VisualBib: A novel Web app for supporting researchers in the creation, visualization and sharing of bibliographies

DOI: https://doi.org/10.1016/j.knosys.2019.07.031

VisualBib: a novel Web app for supporting researchers in the creation, visualization and sharing of bibliographies

Antonina Dattolo[a,*], Marco Corbatto[a]

[a]*SASWEB Lab, Department of Mathematics, Computer Science and Physics, University of Udine*
Via Santa Chiara, 1 - Gorizia, Italy
http://sasweb.uniud.it

Abstract

In this paper, we present VisualBib, a Web application, which allows users to create, visualize, modify, explore, and share bibliographies and the related citation networks, using innovative diagrams, called *narrative views*. The metadata are retrieved in real-time from four existing bibliographic indexes, Scopus, OpenCitations, and CrossRef/Orcid. Bibliographies and views are formally described and modelled using zz-structures, a semantic, not-hierarchical data model. VisualBib has been evaluated through two evaluation studies, one focused on the quantitative side and another on the qualitative side. Taking into account both studies, they evaluate the tool regarding the effectiveness performing tasks, usability, graphic layout and other questions specific to the VisualBib features. The evaluation throws positive significant results in all areas when compared to Scopus searching features.
Keywords: Visual bibliographies, Visualization, Narrative views, Citation networks, Zz-structures, Bibliographic indexes, Web application, Visual organizers.

1. Introduction

A common task for researchers consists in the exploration of the scientific literature for creating and saving, in a reusable format, significant scientific references on specific research topics. The searches are generally carried out on big citation indexes like Scopus, Web of Science (WOS), CrossRef, Google Scholar, Microsoft Academic, OpenCitations and others,

*Corresponding author
Email addresses: antonina.dattolo@uniud.it (Antonina Dattolo), marco.corbatto@uniud.it (Marco Corbatto)
[1]Both the authors contributed equally to this research.

by specifying a set of keywords, the title of a publication or the author names. The results are generally presented in a long list of items, where it is not simple to identify the relations between papers (for example, co-authors, co-citations, temporal order of publications), the typologies of publications (journal, conference, book) or to get a general idea of a specific author's production. Furthermore, each subsequent search brings new results; attempting to aggregate the data would require a considerable effort for the researcher who should manually examine them, find the connections and discard duplicate entries in order to consolidate a set of significant papers.

The idea of adopting visual representations, to show bibliographic data and support users in their analysis and interpretation, has been widely studied [1] and several tools for the visualization of citation networks have been proposed [2]. Unfortunately, all these tools work on bibliographic datasets, which must be retrieved in advance from specific citation indexes. They do not manage multiple sources of data, except through a manual merge of specific datasets; they do not offer rapid methods to share a bibliography.

Our proposal is to offer a new Web application called VisualBib, that interfaces directly with four large bibliographic data providers through API (Application Programming Interface) services in order to retrieve updated information about papers, authors and citations.

From a task-oriented point of view, we have identified three main purposes for using VisualBib:

- **The visual representation of bibliographies** through an overall and interactive view that highlights the semantic connections between authors and papers, the distribution of publications and publication types over time and some collaboration metrics. The proposed VisualBib's main interface, called *narrative view*, is presented in Section 3 and formally defined in Section 6.

- **The support for the creation of bibliographies** through the real-time query of multiple bibliographic indexes, the selection and integration of the retrieved data, the exploration of the citation networks, the importing of external BibTeX archives and the progressive refinement of the bibliography.

- **The sharing of bibliographies** via the Web, their saving on a cloud space, their embedding in external Web pages or their exporting in a standard format.

VisualBib has been designed and implemented as an *personal assistant for researchers* who would like to represent, communicate and share their selections of the scientific production in a certain topic/domain related to a given author (or set of authors); it represents a visual tool able to query in real-time different bibliographic indexes and provide visual analytics insights.

Two previous papers on VisualBib [3, 4] introduce the first prototype of VisualBib and some initial experimental results. In this work, we describe formally and informally the VisualBib app. The new contribution of this work is mainly focused on:

- formal description of VisualBib in terms of zz-structures, with the formal definitions of two new *deep* and *narrative* views;
- integration of the two new bibliographic indexes, CrossRef and Orcid, used in a combined way;
- introduction of new features, like *MachtAuthor*; *Import/Export BibTeX*; the insertion of a *histogram*, representing the distribution over time of the publications; an *area chart*, illustrating the distribution of the publications types; and the possibility of *multiple selection of authors* to highlight the publications in common and their number;
- proposal of some new graphic aspects of the application, like the organization of the search section, the disposition of the items in the narrative view, the highlight of search result in the view and in the list of papers;
- comparative analysis of the API provided by the major current bibliographic indexes;
- a new quantitative and qualitative evaluation, on larger group of participants, and on a more extensive and articulated questionnaire.

A dedicated Website `http://visualbib.uniud.it` documents the evolution of the application and allows interested people to use it.

The rest of the paper is organized as follows: Section 2 discusses related work, emphasizing open issues and challenges in visual representation of bibliographies; Section 3 presents our tool, VisualBib, introducing its basic functionalities and user interface; Section 4 presents a list of low-level data analysis tasks supported by our tool in comparison with existing bibliographic indexes and analyzes the API services offered by them; Section 5 proposes a formal semantic data model of VisualBib based on zz-structures; Section 6 formalizes two new original zz-views, the *deep view* and the *narrative view*, integrated in the user interface of VisualBib; Section 7 describes the architecture of VisualBib and some implementation details of the modules; Section 8 illustrates a comparative (VisualBib-Scopus) quantitative and qualitative user evaluation of VisualBib. Conclusions and future work end the paper.

2. Related work

A general survey [1] examines 109 different visual approaches to analyze scientific literature and patents, that came to light between 1992 and 2016; this work, together with an interactive visual survey [5] of 400 different techniques for text visualization and with a graphical review [6] of the research on visual languages from 1995 to 2014, highlights the fundamental role of visual representations for a meaningful use of publications' metadata for scientific communities. Starting from these studies and from the rest of current literature, we focus our attention on:

- a set of interesting visual tools, which propose graphic representations of bibliographic data and support researchers in exploring bibliographies. However, it must be said that the purpose of these tools is the visual analysis of large bibliographic dataset in order to cluster information and highlight relations and data patterns, while VisualBib is a tool conceived to support researchers in building up and managing small and medium-sized bibliographies as dynamic spaces where it is possible to add, delete, explore and merge new data. For this reason, the application we propose is not directly comparable with these tools; hence, we discuss and compare only some specific aspects;
- ten, widely-used bibliographic indexes, as, for example, AMiner, Google Scholar, MS Academic, Scopus, WOS.

Comparison with ten visual tools. Table 1 compares ten tools considering the proposed views, the typology of implementation (Web or stand-alone application), the modality of retrieving

Applications	Views	Implementation	Real-time data	Data Integration	Under Development
CiteSpace [7] 2004-today	Interactive visualizations of structural and temporal patterns	Stand-alone	Pre-built, static dataset	No	Active
PaperLens [8] 2004-2005	Views across papers, authors and references	Stand-alone for Windows	Pre-built, static dataset	No	No
BiblioViz [9] 2006	Table and network	Stand-alone	Pre-built, static dataset	No	No
CiteWiz [10] 2007	Author, citation, and metadata views. Concept map for keywords	Stand-alone for Windows	Pre-built, static dataset in XML-based format	No	No
VOSviewer [11] 2007-today	Label, density, cluster density and scatter views	Java stand-alone	Pre-built, static dataset + limited API access	Partially	Active
PaperCube [12] 2009-2010	Views based on graphic, hierarchy, and timeline structures	Web application	Pre-built, static dataset	No	No
Cybis [13] 2011	3D cylinder view	Java Web application	Pre-built, static dataset	No	No
Citeology [14] 2012	Generalized fisheye view	Java applet	Pre-built, static dataset from the ACM DL	No	No
PivotPaths [15] 2012	Interactive pathways	Web application demo	Pre-built, static dataset from MS Academic Search	No	No
CitNetExplorer [16] 2014	Citation networks	Java stand-alone	Pre-built, static dataset generated by WOS	No	No
VisualBib 2017-today	Narrative views	Web application	By API from multiple sources	Yes	Active

Table 1: Comparing some technical aspects of 11 visual tools.

data (using pre-built datasets or querying bibliographic indexes in real-time) and eventually integrating them from multiple sources, and the status of the development. The last row of Table 1 is dedicated to VisualBib.

Some general considerations are possible: the majority of them emerged some years ago, and is no longer under active *development*: the only two active projects are CiteSpace [7] and VOSviewer [11], two freely available domain visualization tools for analyzing emerging trends and changes in scientific literature. None of them is a *real-time* application, in the sense that they work on pre-built datasets, or allow the user to upload limited datasets obtained from WOS or other repositories; only VOSviewer can download data through API (i.e., CrossrefAPI, Europe PMCAPI, and several others). The possibility of *data integration* from multiple sources is not supported (only partially by VOSviewer); the tools accept specific input formats (for example, datasets generated by WOS, or by Scopus, or created manually). Furthermore, the majority of them have been conceived not as Web *applications* but as stand-alone applications, so requiring the download and the installation.

In experiments performed by Klein et al. [17], they observed that "switching between completely different visualizations confuse the users"; the variety of views proposed by several tools becomes a limitation for their usability; some tools, among them, BiblioViz [9] limits the possible views to only table and network 2D/3D views of bibliographic data; VisualBib uses a unique comprehensive, holistic view, while the major part of them propose several, also very different, views. PaperLens [8] tightly couples views across papers, authors and references in order to empathize the popularity of a topic, the degree of separation of authors and the most cited papers/authors; CiteWiz [10] deals with authors, citations and metadata, features three different views; PaperCube [12], an evolution of CircleView, offers a suite of alternative visualizations based on graphic, hierarchy, and timeline structures. Cybis [13] uses a visive metaphor representing both papers and terms in a cylinder located in the 3D euclidean space. The genealogy of citation patterns, Citeology [14] connects the titles of papers organized in a chronological layout, using a generalized fisheye view; PivotPaths [15] uses a graph representation of authors, publications, and keywords, all integrated in an attractive interface with smooth animations; the demo available online works on a limited dataset of papers in the fields of HCI, information retrieval, and visualization (up to 2012). CitNetExplorer [16] allows visualization of citation networks, offering expansion and reduction operations and clustering of publications. Since most relevant features of VisualBib are the opportunity for the users to manage bibliographies, creating, updating, and editing them, exploring cited/citing papers, exporting and sharing them, we consider these aspects, compared to the ten application of Table 1. We noted that all these tools do not allow users to dynamically choose the list of papers to insert in the bibliography; they use static datasets. The editing of the bibliography, for example, identifying and merging duplicate entries for a same author is only possible editing the dataset; the list of cited/citing papers is often browsable, but only limited modalities exist to choose each of them and extend the set of papers in the current view; only three tools [7, 11, 16] enable users to export and only one [16] to share their bibliographies.

