RainBio: Proportional visualization of large sets in biology

Lamy Jean-Baptiste, Tsopra Rosy

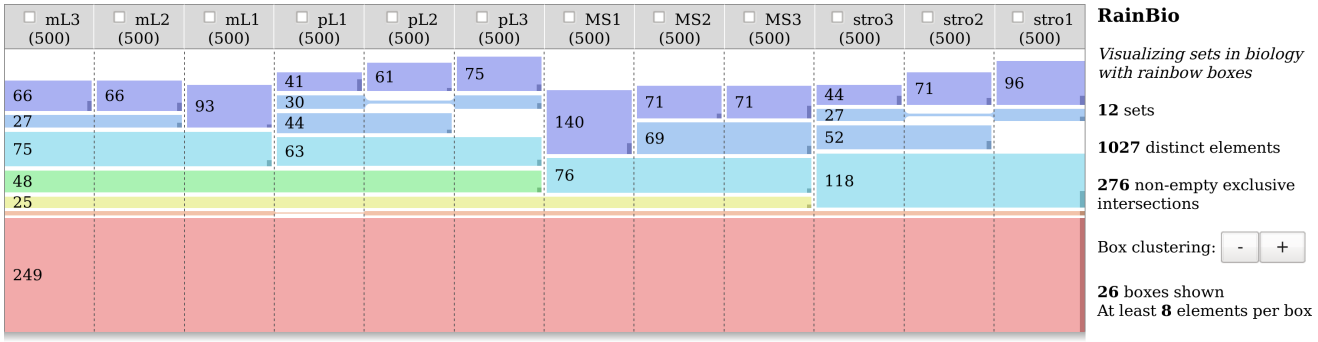

Fig. 1. RainBio displaying the comparison of 12 sets, after clustering. Each set includes the most expressed genes on a mammary tissue sample. Sets are displayed in columns and are ordered by similarity, showing that there are 4 types of tissue. The dataset has 276 non-empty exclusive intersections. Here, the 26 biggest ones are represented by colored boxes. Box color indicates the intersection degrees. The height of the box is proportional to the number of elements after clustering, while the darker bar on the right is proportional to the number of elements before clustering (*i.e.* exclusive elements). For example, there are 249 genes common to all sets, and 118 shared by $stro$ tissues, of which about 30% are exclusive to the three $stro$ tissue samples.

Abstract—Set visualization is a well-known task in information visualization. In biology, it is used for comparing visually sets of genes or proteins, typically using Venn diagrams. However, limitations of the Venn diagram are well-known: they are limited to 6 sets and difficult to read above 4. Many other set visualization techniques have been proposed, but they have never been widely used in biology. In this paper, we introduce RainBio, a technique for visualizing sets in biology and aimed at providing a global overview showing the size of the main intersections, in a proportional way, and the similarities between sets. We adapt rainbow boxes, a technique for visualizing small datasets, to the visualization of larger sets, using element aggregation and intersection clustering. We present the application of RainBio to three datasets, with 5, 6 and 12 sets. We also describe a small user study comparing RainBio with Venn diagrams, involving 30 students in biology. Results showed that RainBio led to significantly fewer errors on 6-set dataset, and that the majority of students preferred RainBio. RainBio is proposed as a web-based tool for up to 15 sets.

Index Terms—Gene set comparison, Set visualization, Venn diagram, Bioinformatics.

1 Introduction

Set visualization is a well-known task in information visualization. It considers some elements and several sets containing one or more of these elements. The sets may represent categories of elements, shared properties, or subsets of elements associated with a given condition. While the problem is intuitively simple, the number of possible set combinations increases exponentially with the number of sets, and thus the visualization becomes rapidly complex beyond 4 sets. A large literature exists on set visualization [1] and many techniques have been proposed.

In biology, set visualization is commonly used for comparing visually sets of genes or proteins. Usual datasets have a high number of elements (*e.g.* thousands of genes) but a small-to-medium number of sets. The two typical situations are: (a) Genes (or proteins, or gene clusters) are isolated in several biological samples (*e.g.* various species, tissues or health statuses). Biologists would like to visualize the genes specific to each sample, or shared by two or more samples. Here, genes are the elements and samples are the sets. (b) Genes (or proteins, *etc*) are identified as biomarkers for a given disorder using several methods. Biologists would like to compare the results obtained with the various methods and/or to compare the methods between themselves. Here, genes are the elements and there is one set per method.

Currently, the Venn diagram is the most used approach to visualize gene sets in biology. However, it has well-known limitations [2], [3]: it is difficult to generate automatically and to read when the number of sets increases. In practice it is limited to 6 sets. Other more recent set visualization techniques, such as UpSet [4], are less commonly used in biology, possibly because they often focus on detailed data mining. On the contrary, biologists sometimes expect a quick "one-screen" overview of the entire dataset or a "big picture" easy to publish in scientific journals.

Recently, we introduced *rainbow boxes* [5], [6], a technique originally able to visualize 2-25 elements and 5-100 sets. Figure 2 shows an example of rainbow boxes displaying a small dataset on planets. The elements are shown in columns, and the sets are represented by rectangular boxes placed below column headers.

• *Authors were with the LIMICS, Université Paris 13, Sorbonne Université, Inserm, 93017 Bobigny, France.*
E-mail: jean-baptiste.lamy@univ-paris13.fr, rosy.tsopra@aphp.fr

Manuscript received XXX, 2018.

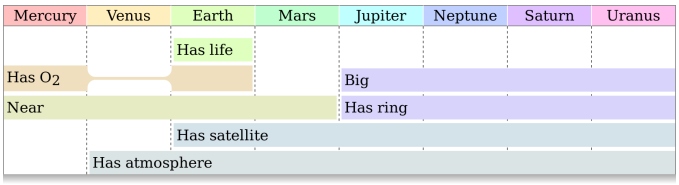

Fig. 2. Example of rainbow boxes showing the 8 planets of the solar system (elements/columns) and 7 properties (sets/boxes), *e.g.* the set of "Big" planets contains Jupiter, Neptune, Saturn and Uranus.

Each box covers the columns corresponding to the elements belonging to its set. The column order is computed using a heuristic optimization algorithm (hence planets are not ordered as usual in Figure 2). This algorithm tries to order the columns so as the elements belonging to each set are contiguous. When it is not possible to have them contiguous for a given set, a "hole" is present in the corresponding box and a small thread links the two parts of the box (*e.g.* in Figure 2, the "Has O_2" box has one hole). Colors are added to column headers and boxes: rainbow colors are associated with column headers, ranging across the spectrum, and the color of a box is the mean of the colors of the columns it covers. Finally, boxes are stacked vertically, with the largest boxes at the bottom. Two boxes can be next to each other, as long as they do not occupy the same columns. Rainbow boxes have already shown their utility in pharmaceutical domain [6]. Later, we proposed a proportional version of rainbow boxes [7] for representing artificial neurons.

In this paper, we present RainBio, a rainbow boxes-based tool for set visualization in biology. The main focus of RainBio is to provide a global overview or a "big picture" of a dataset, *i.e.* to visualize the main intersections and the potential set similarities in a diagram that can be displayed on a single screen of typical size, without the need for scrolling. Our main contributions are: (a) we adapt rainbow boxes to the visualization of large datasets (in terms of elements), supporting up to 15 sets and 40,000 elements, (b) we propose to cluster set intersections, which is a new approach in large set visualization, (c) we achieve the exact proportional visualization of 6 sets, which was not reported yet, (d) we compare our approach with others on several datasets and we present a small user study *versus* Venn diagrams.

The rest of the paper is organized as follows. Section 2 presents background on set visualization and describes the typical tasks required for biologists. Section 4 presents how we adapted rainbow boxes to the visualization of large sets. It also describes interactivity and implementation details. Section 5 illustrates the use of RainBio on three biological datasets. Section 6 argumentatively compares RainBio with other approaches. Section 7 describes a user study comparing RainBio with Venn diagrams on 5 and 6-set datasets. Finally, section 8 discusses the results, the limits of our approach and proposes perspectives.

2 Related works

Alsakallah *et al.* [1] reviewed techniques for overlapping set visualization. They distinguished 6 approaches: (1) Euler and Venn diagrams and their variants, (2) overlays on a map, (3) node-link diagrams, (4) matrix-based techniques, (5) aggregation-based techniques, and (6) scatter plot-based techniques. Another possible classification [4] is to distinguish element-centric approaches, in which elements are shown individually, and set-centric approaches, in which elements are aggregated and only sets are individualized. Table 1 shows the classification of the techniques mentioned below.

Table 1
Classification of set visualization techniques (E: element-centric, S: set-centric, (x): the approach is proportional but either displays only aggregated data (Set O'Gram) or does not relate visually all proportional intersections to their corresponding sets (Radial Sets, PowerSet)).

Technique	Category	-centric	Proportional	Similarity	Overview
Euler diagram	Euler/Venn	E	-	-	X
Venn diagram	Euler/Venn	S	-	-	X
Proportional Venn d.	Euler/Venn	S	X	-	X
LineSets	overlays	E	-	-	X
Bubble Sets	overlays	E	-	-	X
BiSet	node-link	E	-	-	-
Circular itemsets	node-link	S	-	-	X
Bicentric diagram	node-link	S	-	-	X
Linear diagram	matrix	E	-	-	X
Mosaic diagram	matrix	E	-	-	X
Rainbow boxes	matrix	E	X	X	X
Radial Sets	aggregation	S	(x)	-	X
Set O'Grams	aggregation	S	(x)	-	X
ConSet	aggregation+matrix	S	-	X	-
UpSet	aggregation+matrix	S	X	-	-
AggreSet	aggregation+matrix	S	X	X	-
PowerSet	aggregation	S	(x)	-	X
RainBio	clustering+matrix	S	X	X	X

Euler and Venn diagrams are one of the oldest approaches [8]. They are often used to teach set theory. In Euler diagrams, each set is represented by a closed-area [3]. The areas overlap in various regions that represent the (exclusive) intersections of the sets. A Venn diagram is a kind of Euler diagram showing all the $2^n - 1$ possible combinations of overlaps, where n is the number of sets. In a *proportional* Venn diagram, the size of the various regions is proportional to the number of elements in each regions. Venneuler [9] is a tool drawing such diagrams using circles, and nVenn another tool drawing quasi-proportional Venn diagrams [10] using closed shapes made of several circles. The automatic drawing of these diagrams is still a challenge above 4 sets [3].

