**—Multi-valued attributes, sets, visualization, set visualization, data exploration, interaction, design, scalability.

## 1 INTRODUCTION

Many real-world data collections consist of elements with multiple attributes. Some of these attributes may take multiple categorical values; for example, movies may have multiple genres, recipes have multiple ingredients, students take multiple courses, and publications typically have multiple keywords and authors. These multi-valued categorical attributes are commonly referred as *set-typed* since they implicitly describe set memberships over elements.

Set-typed data has recently received considerable attention in the field of information visualization, with visual representations based on linear lists of set intersections [24], radial node-link diagrams [4],

and element matrix compositions [32]. However, common between these and other visual set exploration approaches in the literature is that: (i) they scale to a relatively small number of sets; (ii) they are optimized for particular set exploration tasks; and (iii) they either do not support other element attributes beyond set membership, or the visualization and interaction is designed differently and ad-hoc for other attributes, decreasing consistency.

We present AggreSet, a novel set exploration technique that solves these challenges through an integrated design of linked visualizations of multiple data dimensions with rapid selection, filtering, and comparison (Figure 1). We address the challenges above as the following: (i) To improve scalability, AggreSet uses a matrix-based visualization for set relations. Scalability in the number of sets is achieved by the non-overlapping and zoomable nature of the set-matrix. Scalability in the number of elements is achieved by aggregation. (ii) Based on our analysis of set-typed data exploration and design guidelines, AggreSet is designed to achieve richness of supported tasks, design efficiency and consistency. (iii) AggreSet embeds the set-matrix in a multi-view layout consisting of histogram-based visualizations that are brushed and linked in a

• M. Adil Yalçın, Niklas Elmqvist, and Benjamin B. Bederson are with University of Maryland, College Park. E-mail: {yalcin, elm, bederson}@umd.edu.

Manuscript received 31 Aug. 2015; accepted 01 Aug. 2015. Date of Publication xx Aug. 2015; date of current version 25 Oct. 2015.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.

design that does not differentiate between set-typed and multivariate attributes. Specifically, AggreSet achieves:

**Scalability:** AggreSet supports concurrent analysis on numerous sets (50+) and many aggregated elements (100,000+) across multiple dimensions. Its scalability comes from non-overlapping visualizations of aggregations over elements, and a scrollable and zoomable matrix view for visualizing relations between sets.

**Richness:** AggreSet supports a plethora of tasks for exploring relations in set-typed attributes and elements with minimal visual and interaction components. Its multi-view and linked design enables higher-order analysis (e.g. intersection of three or more sets), surpassing the limitations of static 2D set-matrix layouts.

**Consistency:** The visual and interaction design of AggreSet is consistent across all attribute types; i.e. it does not differentiate between aggregates for sets, set-degrees, set-intersections and other attributes, when applicable.

**Rapid exploration:** The user can observe many relations on tightly coupled visualizations without performing explicit state changes that slow down interaction. Our visual and interaction design encourages an overview-to-detail exploration.

**Matrix design for set relations:** AggreSet’s set-matrix visualizes set-specific relations: empty, identical and sub-sets. It also presents a new set similarity metric, and a new method for set ordering to perceptually emphasize intersections of set groups.

Using our web-based implementation of AggreSet technique, we present screen-captured results using datasets from different domains with various characteristics in data size and relations through this paper: movies (genres) [24], character co-occurrences per chapter in *Les Misérables* [22], recipes (ingredients) [2], data breaches (record types) [35], border countries (neighbors) [10], and undergraduate student records (course enrollments). To evaluate our technique with external participants, we conducted an expert review with three visualization experts, and a case study with undergraduate assistant deans at a major university using student enrollment data.

## 2 RELATED WORK

We review the related work on set visualization based on a categorical approach of visualization types from a recent survey [5]. We refer the reader to this survey for a more thorough analysis. We present a focused comparison and discussion of selected recent techniques in Section 8 after presenting AggreSet in full.

**Euler Diagrams:** Sets can be drawn as enclosing boundaries around elements, generating Euler diagrams. Given few set and element counts, Euler diagrams are powerful and can intuitively demonstrate set concepts. However, scalability is an issue. Proposed improvements, such as untangling [29], cannot avoid the inherent visual complexity beyond a few hundred elements and only a few sets, especially when the sets are densely intersecting. An extensive survey of Euler diagrams is presented by Rodgers [31].

**Overlays:** Sets can also be overlaid on existing visualizations that define element positions (layout) by other attributes [3], [12], [13], [27]. Isocontours are commonly used to enclose elements within sets. Their scale is limited by the element count when elements are not aggregated. Elements appearing in many sets also increase visual overlaps and complexity as in Euler diagrams.

**Node-Link and Chord Diagrams:** Node-link diagrams visualize set relationships by mapping sets to nodes and set-pair (second degree) intersections to edges. Visual scalability is primarily influenced by the set (node) count and link sparseness (edge count). Circular layouts (chord diagrams) position set nodes along a circle to bring a spatial structure visually. To allow for richer set exploration on such diagrams, RadialSets [4] is based on an interactive circular layout with degree histograms on the set nodes, and uses edges to represent intersections of two or more sets. It is included in our focused comparison. AggreSet’s design follows previous studies that have shown that when graphs (connected entities) are bigger than twenty nodes, matrix-based visualization performs better than node-link diagrams on many tasks [17].

**Matrix-Based Diagrams:** A matrix layout is made of rows and columns that list values of a data type. Co-occurrence matrices use the set list on both axes, and cells show set pair intersections. Intersections metrics, such as element count, are commonly visualized using color (*heatmaps*). The resulting visualizations are non-intersecting and easy to read. However, such matrices hide information about higher-order set intersections [23]. AggreSet improves on the set-matrix design with its interactive, multi-dimensional approach. Matrix-based diagrams can also be built using different data dimensions for rows and columns. ConSet [21] uses a matrix with rows from elements and columns from sets. Since elements are not aggregated, its matrix view is not scalable by element count. Among the other approaches, UpSet [24] and OnSet [32] are discussed in our focused comparison.

## 3 SET EXPLORATION MODELING

Set exploration is conceptually non-trivial; there are many tasks that involve intersections and relations between multiple sets and other element attributes [5]. To support a rich and comprehensive ability to explore set-typed data, we present a new modeling for data representations, low-level actions, and high-level tasks. Our data and low-level action model is shown in Figure 2. Higher-level tasks, such as comparing across element selections and exploring set relations based on shared elements, are discussed below.

To exemplify the execution of our model, let us consider a movie dataset where each movie (element) has multiple genres (sets), an average rating, and a country of origin. What are the genres, the countries, and the range of ratings in the dataset (**Analyze** within aggregates)? What are the genres and the rating of the movie *Wall-E* (**Retrieve**)? What are the two most common genres (**Analyze** within genres, **Find**)? How many genres does a movie have at most (the maximum genre degree) and what is the degree distribution? (**Analyze** within genre degrees). Such overview reveals basic patterns. Then, exploration expands through selections. What are the drama movies? Movies that have at least three genres? Movies with highest ratings? Such exploration commonly starts with a **Select**, is followed by **Sync** that retrieves and aggregates selected element attributes, in order to **Analyze** data characteristics in multiple data