Comparison with ten bibliographic indexes. We consider the presence of graphical views in existing bibliographic indexes, which we will analyze in details, on a different perspective, in Section 4. Some of them begin to propose some visual representation, mainly relatively to some metrics (such as h-index, for example). Unfortunately, they do not offer comprehensive and general views on a bibliography, but visualize single metrics; neither the opportunity to compare two or more authors at run-time: AMiner [18] is (together WOS, see below) the most graph rich: it shows some author statistics in a radar diagram, the research interests in a river diagram, all the co-authors in a star graph, and the scholar's trajectory in a map. Google Scholar [19] shows, chosen an author, the histogram of citations over years; Microsoft Academic [20] uses a similar histogram, adding also the number of papers; Scopus visualizes for the authors their h-index graph, a histogram for citations, and various pie/line charts related to their production (documents by sources, by type, by year, by subject). Scopus [21], WOS [22] and Google Scholar are the only which allow users to create lists of selected papers: Scopus offers for this specific bibliography pie/line charts, while WOS proposes for 16 different components (like WOS Categories, Publication Years, etc.) 16 different tree (or alternatively bar) diagrams, and a set of separated histograms to describe so-called citation report. CiteSeerX [23], CrossRef [24], DBLP [25], OpenAire [26], OpenCitations [27], and Orcid [28] do not present visual representations of (meta)data.

In summary: considering the specific tools, the active projects are few; none of them uses "live" repositories; few tools enable users to dynamically update, edit, export and share their bibliographies; the interfaces are not always usable and the system are not Web applications; considering the existing bibliographic indexes, they not offer interesting visual support for a researcher, interested in the creation, management and sharing of a bibliography.

3. Basic functionalities and some screenshots of VisualBib

VisualBib is an online application, freely available for research and teaching, not for commercial purposes. The first prototype has been released in September 30, 2017; the version 1.0 in February 15, 2018 and version 2.0 in September 1, 2018. In the version 2.2[2], described in this paper, VisualBib retrieves data in real-time from the Scopus, OpenCitations and CrossRef/Orcid repositories. For querying Scopus, being a commercial service, it is

[2]The detailed phases of the app development are available online `http://visualbib.uniud.it/en/development/`

necessary to navigate in VisualBib from a subscriber's domain in order to get the required data from the Scopus API.

The data providers available in the current version were chosen after evaluating the eligibility of the metadata provided by various data sources (as illustrated in next Section 4).

A user can create or enrich a bibliography in various ways:

1. searching for a paper, by its DOI (Digital Object Identifier), or, in Scopus, also by Scopus id;

2. searching for an author by nominative, or ORCiD (Open Researcher and Contributor ID), or, by the identifiers applied by the chosen index, in order to retrieve the list of his/her papers and then by selecting what to import;

3. uploading a set of references in .bib format;

4. starting from a paper in the bibliography, retrieve and explore its cited/citing papers.

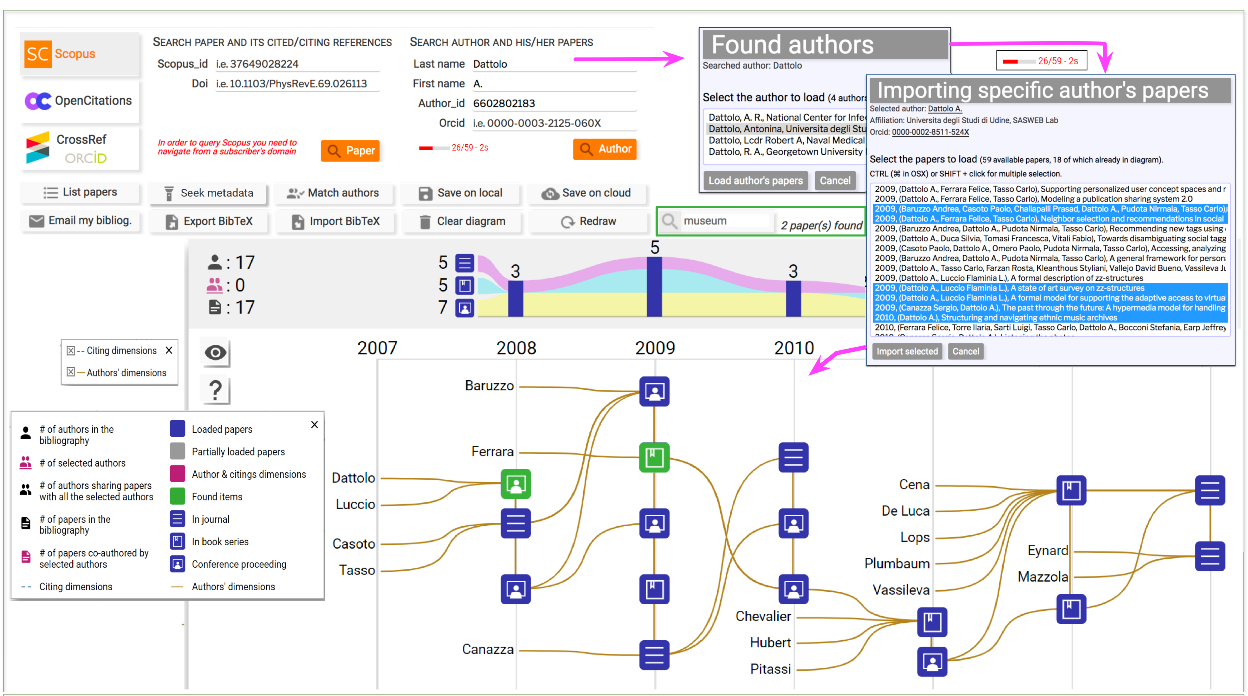

Figure 1: The main interface of VisualBib.

In the first two cases, user starts choosing one of the 3 available indexes (in Figure 1, top-left, the selected index is Scopus), and fills the appropriate fields of the form; in the third case, the user selects the button `Import BibTeX` and upload a file in BibTeX format; in the last case, the user explores the cited/citing papers, following the procedure discussed below and illustrated in the Figure 2. The result will be the visualization of the bibliography in

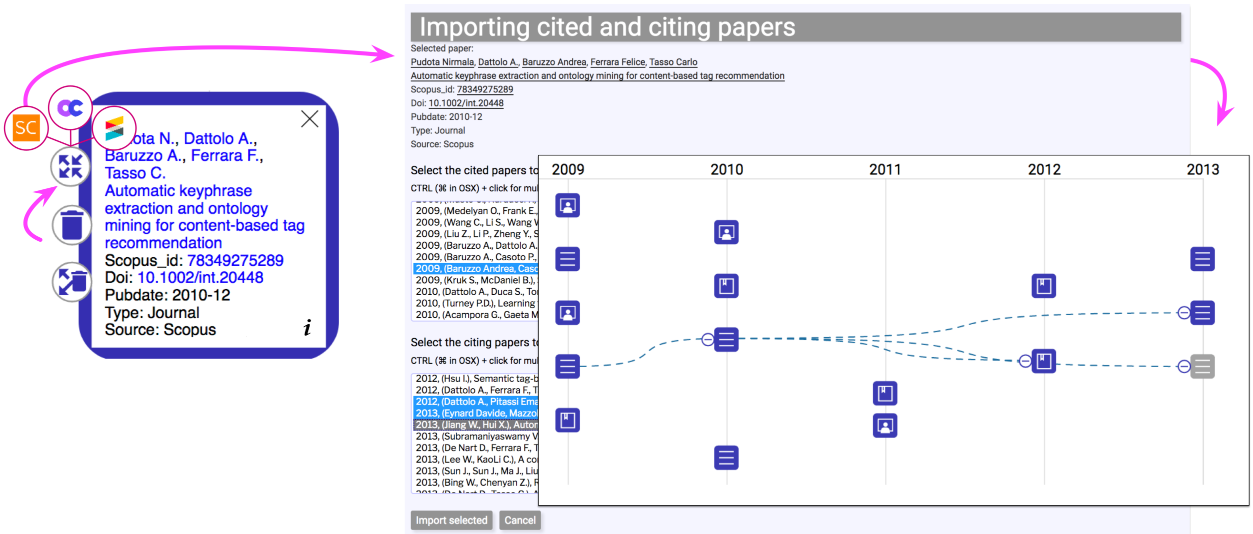

Figure 2: Adding cited/citing relationships.

a narrative view. Figure 1, bottom, displays the narrative view generated by searching the name of a given author ("Dattolo A."): in order to disambiguate between homonyms, the list of 4 found authors is enriched with their name, and, if present, ORCiD, affiliation, subject areas, and OpenCitations's id. Once the user selects the author, the system will fetch the list of the publications, showing a progress bar and an estimation of the residual loading time. In the case of long lists, a stop button is displayed to allow user to interrupt the loading process and examine the partial set of retrieved data. A form containing the ordered list of found papers is then presented to the user who can choose the subset of the publications to import in the narrative diagram, shown at the bottom of Figure 1.

The narrative view (see next Section 6) is a 2-dimensional space: the horizontal dimension is the time, discretized by years; the vertical dimension is spatial and is used to properly organize authors' names, papers and their relationships. The diagram includes the last names of the authors involved in at least one paper of the current set: each author is associated with a goldenrod line that connects all his/her papers, from the oldest to the newest, giving an indication of the author's professional path (clearly limited to the set of imported publications) over the years. The papers are represented by colored, round-cornered square items (in blue and green in Figure 1). Following the second column of the legend of Figure 1, on the bottom-left:

- the three different icons associated to publications distinguish between *journal papers*; *books* or *book chapters*; *conference* or *workshop proceedings*. If the type is different or unknown, an empty icon is associated to the item;

- the color of the icons indicates the papers state: *blue* is associated to a completely loaded paper (all the needed data and metadata have been loaded); *gray* indicates a partially loaded paper, which has been retrieved during a cited/citing search (this operation returns only a subset of papers metadata); *magenta* is used to emphasize semantic relationships during user interaction, as described later; and, finally, *green* marks the papers found by means of a textual search (in Figure 1, they are the papers found looking for "museum" - 2 papers found).

3.1. Visual analytics and information discovery

VisualBib enables users to accomplish some visual analysis tasks on the current bibliography by interacting with the narrative view diagram. In the following we illustrate how our tool supports the extraction of bibliographic data and the exploration of the relationships and the citations network.

Distribution of papers and publication types over time. The histogram above the timeline in Figure 1 shows the distribution of the papers by year of publication, while the overlying area chart, the frequencies of the three publication types: moving the mouse over the colored areas or on the three (publication types) icons, the histogram changes for showing the corresponding counters.