In biology, many Venn diagram-based tools have been proposed: GeneVenn [11] (Venn diagram, limited to 3 sets), BioVenn [12] (proportional Venn diagram, limited to 3 sets), VennMaster [13], [14] (approximately proportional Venn diagram), JVenn [15] (Venn diagram, limited to 6 sets), InteractiVenn [16] (Venn diagram, limited to 6 sets, allows the analyze of set unions interactively), VennDiagramWeb [17] (Venn and Euler diagrams, limited to 5 sets) and VennPainter [18] (Venn diagram and nested Venn diagram, up to 8 sets). Most use Edward-Venn diagrams [19].

Overlay-based techniques are suited for datasets including a spatial component. Examples are LineSets [20], which display elements as points in the space, and sets by lines joining these points, and Bubble Sets [21], which display sets as bubbles including the corresponding elements. Extended LineSets [22] are a variant of LineSets for non-spatial datasets, such as biological pathways.

BiSet [23] is an improvement of node-link diagrams, in which the edges are bundled together to facilitate their reading and their manipulation. Circular itemsets [24] is another node-link technique, which represents intersections in concentric circles. The sets are positioned on the outer circle, then the intersections of 2 sets are positioned on the second circle, *etc*. Bicentric diagram [25]

Table 2
The typical tasks identified for set visualization in biology. The table indicates in which situations each task was encountered (comparison of several samples and/or comparison of several methods, as detailed in introduction), and the corresponding tasks in Alsallakh *et al.* classification [1].

#	**Task**	Samples	Methods	Alsallakh
1	How many elements belong to all sets?	X	X	B10
2	How many elements belong only to a given set X?	X	X	B12
3	What is the set X that contains the highest number of elements belonging to no other set?	X	X	B12
4	What is the biggest exclusive intersection (in general or limited to degree y)?	X	-	B10, B7
5	What is the number of elements in a given exclusive intersection?	X	-	B8, B10

considers two groups of two concentric circles.

Linear diagram is a matrix-based diagram [2], [26], introduced by Leibniz in 1686 [27]. Elements are displayed in columns and sets in rows. A piece of horizontal line is drawn in each cell at the intersection of an element that belongs to a set. Thus, a set is represented by one or more horizontal segments. Colors are usually added to identify the lines of a given set. Mosaic diagram [28], [29] is a space-filling variant of linear diagram. Rainbow boxes [5], [6] can be seen as an evolution of linear diagram, despite the fact that they move away from matrices, by allowing the representation of several sets in a single row. They also permit a proportional variant [7].

In Radial Sets [30], only aggregated information on sets is displayed, using bubbles and histograms organized in a ring. Another aggregation-based technique is Set O'Grams [31]. It represents sets in a bar chart, the height of each bar indicating the cardinality (*i.e.* number of elements) of the set. Bars are divided in several segments, each segment containing the elements that belong to a fixed number of sets (*e.g.* elements belonging to a single set, to 2 sets, *etc*). This technique heavily relies on interaction for relating the various segments and identifying intersections.

Several techniques combine matrix-based visualization with the display of aggregated values using charts. ConSet [32] represents set intersections as pies and sets as a permutation matrix. The tool is interactive, allowing reordering the rows and columns of the matrix. It also displays fan diagrams, a simplified form of the Venn diagram, for subsets of the data. UpSet [4] combines a matrix-based approach, showing the various set combinations similarly to a linear diagram, with aggregated values, such as a bar chart showing the cardinality of each intersection. It also supports advanced interactive queries for the creation of user-defined aggregations. For each set, the user can restrict the visualization to elements belonging, or not, to that set. Finally, UpSet supports the visualization of element attributes (*i.e.* set-typed data). AggreSet [33] is another approach for set-typed data. It creates aggregations for set intersections, set pairs and set degrees, and represents the cardinality of each aggregation. A co-occurrence matrix is used for visualizing set pairs. Interactive options are provided for selection and filtering.

PowerSet [34] is based on Treemaps [35], and displays intersections as rectangles, the area being proportional to the intersection cardinality. Intersections are sorted by degree (*i.e.* the number of sets they involve). Over- and underrepresented intersections are highlighted with colors. PowerSet allows distinguishing very well the main intersections, but the identification of sets is more difficult, since a given set is divided into several, unrelated, rectangles.

3 Requirement and Tasks Analysis

Alsallakh *et al.* [1] proposed a task classification for set visualization. It includes 3 main categories: (A) tasks related to elements, (B) tasks related to sets and (C) tasks related to element attributes. For identifying tasks in biology, we first gathered published papers in bioinformatics [16], [36] and biology [37]–[40] presenting Venn diagrams. These papers do not contain task descriptions, but we extracted the diagrams and the insights mentioned in the text citing the figures. Then, we derived questions and tasks from these insights, and we mapped them to Alsallakh *et al.* classification. For example, in [16], a sentence introduces a Venn diagram and then states that "all methods retrieved 38 common proteins". We derived this into task #1, "How many elements belong to all sets?". We also reviewed the task classification proposed for biological pathway visualization [41]; although mostly graph-based, pathway visualization sometimes involves set visualization (*e.g.* task R3 in [41]). Finally, we completed this information with our expertise in the field, to identify 5 typical tasks for gene set visualization in biology (Table 2). All five tasks belong to category B (tasks related to sets). This was expected, since the elements (*i.e.* genes) are too numerous to be visualized individually (for category A), and there is no per-element attributes (for category C). The number of tasks in Table 2 is limited and they cover only four tasks in Alsallakh *et al.* classification: B7 (identify the set involved in a certain intersection), B8 (identify set intersections belonging to a specific set), B10 (analyze and compare set and intersection cardinalities) and B12 (analyze and compare set exclusiveness). Some tasks in Table 2 correspond to several tasks in Alsallakh *et al.* classification, *e.g.* for task #4, after finding the biggest intersection (B10), a biologist usually wants to know which sets it involves (B7).

In addition, task B11 (analyze and compare set similarity, *e.g.* through a similarity measure) is probably of interest for biologists. However, since this task is not supported by the Venn diagram [1] they commonly use, it is not expressed as such in papers. Biologists often identify similarities between set through the finding of the largest intersections. When comparing gene sets from several samples, a large intersection between two or more sets means that the corresponding samples share many genes, and thus are similar with regard to those genes. On the contrary, the small intersection between two or more sets does not necessarily imply the absence of similarity, because most set visualization techniques (including Venn diagrams) actually represent *exclusive* intersections, *i.e.* elements belonging to the intersection of some sets *and* not belonging to any other intersection of a highest degree. For example, two sets s_1 and s_2 may have a small (exclusive) intersection, while being similar because the intersection of s_1, s_2 and another set s_3 is large. Consequently, biologists are usually more interested in the largest intersections than in the smallest ones.

Alsallakh *et al.* tasks related to inclusion, hierarchy, pairwise intersection and subset selection seems less important in biology. The limited task coverage suggests that the needs of biologists are actually focused on rather specific tasks.

Many existent tools can be used to perform tasks B7, B8, B10 and B12, including the Venn diagram and UpSet, the latter supporting a very wide range of tasks. However these tools are not specifically optimized for the needs of biologists. We identified the main characteristics of the ideal set visualization approach in biology: (a) It should be set-centric, because the number of elements is often very high (several thousand or more). (b) It should provide a global "one-screen" overview, and thus should avoid displaying one intersection per row/column because the number of intersections increases exponentially with the number of sets. (c) It should facilitate the identification of set similarities. This can be done through a proportional visual approach, in which similar sets have similar shapes, like in the proportional Venn diagram, and/or through a set similarity measurement, typically by reordering the rows or the columns of a matrix according to similarity.

Table 1 indicates how the various approaches support these characteristics. No approach satisfies all the criteria listed above. Near misses are the proportional Venn diagram (but it cannot be generated exactly above 4 sets), rainbow boxes (but element-centric and thus unable to display large sets), ConSet, UpSet and AggreSet (but they provide limited global overview, due to their "one intersection per row" approach).

Another particularity of biological datasets should be taken into account in the design: when a given gene is present in two samples, each sample has its own copy of the gene (and not a single copy shared by all samples). Consequently, when considering proportional set visualization, one may consider an area proportional to the number of gene copies, rather than proportional to the number of distinct genes, *e.g.* a gene present in two samples would occupy an area twice larger than the area occupied by a gene present in a single sample. Existing proportional visualizations (including proportional Venn diagram) do not take into account this point.

In the present work, we propose to adapt rainbow boxes in order to produce a new set-centric proportional approach, supporting set similarity and overview, targeting the typical tasks we identified in biology and considering the number of gene copies for proportionality.

4 Adapting Rainbow Boxes for the Visualization of Large Sets

When comparing sets in biology, the number of elements is usually high and the number of sets is limited. This would result in rainbow boxes with 1000-10,000 columns and 2-6 boxes, which is impractical: a standard screen cannot show so many columns simultaneously. Consequently, in this section we adapt rainbow boxes to the visualization of datasets with a high number of elements, by displaying aggregated data instead of showing each element individually. We also propose a novel method for intersection clustering, we define a new color scheme and we add interactivity.

4.1 Columns and boxes

The general principles we propose for visualizing large sets in RainBio are the following (see example in Figure 3). Each column corresponds to a set, and each box to a set combination. The box covers the columns corresponding to the sets in its set combination, *e.g.* the box for the set combination $\{A, B\}$ will occupy the two columns A and B. The box is labeled with the cardinality of the *exclusive intersection* of the sets in its combination. We call *exclusive intersection* of some sets, the elements belonging to the set intersection *and* not belonging to any other set. Exclusive intersections correspond to the regions of the Venn diagram; for clarity, we will refer to them as "intersections" in the following, in opposition to "standard intersections". The height of the box is proportional to the cardinality of the intersection, and the color of a box indicates the intersection degree. When they are too numerous, boxes are clustered: smallest boxes are removed and their elements are moved to taller boxes (this will be described in section 4.2).