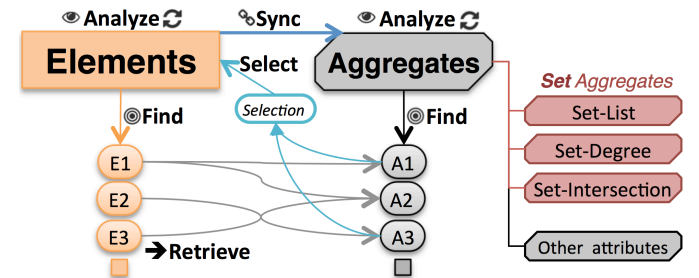


Fig. 2. Our set exploration model for data and low-level actions. Elements are mapped to aggregates, and actions are defined across data types. A set-typed attribute is decomposed into three forms of element aggregates: set-list, set-degree, and set-intersection. This model distinguishes the explicit set-list from set-intersections, and allows for exploration using set-degrees directly. Given a group of elements/ aggregates, you can **Find** an element/ set with some characteristic, or **Analyze** the group overview to detect the range of values and patterns. Given an element, you can **Retrieve** the aggregates that include the element. Given a selection of one or more aggregates, you can **Select** the elements that satisfy the selection. We do not differentiate *how* selection is actualized (i.e. highlighting or filtering). Lastly, given a selected element group, **Sync** is a global action from all elements to all aggregates to reflect underlying element characteristics. **Sync** action generalizes **Retrieve** for selected elements to enable **Analysis** within all aggregates. Sequencing these low-level actions on set list, degree and intersections allows expression of complex queries by creating flexible type-agnostic paths.

dimensions. What is the rating distribution of children’s movies (genre to rating)? What are the common genres of high-rated movies (rating to genres)? What other genres do documentary movies have (genre to genres - set relation)? Which genres have more multi-genre movies (genre degree to genres)? Which genre pairs are more common, which genre pairs include no movies (empty intersections), and which genres always appear together (are subsets) (**Analyze** within set intersections)? We can then compare different selections. How do ratings compare across horror vs. documentary movies (**Select** horror  $\rightarrow$  **Sync**, **repeat** for documentary and **Analyze** for comparison within rating)? We can expand our inquiry by looking at intersections of multiple genres. AggreSet supports all such queries through its single aggregate-based exploration modelling.

Many exploratory questions depend on the **Select** action based on some criteria. Rich data exploration is only possible through flexible selection models, ideally with ease of expression. Selection for set-typed data can include multiple attributes (high-rated drama movies) and multiple set values can be selected using different modalities (family and comedy movies without action), representing intersection ( $\cap$  - and), union ( $\cup$  - or), and complement ( $\setminus$  - not).

Comparison of data characteristics under different selections is a more complex form of exploration. To support comparisons across different element selections, **Select** $\rightarrow$ **Sync** $\rightarrow$ **Analyze** pipeline needs to be executed under each selection, and the resulting distributions need to be saved and visualized. Exploratory comparison then follows visualizations of multiple distributions.

Set-typed data also implicitly define relations between sets (A, B) based on their intersection ( $Q=A\cap B$ ), ordered by increasing strength in Figure 3: disjoint sets, partial sets, proper subsets and identical sets. Revealing these relations are among set visualization goals. *Disjoint* relation ( $Q=\emptyset$ ) represents empty intersection. It is very common in sparsely connected sets. *Identity* relation ( $A=B=Q$ ) represents the strongest connection. It requires both sets to contain the same elements. *Proper subset* relation is the strongest relation when sets have different number of elements. One set subsumes the other, i.e. all elements that appear in the smaller set are also in the larger set ( $A\subset B$ ,  $Q=A$  or  $B\subset A$ ,  $Q=B$ ). Many set-pairs are in *partial* relation. The sets have some shared items, and each set has some unique element compared to the other ( $Q\neq\emptyset$ ,  $A\setminus B\neq\emptyset$ ,  $B\setminus A\neq\emptyset$ ).

To model relations between sets, we define the *strength* of a set pair  $\{A,B\}$  on a continuous scale from disjoint (0) to subset (1), computed as  $|A\cap B|/\min(|A|,|B|)$ . The set-pair intersection gets stronger as the sets share more elements, and the strength reaches 1 when the sets share all the elements they can share. This metric presents a normalized context to set-pair relations, a form of similarity, and is an alternative to characterisation by element count, an absolute value on an unbounded scale.

In contrast, the Jaccard Index, a common set-relation metric, normalizes the intersection size of two sets with their union size ( $|A\cap B|/|A\cup B|$ ), also ranging from 0 (disjoint) to 1 (identical). However, this metric produces an unbalanced distribution since high values (toward equity) are much less likely to occur than strength metric (toward subset-ness) given varying set sizes. There are also other similarity metrics representing deviation from expected values using statistical inference assuming a marginal independence between sets [4], [24]. Such metrics return positive or negative values depending on whether the observed element count is higher or lower than expected. Deviation results can be compared relatively across sets and their intersections, while the *strength* metric is meaningful in absolute form (*subset-ness*) as well as for comparison.

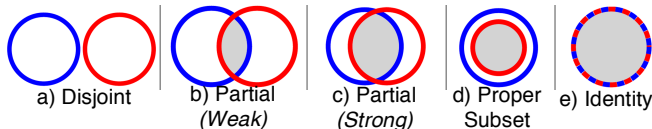


Fig. 3. Relations between two sets based on shared elements.

## 4 DESIGN GUIDELINES FOR AGGRESET

We present a selection of our design guidelines to discuss *how* AggreSet addresses set exploration challenges in scalability and usability through design. The included references present extended analyses and supporting guidelines and discussions.

**Aggregate.** Aggregated visualizations scale to larger data than non-aggregated ones, i.e. screen space limits visualizations of individual data points. Filtering is commonly used to decrease data volumes to manageable sizes, yet overviews are still required at the beginning. We also use aggregation to reduce overlaps and clutter.

**Be consistent** [33], visually and interactively, across similar tasks and representations of different data types. For example, clicking on an aggregate is filtering, and it applies for *all* aggregate representations. Likewise, the selection highlight color applies to *all* affected components in our design.

**Have a tightly connected interactive design.** AggreSet reflects changes in one data dimension to all related visible components. This guideline relates to the **Sync** action in our modelling, and it is applied automatically. Multiple connected views have become a common design pattern applied over many domains and applications [30]. Our tightly connected design also reveals how interface components work as well as the relations in data.

**Avoid overlapping.** Overlap introduces clutter, limits scalability, and makes it harder to observe values and relations [14]. While there are techniques that bring more structure in case of overlaps, improvement in scalability is limited even with advanced methods. We therefore chose non-overlapping matrices for set exploration.

**Avoid duplication.** Having multiple visual representations of a single item uses more screen space, limits scalability, and requires the viewer to encode relations between multiple representations of the same data. While there are methods proposed to reduce overlaps by introducing duplications [20], such methods have not yet received wide adoption or support.

**Have fluid interaction.** *How fast can the user express exploratory or targeted queries?* Fluid design makes information available rapidly and smoothly upon interaction and avoids substantial transformations and explicit mode changes [15]. This guideline influenced our aggregate selection interaction design. In AggreSet, visual changes are animated [19], and visual cues/previews are used to improve the flow of interaction.

Among more generic guidelines, Dieter Rams states, “*Good design is as little design as possible*” [36]. The rule of simplicity in UNIX philosophy emphasises decomposing problems into small, straightforward pieces [28]. The reflection of such guidelines on AggreSet is the explicit separation of set-typed data components, and the minimalist design with few key configurations.

## 5 AGGRESET

We designed AggreSet based on our set exploration modelling and design guidelines. We first present how this technique enables exploration of set, set-pair, and higher order set relations through interaction. Then, we present its features for exploring comparison of selections, relative distributions, disjoint and sub-set relations, relation strength, and set ordering for improved matrix layout.