Paper metadata. Clicking over a paper icon, a pop-up window (see Figure 2, top-left) shows its bibliographic data, where the authors names, the title, the id and the DOI of the paper are links towards dedicated Web pages. An extended description is accessible by clicking on the *"i"* icon: the amount of metadata presented depends on the source index and can be integrated, querying the Scopus API through the *"Seek metadata"* feature, discussed later.

Citation network. A click on the four-arrows icon (see Figure 2, top-left) and the next choice of the index where to search citations (in our example Scopus), loads, in a separate form, the list of cited/citing papers (Figure 2, center) found on the selected data source. Within each selection list, containing respectively all the cited/citing papers, each item already in the diagram, appears highlighted in blue. Users may select, from the two lists, the documents of their interest and import them (with the relative relations) in the diagram. In the example of Figure 2, the user chooses to import the three pre-selected papers plus a new one and, as shown on the right of the same Figure 2 the cited/citing relations become visible as blue dashed lines.

The production of an author and some collaboration metrics. By clicking over an author label (Figure 3, left), the application emphasizes in magenta the path connecting all his/her papers in the current bibliography, from the oldest to the newest. The number in brackets,

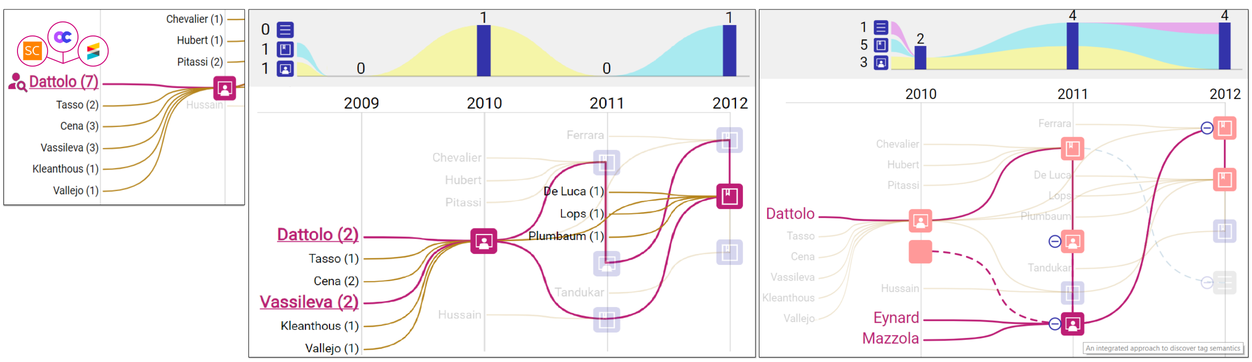

Figure 3: Visual analytics of collaboration networks and views on the papers related to a given selection.

appended to the author's labels, indicates, for the selected author, the number of papers in the bibliography and, for all the other authors, the number of publications in common with the first selected one. The authors without collaborations with him/her are temporarily obscured by applying partial transparency to the corresponding labels, as shown in Figure 3, left, for the author placed under the paper icon. Furthermore, when we select an author, a search icon appears on the left of the name making possible to search the papers available on a specific index (the three icons which appear on the name Dattolo in Figure 3, left).

Co-authored papers and multiple collaborations. When users select more than one author, by clicking on each name (Figure 3, center) all the papers in common between the selected authors are emphasized in magenta and the number of papers written in collaboration with

the current set of the selected authors is reported near each name. Also in this case the authors without collaborations with the selected ones are temporarily obscured. When an author is selected, or deselected by clicking again on the label, all collaboration counters, the histogram and the area chart are updated to reflect the current set of co-authored papers.
Related papers. Finally, by moving the cursor over a paper icon, the narrative view highlights the network of cited/citing papers and the paths relative to the co-authors and their papers (related papers). The focus paper is emphasized in magenta while a salmon color is applied to related papers (Figure 3, right).

3.2. Other features

In addition to visual analysis of bibliographies, VisualBib offers a dynamic environment to progressively expand and refine a them through some features we illustrate in the following.

List papers. To generate the list of the references of all the loaded papers, in textual format.
Seek metadata. To check for the availability of additional metadata for each paper in the bibliography imported from Scopus comprising the list of assigned keywords, the abstract, the citation count, the open access flag, the publication name, the publisher and others.
Match authors. Since VisualBib retrieves papers and authors metadata from multiple providers, the automatic matching of the same author is only applied if a univocal ORCiD code is available.
The `Match authors` feature offers users a wizard in order to merge redundant data.
Save on local. It performs the saving of the current bibliography into the local storage of the browser. At the next load of the application page, if a bibliography is available in the local storage, the system will ask the user whether to reload or ignore it.
Save on cloud. It offers users the possibility to save the bibliography in a remote database. A saved bibliography can be accessed from anywhere if provided by a specific link: the user can share the bibliography in read-only (or in read-write) mode with students or colleagues. VisualBib generates also an embed code which allows users to embody a bibliography, in the narrative view format, into an external web page.
Email my bibliographies. Entering an email address, the list of the owned bibliographies, including all the read-write and read-only links will be sent to the user. In the message user

will also find the links to permanently delete any of the listed bibliographies.

Export/Import BibTeX. It allows users to automatically export (resp. import) a bibliography in the .bib format. In the BibTeX file produced by export, some not standard fields are added to encode the authors and the citation information, necessary to recover, during a next import, all the relations between items.

4. Data analysis

In previous Section 2, we analyzed the lacking of graphical components in ten, well-known, bibliographic indexes; since they index millions of papers, authors and related metadata, they represent for us the real-time datasets from which potentially extract the metadata necessary for using our tool. In the Table 2, applying the analytic task taxonomy in information visualization proposed in [29], we list, in the first column, the low-level data analysis tasks, supported in VisualBib; in the second, the corresponding user visual interactions and, in the third, the related items in the taxonomy. Then, for each user task, we indicate if it is supported in ten of the major bibliographic indexes. We note that, excluding Scopus and WOS, the other indexes support, in a sparse way or partially, these user tasks. At this level of analysis, we considered the functionalities usable directly by the Web platform of these indexes, without considering that some metadata or tasks could be performed combining metadata, provided, for example, by API; a limiting case is OpenCitations, that only manages simple search operations, but provides an interface for executing SPARQL queries.

In order to investigate the feasibility of integrating in VisualBib various bibliographic indexes as real-time data sources, we individuate the following list of specifications:

Paper's metadata by DOI - starting from the DOI of a publication, VisualBib needs to retrieve metadata such as the title, the publication year, the list of co-authors (possibly complete of unique identifiers), the type of publication (i.e., conference paper, journal paper, book or book chapter, etc.) and, possibly, other metadata such as keywords, external links, abstract, metrics;

Citations by DOI - starting from the DOI of a publication, VisualBib would like to retrieve the lists of cited and citing papers, possibly complete with the DOIs and the lists of uniquely identified authors;

VisualBib' user tasks	Visual interactions	Taxonomy	AMiner	CiteSeerX	CrossRef	DBLP	Google Scholar	Microsoft Academic	OpenAire	Open Citations	Scopus	WOS
Find papers metadata(i.e. title, authors, DOI, etc.)	- mouse over papers - click on papers - list papers	Retrieve value	●	●	●	●	●	●	●	●	●	●
Search / import papers from data sources	- search for paper - search for author - search for citations & references	Filter	◐(a)	●	◐(a)	◐(a)	●	●	◐(a)	◐	●	●
Find papers of an author	- author selection	Filter	●	●	●	●	●	●	●	●	●	●
Get co-authored papers	- multiple authors selection	Filter, Cluster	○	○	○	●	○	○	○	○	●	●
View the collaboration network	- author selection - multiple authors selection	Filter, Cluster	●	○	○	●	○	●	○	○	●	●
Find papers by keywords	- local search	Filter, Cluster	●	●	●	●	●	●	●	◐	●	●
Hide/show references of a paper	- click on -/+ on the paper icon	Filter	●	●	○	○	○	●	○	○	●	●
Get summary data (b)	- author selection - multiple authors selection - data summary - list papers	Compute derived value	●	●	○	◐	●	●	◐	○	●	●
Get # of papers per year	- examine histogram	Compute derived value	●	●	●	●	●	●	◐	○	●	●
Order papers by year / get most productive year	- examine histogram - examine timeline - author selection	Sort, Find extremum	●	◐	●	◐	●	◐	◐	◐	●	●
View document types distribution	- examine histogram - interact with area chart	Characterize distribution	○	○	○	○	○	○	○	○	●	●
Determine timespan of a bibliography	- examine timeline	Determine range	●	◐	◐	●	◐	◐	◐	◐	●	●
Find duplicate authors	- match authors	Find anomalies	○	○	○	○	○	○	○	○	○	○
Get related papers	- mouse over a paper icon	Cluster	●	◐	○	○	○	●	●	○	●	●

Legend: ● full provided; ◐ partially provided; ○ not provided

(a) Citations or references not provided (b) Number of papers, authors, co-authored papers, collaborations, citations, etc.

Table 2: List of the low-level data analysis tasks supported by VisualBib compared with those provided by a set of well-known bibliographic indexes.

Authors by name - starting from the name/surname of an author, VisualBib needs to retrieve the list of corresponding authors, enriched with their *ORCiD (Open Researcher and Contributor ID)* and/or internal unique identifier;

Author's papers by ORCiD - starting from an ORCiD or another author unique identifier, VisualBib needs to retrieve author's metadata related to the current affiliation, subject areas, list of the publications, complete of DOIs and lists of authors, possibly uniquely identified.

Table 3 summarizes, for each platform, the availability of specific data retrieval API functions with reference to the above specifications. The evaluations have been formulated analyzing the online available documentation, in some cases rather incomplete, and, where possible, directly testing the services.

Platforms	Paper Metadata (by DOI)				Citations (by DOI)		Authors (by name)		Papers (by ORCiD)		
	title	year	auth.	auth. ids	cited	citing	ORCiD - local ID	aff.	aff. - area	list of papers	list of authors
AMiner	?	?	?	?	?	?	?	?	?	?	?
CiteSeerX	○	○	○	○	○	○	○	○	○	○	○
CrossRef	●	●	●	◐	◐	○	○	○	○	●	◐
Orcid	○	○	○	○	○	○	●	●	●	●	○
DBLP	○	○	○	○	○	○	◐	○	○	○	○
Google Scholar	○	○	○	○	○	○	○	○	○	○	○
MS Academic	●	●	●	●	●	?	●	●	◐	◐	◐
OpenAire	●	●	●	○	◐	○	◐	○	○	○	○
OpenCitations	●	●	●	●	●	●	◐	●	●	●	●
Scopus	●	●	●	●	●	●	●	●	●	●	●
WOS	●	●	●	●	●	●	●	●	●	●	●
WOS Lite	●	●	●	●	○	○	●	?	?	?	?