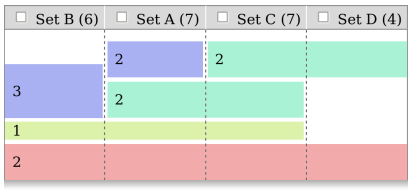

Fig. 3. RainBio showing a trivial 4-set dataset.

Consequently, the set membership of an intersection is encoded spatially, by the horizontal position of the corresponding box. The cardinality of an intersection is also encoded spatially, by the vertical dimension of the box. Finally, intersection degree is encoded by both the box color and vertical position (with intersections of higher degree at the bottom). The rest of the section gives a detailed description.

A set dataset can be formalized as a set of sets $S = \{S_1, S_2, ..., S_i, ..., S_n\}$, where n is the number of sets ($n \geq 2$). Sets S_i can be overlapping, *i.e.* a given element may belong to several sets. $E = \bigcup_{i \in \{1,...,n\}} S_i$ is the set of all elements. The intersection of one or more sets in S is the set of elements that belong to the standard intersection of those sets and that belong to no other sets in S. The function $X()$ computes the exclusive intersection:

$$X(c \subseteq S) = \bigcap_{x \in c} x \setminus \bigcup_{y \in S \setminus c} y$$

Let $C = \{c \subseteq S : X(c) \neq \emptyset\}$ be the set of combinations of S with a non-empty intersection. When the number of such combinations is low, all of them can be displayed in a separate box, and clustering is not needed. For a given box, identified by its set combination, the box function $B()$ returns the elements represented in that box. Without clustering, the elements in a box are simply the exclusive intersection: $B = X$.

In order to have proportional rainbow boxes, the height of the box for a set combination c is $H_c = \max(|B(c)| \times k, H_{min})$ where $|B(c)|$ is the cardinality of the box. k is a scaling factor, inversely proportional to the number of elements in E. It allows maintaining a similar global height, whatever the number of elements is. H_{min} is the minimum allowed height. H_{min} prevent boxes being too small when $|B(c)|$ is very low. Each box is labeled with the number of elements it represents, *i.e.* $|B(c)|$, provided that the box is tall enough to include a label.

Finally, we define a new color scheme for rainbow boxes. The color of the box for set combination c depends on the number of sets in c: we use a gradient of hues, from blue (a single set) to red (the maximum number of sets). Thus, hotter colors are attributed to boxes involving more sets. In addition, boxes are organized vertically by colors: "hotter" boxes involving more sets are lower.

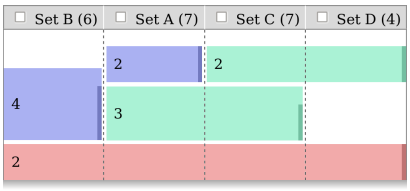

Fig. 4. Same dataset as Figure 3, after clustering (with $t = 2$).

4.2 Intersection clustering

When the number of intersections is too high, it is not possible to display one box for each while keeping a readable visualization. In this case, we propose to cluster intersections. We define a clustering threshold $t \geq 1$. Intersections with at least t elements are used as "seeds" and will be associated with a box. In addition, intersections involving a single set are always selected as seeds, for two reasons: (a) they are associated with boxes of length one, which cannot have holes, and (b) it ensures that all elements will be clustered in at least one box, consequently, no element disappears during clustering. Let $C_s = \{c \subseteq S : |X(c)| \geq t \vee |c| = 1\}$ be the set of seed combinations.

Set combinations $d \in C \setminus C_s$ have less than t elements in their intersection and will not have their own box. The elements in their intersections $X(d)$ will be displayed in the boxes of the seed combinations that are the biggest available subsets of d. Consequently, with clustering, boxes display subsets of elements that are somewhat in-between exclusive and standard intersection. For a set combination c in C_s, the corresponding box displays not only the exclusive intersection of c, but all elements that belong to the standard intersection of c and that do not belong to another box associated with a set combination that is a (strict) superset of c. The box function is now:

$$B(c \subseteq S) = \begin{cases} if\ c \in C_s : & \bigcap_{x \in c} x \setminus \bigcup \{B(c') : c' \in C_s \wedge c \subset c'\} \\ otherwise : & \emptyset \end{cases}$$

$B(c)$ can be computed recursively, starting with the largest combinations c. The height of boxes is proportional to $|B(c)|$, the number of elements after clustering. In addition, we add a small dark bar on the right of the box, whose height is proportional to $|X(c)|$, the number of exclusive elements (*i.e.* before clustering). This bar indicates "how exclusive" the box is. It is deliberately subtle, in order to limit visual clutter. It uses the color value visual variable, while the color hue is used to indicate the intersection degree. Those two visual variables are selective according to Bertin's semiology of graphics [42], allowing the user to focus his attention on one of these variables and ignoring the other. For example, Figure 4 shows the dataset of Figure 3 after clustering with $t = 2$. The yellow box with 1 element has been removed, and its element is now counted in the two boxes that were just above the yellow one.

4.3 Optimizing column order

In rainbow boxes, column order must be optimized for minimizing the number of holes in the boxes. Holes add visual clutter, thus boxes with holes tend to be harder to read. This can be particularly problematic if an important box has holes. Since biologists are often interested in the largest intersections, we improved the optimization criteria proposed previously [6] by taking into account H_c, the height of the boxes, during optimization. The cost for adding a hole in a box is equal to H_c. This prevents holes in taller boxes, to the detriment of the smallest ones.

For a given set order O, the total hole cost is the sum of the number of holes multiplied by the corresponding box height. It is computed by the function h defined as follows:

$$h(O) = \sum_{c \in C_s} H_c \times \Big| \{ i \in I = \{ind(O, s \in c)\} : i + 1 \notin I \wedge i \neq \max(I) \} \Big|$$

where $ind(O, s)$ is a function that returns the index of a given set s in the order O (starting at index 1). The optimization process aims at finding the best order $O^{best} = \arg\min_O (h(O))$.

When the number of columns is below 10, column order can be optimized with a brute force algorithm that tests all possible orders. When the number of columns is above 10, we proposed a metaheuristic algorithm [43].

4.4 Adding interactivity

Interactivity was added for two purposes. First, we used detail-on-demand to display additional information. When the mouse cursor is over a box, a popup label displays the sets involved in that box and the number of elements. This is especially interesting for boxes that are too small to display a label. When the user clicks a box, a new window is open, listing the elements in this box. If several intersections were clustered, the list is organized by intersections. When the mouse cursor is over a column header, a popup label displays the number of intersections, before and after clustering, involving the corresponding set.

Second, two options have been added for filtering out boxes. The first one consists of controlling clustering. We added a panel on the right of rainbow boxes (shown in Figure 1). It displays various statistics, such as the total number of elements and sets, and allows controlling the clustering threshold t with the two buttons "-" and "+". Buttons can be held down in order to display the various steps of clustering as an animation. The minimum allowed value for t is the lower one that produces at most 64 boxes, *i.e.* we limited the visualization to at most 64 boxes.

The second filtering option aims at selecting boxes involving one or several particular sets. We added a checkbox for each set, in the column header. By default, they are unchecked. When a checkbox is checked, boxes corresponding to set combinations that do not involve this set are faded (see example Figure 8). When several checkboxes are checked, their effect is additive: all boxes that do not involve all the selected sets are faded.

4.5 Complexity

Intersections can be determined exponentially with n, since the number of set combinations is 2^n. However, it can also be performed linearly with the number of elements $m = |E|$, because the number of non-empty intersections is at most m (in fact, exclusive intersections are a partition of E). When n is high, the linear complexity according to m is usually preferable since it grows slower. Thus, the complexity is $\mathcal{O}(m)$ or $\mathcal{O}(2^n)$, depending on the chosen algorithm (both can be implemented for selecting the most appropriated one). Similarly, clustering can be performed in $\mathcal{O}(s \times m)$ or $\mathcal{O}(s \times 2^n)$, where $s = |C_s|$ is the number of boxes after clustering (limited to 64 in our implementation).

For column ordering, the brute force algorithm has a factorial complexity $\mathcal{O}(n!)$. The metaheuristic algorithm finds a near-optimal column order in a much faster computation time, however, its complexity is almost impossible to assess.

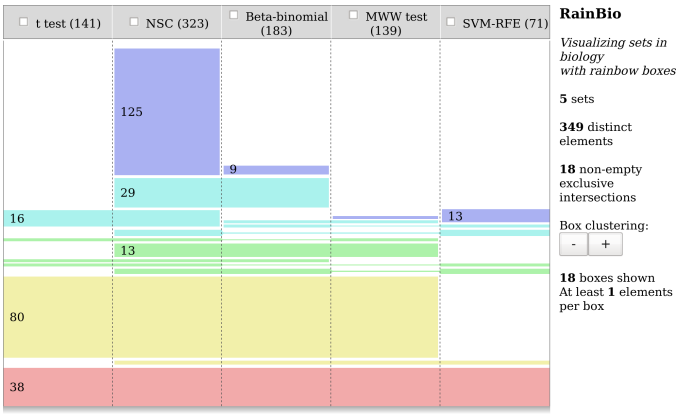

Fig. 5. RainBio showing the prostate biomarker dataset (5 sets).

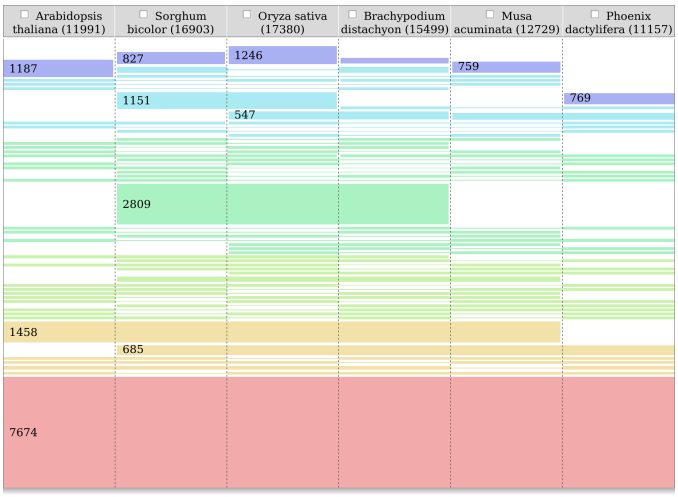

Fig. 6. RainBio showing the banana dataset (6 sets), without clustering (the right panel with clustering control is not shown).