Exploration with AggreSet encourages the overview-to-detail flow of the information seeking mantra [1]. Its approach can be explained in four levels with increasing depth and richness. (i) AggreSet displays sets as a linear list, aggregates elements within sets, and visualizes the distribution of elements. It orders sets with larger element counts first by default (Figure 4 and 8-a). By selecting a specific set, the user can interactively explore (highlight, filter, compare) distributions of elements of the selected set, also revealing its intersections. (ii) AggreSet summarises the set-degree of elements. Selections on this dimension can be used to reveal higher-order set relationships (e.g. intersections of  $>3$  sets) (Figure 4). (iii) AggreSet introduces the set matrix to visualize the distributions in set-pair intersections and set relations (strength) using circle glyphs. The interaction design (highlight, filter, compare) seamlessly extends



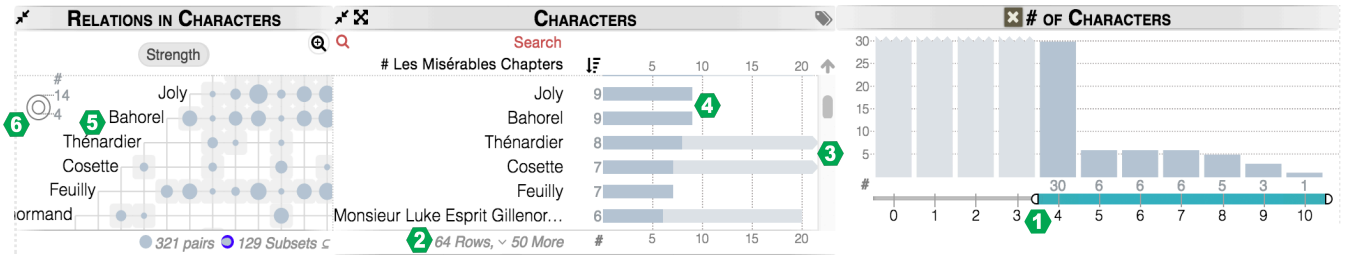











Fig. 4. Character co-occurrences in Les Misérables, with 80 characters (sets) in 356 book chapters (elements). ① Data is filtered to chapters that have at least 4 characters. ② The related 64 characters are reordered by the number of book chapters they occur. ③ Thénardier and Cosette, have ghost-bars (gray extensions), showing that these characters also appeared in chapters with <4 characters while ④ Joly and Bahorel appear only in the chapters with  $\geq 4$  characters. ⑤ Thénardier is one of the common characters, yet he does not appear with Bahorel, Feuille, and some other characters outside of the view (disjoint set pairs). ⑥ The legend shows circle size mapping.

to this matrix. (iv) Intersections beyond second degree (set-pairs) are explored through selections. At all levels, the result list can show all, or filtered, elements (Figure 1), and other categorical and numeric attributes are presented with the same core design as set dimensions.


AggreSet uses element aggregation to scale on element count by design. Element are aggregated per set, per set-degree and per set-pair intersection, as modelled in Figure 2. Since set-pair aggregation is independent of the set order, the set matrix uses half of the matrix and avoids visual duplication. The intersections of a set are captured along two set-lines, one vertical and one horizontal. For example, in Figure 1, action movies are selected and two orange lines in matrix pass through the intersections of this set. The empty half of the matrix displays set labels (for easy identification of sets involved in intersection circles) and visual legend for matrix.






AggreSet visualizes the aggregate distributions as summary statistics using non-overlapping glyphs per each aggregate, using bars for set-list  and set-degree , and circles for set-pair intersections . This visual design builds on studies on human perception [11], [18], [26] to achieve an effective language for exploratory analytical tasks. The length encoding on a shared baseline, which ranks as one of the perceptually strongest visual variables, is used for the primary characteristic where possible: the element count within aggregates. The 2D basis of the matrix view limits the applicability of a length-encoding scheme on a single shared baseline, thus our design uses a circular area mapping for intersection glyphs. In contrast, heat-map designs primarily use the color encoding that is perceptually more limited than the length or size encoding. AggreSet uses color consistently across the interface: to highlight selected element distributions (orange-fill , to allow comparison (black-line , and to show subset relations (blue-border , disjoint sets (grey background ) and set intersection strength (purple shades, ). Following interaction, visualizations are animated to reflect new distributions, and length and area scales are adjusted to fit the ranges of the filtered data.

To explore a high number of sets that cannot fit within the linear and matrix view on a limited screen size, AggreSet matrix supports scrolling and panning, as applied in Figure 1 and 4. Scrolling is a more fluid action to observe limited parts of the dataset compared to

explicitly selecting active sets one-by-one such as applied in Upset [24] and Onset [32]. When the set-list is scrolled, the set matrix follows along its diagonal line so that for all the sets visible on the list, their intersections are also visible on the set-matrix. The intersections involving sets that do not appear in the set-list are outside the diagonal. AggreSet allows these intersections to be explored by panning the matrix view by mouse drag. Notice that sets below the view cannot have any intersections within the matrix view by design. Also, panning reduces the unused portion of the set-matrix view. AggreSet also supports adjusting the matrix cell size (zooming -  button) to make the circles easier to read, or to show more set-pair intersections in a single view (Figure 8).

### 5.1 Search and Analysis through Aggregate Selections

Understanding characteristics of selected elements is fundamental in data exploration. To enable this goal rapidly, AggreSet features a two-step hover-and-click interaction that applies across all aggregate types. When the mouse hovers on an aggregate glyph, AggreSet highlights the characteristics of elements within that aggregate visually across all other aggregates (Figures 1, 5). We call this linked brushing feature the *result-preview*. After previewing an aggregate selection by hovering, click action filters into the selected elements and exploration continues on the filtered data. Clicking on  of the aggregate sets it for comparison across other selections.

**Result-Preview:** The *result-preview* visualizes the attribute characteristics of elements within a hovered aggregate. Given a set-typed (multi-valued) attribute, by hovering on a specific set, the result preview visualizes its relations to other sets (Figure 1, movie genres), a simple, interactive design alternative to set-o-grams [16]. The *result-preview* is activated on hover, a precursor to filtering on clicking. This interaction is immediate and fluid; sweeping the mouse over aggregates visualizes distributions of aggregated elements rapidly. On bar charts, the *result-preview* is visualized with orange bars along the shared linear scale . In the set-matrix, the *result-preview* is visualized with a sweeping arc on circles with 12 o'clock alignment, producing a pie chart with single pie . Our design uses a sweeping arc (instead of radius mapping) to emphasize part-of relations within intersections.  ( $\frac{1}{4}$ ),  ( $\frac{1}{2}$ ), and  ( $\frac{3}{4}$ )

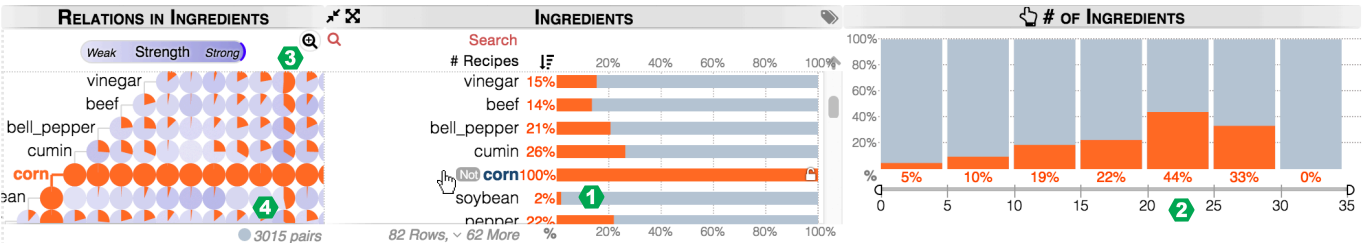



Fig. 5. 313 ingredients (sets) in 5,000 recipes (elements). The *relative-mode* is active; each aggregate glyph is scaled to its maximum size (length or radius), creating a shared percent scale. The orange *result-preview* shows the distribution of selected  corn among all aggregations in percentage. ① Corn is rarely used with soybean; 2% of recipes with soybean have corn. Corn is popular in recipes with high number of ingredients. ② At the peak, 44% of recipes with 20 to 25 ingredients have corn. In the matrix, recipes with the second rightmost ingredient (positioned above the view, tomato) frequently have corn, such as ③ half of vinegar-tomato, and ④ half of soybean-tomato recipes.

serve as easily recognizable visual anchors for comparison of previews to (filtered) element count. If radius mapping by area is used to reflect selection areas, such ratios are harder to perceive, such as  $\frac{1}{4}$ ,  $\frac{1}{2}$ , and  $\frac{3}{4}$ . We notice that the visual distance between circles and the lack of a shared basis can be limiting factors for effective comparisons across set intersections within the matrix.