Legend: ● full provided; ◐ partially provided; ○ not provided; ? not well documented, to be verified.

Table 3: Data provided through API services.

Although AMiner provides comprehensive search and mining services for researcher social networks, and an API service exists [30], at the moment the documentation is not adequate to use it. Furthermore, neither CiteSeerX, neither Google Scholar provide an API service for supporting third-party applications, while CrossRef and Orcid appear be almost complementary: in fact, the API Rest services [31] of CrossRef retrieve the metadata of a publication given its DOI and, if available, get the references to the cited papers. It is also possible to retrieve the list of publications of an author given the ORCiD but not search by name/surname. On the other hand, Orcid provides search service of authors that have registered an ORCiD code: in this case it is possible to retrieve their metadata and a list (possibly incomplete) of their publications in the form of collections of DOIs. Orcid APIs do not provide any service to retrieve metadata about papers.

DBLP offers an API service for the research of authors and publications. Unfortunately the service is based on textual queries and the DOI or ORCiD identifiers cannot be used as search keys although these information are often provided in the responses. This fact, together with the lack of data concerning the cited/citing documents, makes its integration in VisualBib difficult.

According to the documentation, the integration with VisualBib of Microsoft Academic could be feasible through the *Evaluate* method of *Paper* and *Author* entities although some significant fields, like ORCiD and citing references, appear to be absent. We plan to verify the feasibility of an effective interfacing with the Microsoft Academic APIs in the future work.

The current API version of OpenAire offers the retrieval of publications metadata but provides limited features for author search and disambiguation. However, a forthcoming API release has been announced with improvements; we will analyze the new services in the future work.

OpenCitations provides a REST API service to query the internal corpus. For each data retrieval task, VisualBib prepares and submits a list of specific SPARQL queries to the single OCC API endpoint and extracts the needed metadata from the JSON responses. Scopus offers a rich set of API [32] to retrieve authors and papers metadata, including references and citations. The API services of Scopus require that the HTTP API calls must originate from an IP address inside the domain of a subscriber organization, in order to be processed. For subscribers, the data provided by Scopus API are complete, returning rich metadata about papers, authors, references and citing documents.

Also WOS [22] provides a rich set of task based APIs to query more than 70 million records. The recently published WOS API Expanded is a commercial service that could provide all the VisualBib's needed data. A problem arises from the mechanism used to authorize applications to access the indexes, based on an API key. In absence of other protection mechanisms, the integration of the WOS API in VisualBib, would result in a exposure of WOS's data to not subscribers users, in contrast to their data policy. For this reason, now we are exploring possible alternative technical solutions that are compatible with the terms of use of the service. Unfortunately, the service offered by WOS API Lite returns only a restricted subset of WOS metadata: for example, all cited/citing references are not provided.

Based on this analysis, we decided to implement at first the procedures to query and retrieve data from Scopus and OpenCitations as these fully meet all the specifications; then, observing the complementarity of the data provided by CrossRef and Orcid, we decided to use them in combination, performing authors searches on Orcid and papers metadata retrievals from CrossRef.

5. Introducing zz-structures and modelling VisualBib

We model VisualBib using contextual data structures, called zz-structures.

5.1. Zz-structures

Zz-structures were first proposed by Ted Nelson [33, 34] and then revisited in successive works [35, 36]: they introduce an intrinsically non-hierarchical, graph-centric system of conventions for data and computing. As proved in [37], zz-structures are general data structures, since subsume lists, 2D arrays, trees, polyarchies, and all edge-coloured directed multigraphs. Furthermore, they enable to associate semantic interconnections between vertices (for example, papers and authors); manage, in a holistic view, different contextual dimensions which can be simply highlighted on demand; focusing the attention on a specific item, offer local comprehensive view; and, finally, represent a general conceptual model to, formally and informally, describe our knowledge domain.

A zz-structure can be thought as a space filled with cells, called *zz-cells*, connected into linear sequences. Cells are connected together with links of the same color into linear sequences called *dimensions*. A single series of cells connected in the same dimension is called *rank*: a rank is in a particular dimension and a dimension may contain many different ranks. The starting and the ending cells of a rank are called *headcell* and *tailcell*, respectively, and the versus from the starting (ending) to the ending (starting) cell is called *posward* (*negward*). A simple example of zz-structure is proposed in Figure 4, left. Normal, dotted and thick edges

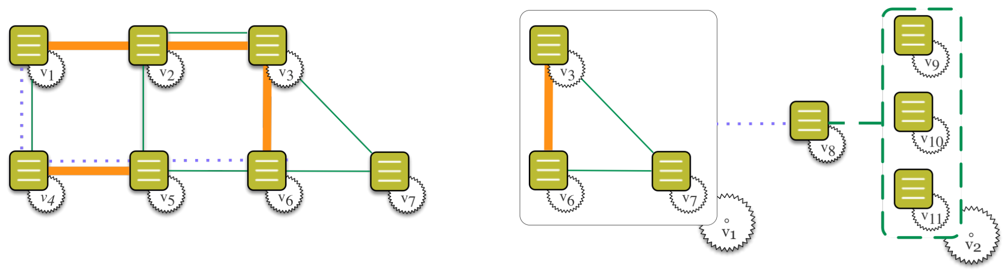

Figure 4: An example of zz-structure (left) - Two compound cells are connected with v_8 (right).

represent three different spatial dimensions [37], and describe different semantic ties [35]. In our example, *d.thick* is composed of two ranks $\{v_1, v_2, v_3, v_6\}$ and $\{v_4, v_5\}$, and one isolated vertex v_7; *d.normal* is composed of two ranks and no isolated vertex: the first rank is a path $\{v_1, v_4\}$, the second one is a ringrank $\{v_2, v_3, v_5, v_6, v_7\}$; finally, *d.dotted* contains a rank $\{v_1, v_4, v_5, v_6\}$ and three isolated zz-cells v_2, v_3, v_7.

Special zz-cells are *the maincells*, headcells of tailcells, which stand for the whole (a maincell is expected to be connected directly to its supporting cells), the *composite* cells, which contain more than one type of data; and *compound* cells, which contain cells or, in general, zz-structures. In this paper, a **compound** cell is addressed by the following notation: a ring over the letter $\mathring{v}$ denotes a compound cell, and $\mathring{V}$ a set of compound cells. In Figure 4, right, is shown an example containing two typologies of compound cells $\mathring{v}_1$ and $\mathring{v}_2$: the first is composed by a zz-structure and is linked to the cell v_8 by the dimension *d.dotted*; the second groups a set of cells semantically joint from the dimension *d.dashed*, the same that connects v_8 to it.

5.2. Modelling a visual bibliography using zz-structures

A bibliography can be thought of as a network of authors and papers, interconnected by citing/cited dimensions and containing, for each paper, details for associating to it *author-*

ship, *title*, *DOI*, *editorial collocation*, *repositories* on which it is indexed, etc.. Preliminary formal definitions are present in [35].

Definition 1. Visual Bibliography - A visual bibliography *VB* is as a zz-structure, where

- $V = \{A, P, DE, \mathring{P}C, \mathring{C}P\}$, the finite set of vertices, is composed by:
  - a finite set of authors A, papers P, and details DE, associated to the papers. DE is constituted by composite cells;
  - $\mathring{\text{P}}$C and $\mathring{\text{C}}$P are compound cells, containing, for each paper:
    - the set of *PC* - *Papers Cited* by it;
    - the set of its *CP* - its *Citing Papers*.
- $D = \{d.a_1, \ldots, d.a_n, d.time, d.cited, d.citing, d.details, d.author\text{-}l, d.title\text{-}l, d.ID\text{-}l, d.doi\text{-}l\}$, where:
  - $d.a_1, \ldots, d.a_n$ identify the dimensions, which group the publications of the authors $a_1, \ldots a_n$; each of these last is the maincell of each dimensions;
  - *d.time* is constituted by the parallel ranks $d.t_1$, ..., $d.t_m$, which group the papers of the bibliography published during each year $t_1, \ldots, t_m$; these time marks are the maincells of each rank;
  - *d.cited* and *d.citing* respectively connect each paper to the sets of the papers cited by it and that cite it; these two dimensions are constituted by parallel ranks, one for each paper;
  - *d.details* is constituted by parallel ranks. Each paper is associated to a composite cell, containing specific details, which link to external information:
    - *d.author-l*: this dimension is constituted by parallel ranks: each author is connected to a related Web page on the repository which the paper has been retrieved from (Scopus, OpenCitations or CrossRef/ORCiD);
    - *d.title-l* and *d.ID-l* link to the paper's Web page on the repository, respectively using as access key the title or the ID of the paper in the specific repository;

∗ *d.doi-l* links to the DOI webpage of the paper.

Futhermore, *VB* becomes operational thanks the two sets of views and mechanisms, introduced in next Section 6.

Figure 5 proposes an example of bibliography, represented in terms of a zz-structure. The

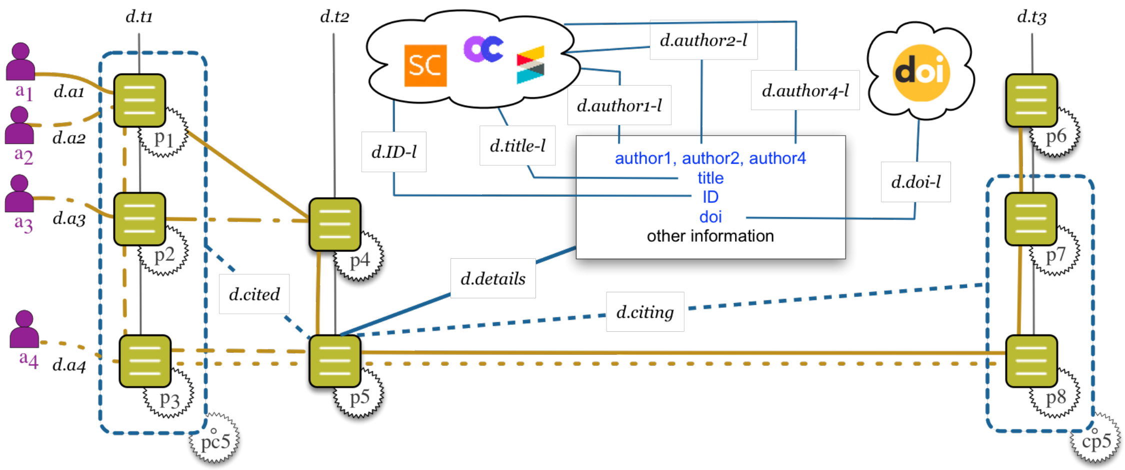

Figure 5: An example of bibliography represented using a zz-structure.

zz-cells of the bibliography are 4 authors $\{a_1, \ldots, a_4\}$, 8 papers $\{p_1, \ldots, p_8\}$, and 2 compound cells $\{\mathring{p}c_5,\ \mathring{c}p_5\}$, composed by subsets of papers, respectively cited by/citing the paper p_5. Following Figure 5, we analyze the involved dimensions:

- $a_1, \ldots a_4$ represent the maincells of the four dimensions $d.a_1, \ldots d.a_4$; each of them groups the publications of the related author. For example, the dimension of the author a_1 is composed by $(a_1, p_1, p_4, p_5, p_8, p_7, p_6)$;

- $d.t_1, d.t_2, d.t_3$ represent the three parallel ranks related to the papers published in the years t_1, t_2, and t_3;

- *d.cited* connects the paper p_5 to the compound cell $\mathring{p}c_5$, containing the list of its references (p_1, p_2, p_3), while *d.citing* connects p_5 to $\mathring{c}p_5$, containing the papers (p_7, p_8), which cite p_5;

- *d.details*: p_5 is associated to a composite cell, containing its details, which link to external information:

– $d.author_1$-l, $d.author_2$-l, and $d.author_4$-l connect the 3 authors a_1, a_2, and a_4 to their Web pages on the repository which the papers have been retrieved from;

– $d.title$-l and $d.ID$-l link to the paper's Web page on the repository;

– $d.doi$-l links to the DOI Web page of the paper.

