Visualizing undirected graphs and symmetric square matrices as overlapping sets

Methods and application to character co-occurrences graphs and matrices in novels and DBpedia

Jean-Baptiste Lamy

This is an author postprint of a paper accepted in Multimedia Tools and Applications:
https://doi.org/10.1007/s11042-019-7655-8

Abstract Undirected graphs and symmetric square matrices are frequently found in various domains. An example is character co-occurrence matrices in digital humanities. However, the visualization of these datasets is difficult, especially if the graph is highly connected. In this article, we propose a method for visualizing undirected graphs and symmetric square matrices, by transforming them into overlapping sets, and then by visualizing these overlapping sets using set visualization techniques such as Euler diagram or rainbow boxes. We also propose a clustering approach to simplify the visualization.

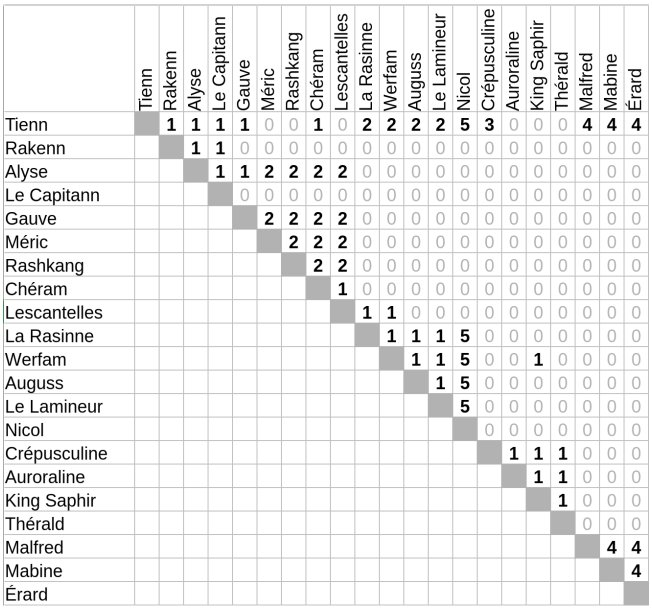
We apply this method to the visualization of various character co-occurrence matrices extracted from novels or DBpedia, ranging from 21 to 114 characters. We show that this visualization allows the finding of several interesting insights. Finally, we discuss the advantages and drawbacks of this method, and we compare it to other approaches in the literature.

Keywords Knowledge visualization · Matrix visualization · Overlapping set visualization · Undirected graph · Symmetric square matrix · Distant reading · Digital humanities · Les Misérables

1 Introduction

Undirected graphs and symmetric square matrices are frequently used to represent co-occurrences in many domains [24]. Any undirected graph can be represented by an adjacency matrix, leading to a symmetric square matrices, and *vice versa*. Such graphs and matrices are frequently encountered in *visual text analysis* [11], which applies information visualization methods to digital humanities. The visualization of large texts such as novel is an important problem. Two approaches exist: *close reading*, in which the text is preserved and visualized, and *distant reading*, in which the text is not

J.-B. Lamy
LIMICS, Université Paris 13, Sorbonne Université, Inserm, 93017 Bobigny, France
E-mail: jean-baptiste.lamy@univ-paris13.fr



	Tienn	Rakenn	Alyse	Le Capitann	Gauve	Méric	Rashkang	Chéram	Lescantelles	La Rasinne	Werfam	Auguss	Le Lamineur	Nicol	Crépusculine	Auroraline	King Saphir	Thérald	Malfred	Mabine	Érard
Tienn	▨	1	1	1	1	0	0	1	0	2	2	2	2	5	3	0	0	0	4	4	4
Rakenn		▨	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alyse			▨	1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Le Capitann				▨	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gauve					▨	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Méric						▨	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Rashkang							▨	2	2	0	0	0	0	0	0	0	0	0	0	0	0
Chéram								▨	1	0	0	0	0	0	0	0	0	0	0	0	0
Lescantelles									▨	1	1	0	0	0	0	0	0	0	0	0	0
La Rasinne										▨	1	1	1	5	0	0	0	0	0	0	0
Werfam											▨	1	1	5	0	0	1	0	0	0	0
Auguss												▨	1	5	0	0	0	0	0	0	0
Le Lamineur													▨	5	0	0	0	0	0	0	0
Nicol														▨	0	0	0	0	0	0	0
Crépusculine															▨	1	1	1	0	0	0
Auroraline																▨	1	1	0	0	0
King Saphir																	▨	1	0	0	0
Thérald																		▨	0	0	0
Malfred																			▨	4	4
Mabine																				▨	4
Érard																					▨

Fig. 1 Example of a symmetric square matrix showing the relations between 21 characters in a novel. Numbers in the matrix indicate the parts of the novel in which the two characters meet for the first time (0: no relation, 1: before the novel starts, 2: during part #1, *etc*).

preserved but important features or characteristics, such as a character network, are extracted from the text and visualized. The present work focuses on distant reading.

In particular, character co-occurrence matrices can be extracted from novels, narrative texts, movies or historical databases, and visualized with various techniques. An example of a symmetric square matrix, representing the relations between the 21 main characters in a novel, is shown in Figure 1. The visualization of such datasets becomes challenging when the number of characters increases, but also when the graph is highly connected: more connections implies more edges, impairing the readability of graph visualization techniques.

In this article, we propose an original method for visualizing highly connected undirected graphs and symmetric square matrices, by transforming the graph or matrix into overlapping sets, and then by visualizing these overlapping sets using set visualization techniques. This method is particularly aimed toward the identification and the representation of subsets of interrelated elements, for example groups of characters that know each other, two by two, in a character network. We will consider two set visualization techniques: the well-known Euler diagrams [8,27] and rainbow boxes [14,15], a technique we recently introduced.

A preliminary version of this work was presented to the international conference on Information Visualisation (iV 2018) [20]. In the present article, we extended it by (1) developing box clustering in rainbow boxes, as suggested by the audience of the conference, (2) testing another set visualization technique (Euler diagrams) in addition to rainbow boxes and (3) applying our method to new datasets extracted from DBpedia.

The rest of the paper is organized as follows. Section 2 presents related works in distant reading and background on rainbow boxes and combinatorial optimization. Section 3 describes the visualization method we propose. Section 4 presents the application of the method to the visualization of character co-occurrences, with various datasets ranging from 21 to 80 characters. It also shows the insights that can be gained from these visualizations. Section 5 discusses the method and the results, and compares our approach to literature, before concluding.

2 Related works

2.1 Visualization of undirected graphs and symmetric square matrices

In the literature, many approaches have been proposed for the visualization of undirected graphs or symmetric square matrices, such as character co-occurrences. First, they can be represented as a graph or a network. However, those graphs often become difficult to read when the number of nodes increases. For instance, word co-occurrences have been visualized using graphs for analyzing scientific literature on patient adherence [34]. Graphs have also been used for visualizing plagiarism in pieces of music, *i.e.* co-occurrences of several identical notes in several pieces [6]. Finally, they have been applied to the visualization of character co-occurrences in novels and movies [4].

Second, these datasets are frequently visualized as colored matrices after reordering rows and columns [30]; for instance, an online matrix visualization of the character co-occurrences in *Les Misérables* can be seen at https://bost.ocks.org/mike/miserables/. An example of matrix-based tools is MatLink [9], which has been applied to the analysis of social networks. Matrix reordering methods allow the identification of interrelated character groups: those groups form squares on the matrix diagonal (or triangle if only half of the matrix is shown). BioFabric [25] displays large networks in semi-matrix style : nodes are represented by horizontal lines and edges by vertical lines. Edge lines start on the horizontal lines corresponding to the source node, and are grouped by destination node. Parallel Aggregated Ordered Hypergraph [29] represents the graph's nodes as rows, and each edge as a vertical line with dots in each rows involved in the edge. Edges can be ordered *e.g.* to represent time.

Third, a character co-occurrence matrix can be treated as a similarity matrix. Therefore, dimension reduction techniques, such as Principal Component Analysis (PCA) or Multidimensional Scaling (MDS) [5], can be used in order to transform a matrix into a two-dimensional scatter plot, a topological landscape or a knowledge map. An example is the Text Variation Explorer [28], which uses PCA to represent various sociolinguistic features in text fragments using a scatter plot. Another example is Memory Islands [31], which integrate hierarchical knowledge, *e.g.* from an ontology.

Fourth, chord diagrams have been proposed for the visualization of character co-occurrences in novels [3]. The chord diagram displays all characters on a ring, and represent co-occurrences by "chords" linking two characters, and located in the middle of the ring. It allows selecting a character and observing its co-occurrences with other characters, but it does not help with the identification of groups of interrelated characters.

Fifth, hierarchical clustering can be applied to the matrix, and visualized as a dendrogram. This approach was proposed for analyzing co-occurrences between MeSH (Medical Subject Heading) terms used to index medical articles [32].