4.6 Implementation

RainBio has been implemented in Python 3. It is available online at the following address: http://www.lesfleursdunormal.fr/static/appliweb/rainbio, with several demo datasets (tested with Mozilla Firefox and Google Chrome). The web application is limited to 15 sets and 40,000 elements, and uses the same file format as InteractiVenn. A video is also available as Supplementary Material.

5 Application to various datasets

In this section, we illustrate the use of RainBio. The first two datasets were initially used for demonstrating InteractiVenn [16]. The third one shows the ability of RainBio to compare more than 6 sets. These datasets were analyzed by JBL, who teaches bioinformatics at university.

5.1 Dataset #1: prostate biomarker (5 sets)

This dataset comes from a study aimed at determining biomarkers for distinguishing two types of prostate cancers [37]. It includes 349 candidate biomarker proteins and 5 sets, each containing the proteins found as valid biomarkers by a given feature selection method: univariate Beta-binomial, semi-multivariate Nearest Shrunken Centroids (NSC), multivariate Support Vector Machine-Recursive Features Elimination (SVM-RFE) and Student's *t*-test and MWW test. The original dataset included 3 methods, and the last 2 were added by the designers of InteractiVenn [16]. 17 intersections are non-empty, out of 31.

Figure 5 shows the dataset in RainBio. If one is interested in comparing the 5 methods, the following insights can be gained through the visualization: (a) No biomarkers were found only by *t*-test (no blue box in the "t test" column). (b) 125 proteins are found as biomarkers only by NSC (the tall blue box at the top); this suggests that NSC produces a lot of false positives (since the four other methods agreed that these proteins are not biomarkers). (c) 80 proteins are found as biomarkers by all methods excepted SVM-RFE (the tall yellow box); this suggests that SVM-RFE produces a lot of false negatives. Consequently, Beta-binomial, *t*-test and MWW are the most consensual methods. If one is interested in finding biomarkers, RainBio allows identifying easily the 38 proteins that are found as valid biomarkers by all the 5 methods (the red box at the bottom). In addition, one may want to consider the biomarkers found by the three most consensual methods.

5.2 Dataset #2: banana (6 sets)

This dataset was designed for comparing the genome of the banana (*Musa acuminata*) with 5 other plant species (*Phoenix dactylifera*, *Oryza sativa*, *Sorghum bicolor*, *Brachypodium distachyon* and *Arabidopsis thaliana*) [38]. The dataset includes 23,143 gene clusters and all the 63 possible intersections are non-empty.

Figure 6 shows the dataset in RainBio. Despite many boxes are small and difficult to individualize, RainBio gives a global overview of the dataset. One can gain the following insights from the visualization: (a) There are 7673 gene clusters shared by the 6 plants (the red box at the bottom). This represents roughly 50-70% of the gene clusters of each species, suggesting some similarities between the plants. (b) There is a high number (2,809) of gene clusters shared by 3 plants (*Sorghum bicolor*, *Oryza sativa* and *Brachypodium distachyon*, the tall green box). These three species are thus close to each other. In addition, clustering can be used to simplify the visualization, by removing the smallest boxes.

5.3 Dataset #3: mammary tissues (12 sets)

This publicly available[1] dataset [44] was designed for comparing 4 types of mammary tissues: fibroblast-enriched stromal (stro), mammary stem cell (ML), luminal progenitor (pL), mature luminal (mL). There are three replicates for each tissue, leading to 12 samples. For each, microarray profiling was used to obtain the expression levels of 28,458 genes. The objective of the study was to determine whether each tissue has a distinct gene expression profile.

In order to produce sets from microarray results, we retain for each sample the most expressed 500 genes. This yielded a dataset with 12 sets (one per sample) of 500 genes each, with 1,027 distinct genes. Figure 7 shows the dataset in RainBio, with the minimum level of clustering allowed (this dataset has 276 non-empty intersections, but as mentioned above we limited RainBio to at most 64 boxes). Figure 1 shows the dataset with a higher level of clustering, resulting in a simpler and more synthetic view. Clustering not only removes the smallest boxes, but also strengthens boxes having a large non-exclusive intersection.

1. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE16997

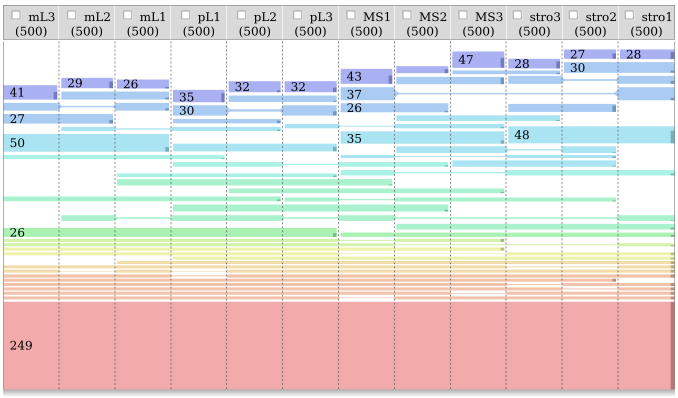

Fig. 7. RainBio showing the mammary tissue dataset (12 sets). This is the same dataset as Figure 1, but with a lower clustering threshold ($t = 3$ leading to 63 boxes).

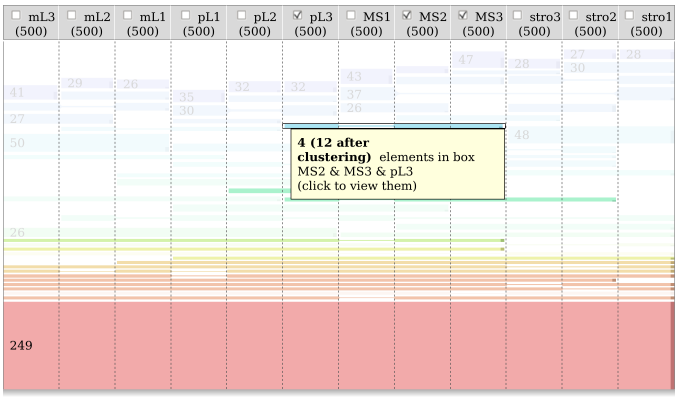

Fig. 8. RainBio showing the mammary tissue dataset, after the user checked 3 checkboxes, for pL3, MS2 and MS3.

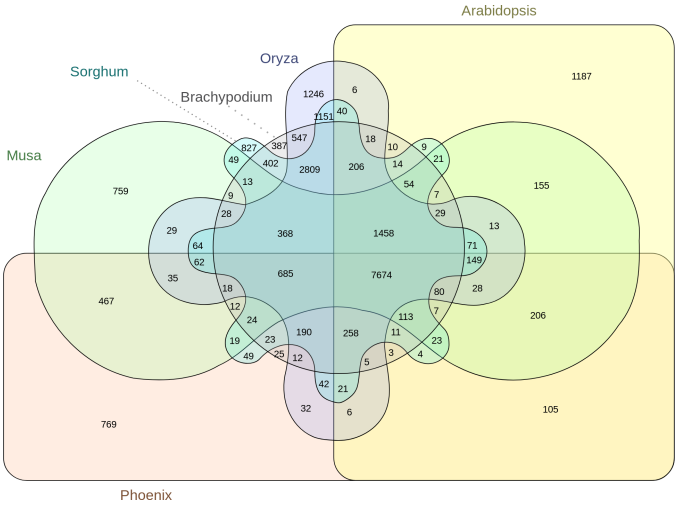

Fig. 9. Venn diagram showing the banana dataset (from [16], CC-BY).

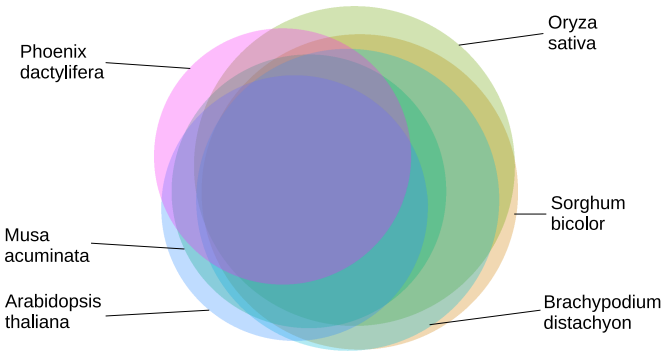

Fig. 10. Proportional Venn diagram displaying the banana dataset.

Column order provides an interesting insight here. Columns were ordered as described previously, by box similarity. The four types of tissue are well-separated from each other: the three stroma samples are grouped in three contiguous columns, *etc*. Therefore, it is possible to respond to the original question that motivated the study, just by looking at the visualization: yes, each tissue has a distinct gene expression profile.

Four tall light blue boxes (labeled 50, 35 and 48 in Figure 7, the last one being unlabeled) correspond to the overexpressed genes shared by the samples of each type of tissue. With a higher level of clustering (Figure 1), the smallest boxes are pruned. The four light blue boxes are taller, because the elements of the missing boxes are moved to other boxes (for example if there is a single gene overexpressed in $pL1$, $pL2$, $pL3$ and $stro1$, it might be clustered with the box for $pL1$, $pL2$ and $pL3$, increasing the size of this box).

Finding a given box can be difficult on huge datasets, especially if the box is small. Figure 8 shows how the checkboxes in column headers can help. In this example, the user wants to identify the box covering $MS1$, $MS2$ and $pL3$. After checking the three checkboxes corresponding to these sets, boxes that do not include at least these three sets are faded. Consequently, the desired box is the top-most non-faded box (unless it is empty).