Besides these main dimensions, the model owns a wider potential, provided by the possibility to identify new ways to semantically connect and visualize relations among authors, papers, and relative metadata. For instance, papers might be related using "subject areas", "keywords", "users' tags" dimensions, generating customized narrative views based on personal labelling of papers.

6. Zz-views in VisualBib

Zz-structures generate a pseudo-space that is somewhat comprehensible visually; there is no canonical viewing mechanism for zz-structures [34], although typical views are the two 2D cursor-centric views, called I-view (or row view) and the H-view (or column view) [34]. These two views make use of 2 spatial dimensions at a time, and locally flatten a subset of the neighbourhood around a cursor (i.e. a selected node). More specifically they visualize the subset of nodes that are connected to the cursor's node via edges along the two chosen dimensions. This subset of nodes is embedded in a (non-Euclidean) 2D manifold, which is then displayed in a flat, 2D view [37]. The concepts and the formal definitions of these views have been introduced in [37, 38], and extended in [39] into n-dimensions H-views and I-views. Many more views are possible. In [40], there are formally introduced other two views, the star view and the m-extended star view.

Below we formally introduce two new, and general, views, conceived and formalized for VisualBib; we remind interested readers to [3, 41] for the formal description of the topological constraints, associated to these two views.

Definition 2. Deep view - Given a zz-structure S, if $\exists$

1. a compound cell $\mathring{v}$, constituted by a set of cells $\{v_1, \ldots, v_m\}$, joint by the unique common color c_k;

2. a cell $v \in V$ and edge of the same color c_k, which links $(\mathring{v}, v)$ (or indifferently, $(v, \mathring{v})$);

then the *deep view* of *focus* v and dimension $d.k$ displays a graph, where:

- $V = \{v, v_1, \ldots, v_m\}$;
- $E = \{(v_j, v) | j = 1, \ldots, m\}$ (or resp. $E = \{(v, v_j) | j = 1, \ldots, m\}$).

In the Definition 2, we use the same letter for the cell v and the compound cell $\mathring{v}$, to indicate that the cells contained in the compound cell $\mathring{v}$ and the v are semantically connected by any same color/dimension.

An example of deep view is shown in Figure 6: on the left we consider an extract of the

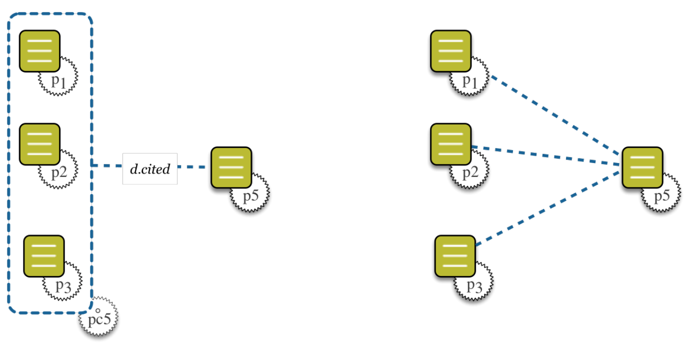

Figure 6: p_5 is connected with $\mathring{p}c_5$ by *d.cited* (left); the deep view explodes the connections (rigth).

zz-structure of the Figure 5, where $\mathring{p}c_5$ and p_5 are connected by the dimension *d.cited*. On the right, is displayed the deep view, where the citation link is exploded in three links.

Definition 3. Narrative view - Given a zz-structure S, which contains:

- a set $V = P \cup A$ of items $p_1, p_2, \ldots, p_{|P|}$, $|P| \neq 0$, each of them attributable to specific agent(s) $a_1, a_2, \ldots, a_{|A|}$, $|A| \neq 0$, and associated to a specific time mark $t_1, t_2, \ldots, t_{|T|}$, $|T| \neq 0$;
- a time dimension *d.time*, containing parallel ranks $d.time = \bigcup_{i=1}^{m} R_i^{t_i}$, one for each time mark;

a complete *narrative view* displays:

1. a horizontal timeline $t_1, t_2, \ldots, t_m$, drawn left to right; under each $t_i, i = 1, 2, \ldots, m$, in vertical, is positioned the parallel rank $R_i^{t_i}$, belonging to the time dimension $d.time$, and containing all the items $p \in P$, marked by t_i;
2. $\forall p \in P$ the dimensions $d.a_1, d.a_2, \ldots, d.a_n$ as linear paths that links any agent to all its items $d.a_i, p$. The linear paths are time ordered;
3. $\forall$ ($\mathring{p}$, p) (or $\forall$(p, $\mathring{p}$)), that respects the constraints 1 and 2 of Definition 2, the relative deep views.

On a narrative view is possible to apply different filters in order to visualize part of the view, excluding from it for example the author dimensions, or one or more deep views.

7. Architecture and Implementation

VisualBib is organized as a single page Web application, based on HTML5, CSS3 and SVG (Scalar Vector Graphics) W3C standard languages and Javascript ES6; it makes use of AJAX techniques to perform HTTP/CORS [42] calls to data providers in order to retrieve needed papers and authors metadata. Although the most of the VisualBib Web application runs on the user's browser, a server-side is provided to offer some cloud services as the saving and retrieving of bibliographies, their indexing and sharing besides the tracing of the errors/exceptions in the application. Figure 7 shows the architecture of VisualBib and its main modules, which we describe in next Subsections 7.1-7.8.

7.1. Data providers

In the following we describe specific features and issues related to each provider.

Scopus. Scopus marks all publications with a unique record id, called Scopus id that is a numeric code which is part of the EID (Electronic id). Authors are marked with a numeric code called author id. Where available Scopus provides (or can be queried by) universal identifiers for papers (DOI) and authors (ORCiD).

VisualBib interfaces with Scopus through a series of API calls [32] to get needed data: *Abstract Retrieval API* to retrieve detailed metadata of a paper given its Scopus id or its

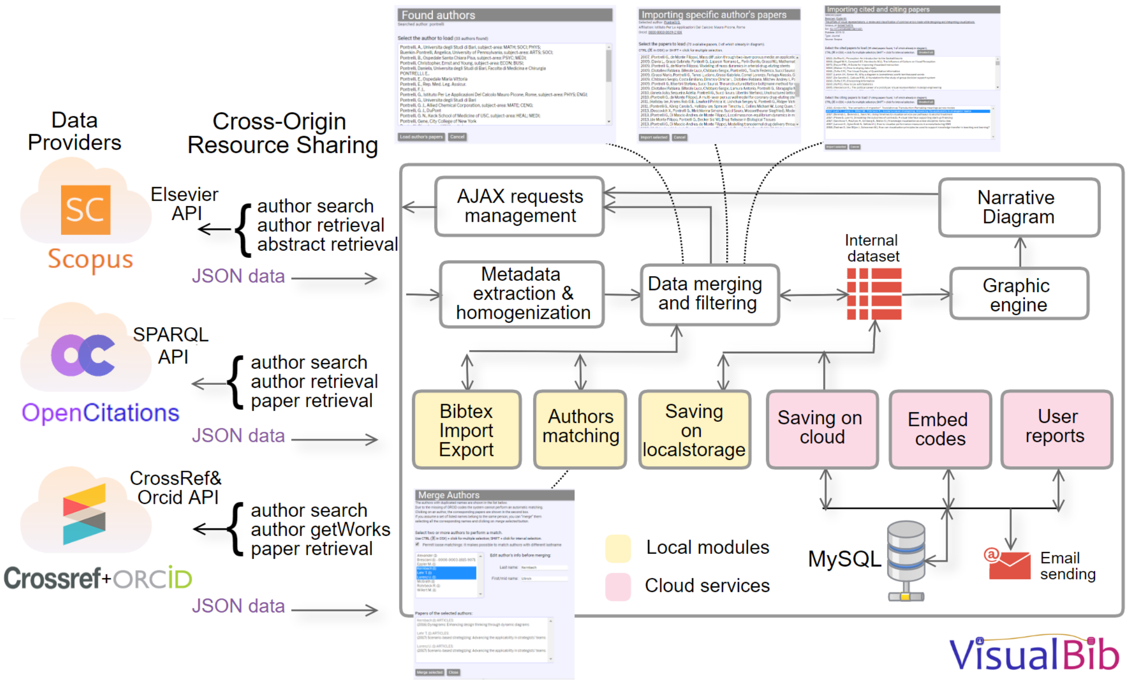

Figure 7: The architecture of the VisualBib application.

DOI; *Scopus Search API* to retrieve the references to cited and citing papers of a specific publication; *Author Retrieval API* to get detailed metadata of an author and a list of his/her publications; *Author Search API* to search authors given their name and surname.

Being the API calls generated on the client side of the application, Scopus only accepts API requests originating from domains of subscribers, according to the data policy. Furthermore, to avoid misuse of data, the platform introduces some limitations to each API endpoint by fixing quotas for the number of queries per week and per second.

OpenCitations. VisualBib implements the retrieve operations from the OpenCitations Corpus [27] submitting queries to a single API endpoint which accepts SPARQL queries. VisualBib uses 8 different query types to get all the needed metadata about papers, citations, authors searched by name or ORCiD and their lists of publications. A known issue in the OCC indexes is the presence of multiple identifiers for a same author: in this case it is necessary to consider all the candidates, examine and import related publications and finally join the duplicate author entries by applying the "Match author" procedure.