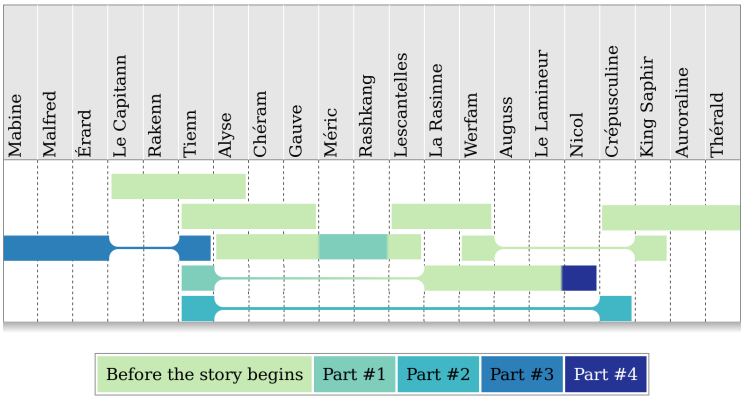


Fig. 2 Rainbow boxes displaying groups of interrelated characters in *Sombre comme l'Aurore*. Box colors indicate the part of the novel during which a character joins the group.

Sixth, various metrics can be derived from the graph or the matrix, such as nodes degree, and visualized using classical plots. This approach was applied to the analysis of social networks in *Alice in wonderland* [1].

Finally, dynamic graphs are commonly visualized either with animated diagrams or with static charts on a timeline [2]. Another option is Massive Sequence View [7].

2.2 Rainbow boxes

Rainbow boxes are a recent set visualization technique [14, 15]. In rainbow boxes (see example in Figure 2), elements are represented in columns and each set is represented by a rectangular box that covers the columns corresponding to the elements belonging to the set (the set label can be shown inside the box, however, in Figure 2 sets are unlabeled). Boxes are ordered vertically by size (with the largest at the bottom) and stacked vertically. Two boxes may be placed next to each other when they share no common column, however, the fact that two boxes appear in the same row has no particular meaning beyond the fact that the two corresponding sets are disjoint.

When the elements of a given set cannot be placed next to each other, holes are present in the box (*e.g.* the yellow box on the left of Figure 2 has one hole of size 2). The column order is optimized for minimizing the number of holes in the visualization (see next section). We initially developed rainbow boxes for the comparison of drug properties [16] and then we applied them to biomedical data and knowledge.

2.3 Artificial Feeding Birds (AFB) metaheuristic

Several visualization techniques require to solve optimization problems, including rainbow boxes. In previous works, we used a heuristic algorithm (described in [14, 15]). This algorithm is a simple construction heuristic. It builds a near-optimal order by starting with a single element (the one belonging to the highest number of set), and then add the other elements one at a time, on the left or the right of the ordering in construction. A score is computed for each candidate insertion (two per remnant element: one on the left and one on the right). However, in case of tied candidate insertions, all candidates are tested. This leads to an increasing number of candidate orders to test. In practice, in a reasonable amount of time, this heuristic is restricted to 20-25 elements.

Here, we will use nature-inspired metaheuristics [33], which are simple, efficient and yet adaptable optimization algorithms. We chose Artificial Feeding Birds (AFB) [21], a recent metaheuristic inspired by the behavior of pigeons. AFB can be adapted to solve both combinatorial optimization and global nonlinear optimization. It was chosen because we showed that it can successfully optimize rainbow boxes beyond 20 elements. On randomly generated datasets with 20 and 30 elements, we found that AFB yields better performances than genetic algorithms (with or without local optimization) and ant colony optimization, two other metaheuristics commonly applied to combinatorial problems [21].

AFB considers a population of artificial birds (by default, 20 birds). The position of each bird represents a candidate solution for the optimization problem. The algorithm performs several cycles; in each cycle, each bird performs one move. Four moves are possible: (1) walk to a random position close to the actual one, (2) fly to a random position, (3) fly to the best position found by the same bird yet, and (4) fly to join the position of another random bird. Move #4 is allowed only for large birds, which represent 25% of the bird population. Moves #3 and #4 are totally independent from the optimization problem. On the contrary, moves #1 and #2 depend on the types of optimization problem. Consequently, AFB can be applied to any optimization problem that is defined by a triplet of functions $(cost, fly, walk)$, where $cost$ is the cost function to minimize, fly is a function that returns a totally random solution and $walk$ is a function that returns a random solution close to another previous solution.

3 Methods

A symmetric square matrix can be formalized as $M = (M_{i,j})_{1 \leq i \leq n, 1 \leq j \leq n} \in \mathbb{R}$, with $M_{i,j} = M_{j,i}$. The proposed method for visualizing M consists of three steps: first, produce overlapping sets from the matrix, second, optionally reduce the number of sets through clustering, and third, visualize these overlapping sets using set visualization techniques such as rainbow boxes or Euler diagrams.

3.1 Translating a symmetric square matrix into overlapping sets

For translating the matrix into overlapping sets, we assume that (1) each row/column of the matrix (*i.e.* each index $i \in E = \{1, 2, ..., n\}$) represent an element (*e.g.* a character), and (2) the matrix expresses relations between some (but not all) of these elements. The selection function $select : \mathbb{R} \rightarrow \{True, False\}$ indicates, for a given value in the matrix, if the two elements are considered as related, *i.e.* i and j are related if (and only if) $select(M_{i,j}) = True$. For example, in the matrix of Figure 1, two elements are considered as related if the value in the matrix is not zero. Overlapping sets will be made of these elements, each set s including only elements that are related to each other (*e.g.* when considering a character matrix, all characters in a given subset s know all the other characters of the subset): $\forall i \in s, \forall j \in s \; with \; i \neq j, select(M_{i,j}) = True$. Only the largest of such sets will be retained.

Algorithm 1 Set clustering algorithm. P is the set of all pairs in S, and (b_1, b_2) is the best pair found, *i.e.* the one with the highest score. For simplicity, we wrote $\big|x : x...\big|$ for $\big|\{x : x...\}\big|$.

```
function $score(s_1, s_2)$:
    $r_{set} = \frac{\big|(i,j) \in (s_1 \cup s_2)^2 : i \neq j \wedge select(i,j)\big|}{\big|(i,j) \in (s_1 \cup s_2)^2 : i \neq j\big|}$
    $r_{el} = \frac{\min\big(\{\big|j \in (s_1 \cup s_2) : j \neq i \wedge select(i,j)\big| \ : i \in (s_1 \cup s_2)\}\big)}{|s_1 \cup s_2| - 1}$
    $r = \sqrt{r_{set} \times r_{el}}$
    return $r$

function $cluster(S, t)$:
    while $True$:
        $P = \{(s_1, s_2) \in (S, S) : s_1 \neq s_2\}$
        $(b_1, b_2) = \arg\max_{(s_1,s_2) \in P} score(s_1, s_2)$
        if $score(b_1, b_2) < t$:
            break while loop
        else:
            remove $b_1$ from $S$
            remove $b_2$ from $S$
            add $b_1 \cup b_2$ to $S$
```

First, we compute S_0, the set of all sets s that include only elements related to each other:

$$S_0 = \Big\{s \subseteq E : |s| > 1 \wedge \forall i \in s, \forall j \in s \ with \ i \neq j, select(M_{i,j}) = True\Big\}$$

Second, we compute S, the set containing only the largest sets s in S_0 (*i.e.* we only keep sets s having no strict superset in S_0).

$$S = \{s \in S_0 : \nexists s' \in S_0, s \subset s'\}$$

As a result, the sets s in S are the largest sets of interrelated elements.

3.2 Set clustering

When the number of overlapping sets produced is high, they can be difficult to visualize. In this case, we propose to cluster similar sets. However, this step is entirely optional.

Set clustering is controlled by a clustering threshold t, with $0 \leq t \leq 1$. When $t = 1$, no clustering occurs; on the contrary, when $t = 0$, all sets are clustered together. Algorithm 1 describes the clustering process. Clustering is performed iteratively. In each iteration, for each pair (s_1, s_2) of distinct sets in S, a score r is computed (described below). If no pair with a score superior or equal to the threshold t is found, clustering terminates. Otherwise, the two sets of the pair (b_1, b_2) with the highest score r are merged into a single set $b_1 \cup b_2$. Finally, another iteration is performed.

The score r includes two components: r_{set}, the set component, and r_{el}, the element component. r_{set} is the ratio of interrelated pairs of elements in $s_1 \cup s_2$, the resulting merged set. r_{el} is computed on a per-element basis, and the final element component is the minimum value found. For each element, we compute the ratio of other elements in $s_1 \cup s_2$ that are related to this element. Both r_{set} and r_{el} are ratios, ranging from

0 to 1. r_{set} favors clustering of two sets that have a high number of related elements, while r_{el} ensures that no element is “sacrificed”, *i.e.* no element in the resulting merged set has a very small number of relations with the other. The final score r is the square root of the product of r_{set} and r_{el} (we use a square root because both r_{set} and r_{el} represent somehow the ratio of interrelated elements in the merged set, thus $r_{set} \times r_{el}$ actually represents the square of that ratio).