**Filter:** Elements in a selected aggregate can be filtered by clicking. The visual distributions follow the *result-preview* in an animated transition. *Ghost-bars* are used to show the original distributions and provide a visual cue for the effect of filtering (Figure 4 and 9). The filtering interaction (mouse click) is explicit and view transforming. Filtering can be removed per selected aggregate (click on set), per-summary (x), or per all (All x). Three filtering modes are supported in the set-list: *and* ( $\cap$  - click on set), *or* ( $\cup$  - click on Or) and *not* ( $\setminus$  - click on Not). The or/not buttons are shown on mouse-over only. Figure 9 shows multiple selection modes applied to courses (sets) taken by students (elements). Clicking on a set-matrix cell (set-pair glyph) enables  $\cap$  filtering with two sets.

**Compare:** AggreSet enables comparison across selections using an interactive one-vs.-many design. When the user clicks on a to select the compared aggregate, the black *compare-lines* (—, —) are inserted (Figure 1). By moving the mouse over different aggregations, the user can visually compare distributions of the compare-lines vs. the result-preview (of other selected aggregate). Once the compare-lines are shown, the comparisons are fluid with the rapid result-preview. Thus, AggreSet encourages one-vs.-many exploration using simple mouse movement. The *compare-lines* support exploration as both targeted and serendipitous activity. Since the *compare-lines* are inserted on top of the *result-preview*, the design has a natural flow. The black lines are the only visual addition that enables richer comparative goals; avoiding visual clutter while preserving the design minimalism. The *compare-lines* are removed when the compared aggregate is unlocked.

AggreSet enables exploration beyond set-pair relations by selection across set dimensions. Figure 7 shows that the *result-preview* selection on a set-pair enables analysis of intersections of three and four sets visually. Set-degree selection also enables higher order analysis. For example, to analyze intersections that involve 4 or more sets, one can filter to elements with degree 4+, as shown in Figure 4. Likewise, selection by an exact set degree will show set relations unique to intersections that only involve as many sets. Quickly iterating through different set-degrees by result-preview can provide a quick overview of higher order relations within the data.

## 5.2 Exploration of Relative Distributions

AggreSet visualizations are based on the absolute value of element count per each aggregate by default. Alternative insightful analysis can focus on the relative frequencies within aggregates. For example, how common (by ratio) are action movies across all movies in 2012,

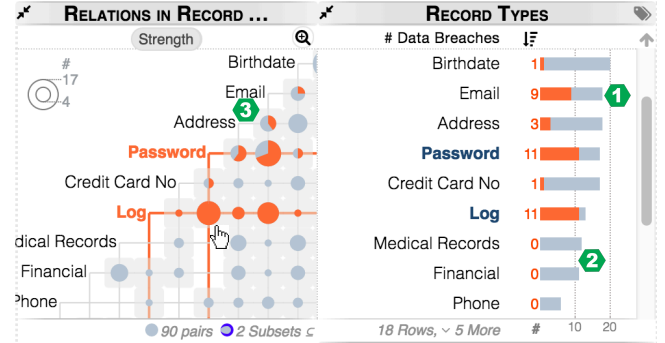


Fig. 7. Record types (sets) compromised in 284 large-scale data breaches (elements). 11 Breaches with *log* and *password* record types are selected using result-preview. The large circle size shows these two record types were commonly compromised together. 3<sup>rd</sup> order intersections ( $\cap$ 's of 3 sets) are shown on the set-list histogram. For example, ① *email* is commonly associated with the selected breaches (9 out of the 11 with *password* and *log*), and ② neither *medical* nor *financial* records were stolen with *passwords* and *logs*. We can also observe intersections of 4 record types. About 35% of *email* and *address* breaches also had *password* and *log* leaks ③.

and what is the trend of this ratio over years? Another form of relative distribution analysis is set-pair relation strength (Figure 3), describing the ratio of elements shared between sets rather than the absolute element count of their intersection.

To enable observations of relative frequencies, AggreSet features the *relative-mode* (Figure 5). This mode enables exploration of the percentage distributions with respect to the total number of (filtered) elements per each aggregate under preview/compare selections, and also visualizes set-pair relation strength in matrix view. In this mode, the scale range of bar and circle glyphs are maximized. This also allows observing aggregates with fewer elements in larger detail, both for bar and circle glyphs.

The *strength* of the relation, as defined in Section 3, is mapped to the circle color and border (Figure 6). Lighter color visualizes a weaker relation than darker color (light blue vs. dark blue). The circle border visualizes subset relations. A full border (dark blue) shows the identity relation, while a half-border (light blue) shows the proper subset relation. The edge connecting the half-circle (upper or right) directs to the larger set. When the sets are ordered by element count, the containing set always appears above since it is larger. Yet, this property may not hold for other ordering approaches and the visual state encodes the direction. The total number of subset relations is also shown below the set-matrix, next to the total number of intersecting set pairs. To maintain design consistency, AggreSet recomputes the set strength metric after filtering. The *relative-mode* is engaged by clicking the bar chart scale axes, designed as a direct interaction with the conceptual change in bar charts, or by clicking

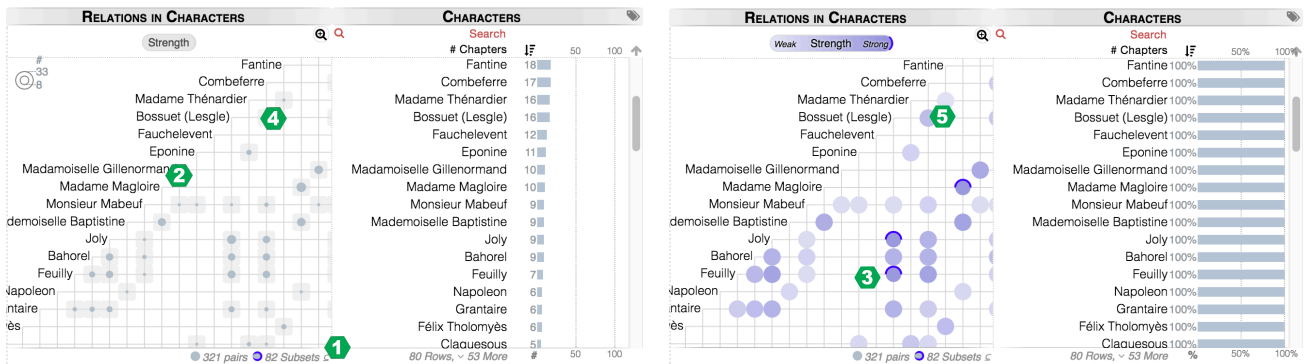


Fig. 6. Character co-occurrences in Les Miserables. ① This dataset has 82 subset relations. **Left:** The circle area maps the number of chapters both characters occurs in. ② Intersections with few chapters appear small and are hard to observe. **Right:** The circles are full and color denotes the character relation strength by the chapters they occur in together. The border is shown when one character always appears with the other character. For example, ③ all of Feuilly's chapters (7) also include Bossuet, who appears in 16 chapters. This suggests a proper-subset relationship, and the border is half. When two characters always appear together, their border is full (not visible in this cross section). We can also observe that while intersection of Madame Thenardier was one of the largest in number of chapters ④, it is *not* one of the strongest ⑤.

the **Strength** button. The strength button changes to **Weak Strength Strong** when relative-mode is enabled, describing the visualization of the strength relation with its gradient, and the blue border at the strong end. This design is limited for analysis of hierarchies of subsets, although hierarchies can be traced using the set matrix step by step.

When all circles (non-empty intersections) are scaled to full-size in the *relative-mode*, the disjoint-sets (of empty space) become visually more distinctive. The matrix layout creates a spatial context for observing sparseness of set intersections. In the absolute mode with varying circle size, AggreSet uses the grey cell background to help the viewer distinguish the small circles (few elements) from empty intersections (cells). Some sets may also be disjoint from *all* others (like disconnected network nodes). To distinguish such *isolated* sets, AggreSet removes their grid-lines, suggesting that there is no line to follow to uncover set-relations. This design reduces

chart ink and makes existing lines easier to perceive.