CrossRef and Orcid. Observing the complementarity of the APIs provided by CrossRef and Orcid APIs, we combined their services to implement a third data source. In particular VisualBib performs the search of authors by name using Orcid API [43] and then, after the disambiguation of the results, queries both Orcid and CrossRef APIs to retrieve the list of the papers' DOIs of the selected author. The results from the two sources are merged and their metadata are retrieved, one by one, from CrossRef, querying it by DOI. Due to this, the retrieval of the publications of an author from CrossRef/Orcid can result slow because it requires a http request/response for each paper: a stop button is available to interrupt the current operation and obtain a partial list of results. About the search of cited/citing papers of a selected one, currently CrossRef can only provide, if available, the list of its cited papers and not those citing it. Consequently the corresponding list in the selection form, see Figure 2, will always result empty.

7.2. AJAX requests management

The interaction with external sources to get real-time data implies the execution of cross-site HTTP calls that must be managed through CORS [42] specific headers to overcome the browser security restrictions. VisualBib must receive the *Access-Control-Allow-Origin* header in order to access the data in each response. Being OpenCitations, CrossRef and Orcid APIs open services, the header above is automatically generated by servers without the need of authentication. Scopus APIs need to receive a previously registered API key in each request in order to authorize the specific domain, declared through the *Origin* header that must match the registered one. This module is also responsible for the preparation of the AJAX requests for the various data provider, each characterized by specific endpoints, parameters and request formats and for the retrieval of long lists of results, generally returned in small chunks, through a series of repeated requests to be triggered recursively.

7.3. Metadata extraction and homogenization

This module provides the necessary data extraction from the JSON flow of data coming from the various data providers. The loading of long data streams is accomplished by multiple recursive requests managed by the module described above. In order to avoid simultaneous API calls and achieve a clearer status reporting to users, this module blocks new search requests until the current data receiving and extraction is complete. Furthermore this module is responsible of the homogenization of the data from different sources, for example by decoding the identifiers of the typologies.

7.4. Data merging and filtering

During the loading of new authors and papers, users are asked to filter the retrieved data by means of selection lists in order to pick the significant items from the results.

This module performs the integration of new data through a series of checks with the current dataset in order to correctly insert and merge the new information, possibly avoiding the introduction of redundancy. It also manages the exceptions that could arise and ensures the consistency of the dataset during the importing and the deletions of papers, the merging of duplicated authors and the import from external BibTeX files.

7.5. Internal dataset

The internal dataset contains all the data needed to represent the current bibliography. The dataset is organized through a set of object oriented Javascript data structures that reflect the zz-structure described in previous Section 5. At this lower abstraction level, zz-cells are objects, organized in arrays and connected each others through references.

For papers, VisualBib manages a set of significant metadata: title (with external link), publication year, abstract, authors (with links), subject areas, Scopus or OC ids (with links), DOI (as link), issn, references list and source list. The links connect the metadata to the corresponding resource on the external index.

For authors, the dataset contains: first, middle and surname (with link), preferred name, affiliation, ORCiD, local id, subject areas and the list of the considered papers.

7.6. Graphic engine and Narrative diagram

To implement VisualBib we have chosen D3.js [44, 45], a the modular JavaScript library for creating interactive documents with a strong emphasis on the HTML, SVG, and CSS Web standards. Similarly to JQuery, D3 provides a powerful DOM selection mechanism based on declarative CSS patterns, a rich library of methods to create complex graphical representations and layouts and to act, with the same syntax, both on single DOM elements and on sets of them. We recently experimented the D3js versatility in realizing the application AppInventory [46, 47]. In order to generate the narrative diagrams, generally characterized by a limited number of elements, we adopted SVG standard which provides all the needed geometric elements to represent the cells and their interconnections.

The graphic engine maps the internal data model into the narrative view; previously informally shown in the previous Figure 1, Figure 2, and Figure 3, and formally in the Definition 3. The papers items are represented by groups of SVG elements being rendered by round-cornered squares combined with appropriate icons. Each paper has associated a series of event handlers to manage users actions and apply style properties to the current element and to the connected ones, to make visible paper details and icons to trigger further actions. Transition and animation effects have been introduced to improve the user experience through a progressive highlighting of the semantically connected items. The *authorship* and *citation* relations between papers are rendered with a SVG path element which describes *smooth cubic Bezier curves* connecting the respective icons. The paths, as shown in Figure 3, are opportunely stylized and colored and, in order to reduce the complexity of the representation, any multiple relationships (for example same authors for subsequent papers) generates overlapped paths.

7.7. Local modules

VisualBib has a simple `Save on localstorage` feature (see Figure 1): it enables users to store the current bibliography in a permanent (preserved in different sessions), erasable but nor shareable space, useful for frequent savings. Furthermore, are present the `Author matching` wizard and the `Import/Export in BibTeX format` function, introduced in Section 3.

7.8. Cloud services

VisualBib includes some server side modules, in order to store and retrieve the visual bibliographies. User diagrams are represented in JSON format and described by title, email address relative to the owner, last saving date and two unique urls: every time a new bibliography is saved clicking on the `Save on cloud` button (see Figure 1), the user is asked to specify an e-mail address; the system saves the bibliography in a MySql database and generates a couple of unique urls for future accesses and/or for sharing with other users in both read-write and read-only mode. Users may also require, clicking on the `Email my bibliog.` button (see Figure 1), to receive the complete list of their saved bibliographies.

8. User evaluation

We have carried out two different studies in order to evaluate the impact of our approach. In both studies, in addition to VisualBib, we have considered the Scopus Web platform [21] for bibliographic searches in order to evaluate some usability aspects of our visual approach compared to a traditional one. The choice of Scopus as the second platform on which to conduct the tests is motivated by the advanced search features that it makes available and the possibility to evaluate both platforms with a common dataset. The first study, described in [3, 4], was a qualitative study in which the participants were divided in two groups in order to evaluate separately some usability aspects of VisualBib and Scopus Web platforms. The second study, described in this paper, involves a larger group of participants who faced with a series of bibliographic search problems on both platforms: we also collected quantitative data about the execution times of each task as well as some feedback on their user experience. In this case, in order to collect more aware opinions and ratings, we chose to involve each participant in the evaluation of both platforms to inform them about the distinguishing features and let them experiment the two alternative approaches before evaluating each one.

It is important to clarify that the tasks performed by the participants involved only a subset of the features of the two platforms; for this reason, the usability results apply only to the considered aspects. Both studies were carried out on VisualBib 2.0 version which did not include the histogram, the area chart and the counters of the collaborations of the single authors, introduced in version 2.1.

8.1. Study aims

The main questions to try to give answer are:

- Is the VisualBib application effective to deal with some specific bibliographic searches? We made a comparison with the time employed for the same operations in the Scopus Web platform;

- Is there a significant difference in usability between VisualBib and Scopus, computed by a standard questionnaire such as SUS?
- Is the novel VisualBib interface appreciated by users and considered innovative?
- How important are considered by the users some general features and how much are they perceived present in VisualBib and in Scopus?

8.2. Study design and data analysis

The participants were recruited on a voluntary basis among undergraduate students, students of the last year of high school participating to university orientation programs, and librarians of University of Udine: altogether they were 93, aged between 18 and 56 years.

The questionnaire was organized in two main sections, the first dedicated to VisualBib and the second to Scopus: for both the platforms, the participants were asked to answer, for the quantitative evaluation, 5 questions, after having performed 5 tasks (T1, ..., T5); and, for the qualitative evaluation, to answer the 9 items of the SUS_{-01} test, 6 items on user experience (U1, ..., U4) and aspects of graphical layout (G1 and G2), and finally 7 items (F1, ..., F7) on specific features of the application. In order to get comparable data, the tasks T1 ... T5 were exactly the same in VisualBib and Scopus, except for the provided input data.

Before starting the experiment we organized a live presentation to illustrate the Web interfaces and to demonstrate some use cases to the participants; then we asked them to perform a series of training activities (the same for both the applications), consisting of bibliographic searches, similar to T1 ... T5, described below.

Five quantitative search tasks. The five tasks regarding specific searches required the filling, on a Web form, of a numerical answer. Before performing each task, users were asked to read and understand the question; then to insert the start time (hour, minutes and seconds), find the solution of the search task using VisualBib, insert the answer, and finally report the time at the end of the activity. In order to acquire effective times, the form did not accept wrong or empty answers, forcing the user to find the correct value. In the following T1 ... T5 items, the notation A_i indicates a generic author described by name, surname, Scopus Id, affiliation and subject-area.

T1 *Most productive year*: consider the publications of the author A_1. In which year did he/she write the highest number of papers?

T2 *Number of publications in collaboration*: consider again the publications of the author A_1 in a specified time interval. How many of them were written in collaboration with author A_2?

T3 *Self citations of a paper*: consider the paper P_1 of the author A_1, published in a specified year. How many times has it been self-cited in other papers by A_1?

T4 *Textual search in a set of authors*: consider, besides the previous author A_1, the author A_2 and the papers written by them, independently or in collaboration, in a specified time interval: how many of them contain a given word in the title?

T5 *Typology of papers for a set of authors*: consider again the authors A_1 and A_2: how many papers of type *Book* did they write, in collaboration or independently, in a specified time interval?

Since we consider the tasks T4 and T5 possibly complex to carry out on the Scopus platform, the participants could leave the answers empty: in this case we discarded the time measurements related to both platforms. Figure 8 shows the distributions of the execution times of the five tasks for the two platforms, discarding the higher outlier data to improve readability of the graphs. The boxes represent the interval between 1^{st} and 3^{rd} quartiles, the black line is the median of the distribution while the circles represents outliers. Having verified that the distributions of the time difference were not normally distributed, we applied the non-parametric Wilcoxon signed-rank test to verify if the difference of execution times on the two platforms were significant for each task. Table 4 summarizes the results: considering a 95% confidence interval, we can conclude that the execution times for each task performed on Scopus were significantly higher to those performed on VisualBib platform.
Nine qualitative items related to SUS_{-01}. The perceived usability level of the application using a simplified version of the well-known SUS (System Usability Scale) questionnaire [48].