3.3 Using rainbow boxes for visualization

The sets in S can be visualized using rainbow boxes: each element in E (corresponding to rows/columns in the matrix) will be represented by a column in rainbow boxes, and each set in S will be represented by a rectangular box. The box covers all columns corresponding to the elements in the set.

Set membership does not represent the totality of the information present in the matrix, because set membership is binary (an element either belongs to a given set, or does not belong to). It only represents the part of the information returned by the Boolean selection function *select*. However, some matrices may contain additional information, for example the matrix of Figure 1 indicates not only relations between characters but also the parts of the novel containing their first meeting. In order to aggregate this additional information and to represent it by colors in rainbow boxes, we will consider 2 new functions: the aggregation function and the colorization function.

First, for a given element $i \in E$ in a subset $s \in S$, there can be several values in the matrix. For example, if we consider $s = \{i, j, k\}$, two values are present in the matrix for i and the other elements of s: $M_{i,j}$ and $M_{i,k}$. In the general case, the number of values is $|s| - 1$. For the purpose of visualization, we need to aggregate these values into a single value and then to translate it into a color. The aggregation is achieved by the function $aggregate : \mathbb{R}^p \rightarrow \mathbb{R}$ (with $p \geq 1$). It takes one or more values from the matrix, and returns a single aggregated value. Typical aggregation functions are *min*, *max*, *mean* and *sum*.

Second, the colorization function $colorize : \mathbb{R} \rightarrow color$ translates the aggregated value into a color. The color will be applied to the rectangular box in the corresponding column (*i.e.* colors are defined on a per-set/box and per-element/column basis, thus a given box may have several colors).

Rainbow boxes require to optimize the column order, for minimizing the number of holes in the boxes. This is a complex combinatorial optimization problem with a factorial complexity. Here, we used the Artificial Feeding Birds (AFB) metaheuristic (see section 2.3) for optimizing column order.

3.4 Using Euler diagram for visualization

The sets in S can also be visualized using an Euler diagram. Euler diagrams are known to be difficult to produce automatically when the number of sets is high. Here, we used Multidimensional Scaling (MDS) [5]to project the characters on a two-dimensional plane, as follows. We first create a matrix $M' = (M'_{i,j})_{1 \leq i \leq n, 1 \leq j \leq |S|} \in \{0, 1\}$ whose rows correspond to characters and columns correspond to set in S. $M'_{i,j} = 1$ if the i^{th} character belongs to the j^{th} set, otherwise $M'_{i,j} = 0$. Then, using MDS, we compute $M'' = (M''_{i,j})_{1 \leq i \leq n, 1 \leq j \leq 2} \in \mathbb{R}$, a two-dimensional projection of M'. We used the MDS implementation in the Python module Scikit-Learn [26]. Using this two-dimensional projection, characters were positioned in a 2D space. Finally, the sets, corresponding to character groups, were manually drawn.

4 Application to character matrices

4.1 Application to *Sombre comme l'Aurore*

Figure 1 shows a character matrix for *Sombre comme l'Aurore*, a French yet-unpublished novel written by the author. This matrix includes the 21 main characters of the novel and was produced manually. A very conservative definition of "character relation" was considered: two characters are related if it is plausible that they met together outside of the events directly reported in the novel. Since the novel covers a period of one year, not all events are reported and, in particular, characters that are friends to each other are likely to meet much more often that told in the novel. On the contrary, simple encounters that do not lead to regular relationship are not considered as "relation" in this matrix. Consequently, the matrix is a relationship matrix rather than a co-occurrence matrix. When a relation holds between two characters, the matrix indicates by an integer number the part of the novel during which their relation starts: 1 means before the story begins, and 2-5 correspond to part #1-4 in the novel, respectively. Finally, 0 in the matrix indicates the absence of relation between two characters. Therefore, the selection function selects all pairs in the matrix were the value in the matrix is not zero.

$select : p \mapsto True\, if\, p \neq 0, False\, otherwise$

We used the previously described method for transforming this matrix into overlapping sets (without clustering), and for visualizing these sets using rainbow boxes.

4.1.1 Rainbow boxes visualization

We used *min* as the aggregation function. This is consistent with the content of the matrix, which indicates the *first* part of the novel where two characters have a relation.

$aggregate : p_1, p_2, ... \mapsto \min(p_1, p_2, ...)$

The colorization function produce a bright color, whose hue encodes the first part of the novel in which a character joins a group, using a color key generated using Color Brewer 2[1] and shown at the bottom of Figure 2. Yellow/dark colors correspond to the beginning/end of the story, respectively.

$$colorize : a \mapsto \begin{cases} yellow & if\, a = 1 \\ . & if\, a = 2 \\ . & if\, a = 3 \\ . & if\, a = 4 \\ dark\ blue & if\, a = 5 \end{cases}$$

Figure 2 shows the sets extracted from the matrix as rainbow boxes. The 21 characters are represented by columns. Each box represents a group of interrelated characters, *i.e.* any character in a given group is related to any other character of the group. For

[1] http://colorbrewer2.org

example, the top-most box indicates that Le Capitann, Raken, Tienn and Alyse are related to each other, two by two. In addition, the colors indicate at which moment a character joined the group. The top-most box is entirely yellow, indicating that the four characters were related to each other before the story begins. On the contrary, the box covering Alyse, Chéram, Gauve, Méric, Rashkang and Lescantelles has two colors: Alyse, Chéram, Gauve and Lescantelles formed a group before the story begins (yellow color) while Méric and Rashkang joined this group in part #1. Notice that, since we used the minimum function as aggregation function, the fact that Alyse is in yellow in this box *does not* imply that she is related to all other characters in the group before the story begins: she may be related to only some characters in the group before the beginning, and then develop relations with the others later.

Figure 2 suggest several interesting insights: (1) Most groups already exist before the story begins (the violet color is dominant). (2) Tienn is the character that belongs to the highest number of groups (five rectangular boxes in his column). Indeed, Tienn is the hero of the novel. (3) Tienn belongs to two groups before the story begins (the two violet boxes in his column). (4) In the first part of the novel, Tienn joins an already formed group of characters (La Rasinne, Werfam, Auguss and Le Lamineur). In fact, this corresponds to his colleagues in his first job. (5) This group will be joined by Nicol in part #4 (dark blue segment in that box). He is a new colleague. (6) In part #2, Tienn encounters Crépusculine (the middle blue box at the bottom). In addition, we can see that Tienn and Crépusculine have no common friends at all (this box includes no other characters), and that they are very distant in the visualization. Indeed, they come from two different worlds, and they fall in love. (7) Moreover, we can search the shortest path between Crépusculine and Tienn (ignoring their direct relation). This path is Crépusculine → King Saphir → Werfam → Tienn. (8) In part #3, Tienn joins a new group of three characters (Mabine, Malfred and Érard) that have no relations with any other characters. Indeed, those characters live isolated, in a kind of ghetto.

4.1.2 Euler diagram visualization

Figure 3 shows the sets extracted from the matrix as an Euler diagram with the characters positioned by MDS (see section 3.4 for details on the generation of the Euler diagram). Many insights previously obtained with rainbow boxes cannot be found on the Euler diagram. In particular, Euler diagram cannot represent per-element-set-membership variables, and thus the Euler diagram does not display the part of the novel in which a character joins a group. Moreover, the important distance between Tienn and Crepusculine does not appear on the Euler diagram.

However, interestingly, other insights can be obtained from the Euler diagram. In particular, thanks to the use of MDS, distance represents the (dis)similarity between characters. First, Tienn and Werfam, are very distant and located on two opposite corners of the visualization. These two characters seem in opposition, despite the set that relates them. Indeed, Tienn and Werfam are two political enemies, and they oppose during part #4 of the novel. Second, the dashed gray line separates the character of the two afore-mentioned worlds. On the contrary, in rainbow boxes, the characters of the world of Crépusculine were positioned at both extremities.