### 5.3 Perceptual Set Ordering for Set Matrix

The Gestalt principles state that our perception is influenced by similarity, continuation, closure, and proximity. Jacques Bertin says “simplification is no more than regrouping similar things” [8]. Characteristics of set visualizations and visually emphasized patterns therefore depend on the set order. To reveal patterns among sets that are closely related, AggreSet includes a perceptual set ordering method aimed for the set-matrix layout. Figure 8 shows that ordering sets on element count may create salt and pepper pattern within the set matrix, and perceptual ordering can improve visual structure by placing connected sets along the diagonal.

Matrix reordering methods have been long studied [25]. Greedy heuristics and clustering are commonly used approximate solutions since ordering optimization is NP-complete in the general case given  $\#sets!$  combinations. In AggreSet, set ordering is solved once as an approximate global layout optimization, since both matrix axes use the same order. We translate set ordering to the Minimum Spanning Tree (MST) problem by using sets as nodes, and set-pair intersections as undirected edges. The edge weight between two sets for MST is the total dissimilarity in their relation to all sets, such that  $\|AB\| = \sum_x |A \cap X| - |B \cap X|$ , where  $A, B, X \in \mathcal{U}$ . The intersection size  $|a \cap b|$  is used as the visual characteristic of the set-pair, i.e. the metric to optimize the matrix layout. To reduce the number of edges to be processed, we consider only intersecting set-pairs, such that  $A \cap B \neq \emptyset$ . This edge weight is defined for the MST algorithm to optimize the layout globally, and is not exposed visually otherwise.

To generate MST(s) of the set-intersection graph, we use Kruskal’s algorithm, which greedily inserts edges with smaller weight (higher set similarity) to MST(s). We generate the linearized set ordering by a breadth-first traversal of MST(s), starting with the largest tree in terms of the number of nodes (sets). To have a consistent linearization with larger sets within a tree appearing before smaller ones, larger nodes need to be traversed first. To achieve this, we modify Kruskal’s algorithm such that when two nodes are connected, the node (set) with more elements becomes the new root. Our open-source implementation provides more details.

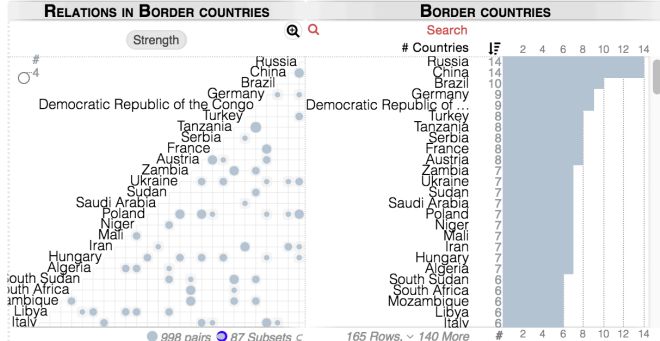
### 6 IMPLEMENTATION NOTES

Our implementation runs on modern web browsers based on HTML5, CSS3, SVG, and JS. We used D3 [9] to link data to interface. Datasets are fully loaded, indexed, and filtered in memory. To enable fast selections, aggregates index their elements, and elements links to their aggregates. Elements are incrementally added and removed upon selection. Our implementation, source code, and 20 datasets are publicly available at [www.keshif.me/AggreSet](http://www.keshif.me/AggreSet).

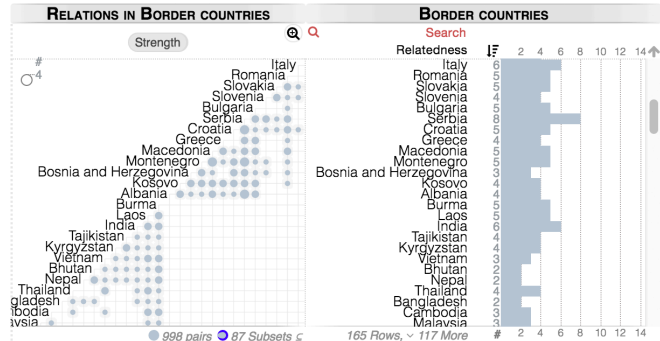
In this paper, our scalability discussions are based on the visual scalability of the design. The rendering and selection speed of web browsers, which our implementation is based on, is a limiting factor on scalability of our implementation. On a MacBook Pro laptop with 2.3GHz Intel Core i7 processor and 8GB DDR3 RAM using the Safari browser, 175,000 elements spread across 131 sets with 2,300 pairwise intersections can be interactively explored. The performance, however, varies between browsers and data characteristics, such as set-relation sparseness. To improve the performance, future implementations can use higher performance graphics APIs, e.g. OpenGL, and distributions based on aggregate selections can be computed by server-side implementations.

### 7 EVALUATION

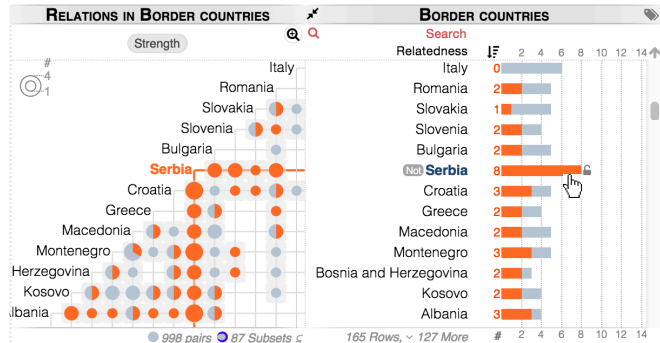
To evaluate our design, we conducted user studies with two complementary approaches. First, we conducted expert reviews to identify strengths and weaknesses of AggreSet as observed by visualization experts using multiple datasets. Expert reviews in visualization have been shown to help detect usability and design issues, and yield qualitative results [34]. Second, we conducted a



(a) A zoomed-out view sorted by decreasing element (neighbor country) count. This view emphasizes countries with more neighbors. Pair-wise relations between 25 countries are visible. Notice the salt-pepper pattern in the set-matrix.



(b) Countries are reordered using a perceptual set ordering approach. The new ordering follows their geographical closeness, for many countries, and forms visual clusters along the diagonal.



(c) A group of 13 countries is focused by adjusting the matrix zoom. In this group, Serbia has the most neighbors, and is selected by mouse-hover. This selects the neighbors of Serbia, and the preview shows the neighbors of those countries.

Fig. 8. Exploring country neighborhood relations. The list aggregate number shows the number of neighbors per each country (set).



case study where domain experts analyzed complex data, with the aim of uncovering the usability and usefulness of AggreSet and analysis strategies. In both evaluations, we collected qualitative feedback on usability and design features during the studies and in semi-structured post-study interviews. We used the feedback to improve AggreSet design and to identify future work.

## 7.1 Expert Review

We recruited three visualization experts (senior researcher P., graduate student D., and industry professional F.) and asked for their honest feedback in 1.5-hour sessions. We first used the movie dataset to demonstrate set exploration in multiple dimensions and set-pair strength. We followed with the Les Miserables characters dataset to demonstrate subset relationships and perceptual set ordering. We encouraged the participants to think aloud, and interrupt at any point to ask questions, make, and share observations. The following summarizes some of their comments and observations.

Before introducing the matrix view, we asked D. which movie pair would have the biggest intersection, to which he replied “I cannot tell, I don’t have the overview. If I knew which ones to compare, I’ll use (selection), but I don’t know. You need other ways to see which pairs are most interesting”. With genre matrix enabled and high-rated movies previewed, he said, “The *drama and war* (movies) seems to be very good... I immediately found (the intersection). Now I want to see the release date of war and drama, and 4-star rating”. By filtering and selection, he found some movies he liked. This exemplifies the utility of set-matrix view.