Other insights can be obtained from the visualization: (a) There are 249 genes shared by all samples (red box); this high number is expected since all tissues are mammary. (b) pL and mL tissues share several overexpressed genes (the green box labeled 26 in Figure 7, and 48 in Figure 1); since both tissues are luminal, it is not surprising that they are similar. (c) $stro$ seems the most different type of tissue (higher number of specific genes shown by the tall dark bar in its light blue box, and no tall green or yellow box indicating an important number of genes shared with other types of tissue).

6 Comparison with other approaches

6.1 Venn diagrams

Figure 9 shows dataset #2 (banana) using Edward-Venn diagram (drawn with InteractiVenn). Identifying the previously mentioned large 3-set intersection (with 2809 genes) is more difficult in the Venn diagram, while it literally "pops out" in RainBio (Figure 6) thanks to its proportional nature. On the other hand, the Venn diagram allows a better reading of the smallest regions.

6.2 Proportional Venn diagrams

Figure 10 shows dataset #2 (banana) using a proportional Venn diagram (generated with Venneuler [9]). The large overlaps between sets make the diagram complex to read. The previously mentioned large 3-set intersection can be identified on the right of the diagram. Since no method exists for drawing an exact proportional Venn diagram of 6 sets, Figure 10 is approximate: it displays only 30 intersections out of 63. In particular, the 759 gene clusters found only in *Musa acuminata* are not shown: the "Musa" circle has no

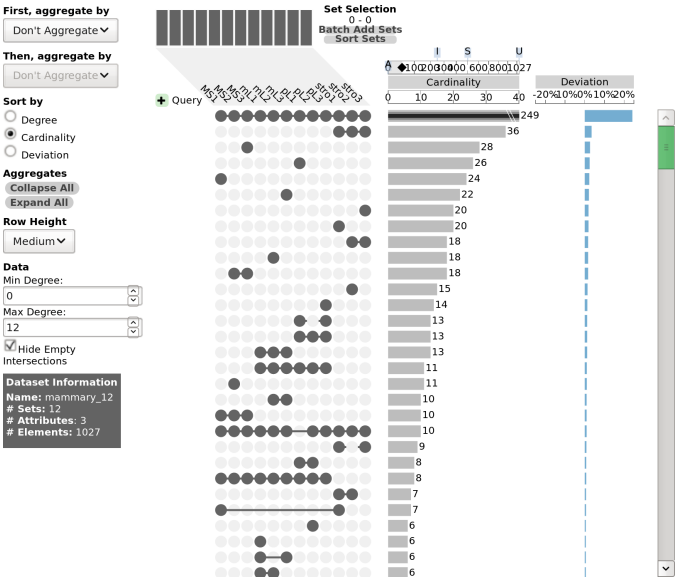

Fig. 11. UpSet displaying the mammary tissue dataset.

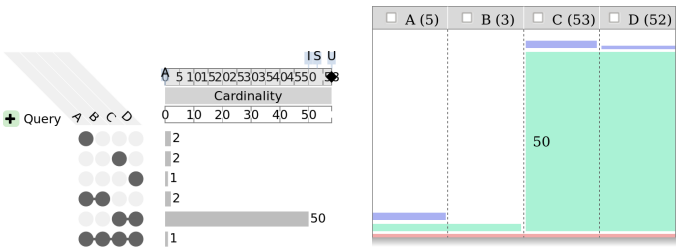

Fig. 12. UpSet and RainBio displaying the same small dataset with huge disparities in intersection cardinalities.

region that does not overlap any other circles. This is especially problematic here, since the objective of the initial study was to compare *Musa acuminata* to other plants. Using a proportional Venn diagram, biologists might wrongly conclude that banana has no specific gene clusters. On the contrary RainBio allows an exact proportional visualization with 6 sets (Figure 6).

6.3 UpSet

UpSet [4] is a recent set visualization technique. Figure 11 shows dataset #3 (mammary tissues) with UpSet. It displays the 30 largest intersections. Both RainBio and UpSet are matrix-based set visualization and thus RainBio is closer to UpSet than to Venn diagrams. The main difference is that UpSet encodes the sets involved in an intersection and the intersection cardinality in two separated visual elements: (1) connected dots in a matrix, and (2) a bar chart aligned with the matrix. On the contrary, RainBio combines these two pieces of information in one: the connected dots are replaced by a rectangular box whose height encodes the intersection cardinality. With UpSet, the user has to put the matrix in relation to the bar chart and the column headers, which requires to focus visual attention on three places. With RainBio, the user has only to focus his attention on boxes and column headers. This

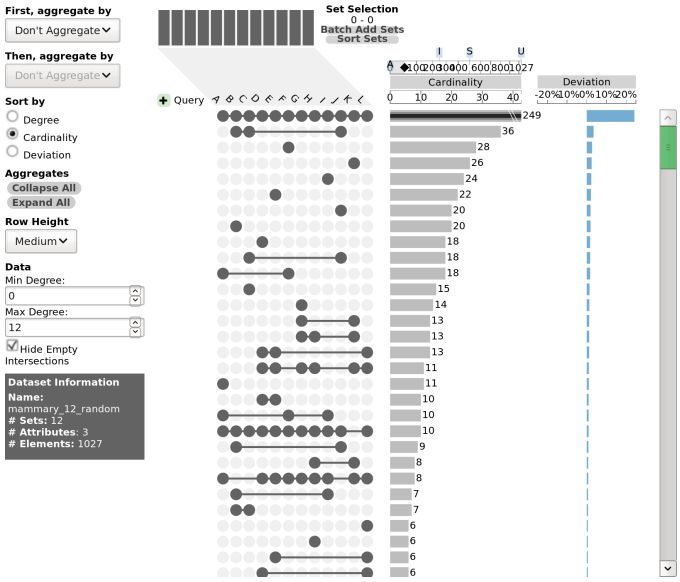

Fig. 13. UpSet displaying the mammary tissue dataset, when the order of the columns/sets is unknown.

arguably facilitates the reading of the intersections by removing an extra level of indirection.

In addition, since there is no bar chart aligned with the matrix in RainBio, it allows packing multiple intersections (*i.e.* boxes) next to each other, in the same "row", as long as the boxes do not overlap. This has two advantages. First, it makes RainBio potentially more compact than UpSet. For example, in Figure 7, RainBio displays 63 intersections while, in Figure 11, UpSet displays only 30, out of 276. However, RainBio shows only 21 cardinality numbers while UpSet shows 30. Second, packing multiple intersections in the same "row" may facilitate the identification of intersection disjointedness: any boxes that are next to each other do not overlap and thus are necessary disjoint.

However, the layout of UpSet is simpler and more uniform than in RainBio. Both are biased towards large intersections, in different ways: RainBio displays larger intersections in bigger boxes while UpSet shows them first. But this bias is higher in RainBio. This is an advantage of UpSet, especially when there is a huge disparity in intersection cardinalities: UpSet can display all cardinalities with numbers while RainBio cannot (see example Figure 12).

A second difference is that UpSet provides intersection aggregations by degree, sets, deviations and overlaps, while RainBio provides intersection clustering. UpSet displays the largest intersections at the top and the user has to interact with a scroll bar to display the others. For example, in Figure 11, only 662 elements are represented, out of 1027 (about 64%). But scrolling can be tedious, especially if there is a "long tail" of many small intersections. Here, Figure 11 is the first screen out of 9, as suggest the size of the vertical scroll bar holder. On the contrary, the use of clustering in RainBio allows displaying at least partial membership information for *all* elements in a single screen.

A third difference is that UpSet has no automatic sets/columns ordering while RainBio automatically orders them by similarity using a metaheuristic. Similarity-based set ordering can greatly help users. Let us imagine that biologists are not aware that there are 4 types of mammary tissues in dataset #3, and that they want

Table 3
The questions with the correct answers. Questions marked by (*) correspond to insights mentioned in the original publications.

		Question	Correct answer
5-set dataset	-	How many biomarkers are found only by SVM-RFE?	13
	#1	How many biomarkers are found by all the 5 methods? (*)	38
	#2	How many biomarkers are found only by beta-binomial?	9
	#3	Which method finds the highest number of biomarkers that are not found by other methods? (*)	NSC
	#4	What are the two methods finding the highest number of common biomarkers, not found by any other method?	Beta-binomial and NSC
	#5	How many biomarkers are found by both SVM-RFE and NSC (and no other method)?	6
6-set dataset	-	How many gene clusters are present only in Arabidopsis?	1187
	#1	How many gene clusters are shared by the 6 plants? (*)	7674
	#2	How many gene clusters are present only in M. acuminata? (*)	759
	#3	Which species has the most specific gene clusters (=not present in other species)? (*)	Oryza
	#4	Which are the 3 plants that share the highest number of gene clusters (present only in these 3 plants)? (*)	Arabidopsis, Oryza, Sorghum
	#5	How many gene clusters are shared by Phoenix, Musa and Arabidopsis (and not shared with other species)?	206

to *discover* those tissue types. In Figure 11, sets were ordered alphabetically, which resulted in grouping them by tissue types (*e.g.* stroma, mL,...). This is no longer possible if tissue types are unknown. Figure 13 shows the dataset with UpSet, using a random set order. In this figure, discovering the four tissue types is more difficult. It requires to find visually that intersections B-C-J ($stro$ previously), G-H-K (pL), D-E-L (mL) and A-F-I (MS) are large *and* that they make a partition of the set of all sets, *i.e.* the intersections include each set once and only once. This is more difficult in UpSet than in RainBio, because UpSet represents intersection cardinalities and sets separately (as said above), and it is even more difficult when sets are not ordered by similarity. Manually ordering the sets to simplify the visualization in UpSet would be tedious, because there are $12! = 479,001,600$ candidate orders. On the contrary, discovering the four tissue types is easier in RainBio: the fact that some boxes are next to each other facilitates the identification of disjoint intersections (which are requirements for partitions), and sets are automatically ordered by similarity. Thus, the set order in the dataset has no impact, and the visualization remains the same as in Figure 1 and 7. This example shows that UpSet depends a lot on the set order (arbitrary order in the dataset file or alphabetical order). However, a good visualization should be independent from any arbitrary order.