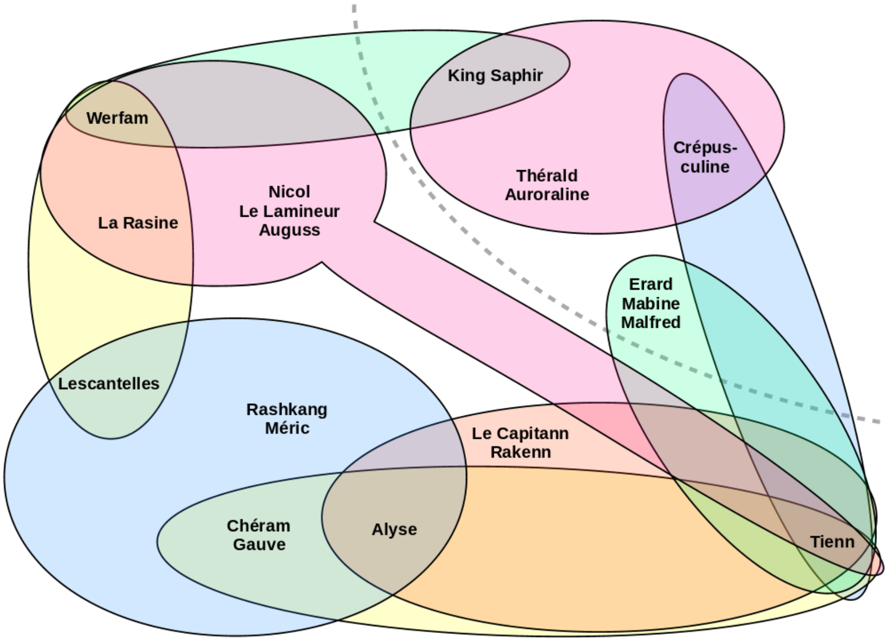


Fig. 3 Euler diagram displaying character co-occurrences in *Sombre comme l'Aurore*.

4.2 Application to *Les Misérables*

Les Misérables is a famous French novel written by Victor Hugo. A well-known dataset from the Stanford GraphBase [13] includes a character graph for this novel. The graph contains 80 characters and indicates all their co-occurrences in the novel. Each co-occurrence is a separate relation in the graph, and is labeled with the part and chapter in which it occurs. We transformed this graph into a symmetric squared matrix, with characters as rows and columns. For each pair of characters i and j, the matrix contains a pair of values $M_{i,j} = (p, c)$ where p is the parts of the novel in which the two characters co-occur for the first time, and c is their total numbers of co-occurrence. If the two characters never co-occur, then $M_{i,j} = (0, 0)$.

We used the following selection, aggregation and colorization functions:

$$select : x \mapsto True\ if\ x \neq (0,0), False\ otherwise$$
$$aggregate : (p_1, c_1), (p_2, c_2), ..., (p_m, c_m) \mapsto (\min(p_1, p_2, ..., p_m), c_1 + c_2 + ... + c_m)$$
$$colorize : (p, c) \mapsto color\left(hue = \begin{cases} blue & if\ p = 1 \\ violet & if\ p = 2 \\ red & if\ p = 3 \\ orange & if\ p = 4 \end{cases}, brightness = \frac{c}{\max(c)} \right)$$

The aggregation function retains the minimum value for the part of the first co-occurrence, and the sum for the number of co-occurrences. The colorization function encodes the part of the first co-occurrence using hue, from blue to red, and the total number of co-occurrences using brightness: characters having more co-occurrences with

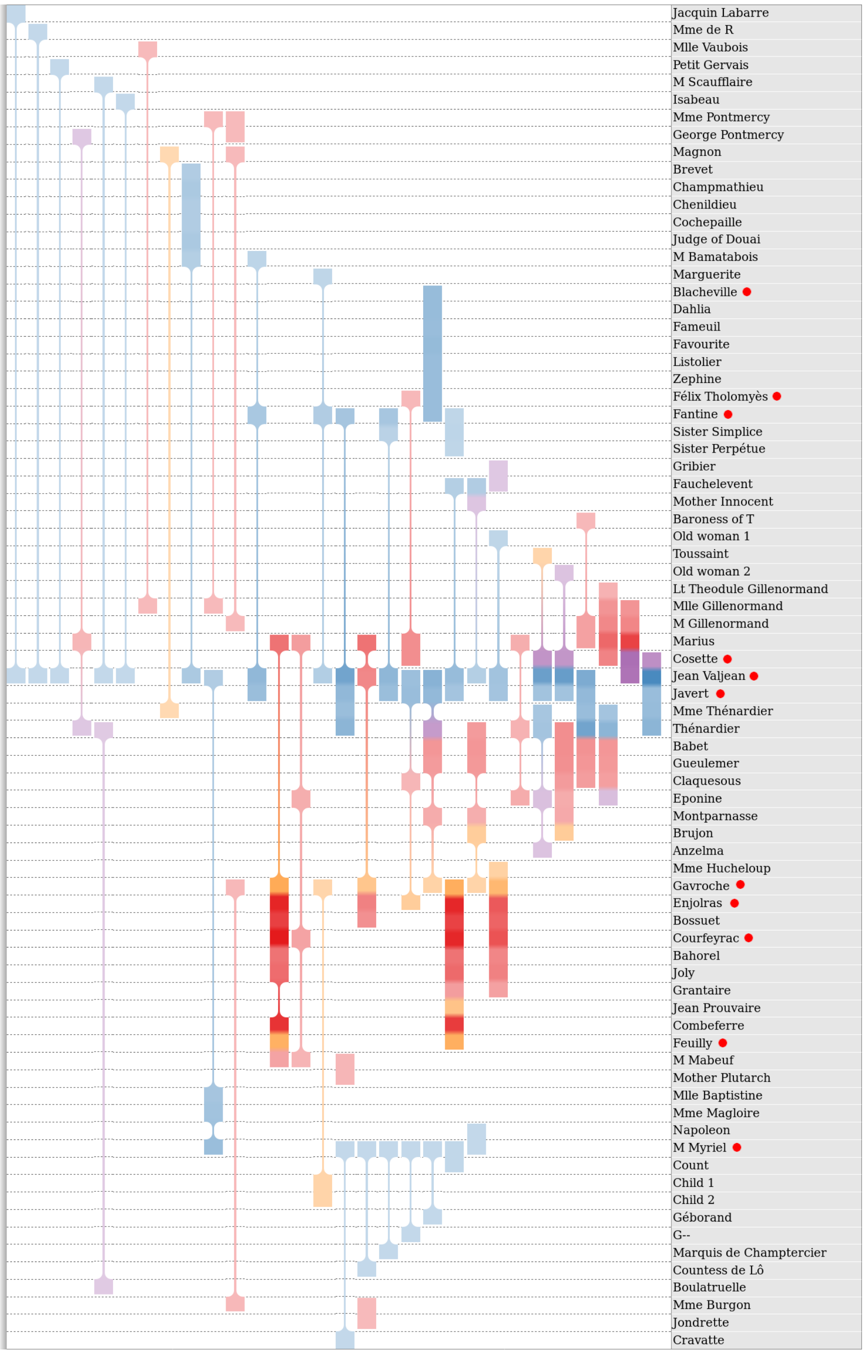


Fig. 4 Rainbow boxes displaying the character co-occurrences in *Les misérables*. Color (blue, violet, red, orange) indicates the parts (1-2-3-4) of the book (respectively) and brightness the number of co-occurrences. Characters mentioned in the text are marked with red dots.

other characters in the group are represented with brighter colors. Since no character joined a new group in part #5, the colorization function considers only 4 different hues.

In addition, the rainbow boxes column optimization was biased in order to favor holes in boxes associated with fewest co-occurrences. For each box, we computed c_m the mean number of co-occurrences. Then, the cost of a hole in a given box was proportional to the corresponding c_m. This prevented splitting boxes representing the most important (in terms of the total number of co-occurrences) groups of characters.

The resulting rainbow boxes are shown in Figure 4. It suggests the following insights: (1) Most of the new character-group relations occur in parts #1 and #3 of the novel (blue and red colors are dominant). (2) Jean Valjean appear as the "central" character: he is the character that belongs to the highest number of groups. Indeed, he is the hero of the novel. Moreover, in most groups he belongs to, Jean Valjean is the most interacting character (brighter color in his column). (3) Javert is placed next to Jean Valjean (they have 11 groups in common). This is expected, because Javert is a policeman who tracks Jean during most of the novel. (4) Cosette appears in part #2 and also joins new groups in part #3. (5) Myriel has several relations, but with very few co-occurrences (non-bright color). (6) Gavroche appears in part #3, in a single group, but joins many other groups during part #4, two of them including Jean Valjean. (7) There are some very highly interacting characters from Gavroche to Feuilly, spread over three groups, and Courteyrac and Enjolras seem the most active in these groups. (8) There is a group of 8 interrelated characters (from Blacheville to Fantine, the blue box on the left) that have very few relations with other characters (only Fantine and Félix are belonging to other groups).

In addition, a characteristic pattern of triangular "Christmas tree" (fir) can be observed around Myriel and (to a lesser extent) Jean Valjean. This pattern indicates that the central character (forming the trunk of the tree) has many isolated relations with other characters (forming the leaves), *i.e.* the central character is related to many other characters without having common friends with them. Depending on the global shape of the rainbow boxes, this pattern can be bilateral (for Jean Valjean) or unilateral (for Myriel).