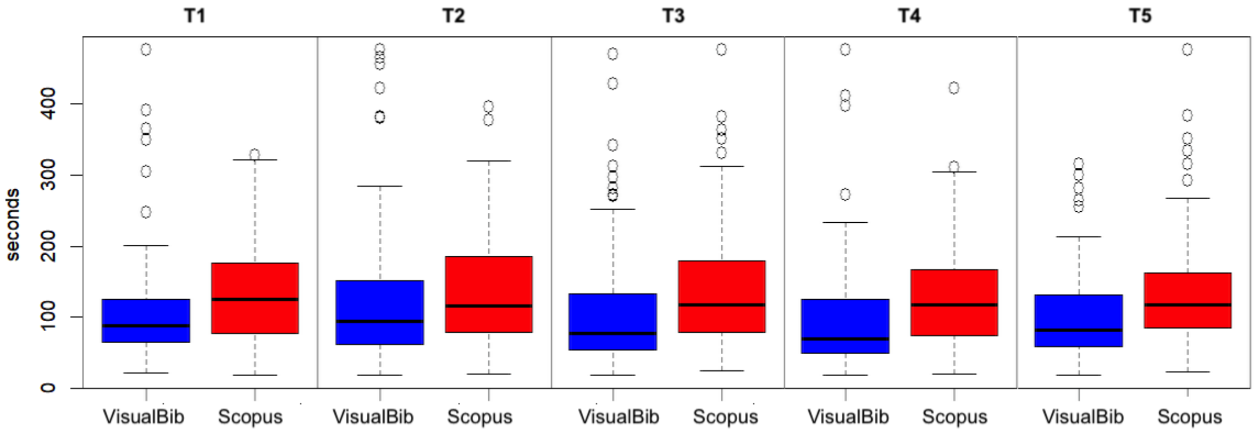

Figure 8: The distributions of the execution times of the five tasks for the two platforms.

task	sample size*	$\mu(t_{sc} - t_{vb})$	W	p-value	significance level	H_0 hypothesis
1	93	18.3s	2631	0.0002168	0.05	**rejected**
2	93	4.3s	2334	0.03789	0.05	**rejected**
3	93	26.7s	2700	0.002174	0.05	**rejected**
4	83	36.1s	2380	0.0003554	0.05	**rejected**
5	84	52.2s	2290	0.0002627	0.05	**rejected**

* Empty answers on tasks T4 and T5 (optional for Scopus) were discarded

Table 4: The results of a Wilcoxon signed-rank test applied to task execution times on VisualBib (t_{vb}) and Scopus (t_{sc}) on the null-hypothesis $H_0 : t_{vb} \geq t_{sc}$

We discarded the first item of the standard SUS, "I think I would like to use this system frequently", to avoid a distortion of the scores in case the system under study is one that would only be used infrequently, and we used the remaining 9 items of the questionnaire with five response options for respondents. Lewis and Sauro [49] studied the effects of dropping an item from the standard SUS questionnaire: specifically, when leaving out the first question, they measured a mean difference from the score the full SUS survey of -0.66 points, considering a 95% confidence interval.

The SUS_{-01} value was computed for each participant and platform, with the formula:

$$SUS_{-01} = \left(\sum_{k=0}^{4} (5 - A_{2k+1}) + \sum_{k=1}^{4} (A_{2k} - 1)\right) * \frac{100}{36}$$

where $A_1, A_2, \ldots, A_9$ are the answers to SUS_{-01} items in the scale 1 (strongly disagree)... 5

Metrics/Platforms	VisualBib	Scopus
Min	33.00	0.00
1st Qu.	58.00	33.00
Median	67.00	42.00
Mean	67.8	43.11
Std. dev.	16.04	15.52
3rd Qu.	81.00	53.00
Max	100.00	83.00

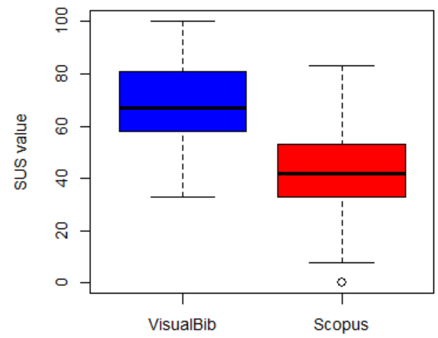

Figure 9: The parameters of the SUS_{-01} distributions (left) and their comparison (right).

(strongly agree); the odd items refer to negative tone questions, even ones to positive tone questions. The distributions, for the two platforms, are summarized in Figure 9. In order to

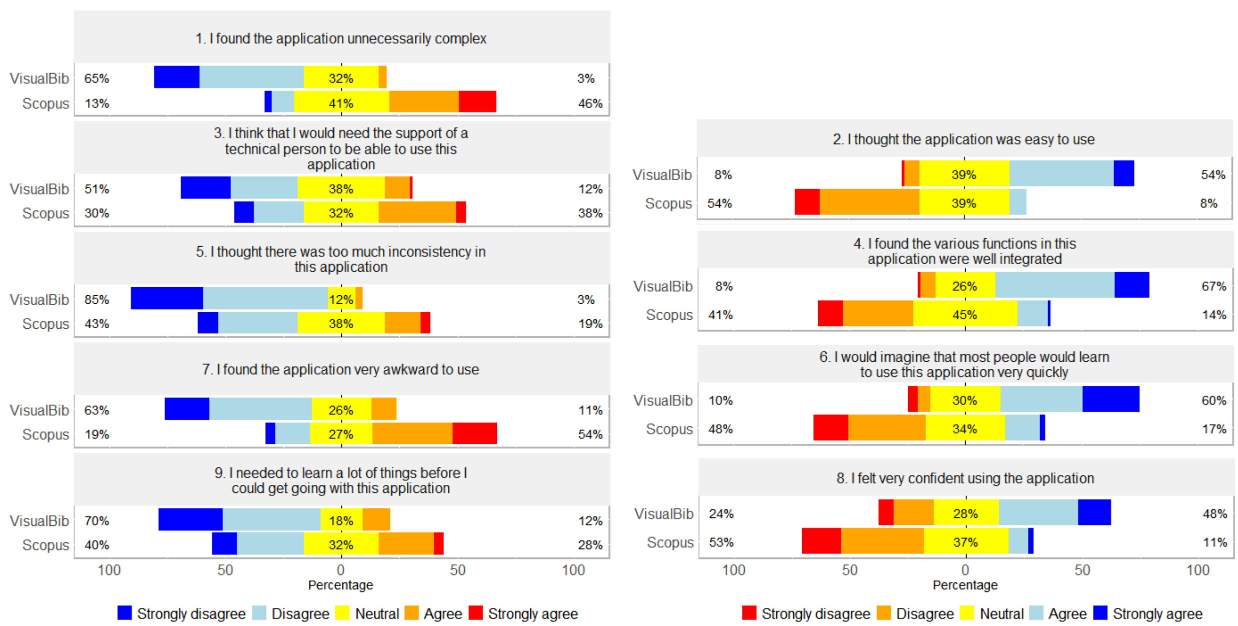

Figure 10: SUS_{-01}: the comparative distributions of answers to the odd items, negative tone (left), and to the even items, positive tone (right).

compare the results we applied a hypothesis t-test for the difference between the means μ_v and μ_s (the VisualBib and Scopus SUS_{-01} means), fixing the null hypothesis $H_0 : \mu_s \geq \mu_v$. We have previously verified the normal distribution of the two samples using the Shapiro-Wilk normality test obtaining the W test statistics $W_{scopus} \simeq 0.987$ and $W_{visualbib} \simeq 0.972$ that are within the 99% acceptance interval [0.9631, 1.0000] of the normal distribution hypothesis.

The test statistic $t \simeq 10.8$ corresponding to a p-value$\simeq 10^{-5}$ which is less than the chosen significance level $\alpha = 0.01$ leading us to reject H_0 in favour of the alternative hypothesis $H_1 : \mu_s < \mu_v$. Regarding the absolute values of SUS_{-01} means, their relatively low values probably reflect the difficulty of a part of the participants in dealing with bibliographic search tasks. Figure 10 shows the comparative distribution of the answers to single SUS_{-01} odd and even items. *Six specific qualitative items.* The items are described in Figure 11; four

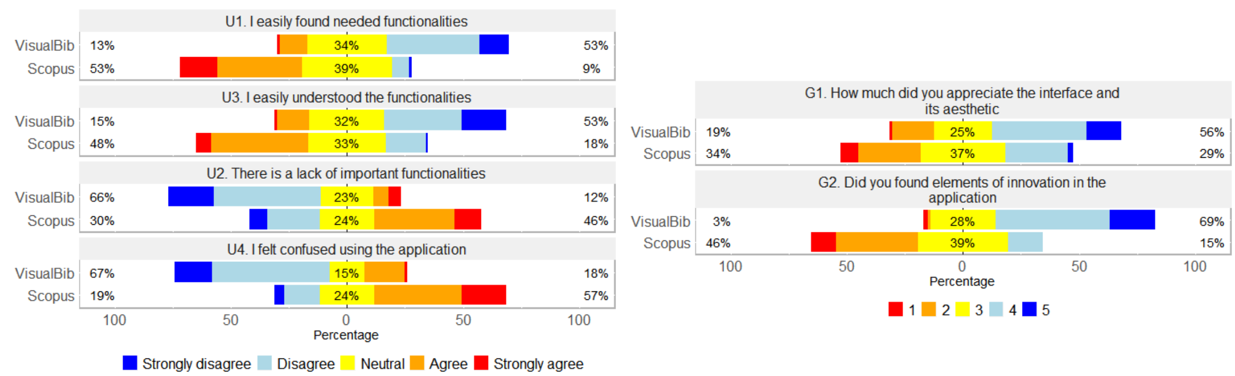

Figure 11: The comparative distributions of answers to the U1, . . . , U4 (left), and to G1 and G2 (right).

of them focused on the user experience (U1, U2, U3, U4) and the last two on the aesthetic and the innovative aspects of the graphical layout (G1, G2). The same Figure 11 shows the distributions of the answers for the U1, U3 (positive tone); U2, U4 (negative tone); and G1 and G2.

Seven features. The participants were asked to attribute a *value* to seven general features (F1, . . . , F7) and then to quantify the perceived level of each features presence in the two platforms. The seven features taken in consideration are present in Figure 12, which also summarizes the given answers with regard to the level of importance and of presence attributed by the users to the 7 identified features. For the value attributed to features, the 5 level descriptors come from *1=not important at all* to *5=absolutely important*, while, for the level of presence in each application, from *1=not present at all* to *5=predominant*.

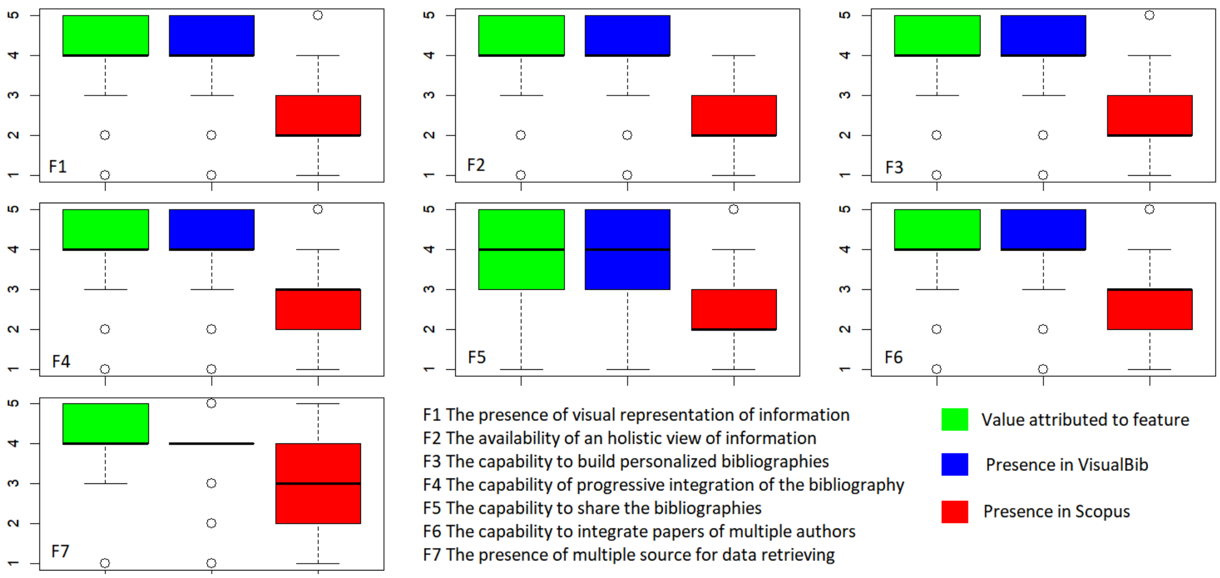

Figure 12: The distributions of attributed values to features and of the perceived level of presence in both platforms.