To conclude, UpSet has a more uniform global layout and provides much more interactive options, such as two-level aggregations or user-defined queries. It also provides intersection deviations and supports the analysis of set-typed data. Thus, it is arguably better for detailed data mining or when intersection cardinalities are very disparate. On the other hand, thanks to tighter packing and clustering, RainBio can display more information than UpSet in a single screen. Therefore, RainBio provides arguably a more comprehensive "one-screen" overview. In addition, similarity-based column ordering may help find relation between sets. These two points correspond to ideal requirements (b) and (c) we identified in section 3.

7 User Study

In this section, we present a small user study. Its objective was to compare RainBio and Venn diagrams for the visualization of 5 and 6-set datasets, with regard to the five typical tasks we identified in biology (Table 2). The Venn diagram was chosen as a comparator, because (a) it is, by far, the most widely used approach in biology (as the large number of Venn diagram-based tools for biology suggests, see section 2), (b) it has also been chosen as a comparator during other studies, *e.g.* for linear diagrams [45] and (c) more recent approaches, such as UpSet, have not yet been evaluated during user study.

7.1 Recruitment

30 biology students were recruited at our university in the first year of master degree (M1). Most of them were female; this is usual in biology courses. About half of them were from foreign countries, thus various cultures were represented. The study was performed during the course of bioinformatics and students were not aware that the study will take place at this moment. They were told that the objective of the study was to compare two tools, but not that the Venn diagram is a well-established technique and RainBio is a challenger. Some students already used Venn diagrams in other courses.

7.2 Protocol

The study was anonymous and no personal information was recorded (such as age or sex). We used a balanced crossover protocol in which each student tested both tools. We used datasets #1 and #2, with 5 and 6 sets, initially produced for demonstrating InteractiVenn. For each, we designed 6 questions: a "warm-up" question (whose results were not analyzed) and 5 other questions (Table 3), one per typical task described in section 3. 6 questions (out of 10) corresponded directly to insights mentioned by the authors of the original publications [16], [37], [38]. The two questions #5 were deliberately designed so as they target small intersections, the cardinality of which is not shown in RainBio boxes (thus requiring user interaction). Each question had a single correct answer.

Students were randomly divided in two groups. The first group had to reply to the questions on the 5-set dataset with the Venn diagram and to the questions on the 6-set dataset with RainBio, and *vice versa* for the second group. All students began with the simpler 5-set dataset. Thus, both groups had the same questions and datasets, in the same order, but not with the same tool. We recorded the error rate and the response time. Consequently, the independent variables were the dataset (5-set or 6-set), the tool used (RainBio or Venn diagram) and the question identifier (#1-5), and the dependent variables were the response accuracy and the response time. Finally, the last question asked the student about his preferred tool, with three possible choices: RainBio, Venn diagram, or no opinion.

7.3 Hardware and software

The study was performed on the classroom desktop computers (Intel Core i5-6500 processors at 3.20 GHz, 8 Gb RAM) running Debian GNU/Linux. They were equipped with 19' screens with a resolution of 1280x1024 pixels.

A dynamic website was developed for the study. The website was responsible for the randomization in the two groups. The first page presented briefly the objective of the study. The second page presented the first tool (depending on the group in which the user was randomized). The next pages corresponded to the questions with the 5-set dataset. Then the second tool was presented, followed by the questions with the 6-set dataset. Venn diagrams were based on InteractiVenn (without the panel for authoring the dataset). In RainBio, clustering was removed, since InteractiVenn does not have it. When the response to a question was a number, students had to enter it. When the response was one or more sets, student had to choose the response from 5 predefined values. Finally, the last page asked the question related to user preference. The website collected error rates and response times.

7.4 Statistical analysis

Statistical analysis was performed using R version 3.3.2 [46]. The significance threshold was set at $\alpha = 0.05$. Error rate was the main criteria; it was compared with Fisher's exact test and Global Linear Model (GLM), considering two factors: tool (Venn diagram or RainBio) and dataset size (5 or 6). Response time was log-transformed to normalize the distribution, and analyzed with Welch two sample *t*-test.

7.5 Results

All students performed the entire study and responded to all questions. We collected 150 responses with each tool.

35 errors were recorded with the Venn diagram (error rate 23.3%) and 25 with RainBio (error rate 16.7%). This difference is not significant (p value = 0.19, Fisher exact test). GLM showed that dataset size has a significant relationship with error rate (p = 0.015) and that there is a significant interaction between tool and dataset size (p = 0.0004). Thus we analyzed each dataset separately. With the Venn diagram, there were 8 and 27 errors for the 5- and 6-set datasets, respectively. With RainBio, there were 14 and 11 errors, respectively. For the 5-set dataset, the difference is not significant (p = 0.11). For the 6-set dataset, the difference is significant (p = 0.0006).

The mean response time was 31.6 seconds with the Venn diagram *vs* 32.6 seconds with RainBio. This difference is not significant (p value = 0.45, Welch two sample *t*-test performed on log(time)). 10 (33.3%) students preferred the Venn diagram, while 17 (56.7%) preferred RainBio (the 3 others indicated no opinion).

In conclusion, RainBio led to significantly fewer errors on the 6-set dataset, and was preferred by the majority of the students.

8 Discussion and Conclusion

In this paper, we presented RainBio, a visualization technique for providing a global overview of large sets in biology. We adapted rainbow boxes for the visualization of datasets with a high number of elements, by using aggregation based on exclusive intersection, by representing sets in columns rather than in boxes, and by defining a new color scheme, allowing the visualization of up to 15 sets. We proposed intersection clustering, a novel approach to set visualization. In particular, we proposed the exact proportional visualization of up to 6 sets, which was not reported yet to our knowledge. We demonstrated the use of RainBio on 3 biological datasets of various sizes, and we compared it with several other techniques (Venn diagram, proportional Venn diagram and Upset). Finally, we compared RainBio with Venn diagrams in a small user study. We showed that students made fewer errors with RainBio on 6-set datasets, and that it was preferred by the majority of the students.

8.1 Visualization technique

In RainBio, we opted for a proportional approach to set visualization: larger intersections are represented by taller boxes. In addition, we gave more importance to largest intersections when determining the column order (section 4.3). Favoring the largest intersection is an approach followed by several other set visualization tools, including PowerSet, Radial Sets and the proportional Venn diagram. In a similar spirit, UpSet allows sorting intersection by cardinality, thus focusing on the largest ones, and BiSets allows hiding individual edges. This proportional approach facilitates the identification of the largest intersections and the discovery of similarities between sets (as seen in section 3), at the price of making the smallest intersections harder to identify, or even hiding them. This might bias the visualization towards the largest intersections. However, when comparing several samples, biologists are often interested in largest gene clusters rather than by smallest ones. In addition, size is a selective visual variable [42], and thus selecting visually the smallest boxes remains possible.

Various designs were considered during the development of RainBio. First, it is known that the Human vision is usually more sensitive to surface than to size [47]. Thus, we tried to encode intersection cardinality as box area (instead of box height). However, rainbow boxes have a "semi-table-like" aspect, with clearly delimited columns but without rows. Thus, it favors a separate interpretation of the horizontal and vertical dimensions, as usual in tables. Moreover, when the number of sets is high, the area-proportional encoding leads to very small boxes for intersections of high degree. In addition, as seen at the end of section 3, when a gene is shared between two samples, each sample has its own copy of the gene. Consequently, using box height for encoding cardinality, actually encodes the number of observed copies as box area: box area represents the number of copies (*i.e.* counting 1 copy per sample) while box height represents the number of distinct genes (*i.e.* without duplicates). For these reasons, we preferred height-proportional encoding of cardinality, especially when visualizing gene sets. Second, we considered the use of variable column widths, proportional to the cardinality of each set. However, variable column widths imply that a box representing the exclusive intersection of two or more sets occupy a largest area in the largest column. This might lead to the biased perception that the largest column plays a more important role with regard to the intersection. Thus, we opted for fixed column widths.

RainBio displays boxes horizontally. An alternative option would be to display boxes vertically (*e.g.* rotating the whole visualization by 90°). Since computer screens are usually wider than tall, a vertical disposition would provide more space for boxes. However, rainbow boxes stack boxes at the bottom of the screen, mimicking the action of gravity: this behavior is more natural than stacking them on the left or on the right. For this reason, we chose to display boxes horizontally.

Table 4
The various tasks related to sets and set relations in Alsallakh *et al.* classification [1]. The last column indicates whether user interaction is required or not (V: visually, Ic: interactively using checkboxes, Id: interactively using details-on-demand, Ig: interactively using clustering).

#	Task	How to perform the task with RainBio	
B1	Find out the number of sets in the set family	Count the number of columns	V
B2	Analyze inclusion relations	Find visually a column where all boxes are shared with another column	V
B3	Analyze inclusion hierarchies	- (not supported)	-
B4	Analyze exclusion relations	Find visually the absence of a box covering two or more columns (checkboxes can help if many boxes are present)	V, Ic
B5	Analyze intersection relations	Find visually the box covering two or more columns (checkboxes can help)	V, Ic
B6	Identify intersections between k sets	Select visually boxes of a given color	V
B7	Identify the sets involved in a certain intersection	Find visually the columns covered by a given box	V
B8	Identify set intersections belonging to a specific set	Select visually the boxes in a given column	V
B9	Identify the set with the largest/smallest number of pairwise set intersections	Find visually the column with the tallest/smallest 2-set boxes (clustering can help by reducing and merging boxes)	V, Ig
B10	Analyze and compare the cardinality of sets and intersections	Find visually the tallest/smallest boxes (detail-on-demand is required for small boxes, and clustering can help)	V, Id, Ig
B11	Analyze and compare set similarities	Select visually adjacent column and/or find column sharing many and tall boxes	V
B12	Analyze and compare set exclusiveness	Compare visually the blue 1-column boxes	V
B13	Highlight specific sets, subsets, or set relations	Use checkboxes to fade out boxes not including one set	Ic
B14	Create a new set using set-theoretic operation	- (not supported)	-

On dataset #3, we showed that RainBio could be used to discover similarities between sets and to group them in clusters. In dataset #3, these clusters corresponded to the already known categories of tissue. However, if those categories were not yet known, RainBio could have been used to discover them. Consequently, RainBio might be used for *clustering* sets and for *unsupervised learning*. Other visual approaches allow unsupervised learning, for example principal component analysis (PCA) associated with scatter plots allows clustering visually objects described by an object-property matrix. Other set visualization approaches, such as ConSet [32], were able to achieve visual clustering through columns and rows reordering, using different algorithms.