4.3 Application to character matrices extracted from DBpedia

4.3.1 Data extraction

Character matrices were extracted from DBpedia [22] version 2016-10, as follows. First, we selected an article of interest. Then, we extracted all linked entities in this article (using "wikiPageWikiLink" property) that belong to the class Person (*i.e.* characters). Finally, we considered two characters as related if at least one wikiPageWikiLink relation exists between them. For a given pair of characters, the value in the matrix is the number of relations between them: 0 (unrelated characters), 1 (unidirectional relation) or 2 (bidirectional relation). We used the Owlready 2 [18, 17] ontology-oriented programming module for Python for loading DBpedia RDF and OWL data, and for performing the extraction. We extracted three datasets, from the following articles: Paris Commune (the events of 1871), Troubadour, and Trouvère.

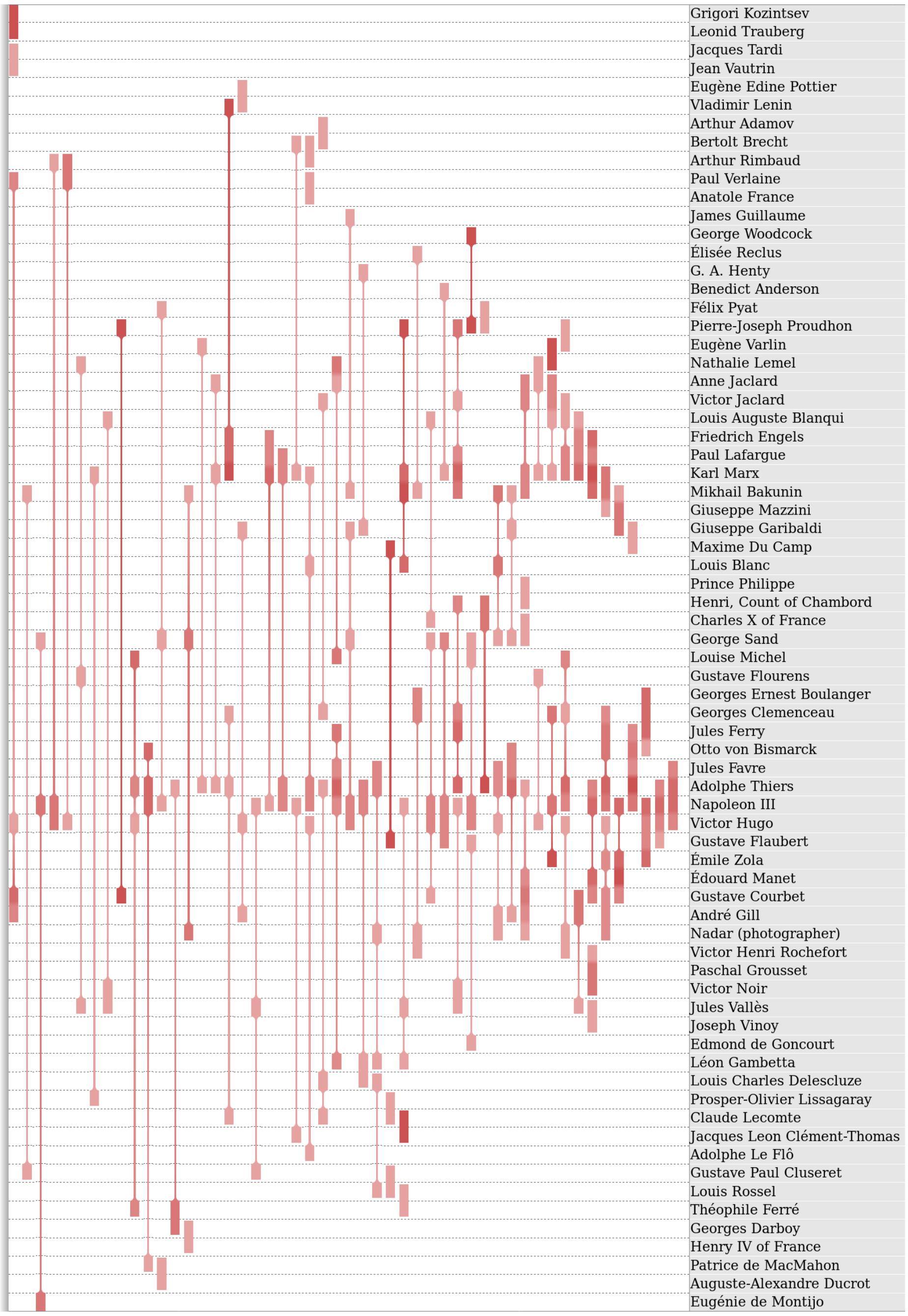


Fig. 5 Rainbow boxes displaying the Paris Commune dataset extracted from DBpedia. Color brightness encodes the relationship: brighter colors represent bidirectional relations while dimmed colors represent unidirectional relationship.

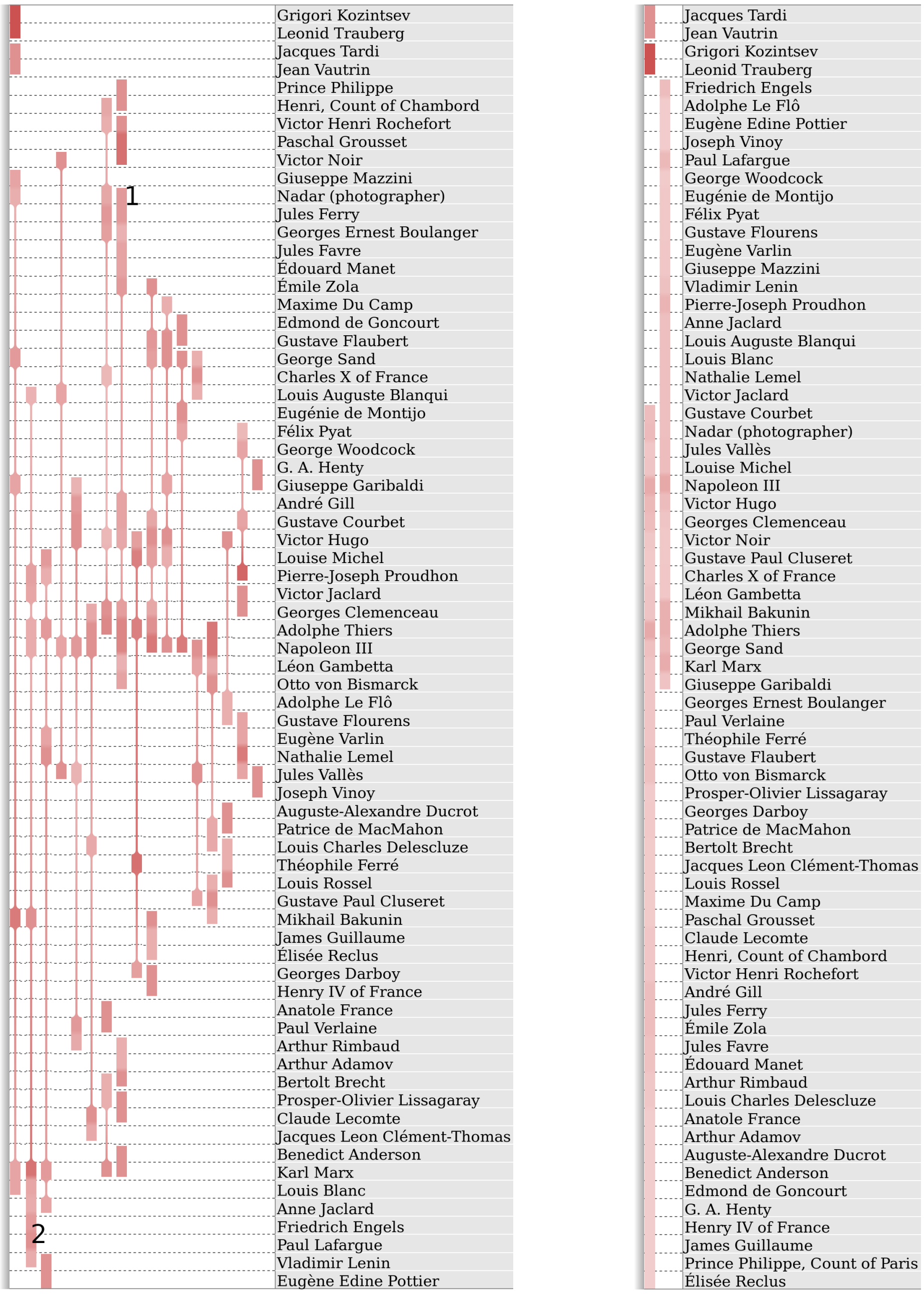


Fig. 6 Rainbow boxes displaying the Paris Commune dataset after clustering with $t = 0.25$ (left) and $t = 0.025$ (right).