The participants also developed strategies to effectively explore data using AggreSet. F. noted, “The bar chart serves as a key to the matrix.” He continued “For navigation, you have the matrix,... the 2D space you are maneuvering in... For interpretation, it is good to look back at the bar chart... That is two of them complementing each other”. Upon selecting a genre-pair intersection and analyzing the selections for a while, F. said, “You are actually showing, out of the intersection of 2 things, multiple set of intersections... It is a little bit of a mind-bender”. D. commented likewise upon selecting comedy, “In other views, it tells me the percentage of *comedy* in those overlaps of the other movies... I am comparing three basically”.

When explaining the potential complexity of the interface, F. said, “It is a lot of information. Once the person masters it, and then they have at their fingertips a lot of information in a very little space. It is just that getting there takes some effort. I understand you are trying to minimize that effort so that the user can quickly master the way to interpret this chart”. This follows our suggestion that intersection characteristics should be queried after the set-list and set-degree, as part of overview-to-detail exploration. As F. notes, “When you hover with your mouse on top of the matrix, showing (previewing) those intersections is when it is a little overwhelming”. Commenting on matrix readability, F. also said, “Interacting with the matrix on the horizontal level and on the vertical level (for a single set), that takes some time. It is not something that comes to you immediately, like differences in (strength) colors do”.

The participants found the zoomed-out matrices dense overall; visualizations on small circles were not easy to observe. However, D. added, “This makes sense. I start with the overview, and then I drill down to the area... It helps me... because I have made some observation based on the high-level small pie chart. I want to confirm, so I will drill down and see exactly how it looks like.”

The relative mode with percentage distributions was favored among all participants; P said, “I like this (percentage) view better for doing... complex queries”. Subset relations were found the most complex concept, although the participants could understand the relation and encoding through some exploration. At the start, F. noted, “I am trying to understand why (circles) have outline... Three states: Total outline, half outline, and no outline.” After exploration,

F said, “This is one that I think some teaching aid would be great.” And P. said, “I like that I was able to do it, but it was hard.”

We implemented several changes to our design following the expert reviews: (i) An earlier design visualized set similarity (strength) by mapping to circle size. This made understanding circle-size mapping harder as it overloaded the element-count mapping. We updated our design to use color-coding for strength metric as suggested, and to use circle size for element count only. (ii) We noted that color-coding was ineffective with varying and small circle sizes with the cell background. Thus, in relative-mode (strength), we chose to use full-size circles and remove cell-background. (iii) We linked *relative-mode* and strength metric, effectively encoding strength as a *relative* set-pair metric. This simplified AggreSet while making it easier to understand and use. (iv) Our earlier design used a 3-second mouse point-wait to select an aggregate for comparison. D. stated, “Hovering means I am thinking, it doesn’t mean I want to compare”, and P. said, “I’d like to turn it off when I don’t want it.” Users converted to using their hands to point things instead of using mouse, changing their behavior to overcome the issues with the specific design. We then designed an explicit control using , which also visually reveals the selected aggregate. (v) Our earlier visual design for comparing distributions (black lines) was an enclosing section () which suggested stacked-charts semantics for some users when previews were enabled () thus complicating the visual language of AggreSet. We changed the bordered design in favor of a simple bar extending from the baseline ().

## 7.2 Case Study

We conducted a case study with two assistant deans of the undergraduate studies department of a large public university analysing student degree and course enrollment data. First, the participants had access for a few months to a version of the visualization without the set matrix, but with histograms and the data preview and selection. This allowed them to look at categorical and numerical aspects of the multivariate student records, including set-typed data using set-lists and set-degrees. They used the tool a few times on their own during this period. After we developed the set matrix, we performed data exploration including the matrix view in a 1.5hr session with the two participants together. Our aim was to capture the cognitive and reasoning processes of novice visualization users with rich data in a limited time using AggreSet. Thus, we used pair analytics [6]. The participants collaboratively formed questions, observed data, and generated insight. One of the authors of the paper acted as “driver”, demonstrating features (from set-list overview to set-matrix detail) and expressing their queries.

First, the participants analyzed 175,000 students and the degrees they received, along with their birth year and gender to provide context. 131 most common majors with at least 100 students were the sets over students (elements). (i) Early in the exploration, the participants wondered why there were multiple majors on “*Math*”. The driver performed a search within the degree-list to select all majors with “*Math*” (a  $\cup$  query by text input). The resulting visualizations supported their hypothesis that one of the “*Math*” sets was “*Applied Math*”. (ii) When the driver previewed the *Economics* selection, they observed the other degrees received by students in *Economics*. (iii) They wanted to explore students who did not receive a degree. First, they tried to generate hypotheses about their distribution trends and what the data represents, such as whether the declared yet unfinished degrees were included in the reported numbers. Upon selecting 0-degree students, they noticed these students were younger, suggesting many were possibly still taking courses. To improve their outlier analysis, they wished for more data context in the browser, such as entry term and majors declared. Upon selecting students with 1-degree, they noted, “Those (selected) are all the people that earned 1 degree... (The rest) are the ones with double majors”. (iv) The driver then enabled relative mode. Upon selecting females, they noted “67% of the sociology students are female. It

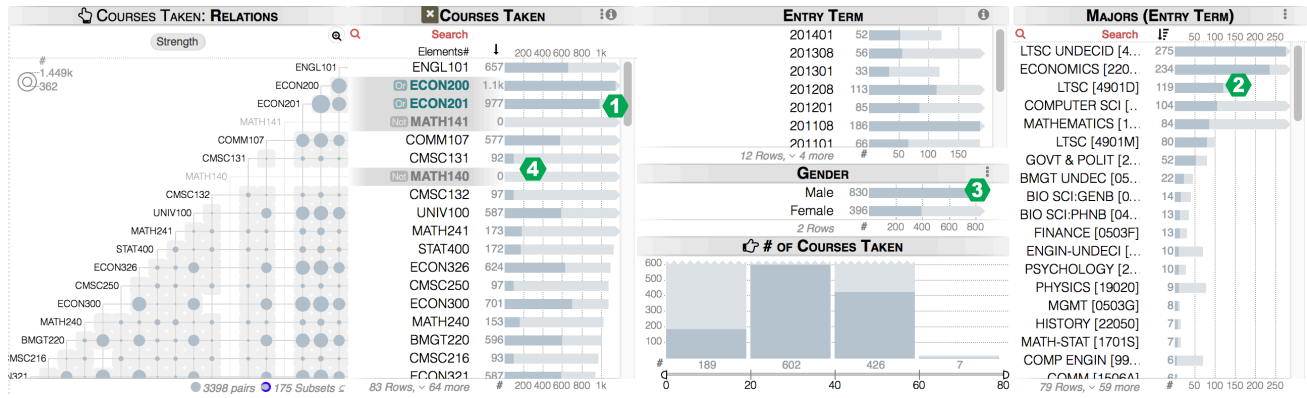


Fig. 9. 83 courses taken by 4,300 students. ① Students who took ECON200&201, but not MATH140&141 are selected. Updated distributions show the other courses taken, the total number of courses, gender, entry term, and the majors of these students. ② Most of these students are in LTSC (undecided), or in ECON, ③ as well as with a male majority. We can notice emerging patterns in the top courses after filtering. ④ In this filtered selection, CMSC courses are less common than other ECON and core courses.

makes more sense this way”. Upon selecting 1-degree students again, they noted some majors had very few students with multiple majors, enriching their knowledge of the more demanding majors.