The main limit of the proposed technique is the visualization of datasets having many sets but few elements. The number of intersections grows exponentially with the number of sets. With more than 10 sets, intersection cardinalities tend to be lower. In extreme cases, there can be as many non-empty intersections as elements, each intersection including a single element. In these cases, the visualization in RainBio is not informative and the clustering method we proposed cannot be applied (since we select the largest intersections as seeds). This situation occurs typically with purely random datasets; on the contrary, the mammary tissue dataset has some large exclusive intersections that "structure" data.

8.2 Intersection clustering

In the literature, existing approaches, including UpSet [4] and AggreSet [33], rely on aggregation (or grouping) rather than clustering: they provide predefined aggregations, such as "grouping intersections by degree", "grouping intersections by set" or "grouping intersections by pair of elements", or allows the user to create his own aggregates. Aggregation gives a more general view of the data. However, the user has little control over the number of aggregates produced: if the number is high, all aggregates cannot fit on the screen and it is difficult for the user to obtain a global overview. On the contrary, in RainBio, intersection clustering gives to the user a better control over the number of clustered intersections, *via* the parameter t.

Clustering leads to an information loss. However, it is not comparable with approximately proportional Venn diagrams, because clustering can be controlled and it ensures that the visible intersections are always the largest ones. The clustering method we proposed guarantees that all intersections with t elements or more are visible, allowing the quantification of the information loss. On the contrary, on dataset #2, the approximately proportional Venn diagram (Figure 10) missed a large intersection of 759 elements, and it is difficult for the user to evaluate how approximated a diagram is.

8.3 Set visualization tasks

Alsallakh *et al.* [1] proposed a task classification for set visualization. Since RainBio follows a set-centric approach and does not support set-typed data, we are only interested in category B (tasks related to sets). In section 2, we identified 5 typical tasks for biologists; RainBio targets particularly those tasks. Nevertheless, it can be used to achieve other tasks beyond those five. Table 4 shows the set-related tasks, and how RainBio allows achieving them. Some tasks in the classification can be divided in subtasks, for example task B10 (cardinality of an intersection) involves sets (*i.e.* $|A|$), standard intersections ($|A \cap B|$) and unions ($|A \cup B|$). In some cases, RainBio does not always support all possible subtasks, for example for task B10, RainBio is efficient for evaluating the cardinalities of intersections, but not for the ones of unions. Similar remarks hold for tasks B2, B4, B12 and B13.

Compared to the Venn diagram (and its task description by Alsallakh *et al.* [1]), RainBio also supports task B1, B6, B11 and B12. In the literature, other set visualization techniques have achieved a higher coverage of Alsallakh *et al.* task classification: for instance, UpSet supports 23 of the 26 tasks [4], including all tasks of category B excepted B11 (set similarity). Task B11 is not supported by the Venn diagram and UpSet, while it is by RainBio.

RainBio is focused on the global overview of the dataset. Consequently, in Table 4, most tasks can be achieved visually, without interacting with the system. Interaction is required for only one task (B13), and four other tasks require interaction only in certain circumstances (*e.g.* small boxes) or may be enhanced through interaction. This limited use of interactivity could facilitate the use of RainBio by biologists in scientific publications, because it is much easier to publish a still picture than an interactive interface, for technical reasons (some journals are paper-based)

but also because readers may not be trained in the use of the interactive interface and therefore may not be able to find the expected insights from the visualization. In our opinion, this point might explain why the Venn diagram is still widely used in biology, despite its limitations are known by biologists themselves (*e.g.* the Venn diagram is "*not effective for presentation of more than four categorical groups*" [48]).

On the contrary, many recent tools such as UpSet or AggreSet are more focused on the interactive analysis of set data. Those tools allow the creation of user-defined aggregations, but require more complex user interaction. They also frequently use scroll bars, making harder to obtain a global overview of the data.

8.4 User study

For the user study, we deliberately reused datasets produced for InteractiVenn, in order to limit biases in the selection of the datasets (*i.e.* we did not design specific datasets that would favor RainBio). We used a crossover protocol, in which each subject tested both tools on different datasets, and thus can be his own control. Crossover protocols are known to reduce the inter-subject variability [49], which is important when the number of subjects is rather low. We showed that students made fewer errors with RainBio on a 6-set dataset, compared to the Venn diagram. On the contrary, on a 5-set dataset, we observed fewer errors with the Venn diagram, although the difference was not significant. Some questions on the 5-set dataset might have favored the Venn diagram, for example, the two sets involved in question #5 were next to each other on the Venn diagram but not on RainBio. In addition, surprisingly, we observed slightly fewer errors with RainBio on the 6-set dataset than on the 5-set one (although the difference is not significant).

Response time was slightly higher (although not significantly higher) with RainBio than with Venn diagrams. This might be explained because RainBio requires user interaction for answering some questions (such as finding the number of genes in a given exclusive intersection, questions #5 in Table 3), while the Venn diagram does not. The fact that some students already used the Venn diagram might also be an explanation. Finally, for the 6-set dataset, the response times can difficultly be analyzed without considering the difference in the number of errors.

The presented user study showed that RainBio could advantageously replace the Venn diagram for 6 sets. However, the study did not evaluate the new features proposed by RainBio, such as clustering and the ability to visualize a higher number of sets than the Venn diagram. Future user studies should consider larger datasets and be focused on clustering, probably using another comparator than Venn diagrams.

In the literature, the limits of Euler and Venn diagrams have already been shown, compared to matrix-based approach such as linear diagram. Chapman *et al.* [45] compared four techniques for set visualization: Venn diagram, standard Euler diagram (using complex shapes), Euler diagram with shading (*i.e.* Euler diagram using ellipses, empty regions being shaded) and linear diagram. They measured response times and error rates, and they showed that the linear diagram performed the best. In our user study, we obtained similar results on error rates for 6-set dataset (but not in response times). However, our study differs since we compared proportional rainbow boxes to Venn diagrams showing numbers, while in the Chapman *et al.* study, intersections of Venn diagrams were empty, and there exists no proportional version of the linear diagram. Moreover, contrary to Chapman *et al.*, we measured user preferences in addition to error rates and response times. In user studies, effectiveness (measured here by error rates), efficiency (error rates / response times ratio) and satisfaction (*e.g.* user preference) are not necessarily correlated [50], and thus each of them should be assessed independently.

8.5 Perspectives

In this paper, we focused on applications in biology for comparing gene or protein sets. However, RainBio could be used beyond to biology. Thus, a first perspective of this work is to adapt RainBio for set visualization in other domains. Examples include co-authorship relations in bibliographic databases or cloned software systems [51]. Another perspective is the use of RainBio for unsupervised learning, either visually, or even entirely automatically, by identifying similarities between sets. A third perspective is to develop a set visualization tool for data mining, combining the global overview proposed by RainBio with advanced queries and filtering such as those proposed in other tools.

REFERENCES

[1] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "The State-of-the-Art of Set Visualization," *Computer Graphics Forum*, vol. 35, no. 1, pp. 234–260, 2016.
[2] Gottfried B, "Set space diagrams," *Journal of visual languages & computing*, vol. 25, no. 4, pp. 518–532, 2014.
[3] Rodgers P, "A survey of Euler diagrams," *Journal of Visual Languages and Computing*, vol. 25, no. 3, pp. 134–155, 2014.
[4] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, "UpSet: visualization of intersecting sets," *IEEE Transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1983–1992, 2014.
[5] J. B. Lamy, H. Berthelot, and M. Favre, "Rainbow boxes: a technique for visualizing overlapping sets and an application to the comparison of drugs properties," in *International Conference Information Visualisation (iV)*, Lisboa, Portugal, 2016, pp. 253–260.
[6] J. B. Lamy, H. Berthelot, C. Capron, and M. Favre, "Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain," *Journal of Visual Language and Computing*, vol. 43, pp. 71–82, 2017.
[7] J. B. Lamy and R. Tsopra, "Translating visually the reasoning of a perceptron: the weighted rainbow boxes technique and an application in antibiotherapy," in *International Conference Information Visualisation (iV)*, London, United Kingdom, 2017, pp. 256–261.
[8] Baron ME, "A note on the historical development of logic diagrams: Leibniz, Euler and Venn," *The Mathematical Gazette*, vol. 53, no. 384, pp. 113–125, 1969.
[9] Wilkinson L, "Exact and approximate area-proportional circular Venn and Euler diagrams," *IEEE Transactions on visualization and computer graphics*, vol. 18, no. 2, pp. 321–331, 2012.
[10] J. G. Pérez-Silva, M. Araujo-Voces, V. Quesada, and J. Wren, "nVenn: Generalized, quasi-proportional Venn and Euler diagrams," *Bioinformatics*, vol. 34, no. 13, pp. 2322–2324, 2018.
[11] M. Pirooznia, V. Nagarajan, and Y. Deng, "GeneVenn - A web application for comparing gene lists using Venn diagrams," *Bioinformation*, vol. 1, no. 10, pp. 420–2, 2007.
[12] T. Hulsen, J. de Vlieg, and W. Alkema, "BioVenn - a web application for the comparison and visualization of biological lists using area-proportional Venn diagrams," *BMC genomics*, vol. 9, p. 488, 2008.
[13] H. A. Kestler, A. Müller, T. M. Gress, and M. Buchholz, "Generalized Venn diagrams: a new method of visualizing complex genetic set relations," *Bioinformatics (Oxford, England)*, vol. 21, no. 8, pp. 1592–5, 2005.
[14] H. A. Kestler, A. Müller, J. M. Kraus, M. Buchholz, T. M. Gress, H. Liu, D. W. Kane, B. R. Zeeberg, and J. N. Weinstein, "VennMaster: area-proportional Euler diagrams for functional GO analysis of microarrays," *BMC bioinformatics*, vol. 9, p. 67, 2008.
[15] P. Bardou, J. Mariette, F. Escudié, C. Djemiel, and C. Klopp, "jvenn: an interactive Venn diagram viewer," *BMC bioinformatics*, vol. 15, p. 293, 2014.
[16] H. Heberle, G. V. Meirelles, F. R. da Silva, G. P. Telles, and R. Minghim, "InteractiVenn: a web-based tool for the analysis of sets through Venn diagrams," *BMC bioinformatics*, vol. 16, p. 169, 2015.