4.3.2 Paris Commune dataset

This dataset includes 82 characters. Figure 5 shows the dataset using rainbow boxes, using the following selection, aggregation and colorization functions:

$$select : p \mapsto True\ if\ p \neq 0, False\ otherwise$$
$$aggregate : p_1, p_2, ..., p_m \mapsto \frac{p_1 + p_2 + ... + p_m}{m}$$
$$colorize : p \mapsto color(hue = red, brighness = p)$$

We can identify two "Christmas tree" patterns. The larger one has two "trunk" characters: Napoleon III and Adolphe Tiers. They correspond to the two governmental leaders, before the Paris Commune and after, respectively. On the contrary, Paris Commune did not have a single, clearly identified, leader. Despite the Commune had several iconic characters, such as Louise Michel, they have surprisingly few relations in DBpedia. A second "Christmas tree" pattern can be seen around Karl Marx. Although not directly involved in the events, he supported the Paris Commune. The four left-most characters are totally disconnected from the other. They correspond to film directors or writers that worked on the Paris Commune, but were not contemporaries of the events.

Figure 6 shows the same dataset after clustering, using the same selection, aggregation and colorization functions as above. With $t = 0.25$ (left), the number of character groups/boxes is lower. We can observe interesting boxes. For example, the box labeled "1" includes many artists and writers, *e.g.* Nadar, Victor Hugo, Émile Zola, André Gill, Gustave Courbet, Édouard Manet. The box labeled "2" includes most of the socialist and communist theoreticians, such as Auguste Blanqui, Karl Marx, Friedrich Engels, Mikhail Bakunin, Vladimir Lenin, Pierre-Joseph Proudhon, Louis Blanc,...

Clustering with a very small threshold $t = 0.025$ provides an interesting results (Figure 6, right). Here, only four groups remain, two of them being the two small isolated groups mentioned above. The two other groups gather most of the characters and overlap partially. The group on the left includes most of the characters that supported the Paris Commune (*e.g.* Eugène Pottier, Eugène Varlin, Auguste Blanqui, Nathalie Lemel, Gustave Courbet, Jules Vallès, Louise Michel,...) while most of the characters on the right did not (with two notable exceptions: Verlaine and Louis Charles Delescluze).

As a matter of comparison, Figure 7 shows the Paris Commune dataset as a graph. It was generated with the NetworkX Python module using the Kamada-Kawai algorithm [12]. Compared to the proposed visualization, the graph facilitates the identification of isolated characters (corresponding to the nodes at the periphery). On the contrary, the highly connected part at the center of the graph is difficult to read, while interrelated groups of characters are easier to observe on the approach we propose. Similarly, Figure 8 shows the dataset using a chord diagram (draw with Bokeh[2]). Here, with 82 characters/nodes and many edges, the readability is quite low. In the literature [3], chord diagram was applied to character relations, but on smaller datasets (about 20 characters). In addition, chord diagram is more interesting when quantitative data are available.

[2] https://bokeh.pydata.org

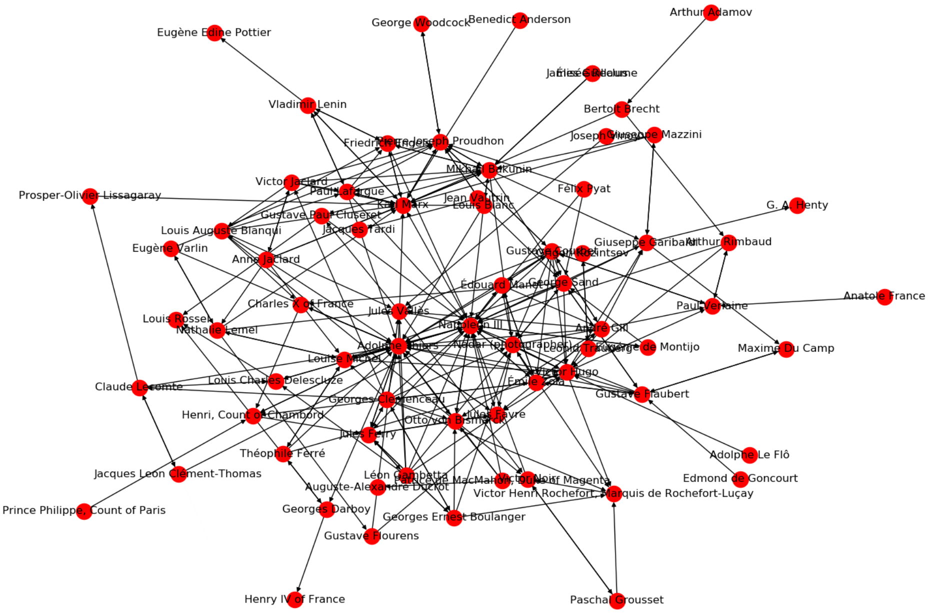


Fig. 7 Graph displaying the Paris commune dataset.

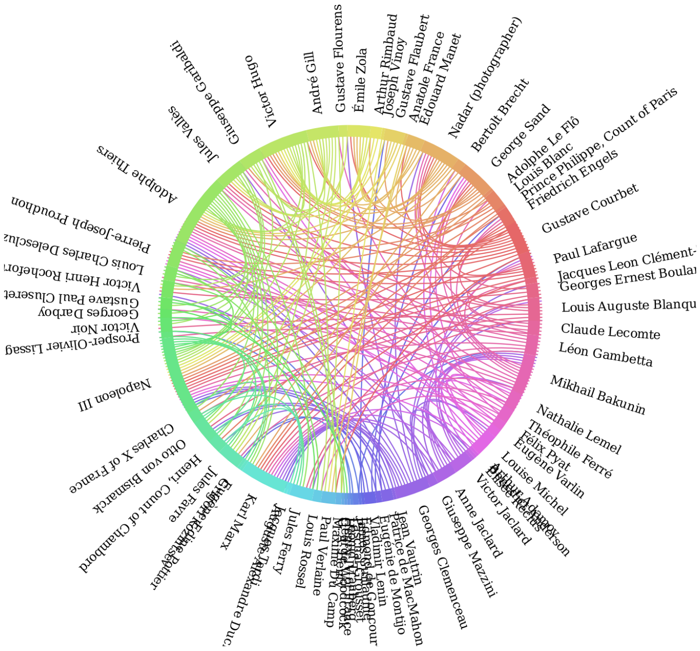


Fig. 8 Chord diagram displaying the Paris commune dataset.

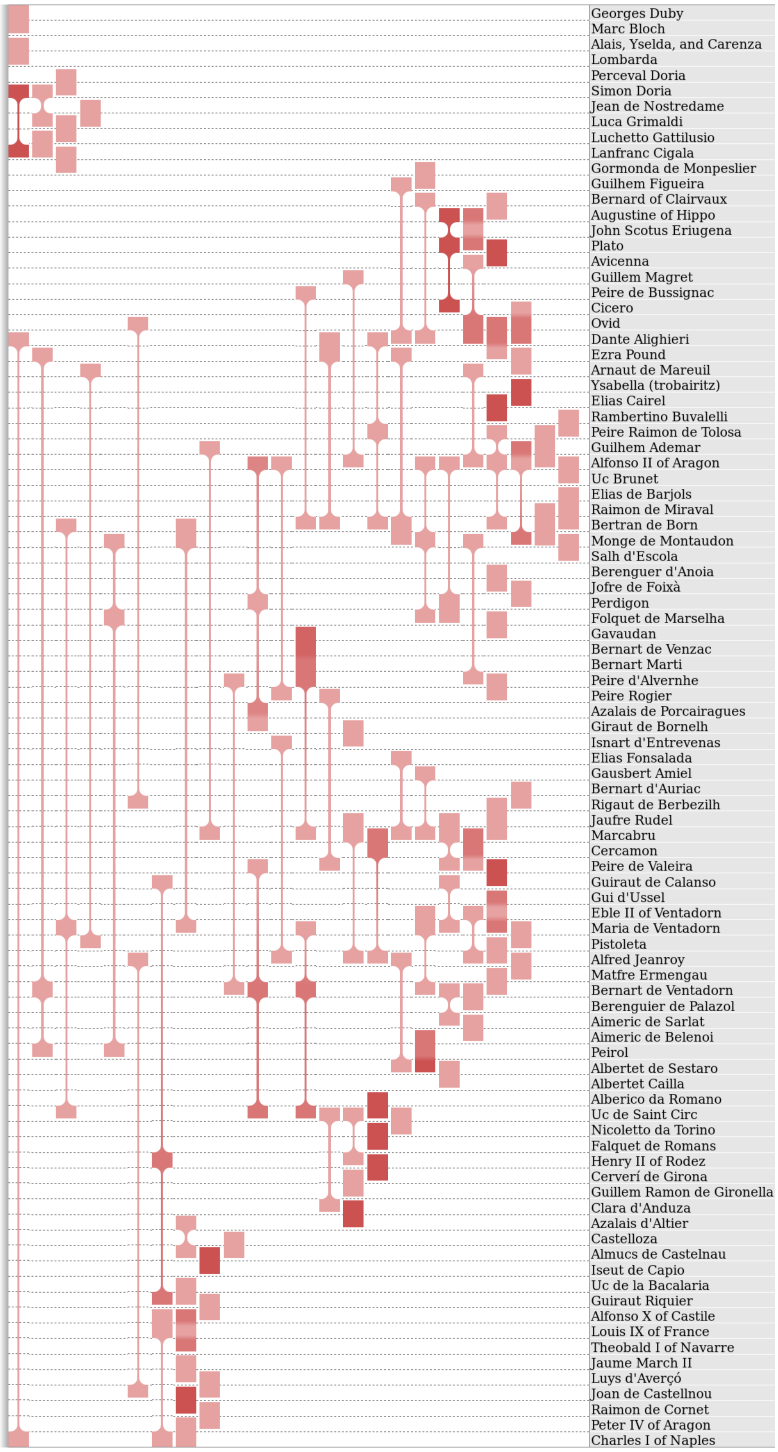


Fig. 9 Rainbow boxes displaying the Troubadour dataset.