9. Conclusions and future work

In this paper, we proposed VisualBib, a Web application which offers some original features to support the researchers in creating, saving and sharing their bibliography, starting from a set of papers and authors. The user evaluation carried out and presented in this work and in [3] highlights the positive impact of our visual model and the usability of the narrative views. We are already working on new features and improvements of VisualBib: we planned to expand the list of the repositories to query, to extend the set of considered metadata, to provide statistics data, to introduce tags and semantic data for papers to cluster and reorder them and, finally, to experiment with new forms of visual representation in order to improve the fruition of bibliographic information also for wider bibliographies.

10. Acknowledgements

We would like to thank Meshna Koren and Dave Santucci for confirming the Elsevier/Scopus interest in our project and for the technical support; Mirco Ianese for his contribution in the development of some features of VisualBib. This work has been partially supported by the project SMARTLAND, financed by the University of Udine (2017-19).

References

[1] P. Federico, F. Heimerl, S. Koch, S. Miksch, A survey on visual approaches for analyzing scientific literature and patents, IEEE Transactions on Visualization and Computer Graphics 23 (2017) 2179–2198.

[2] N. J. Van Eck, L. Waltman, Visualizing bibliometric networks, in: Measuring scholarly impact, Springer, 2014, pp. 285–320.

[3] M. Corbatto, A. Dattolo, A web application for creating and sharing visual bibliographies, LNCS 10959 (2018) 78–94. doi:10.1007/978-3-030-01379-0_6.

[4] A. Dattolo, M. Corbatto, Visualbib: Narrative views for customized bibliographies, in: Proceedings of the 22nd International Conference Information Visualisation (IV), July 10-13, Salerno, Italy, 2018, pp. 133–138, IEEE. doi:10.1109/iV.2018.00033.

[5] K. Kucher, A. Kerren, Text visualization techniques: Taxonomy, visual survey, and community insights, in: 2015 IEEE Pacific Visualization Symposium (PacificVis), 2015, pp. 117–121.

[6] G. Costagliola, M. De Rosa, V. Fuccella, S. Perna, Visual languages: A graphical review, Information Visualization 17 (2018) 335–350.

[7] C. Chen, F. IbekweSanJuan, J. Hou, The structure and dynamics of co-citation clusters: A multiple-perspective co-citation analysis, Journal of the American Society for Information Science and Technology 61 (2010) 1386–1409.

[8] B. Lee, M. Czerwinski, G. Robertson, B. B. Bederson, Understanding research trends in conferences using paperlens, in: CHI EA '05 Extended Abstracts on Human Factors in Computing Systems, ACM, 2005, pp. 1969–1972.

[9] Z. Shen, M. Ogawa, S. T. Teoh, K.-L. Ma, Biblioviz: A system for visualizing bibliography information, in: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Vol.60, APVis '06, Tokyo, Japan, 2006, pp. 93–102.

[10] N. Elmqvist, P. Tsigas, Citewiz: A tool for the visualization of scientific citation networks, Information Visualization 6 (2007) 215–232.

[11] N. van Eck, L. Waltman, Software survey: Vosviewer, a computer program for bibliometric mapping, Scientometrics 84 (2010) 523–538.

[12] P. Bergström, D. C. Atkinson, Augmenting the exploration of digital libraries with web-based visualizations, in: 2009 Fourth International Conference on Digital Information Management, 2009, pp. 1–7.

[13] G. Costagliola, V. Fuccella, Cybis: A novel interface for searching scientific documents, in: Proceedings of the 15th International Conference on Information Visualisation (IV), 2011, pp. 276–281.

[14] J. Matejka, T. Grossman, G. Fitzmaurice, Citeology: Visualizing paper genealogy, in: CHI '12 Extended Abstracts on Human Factors in Computing Systems, ACM, 2012, pp. 181–190.

[15] M. Dörk, N. H. Riche, G. Ramos, S. Dumais, Pivotpaths: Strolling through faceted information spaces, IEEE Transactions on Visualization and Computer Graphics 18 (2012) 2709–2718.

[16] N. J. van Eck, L. Waltman, Citnetexplorer: A new software tool for analyzing and visualizing citation networks, Journal of Informetrics 8 (2014) 802 – 823.

[17] P. Klein, F. Müller, H. Reiterer, M. Eibl, Visual information retrieval with the supertable + scatterplot, in: International Conference on Information Visualisation, London, UK, July 10-12, 2002, pp. 70–75.

[18] Aminer platform, accessed on July, 16 2019. URL: `https://aminer.org/`.

[19] Google Scholar, accessed on July, 16 2019. URL: `https://scholar.google.it/`.

[20] Microsoft Academic, accessed on July, 16 2019. URL: `https://academic.microsoft.com/`.

[21] Scopus platform, accessed on July, 16 2019. URL: https://www.scopus.com/.

[22] Web Of Science platform, accessed on July, 16 2019. URL: https://webofknowledge.com/.

[23] CiteSeerX, accessed on July, 16 2019. URL: http://citeseerx.ist.psu.edu.

[24] CrossRef platform, accessed on July, 16 2019. URL: https://www.crossref.org/.

[25] Digital bibliography & library (dblp) project, accessed on July, 16 2019. URL: https://dblp2.uni-trier.de/.

[26] OpenAire platform, accessed on July, 16 2019. URL: https://www.openaire.eu.

[27] OpenCitations organization, accessed on July, 16 2019. URL: http://opencitations.net/.

[28] Orcid platform, accessed on July, 16 2019. URL: https://orcid.org/.

[29] R. Amar, J. Eagan, J. Stasko, Low-level components of analytic activity in information visualization, in: Proceedings - IEEE Symposium on Information Visualization, INFO VIS, 2005, pp. 111–117.

[30] Datacite api services documentation, accessed on July, 16 2019. URL: http://doc.aminer.org/en/latest/.

[31] CrossRef platform APIs, accessed on July, 16 2019. URL: https://github.com/CrossRef/rest-api-doc.

[32] Elsevier Scopus APIs, accessed on July, 16 2019. URL: https://dev.elsevier.com/sc_apis.html.

[33] T. H. Nelson, What's on my mind, Invited talk at the *first Wearable Computer Conference*, 1998. URL: http://www.xanadu.com.au/ted/zigzag/xybrap.html.

[34] T. H. Nelson, A cosmology for a different computer universe: Data model, mechanisms, virtual machine and visualization infrastructure, Journal of Digital Information 5 (2004).

[35] A. Dattolo, F. L. Luccio, A formal description of zz-structures, in: Proceedings of the 1st Workshop on New Forms of Xanalogical Storage and Function, held as part of the ACM Hypertext 2009, number 508 in CEUR, Turin, Italy, 2009, pp. 7–11. URL: http://ceur-ws.org/Vol-508/paper2.pdf.

[36] A. Dattolo, F. L. Luccio, A state of art survey on zz-structures, in: Proceedings of the 1st Workshop on New Forms of Xanalogical Storage and Function, ACM Hypertext 2009, CEUR:508, Turin, Italy, 2009, pp. 1–6. URL: http://ceur-ws.org/Vol-508/paper1.pdf.

[37] M. J. McGuffin, m. schraefel, A comparison of hyperstructures: zzstructures, mspaces, and polyarchies, in: Proceedings of the 15th ACM conference on Hypertext and Hypermedia - Hypertext 2004, ACM Press, Santa Cruz, CA, USA, 2004, pp. 153–162.

[38] A. Dattolo, F. L. Luccio, A new actor-based structure for distributed systems, in: Proceedings of the 30th Jubilee International Convention: Microelectronics, Electronics and Electronic Technologies, Hypermedia and Grid Systems - MIPRO 2007, Opatija, Croatia, 2007, pp. 195–201. URL: http://bit.ly/2XLtdW1.

[39] A. Dattolo, F. Luccio, Formalizing a model to represent and visualize concept spaces in e-learning environments, in: Proceedings of International Conference on Web Information Systems and Technologies - Webist 2008, Funchal, Madeira, Portugal, 2008, pp. 339–346. URL: http://bit.ly/2YdFT7T.

[40] A. Dattolo, F. Luccio, A New Concept Map Model for E-Learning Environments, volume 18 of *Lectures Notes in Business Information Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 404–417. doi:10.1007/978-3-642-01344-7.

[41] M. Corbatto, A. Dattolo, Building, enriching and refining a bibliography in the visualbib platform, Journal of Computer Languages (2019).

[42] Fetch Live Standard: the CORS protocol, accessed on July, 16 2019. URL: https://fetch.spec.whatwg.org/.

[43] Orcid platform public APIs, accessed on July, 16 2019. URL: http://members.orcid.org/api/about-public-api.

[44] M. Bostock, V. Ogievetsky, J. Heer, D^3 data-driven documents, IEEE transactions on visualization and computer graphics 17 (2011) 2301–2309.

[45] D3.js, a javascript library for manipulating documents based on data., 2011. https://d3js.org/ (accessed on July, 16 2019).

[46] M. Corbatto, A. Dattolo, Appinventory: A visual catalogue of web 2.0 and mobile applications for supporting teaching and learning activities, in: Proceedings of the 22nd International Conference Information Visualisation (IV), July 10-13, Salerno, Italy, 2018, pp. 530–535, IEEE. doi:10.1109/iV.2018.00098.

[47] M. Corbatto, A. Dattolo, Exploring appinventory, a visual catalog of applications for assisting teachers and students, Multimedia Tools and Applications (2019). doi:10.1007/s11042-019-08000-6.

[48] J. Brooke, Sus: a retrospective, Journal of usability studies 8 (2013) 29–40.

[49] J. R. Lewis, J. Sauro, Can i leave this one out?: the effect of dropping an item from the sus, Journal of Usability Studies 13 (2017) 38–46.