[17] F. Lam, C. M. Lalansingh, H. E. Babaran, Z. Wang, S. D. Prokopec, N. S. Fox, and P. C. Boutros, "VennDiagramWeb: a web application for the generation of highly customizable Venn and Euler diagrams," *BMC bioinformatics*, vol. 17, no. 1, p. 401, 2016.

[18] G. Lin, J. Chai, S. Yuan, C. Mai, L. Cai, R. W. Murphy, W. Zhou, and J. Luo, "VennPainter: A Tool for the Comparison and Identification of Candidate Genes Based on Venn Diagrams," *PloS one*, vol. 11, no. 4, p. e0154315, 2016.

[19] Edwards AWF, *Cogwheels of the mind: The story of Venn diagrams*. Baltimore: Johns Hopkins University Press, 2004.

[20] B. Alper, N. H. Riche, G. Ramos, and M. Czerwinski, "Design study of LineSets, a novel set visualization technique," in *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, 2011, pp. 2259–2267.

[21] 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, 2009.

[22] F. Paduano and A. G. Forbes, "Extended LineSets: a visualization technique for the interactive inspection of biological pathways," *BMC proceedings*, vol. 9, no. Suppl 6 Proceedings of the 5th Symposium on Biological Data, p. S4, 2015.

[23] M. Sun, P. Mi, C. North, and N. Ramakrishnan, "BiSet: Semantic edge bundling with biclusters for sensemaking," pp. 310–319, 2016.

[24] G. Bothorel, M. Serrurier, and C. Hurter, "Visualization of frequent itemsets with nested circular layout and bundling algorithm," in *International Symposium on Visual Computing*, 2013, pp. 396–405.

[25] H. Park and R. C. Basole, "Bicentric diagrams: Design and applications of a graph-based relational set visualization technique," *Decision Support Systems*, vol. 84, pp. 64–77, 2016.

[26] P. Rodgers, G. Stapleton, and P. Chapman, "Visualizing sets with linear diagrams," *ACM Transactions on Computer-Human Interaction*, vol. 22, no. 6, p. 27, 2015.

[27] Couturat L, *Opuscules et fragments inédits de Leibniz*. Felix Alcan, 1903.

[28] S. Luz and M. Masoodian, "Visualisation of parallel data streams with temporal mosaics," in *11th International Conference Information Visualization IV'07*, 2007, pp. 197–202.

[29] ——, "A comparison of linear and mosaic diagrams for set visualization," *Information visualization*, vol. 1473871618754343, 2018.

[30] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser, "Radial Sets: Interactive Visual Analysis of Large Overlapping Sets," in *IEEE Transactions on Visualization and Computer Graphics (Proceedings Information Visualization 2013)*, vol. 19, no. 12, 2013, pp. 2496–2505.

[31] 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, 2008.

[32] B. Kim, B. Lee, and J. Seo, "Visualizing set concordance with permutation matrices and fan diagrams," *Interacting with computers*, vol. 19, no. 5-6, pp. 630–643, 2007.

[33] M. A. Yalcin, N. Elmqvist, and B. B. Bederson, "AggreSet: Rich and scalable set exploration using visualizations of element aggregations," *IEEE Transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 688–697, 2016.

[34] B. Alsallakh and L. Ren, "PowerSet: A comprehensive visualization of set intersections," *IEEE Transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 361–370, 2017.

[35] B. Johnson and B. Shneiderman, "Treemaps: a space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the 2nd International IEEE Visualization Conference*, San Diego, 1991, pp. 284–291.

[36] S. Jahangiri-Tazehkand, L. Wong, and C. Eslahchi, "OrthoGNC: A Software for Accurate Identification of Orthologs Based on Gene Neighborhood Conservation," *Genomics, proteomics & bioinformatics*, vol. 15, no. 6, pp. 361–370, 2017.

[37] Y. Kim, V. Ignatchenko, C. Q. Yao, I. Kalatskaya, J. O. Nyalwidhe, R. S. Lance, A. O. Gramolini, D. A. Troyer, L. D. Stein, P. C. Boutros, J. A. Medin, O. J. Semmes, R. R. Drake, and T. Kislinger, "Identification of differentially expressed proteins in direct expressed prostatic secretions of men with organ-confined versus extracapsular prostate cancer," *Molecular & cellular proteomics : MCP*, vol. 11, no. 12, pp. 1870–84, 2012.

[38] A. D'Hont, F. Denoeud, J. M. Aury, F. C. Baurens, F. Carreel, O. Garsmeur, B. Noel, S. Bocs, G. Droc, M. Rouard, C. Da Silva, K. Jabbari, C. Cardi, J. Poulain, M. Souquet, K. Labadie, C. Jourda, J. Lengellé, M. Rodier-Goud, A. Alberti, M. Bernard, M. Correa, S. Ayyampalayam, M. R. Mckain, J. Leebens-Mack, D. Burgess, M. Freeling, D. Mbéguié-A-Mbéguié, M. Chabannes, T. Wicker, O. Panaud, J. Barbosa, E. Hribova, P. Heslop-Harrison, R. Habas, R. Rivallan, P. Francois, C. Poiron, A. Kilian, D. Burthia, C. Jenny, F. Bakry, S. Brown, V. Guignon, G. Kema, M. Dita, C. Waalwijk, S. Joseph, A. Dievart, O. Jaillon, J. Leclercq, X. Argout, E. Lyons, A. Almeida, M. Jeridi, J. Dolezel, N. Roux, A. M. Risterucci, J. Weissenbach, M. Ruiz, J. C. Glaszmann, F. Quétier, N. Yahiaoui, and P. Wincker, "The banana (Musa acuminata) genome and the evolution of monocotyledonous plants," *Nature*, vol. 488, no. 7410, pp. 213–7, 2012.

[39] G. H. Villarino, Q. Hu, S. Manrique, M. Flores-Vergara, B. Sehra, L. Robles, J. Brumos, A. N. Stepanova, L. Colombo, E. Sundberg, S. Heber, and R. G. Franks, "Transcriptomic Signature of the SHATTERPROOF2 Expression Domain Reveals the Meristematic Nature of Arabidopsis Gynoecial Medial Domain," *Plant Physiol*, vol. 171, no. 1, pp. 42–61, 2016.

[40] D. R. Nelson, B. Khraiwesh, W. Fu, S. Alseekh, A. Jaiswal, A. Chaiboonchoe, K. M. Hazzouri, M. J. O'Connor, G. L. Butterfoss, N. Drou, J. D. Rowe, J. Harb, A. R. Fernie, K. C. Gunsalus, and K. Salehi-Ashtiani, "Chloroidium sp. UTEX 3007 reveal adaptive traits for desert acclimatization," *eLife*, vol. 6, 2017.

[41] P. Murray, F. McGee, and A. G. Forbes, "A taxonomy of visualization tasks for the analysis of biological pathway data," *BMC bioinformatics*, vol. 18, no. Suppl 2, p. 21, 2017.

[42] Bertin J, *Semiology of graphics*. University of Wisconsin Press, Madison, 1983.

[43] Lamy JB, *Advances in nature-inspired computing and applications*. Springer, 2019, ch. Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons, pp. 43–60.

[44] E. Lim, F. Vaillant, D. Wu, N. C. Forrest, B. Pal, A. H. Hart, M. L. Asselin-Labat, D. E. Gyorki, T. Ward, A. Partanen, F. Feleppa, L. I. Huschtscha, H. J. Thorne, kConFab, S. B. Fox, M. Yan, J. D. French, M. A. Brown, G. K. Smyth, J. E. Visvader, and G. J. Lindeman, "Aberrant luminal progenitors as the candidate target population for basal tumor development in BRCA1 mutation carriers," pp. 907–913, 2009.

[45] P. Chapman, G. Stapleton, P. Rodgers, L. Micallef, and A. Blake, "Visualizing sets: An empirical comparison of diagram types," in *International Conference on Theory and Application of Diagrams*, 2014, pp. 146–160.

[46] R team development core, *R: A language and environment for statistical computing*, Vienna, Austria, 2008.

[47] Ware C, *Visual thinking for design*. Burlington, USA: Morgan Kaufmann, 2008.

[48] A. Shade and J. Handelsman, "Beyond the Venn diagram: the hunt for a core microbiome," *Environ Microbiol*, vol. 14, no. 1, pp. 4–12, 2012.

[49] B. Jones and M. G. Kenward, *Design and analysis of cross-over trials (3rd edition)*. Chapman and Hall/CRC, 2014.

[50] E. Frøkjær, M. Hertzum, and K. Hornbæk, "Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated?" in *Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems*, The Hague, The Netherlands, 2000, pp. 345–352.

[51] V. Tenev, S. Duszynski, and M. Becker, "Variant analysis: Set-based similarity visualization for cloned software systems," in *Proceedings of the 21st International Systems and Software Product Line Conference*, 2017, pp. 22–27.

Jean-Baptiste Lamy (PharmD, PhD) is a senior lecturer at University Paris 13 in the LIMICS laboratory. He teaches bioinformatics and medical informatics. His main research interests are information and knowledge visualization, knowledge representation and clinical decision support.

Rosy Tsopra (MD, PhD) is a university hospital assistant at University Paris 13 and hospital Avicenne. Her main research interest is medical informatics.