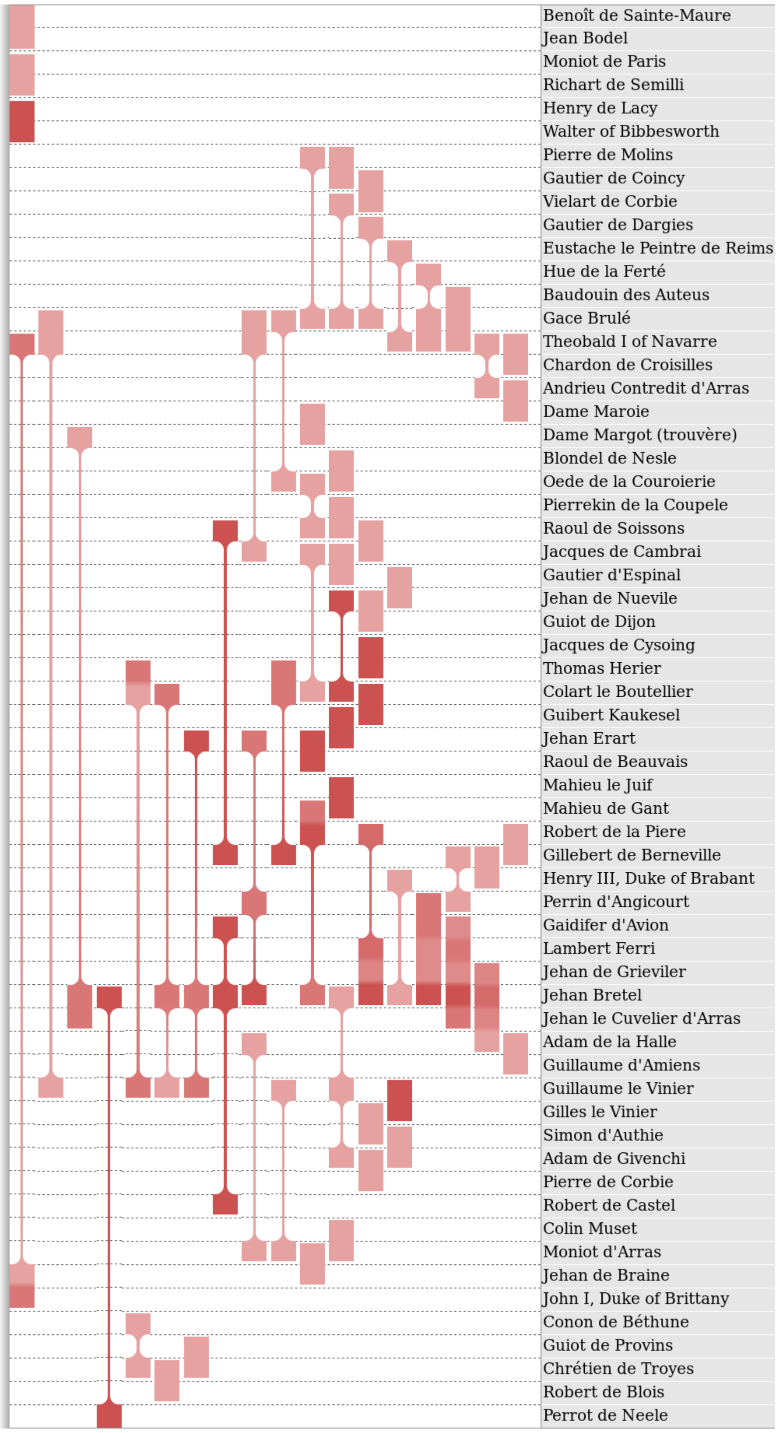


Fig. 10 Rainbow boxes displaying the Trouvère dataset.

4.3.3 Troubadour and Trouvère dataset

Troubadours were poets and songwriters in Old Occitan during the high Middle Ages. Trouvères continued the artistic movement of troubadours, in Oïl language in the north of France. The two datasets include 114 and 76 characters, respectively. They are shown in Figure 9 and 10.

The Troubadour dataset seems not very structured, with no "Christmas tree" pattern nor large boxes. We can distinguish two sets of boxes, one on the left and the other on the right, separated by a white diagonal region in the middle. It does not indicate two isolated sets of characters, because boxes do not represent characters but groups. However, it is caused by the presence of many boxes with long holes on the left columns. Thus, it suggests some kind of heterogeneity in that part of the visualization. On the contrary, on the Trouvère dataset, we can observe two "Christmas tree" patterns around Gace Brulé and Theobald I for the first one, and Jehan Bretel for the second. In addition, these two "Christmas tree" patterns are quite distant one from the other, suggesting that they involve different characters.

The difference between the Troubadour and Trouvère visualization may suggest more structured organization or relationships for trouvères, and more informal ones for troubadours, without prominent "leaders". However, it can also result from a difference in the way troubadours and trouvères pages are written or indexed in Wikipedia.

5 Discussion and conclusion

In this article, we proposed a new method for visualizing undirected graphs and symmetric matrices using overlapping set visualization techniques. We applied this method to the visualization of character co-occurrence matrices extracted from novels or DBpedia. We showed that this method was efficient for identifying groups of interrelated elements and generating new insights. It may be particularly interesting for visualizing highly connected networks, which are usually difficult to visualize using graphs. We also described a characteristic pattern of "Christmas tree" for identifying characters related to many isolated others.

The two datasets extracted from novels we used differ in size, but also qualitatively. The first one was produced manually by the author, using a very conservative definition of "relation" between two characters. On the contrary, the second one was built automatically by considering co-occurrence of characters in the novel. Figure 2 is visually much simpler and easier to read than Figure 4, this can be explained by the smaller size of the dataset, but possibly also by the difference in the nature of the matrices. It would be interesting to extract a co-occurrence matrix from *Sombre comme l'Aurore*, and to compare it with the manually produced matrix.

When building overlapping sets from matrices, we considered only pairwise relations. Thus, the character groups we extracted do not necessarily exhibit higher-level relations : for example, if A is related to B and C, and B is related to C, then A, B and C form a group. However, it does not necessarily imply that A, B and C meet all the three (*i.e.* ternary relation). Consequently, A might be *unaware* that B and C are related, even if they belong to the same group A, B and C. This limitation is inherent to the graph structure of the original data: other graph visualizations, such as reordered matrices, suffer from the same limit. The use of hypergraph could encompass this problem, however, they may be more difficult to manage and to produce. However, contrary to reordered matrices, the proposed approaches could be applied to hypergraph because it can represent differently a group of 3 characters *vs* 3 pairwise-related characters.

When colors are used to display the parts of the novel during which a character joins the group, another limitation is that we do not display the moment at which a character leaves that group (if he does). In theory, time of leaving could be encoded visually using another visual variable beyond color, *e.g.* texture, or by using two colors in boxes (*e.g.* one at the top of the box and another at the bottom). However, the notion of "leaving a group" is difficult to define: if groups correspond to characters that know each other, one still knows someone else even if he no longer meets him. Moreover, the absence of meetings after the initial encounter does not necessarily imply that the characters are no longer related: often, the first meeting between two characters is detailed in the novel, and then, the characters are supposed to meet again regularly but these events may not be narrated explicitly. Consequently, in the present work, we focused on the moment a character joins a group, but not the moment he leaves it.

With regard to the matrix size, the proposed method is limited by the number of columns that can be shown on a screen, and by the maximum number of columns that can be optimized in rainbow boxes. Here, we presented for the first time rainbow boxes with more than 100 columns. However, the optimization took more than one hour on a recent laptop computer (Intel Core i7-7500U 2.70 GHz, 16 Gb RAM, using a single core, i.e. without parallelization). As a consequence, the method is currently limited to matrices with at most 100-150 rows/columns.