Next, the driver showed the major (set) matrix. One participant immediately pointed out “this means there are more people that have accounting and finance. The bigger gray circle means there is more people”. When the driver asked about any trends they detected, one said “All those double majors with X... Department of X would be very interested to see this”. Since only a limited number of majors could be shown at once, one asked, “Does it ever get wider this way?”, suggesting outside the triangle, at which point the driver panned the set-matrix. They explored various departments and their intersections through rapid result previews. Then, the driver enabled major-pair strength visualization. First, enlarged circles made it easier for them to see intersecting majors, as it was a stronger cue than the gray cell background in the default view. One noted, “Darker color means a higher percentage than the one next to it (lighter)”, while the other complimented this statement by saying, “When we looked at that gray view, it was actual numbers.” After further discussion, they concluded, “While there are a lot of marketing and finance (students), there is more accounting in finance, of the total numbers.” Few students received three or more degrees, limiting exploration of higher-order intersections in this dataset.

Next, they analyzed 4,300 students and the 83 most-registered courses (Figure 9). They noticed that few students took 50 or more courses. Note that the sets (courses) are densely connected, and the set-degree distribution has a wide range. By selecting those students, they explored their majors and courses, and generated insights regarding degree programs, and potential effects of course count on student success. They also noted “This isn’t showing courses they are taking *above* what they would have needed”. They needed a new form of set-summary that would show the additional courses the student is taking compared to declared major requirements, a more complex data setup. When the matrix view was shown, they noted large pair intersections of some common core courses (such as English), as well as courses that are prerequisites to others. Noting of their previous experience analyzing this data without the matrix view, one said, “This view would have allowed us to do what we wanted to do more easily than what we did. What courses they take, and what they take together”. When the strength metric was enabled, they noticed courses that had consistent colors among all its intersections, which meant that they had no strong relationships with others. They went on to analyze common properties among students that did not take some specific courses.

## 8 COMPARISON AND DISCUSSIONS

In this section, we present a focused comparison of recent set exploration techniques, including AggreSet. Table 1 summarises the

results under data, task, set-relation, design, and scalability questions. We discuss each technique separately below.

**UpSet** [24] uses a combination matrix and table layout. In the matrix view, columns are (active) sets, rows are all possible intersections of these sets, and cells show the intersecting sets per row. Per each row (intersection), the tabular view shows the cardinality, deviation, and summary attribute statistics using sortable columns. Since UpSet explicitly shows all set intersections, it is effective for analysis of high-degree intersections as well as attribute characteristics per each intersection. UpSet answers  $\cap$ - $\cup$ - $\setminus$  set queries by selecting and grouping intersections that satisfy the query. Grouping and sorting features for intersections extend its linear basis of design, yet these features apply view transformations that may not be intuitive on first use. As the active set count increases (more sets are inserted to the view), the combinatorial growth in number of rows and the widened matrix view reduces its visual scalability. Targeting sparsely connected sets, UpSet can reduce the number of rows by removing empty intersections. Set-attribute filtering is visually separated from filtering other attributes, while AggreSet uses the same selection modalities across data dimensions. UpSet does not visualize element degrees explicitly, although it offers a range filter and grouping by degree. In its element view, it also does not explicitly show, or link to, set memberships. Overall, when set exploration needs to focus on all possible set intersections and their characteristics given some chosen sets, the interactive tabular view of UpSet provides a rich visual exploratory space.

**RadialSets** [4] is based on the circular layout node-link diagram design, thus has the scalability limitations by intersecting edges. The distribution of element degrees is explicitly visualized by length encoding for each set (node), and revealed upon selection for set intersections (links). RadialSets can also visualize intersections of three or more sets using circular glyphs as hyper-edges. The positions of these glyphs are optimized to visually reduce overlaps, or placed in layers sorted by glyph sizes. Thus, understanding higher degree set relations relies either on tracing overlapping edges, or on selecting glyphs to see contributing sets. RadialSets also supports mapping other attribute characteristics to the color of set-intersection glyphs, allowing high-level overviews of differing characteristics of set intersections.

**OnSet** [32] visualizes elements as cells within set matrices. A matrix can represent a single set, or a set combination. Elements are located at the same cell positions across matrices, and can be spatially grouped by bounding boxes. OnSet matrices should be large enough to hold all elements, limiting scalability on element count. Sets can be dropped and merged with direct manipulation. Merge queries support  $\cap$ - $\cup$ - $\setminus$  modalities with hierarchical compositions. When a matrix represents a set combination, cell (element) opacity/color shows the number of sets, of the combination, that the element appears under. Yet, the sets of the elements are not directly



available. To visualize similarity across set matrices, OnSet supports a node-link diagram. This layer is visually limited in the number of (large) matrices because of occlusions. OnSet relies on pan-and-zoom interaction on a 2D zoomable canvas to explore non-trivial number of sets and relations. However, element context can be lost when zoomed out, and controlling the canvas can make the canvas space more complex to navigate and understand [7]. Its matrix design depends on the viewer’s ability to understand which elements are located at which cells across matrices. Yet, element ordering and grouping structure is not explicit, and finding a specific element across multiple matrices with many rows and columns is a non-trivial task.

**AggreSet** supports a high number of sets, visualizes all set dimensions explicitly, enables the tasks consistently across data dimensions and attributes, supports rich, high-level exploratory goals, and avoids major design problems that may affect scalability and usability. It can be used to express the set exploration tasks proposed by Alsallakh et al. [5] through selections of five data dimensions (elements, set-list, set-degree, set-intersection and other attributes), except the three tasks relating to creating new sets from specific element selections, and analysis of inclusion (subset) hierarchies. AggreSet is also different from other multi-view visualization systems [30] with its novel combination of set-matrix view with element aggregations, set-exploration specific features (such as set-pair strength and perceptual set ordering), and interaction design with preview, filter, and compare models. The limitations of AggreSet can be discussed as the following:

	AggreSet	UpSet	RadialSets	OnSet
<b>Scale</b>	# Elements	Aggr.	Aggr.	100s
	# Sets	50+	20-50	N
	# $\cap$ (Intersect.s)	$(\#Row)^2$	#Row	$(\#Set)^2$
<b>Data</b>	Elements	✓	✓	✓
	Sets	✓	✓	✓
	Degrees	✓	Group,filter	✗
	Attributes	✓	✓	✗
	$\cap$ Degree	2-4+	N	N
<b>Actions</b>	$\cap$ as	Cell	Row	Arc/Circle
	Retrieve	✓	✓	✓
	Analyze	Sets & Elements	Sets & Elements	Element focused
	Synchronize	✓	✓	Partial
<b>Features</b>	$\subset$	Yes hierarchy	✗	✗
	$\emptyset$	In-context	Remove	✗
	$\cap \cup \setminus$	Mixed	Mixed	Rich
	Similarity	✓	✓	✓
	Compare Dist.	1-to-many	Tabular	No
<b>Design</b>	Higher-Order	Preview,filter	Visible	Choose 2-4
	Matrix-View	Set x Set	Set x $\cap$	N/A
	Element Aggr.	✓	✓	✓
	Overlapping	✗	✗	Yes
	Animated	✓	✓	✗
<b>Highlight-Select</b>	Hover, brush	Within matrix only	✗	✗

Table 1. A comparison of interactive set exploration approaches. **Scale** group shows practical limitations in scale per data type. **Sets** shows active number of sets.  $\cap$  shows number of intersections that can be visible on the screen. **Data** group shows the data dimensions explicitly shown. In *Degrees*, “Filter, Group” shows that degree is not a primary data type; it is explored by grouping and filtering in separate interface. **Actions** group shows low-level actions. *Partial sync* means not all components in the interface are connected. **Features** enable higher-order and set-specific exploration.  $\subset$  shows whether subsets are explicitly visualized; 1 denotes subset hierarchies are not explicit.  $\emptyset$  (empty sets) can be highlighted in-context, or can be removed from display. *Similarity* of set-pairs includes deviation from expected values. *Comparison of distributions* can be enabled as *1-to-many*, in tabular form, or using color mapping. *Higher-Order* shows how intersections of many sets are explored. **Design** group lists design guidelines. *Matrix* row shows the matrix view construction.

(i) *Higher-order relations*: Exploring relations beyond set-pair are not immediately visualized and such exploration requires selection. In our overview-to-detail approach, this is presented as the final (fourth) level. Since explicitly visualizing higher-order relations increases the number of visualized data items, placing this information on demand through interaction allows our design to visually and seamlessly scale to overviews of more sets.

(ii) *Set intersection*: Element attribute characteristics cannot be shown within the set visualizations directly, while UpSet and RadialSets support such cases. Relations between sets and other attributes are explored through explicit selections in the minimalist design that consistently applies in both directions (set  $\Leftrightarrow$  attribute).

(iii) *Data density*: When aggregation glyphs are small, the visual mappings (size and color) can be hard to distinguish, especially for circles in the matrix view. To mitigate this problem, matrix zooming can be used to enlarge the glyphs, a tradeoff between space and number of data points. In addition, result-preview and set-pair strength uses the same visual channel (color) in matrix view, with the dominant being orange preview. While the strength is occluded on the circle, it is still available in the set-list view, right side of the matrix, in % value. This also highlights how set-list and set-matrix support one another.

(iv) *Scalability*: Given a laptop/desktop display (1280x800 pixels or more), AggreSet can accommodate on the order of 50 sets. Zooming out shrinks set and cell visualizations, and allows showing more data in a fixed display size. Panning allows exploring areas outside the visible matrix viewport. Perceptual ordering can improve the visual structure along the diagonal for some set relations and reduce information outside of the visible matrix area. Scaling to hundreds of sets with dense relations is still not practical, which would require techniques for aggregating sets and their intersections.

## 9 CONCLUSION AND FUTURE WORK

We have presented AggreSet, an interactive visualization technique for exploring relations in set-typed and other attributes of multivariate datasets using a rich, scalable, clutter-free visual interface. AggreSet improves upon existing set visualization approaches using data aggregation that gracefully scales to larger set counts. These aggregations are displayed as a collection of linked data summaries that are synchronized on interaction. The set-matrix improves the non-overlapping co-occurrence matrix design with advanced visual encodings for set-typed data, and with interactions that reveal higher order relationships. Our user evaluations include both an expert review as well as extended case studies with domain experts trying to understand complex multivariate datasets.

In the future, our data model and design can be extended to support set-dependent attributes by storing extra information along with the set membership relation. For example, the simple set-typed data model can encode the club memberships of a person, yet cannot encode the join-date and cost of each membership. Set memberships can also change in time, requiring focused, topological analysis through time dimension. Representing fuzzy set memberships is also another challenge. Finally, we are also interested in exploring how our mouse-based interaction model can be extended to other types of interaction, particularly multi-touch.

## ACKNOWLEDGMENTS

The authors wish to thank user study participants, reviewers, and the curators of the datasets used in this study.

## REFERENCES

- [1] C. Ahlberg and B. Shneiderman, “Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 313–317.

- [2] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Scientific Reports*, vol. 1, Dec. 2011.
- [3] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski, "Design Study of LineSets, a Novel Set Visualization Technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2259–2267, Dec. 2011.
- [4] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser, "Radial Sets: Interactive Visual Analysis of Large Overlapping Sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2496–2505, Dec. 2013.
- [5] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges," presented at the Eurographics Conference on Visualization (EuroVis), Swansea, Wales, UK, 2014, pp. 1–21.
- [6] R. Arias-Hernandez, L. T. Kaastra, T. M. Green, and B. Fisher, "Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics," in *2011 44th Hawaii International Conference on System Sciences (HICSS)*, 2011, pp. 1–10.
- [7] B. B. Bederson, "The Promise of Zoomable User Interfaces," in *Proceedings of the 3rd International Symposium on Visual Information Communication*, New York, NY, USA, 2010, pp. 2:1–2:1.
- [8] J. Bertin, *Graphics and Graphic Information Processing*. Walter de Gruyter, 1981.
- [9] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
- [10] Central Intelligence Agency (CIA), "The World Factbook," *The World Factbook*, 10-Feb-2015. [Online]. Available: <https://www.cia.gov/library/publications/the-world-factbook/>. [Accessed: 10-Feb-2015].
- [11] W. S. Cleveland and R. McGill, "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, Sep. 1984.
- [12] C. Collins, G. Penn, and S. Carpendale, "Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1009–1016, Nov. 2009.
- [13] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg, "Kelp Diagrams: Point Set Membership Visualization," *Computer Graphics Forum*, vol. 31, no. 3pt1, pp. 875–884, 2012.
- [14] G. Ellis and A. Dix, "A Taxonomy of Clutter Reduction for Information Visualisation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1216–1223, Nov. 2007.
- [15] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly, "Fluid Interaction for Information Visualization," *Information Visualization*, vol. 10, no. 4, pp. 327–340, Oct. 2011.
- [16] W. Freiler, K. Matkovic, and H. Hauser, "Interactive Visual Analysis of Set-Typed Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1340–1347, Nov. 2008.
- [17] M. Ghoniem, J. Fekete, and P. Castagliola, "A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations," in *Proceedings of the IEEE Symposium on Information Visualization*, 2004, pp. 17–24.
- [18] J. Heer and M. Bostock, "Crowdsourcing graphical perception: using mechanical turk to assess visualization design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2010, pp. 203–212.
- [19] J. Heer and G. Robertson, "Animated Transitions in Statistical Data Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, Nov. 2007.
- [20] N. Henry, A. Bezerianos, and J.-D. Fekete, "Improving the Readability of Clustered Social Networks using Node Duplication," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1317–1324, Nov. 2008.
- [21] B. Kim, B. Lee, and J. Seo, "Visualizing Set Concordance with Permutation Matrices and Fan Diagrams," *Interact. Comput.*, vol. 19, no. 5–6, pp. 630–643, Dec. 2007.
- [22] D. E. Knuth, *The Stanford GraphBase: a platform for combinatorial computing*, vol. 37. Addison-Wesley Reading, 1993.
- [23] A. Lex and N. Gehlenborg, "Points of view: Sets and intersections," *Nature Methods*, vol. 11, no. 8, pp. 779–779, Aug. 2014.
- [24] A. Lex, N. Gehlenborg, H. Strobel, R. Vuilleminot, and H. Pfister, "UpSet: Visualization of Intersecting Sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1983–1992, Dec. 2014.
- [25] I. Liiv, "Seriation and matrix reordering methods: An historical overview," *Statistical Analysis and Data Mining*, vol. 3, no. 2, pp. 70–91, Apr. 2010.
- [26] J. Mackinlay, "Automating the Design of Graphical Presentations of Relational Information," *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, Apr. 1986.
- [27] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer, "KelpFusion: A Hybrid Set Visualization Technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 11, pp. 1846–1858, Nov. 2013.
- [28] E. S. Raymond, *The Art of UNIX Programming*, 1 edition. Addison-Wesley, 2003.
- [29] N. H. Riche and T. Dwyer, "Untangling Euler Diagrams," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1090–1099, 2010.
- [30] J. C. Roberts, "State of the Art: Coordinated Multiple Views in Exploratory Visualization," in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007. *CMV '07*, 2007, pp. 61–71.
- [31] P. Rodgers, "A survey of Euler diagrams," *Journal of Visual Languages & Computing*, vol. 25, no. 3, pp. 134–155, Jun. 2014.
- [32] R. Sadana, T. Major, A. Dove, and J. Stasko, "OnSet: A Visualization Technique for Large-scale Binary Set Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1993–2002, Dec. 2014.
- [33] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5 edition. Boston: Prentice Hall, 2009.
- [34] M. Tory and T. Moller, "Evaluating visualizations: do expert reviews work?," *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 8–11, Sep. 2005.
- [35] "Breach Level Index," *Breach Level Index*. [Online]. Available: <http://breachlevelindex.com/>. [Accessed: 19-Mar-2015].
- [36] "The Design Ethos of Dieter Rams," *San Francisco Museum of Modern Art*. [Online]. Available: [http://www.sfmoma.org/about/press/press\\_exhibitions/releases/880](http://www.sfmoma.org/about/press/press_exhibitions/releases/880). [Accessed: 10-Feb-2015].