The proposed method focuses on groups of interrelated characters. In the literature, matrix reordering methods have been used for the identification of interrelated character groups: those groups form squares on the matrix diagonal (or triangle if only half of the matrix is shown). For example, in Figure 1, 5 such groups can be seen on the diagonal. However, a character can belong only to a single group in this approach, or at most two if the character is placed in-between two squares on the diagonal. For instance, Tienn has been placed at the beginning of the matrix in Figure 1, but could also be placed at the end (due to the three "4" at the end of the Tienn's row). On the contrary, Tienn belong to not less than 5 groups in the rainbow boxes representation in Figure 2, and a total of 9 interrelated groups have been identified (thus 4 additional groups compared to the matrix reordering methods). Similarly, in Figure 4, we can see several overlapping groups at the top of the visualization, most of them including Jean Valjean. In addition, rainbow boxes allow a compact representation of the dataset: a group of n characters that know each other occupies $\frac{n \times (n-1)}{2}$ cells in the matrix, but the entire group is represented as a single box in rainbow boxes, with a length of n columns.

The proposed method needs to be properly evaluated and compared to other techniques, *e.g.* during case studies, or user studies measuring efficacy, efficiency, and user preference. Diverse tasks should be tested. We expect that the proposed method may be efficient for tasks relating to the identification of interrelated groups, as these groups are clearly identified in our method. On the contrary, it may not be as efficient for other tasks, e.g. when considering the relations of a given element rather than a group.

We proposed two techniques for visualizing the resulting overlapping sets, Euler diagrams and rainbow boxes. However, the latter have two advantages: (1) they are easier to produce automatically, while it is known that Euler diagrams are difficult to generate above 6 sets [8,27], and (2) rainbow boxes are easier to read, as shown by a recent user study on amino acid properties [19]. Nevertheless, Euler diagrams suggested different insights than rainbow boxes, and thus they can be a complementary visualization, especially on small datasets. Other techniques, such as UpSet [23], could also be considered in the future for visualizing these overlapping sets.

Rainbow boxes were previously used only in biomedicine. Here, we presented an application of rainbow boxes to digital humanities, which is a totally different domain. This validates the general nature and the genericity of the visualization technique. In addition, we introduced box clustering in rainbow boxes. Clustering can be used to simplify the visualization, by reducing the number of boxes and thus the vertical space required. Moreover, we have shown on the Paris Commune dataset that it can also be used for performing overlapping clustering on characters when the threshold t is very low. This character clustering abilities is an interesting perspective and need additional evaluation.

Additional perspectives of this work are (1) to adapt the proposed method to the visualization of directed graphs/non-symmetric square matrices and hypergraph, (2) to implement element/column clustering, in addition to set/box clustering, (3) to apply the method to other domains, such as the visualization of FOAF (Friend Of A Friend) graphs in social media, protein-protein interaction matrices in bioinformatics, drug-drug interaction matrices in pharmacology, or matrices in Linkography [10], and (4) to evaluate the proposed approach properly.

References

1. Agarwal, A., Corvalan, A., Jensen, J., Rambow, O.: Social network analysis of Alice in wonderland. In: Workshop on computational linguistics for literature, pp. 88–96 (2012)
2. Beck, F., Burch, M., Diehl, S., Weiskopf, D.: A Taxonomy and Survey of Dynamic Graph Visualization. Computer graphics forum **36**, 133–159 (2017). DOI 10.1111/cgf.12791
3. Bilenko N: The narrative explorer, Technical report, EECS Department, University of California, Berkeley (2016)
4. Bonato, A., D'Angelo, D.R., Elenberg, E.R., Gleich, D.F., Hou, Y.: Mining and modeling character networks. In: International workshop on algorithms and models for the web-graph, pp. 100–114 (2016)
5. Borg, I., Groenen, P.J.F., Mair, P.: Applied multidimensional scaling. Springer (2013)
6. De Prisco, R., Esposito, A., Lettieri, N., Malandrino, D., Pirozzi, D., Zaccagnino, G., Zaccagnino, R.: Music plagiarism at a glance: metrics of similarity and visualizations. In: International Conference Information Visualisation (iV), pp. 410–415. Lisboa, Portugal (2016)
7. van den Elzen, S., Holten, D., Blaas, J., van Wijk, J.J.: Dynamic network visualization with extended massive sequence views. Ieee transactions on visualization and computer graphics **20**(8), 1087–1099 (2014)
8. Gottfried B: Set space diagrams. Journal of visual languages & computing **25**(4), 518–532 (2014)
9. Henry, N., Fekete, J.D.: MatLink: Enhanced matrix visualization for analyzing social networks. In: INTERACT, Lecture Notes in Artificial Intelligence, vol. 4663, pp. 88–302. Springer (2007)
10. Hsieh, T.L., Chang, T.W.: Whether the relationscape of Interaction design strategies during design process can be explained by Linkography. In: International Conference Information Visualisation (iV), pp. 14–19. London, United Kingdom (2017)
11. Jänicke, S., Franzini, G., Cheema, M.F., Scheuermann, G.: Visual text analysis in digital humanities. Computer graphics forum **36**(6), 226–250 (2016)
12. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Information Processing Letters **31**, 7–15 (1989)
13. Knuth DE: The Stanford GraphBase: A platform for combinatorial computing. Addison-Wesley (1993)

14. Lamy, J.B., Berthelot, H., Capron, C., Favre, M.: Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain. Journal of Visual Language and Computing **43**, 71–82 (2017)
15. Lamy, J.B., Berthelot, H., Favre, M.: Rainbow boxes: a technique for visualizing overlapping sets and an application to the comparison of drugs properties. In: International Conference Information Visualisation (iV), pp. 253–260. Lisboa, Portugal (2016)
16. Lamy, J.B., Berthelot, H., Favre, M., Ugon, A., Duclos, C., Venot, A.: Using visual analytics for presenting comparative information on new drugs. J Biomed Inform **71**, 58–69 (2017)
17. Lamy JB: Ontology-Oriented Programming for Biomedical Informatics. Studies in health technology and informatics (STC) **221**, 64–68 (2016)
18. Lamy JB: Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. Artif Intell Med **80**, 11–28 (2017)
19. Lamy JB: A new diagram for amino acids: User study comparing rainbow boxes to Venn/Euler diagram. In: International Conference Information Visualisation (iV), pp. 361–366. Salerno, Italy (2018)
20. Lamy JB: Visualizing symmetric square matrices with rainbow boxes: methods and application to character co-occurrence matrices in literary texts. In: International Conference Information Visualisation (iV), pp. 344–349. Salerno, Italy (2018)
21. Lamy JB: Advances in nature-inspired computing and applications, chap. Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons, pp. 43–60. Springer (2019)
22. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. Semantic web **6**(2), 167–195 (2015). DOI 10.3233/SW-140134
23. Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., Pfister, H.: UpSet: visualization of intersecting sets. IEEE Transactions on visualization and computer graphics **20**(12), 1983–1992 (2014)
24. Leydesdorff, L., Vaughan, L.: Co-occurrence matrices and their applications in information science: Extending ACA to the Web environment. Journal of the association for information science and technology **57**(12), 1616–1628 (2006)
25. Longabaugh WJR: Combing the hairball with BioFabric: a new approach for visualization of large networks. BMC bioinformatics **13**, 275 (2012). DOI 10.1186/1471-2105-13-275
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
27. Rodgers P: A survey of Euler diagrams. Journal of Visual Languages and Computing **25**(3), 134–155 (2014)
28. Siirtola, H., Isokoski, P., Säily, T., Nevalainen, T.: Interactive Text Visualization with Text Variation Explorer. In: International Conference Information Visualisation (iV), pp. 330–335. Lisboa, Portugal (2016)
29. Valdivia, P., Buono, P., Plaisant, C., Dufournaud, N., Fekete, J.D.: Using Dynamic Hypergraphs to Reveal the Evolution of the Business Network of a 17th Century French Woman Merchant. In: 3rd workshop on visualization for the digital humanities (2018)
30. Wu, H., Tzeng, S., Chen, C.: Handbook of data visualization, chap. Matrix visualization, pp. 681–708. Springer (2008)
31. Yang, B., Ganascia, J.G.: Creating knowledge maps using Memory Island. International journal on digital libraries **18**(1), 41–57 (2017)
32. Yang, Y., Wu, M., Cui, L.: Integration of three visualization methods based on co-word analysis. Scientometrics **90**(2), 659–673 (2012)
33. Yang XS: Nature-inspired metaheuristic algorithms (second edition). Luniver Press (2010)
34. Zhang, J., Xie, J., Hou, W., Tu, X., Xu, J., Song, F., Wang, Z., Lu, Z.: Mapping the Knowledge Structure of Research on Patient Adherence: Knowledge Domain Visualization Based Co-Word Analysis and Social Network Analysis. Plos one **7**(4), 1–7 (2012). DOI 10.1371/journal.pone.0034497