Scientometrics manuscript No.
(will be inserted by the editor)

Analysis of Co-authorship Graphs of CORE-ranked Software Conferences

Javier Luis Cánovas Izquierdo · Valerio Cosentino · Jordi Cabot

Received: date / Accepted: date

Abstract In most areas of Computer Science (CS), and in the software domain in particular, international conferences are as important as journals as a venue to disseminate research results. This has resulted in the creation of rankings to provide quality assessment of conferences (specially used for academic promotion purposes) like the well-known CORE ranking created by the Computing Research & Education Association of Australasia. In this paper we analyze 102 CORE-ranked conferences in the software area (covering all aspects of software engineering, programming languages, software architectures and the like) included in the DBLP dataset, an online reference for computers science bibliographic information. We define a suite of metrics focusing on the analysis of the co-authorship graph of the conferences, where authors are represented as nodes and co-authorship relationships as edges. Our aim is to first characterize the patterns and structure of the community of researchers in software conferences. We then try to see if these values depend on the quality rank of the conference justifying this way the existence of the different classifications in the CORE-ranking system.

Keywords Co-authorship graph · DBLP dataset · CORE conference ranking · Scientometrics · Computer Science · Software Engineering

Javier L. Cánovas Izquierdo
UOC, Barcelona, Spain
E-mail: jcanovasi@uoc.edu

Valerio Cosentino
AtlanMod – Inria – EMN – LINA, Nantes, France
E-mail: valerio.cosentino@inria.fr

Jordi Cabot
ICREA - UOC, Barcelona, Spain
E-mail: jordi.cabot@icrea.cat

1 Introduction

In Computer Science (CS) and, in particular, in the software area (including all aspects of software development, software engineering, programming languages, architecture, etc.), publications in international conferences are as important as journals as venues to disseminate research results [20, 2]. Providing better visibility of the work, personal contact with the research community and shorter times between submission and publication are part of the reasons that lean many software researchers to prefer conferences rather than journals. This shift on the publication behavior has motivated the creation of mechanisms for the quality assessment of conferences in CS, similar to what was already in place for journals (e.g., JCR impact factor index) to help researchers promote and justify (in front of evaluation or funding agencies) their scientific results.

In the last years, several conference rankings have been proposed (e.g., MAS[1], SHINE[2] or GII-GRIN[3]). Among them, the Computing Research and Education Association of Australasia (CORE) conference ranking mechanism[4] is the most well-known and widespread one. Created by an association of university departments of computer science in Australia and New Zealand, the CORE conference ranking is an ongoing process that provides yearly assessment of major conferences in the computing disciplines (e.g., software, databases, artificial intelligence, etc.) according to four quality ranks (i.e., A*, A, B and C). Conference rankings are determined by a mix of indicators, including citation rates, paper submission and acceptance rates, and the visibility and research track record of the key people hosting the conference and managing its technical program. Although it tries to use objective measures as much as possible (even if some of the factors that influence the ranking are not), the qualification of a conference does not follow an automatic process, and therefore it is subjected to a degree of arbitrariness, starting with the number of ranks. Given the importance of this ranking and its impact on the evaluation of researchers in CS, we believe it is important to keep assessing whether assigned rankings are justified based on observable conference metrics, for instance with regard to the community patterns and structures linked to the conference.

Social network analysis [22] is an active research field where researchers try to better understand communities of people by looking at their internal relationships and subgroups. A social network is then the representation of these connections [13]. In this sense, conferences can be regarded as a specific case of social networks where the authorship details on papers published at the conference represent connections between researchers in the community.

[1] Microsoft Academic Search Conference Ranking. `http://bit.ly/1WMsg4y`

[2] Simple H-Index Estimator. `http://shine.icomp.ufam.edu.br/about.php`

[3] Initiative sponsored by GII (Group of Italian Professors of Computer Engineering) and GRIN (Group of Italian Professors of Computer Science). `http://valutazione.unibas.it/cs-conference-rating/conferenceRating.jsf`

[4] `http://www.core.edu.au`

By using co-authorship graphs, where authors are represented as nodes and co-authorships as edges, we can analyze conferences from a community dimension perspective.

In this paper we define a suite of metrics for co-authorship graphs to study and characterize the community behind the software conferences included in the CORE ranking list. The graphs are built by mining the metadata provided by DBLP, an online reference for bibliographic information on major computer science publications. Metrics are first applied to each conference individually regardless their CORE rank and we then perform correlation analysis among the metrics. Given the relevance of the CORE ranking system as a way to classify conferences, after we study whether conferences ranked differently also present significant differences on their metrics values, thus aiming at uncovering whether the study of conference communities may give some clues on CORE ranks. This is done by doing an analysis of variance complemented with factor and regression analysis. To ensure the replication of our results, we have made public the process to build the co-authorship graphs, such graphs and the results of our study for further evaluation at [17].

The paper is structured as follows. Section 2 describes the methodology we followed to build the co-authorship graphs and Section 3 presents the metrics we defined to evaluate those graphs. Section 4 describes the results of the analysis considering the full set conferences while Section 5 shows the results of the study once conferences are grouped according to their CORE ranks. Section 6 discusses the main results of the analysis. Section 7 describes the identified threats to validity while Section 8 presents the related work. Section 9 ends the paper and presents some future work.

2 Data Collection

Our analysis has been performed on a dataset built from two data sources: (1) the CORE conference ranking list and (2) the DBLP database. The CORE conference ranking classifies conferences in the computing disciplines in four main categories according to their "quality"[5]: *A** for flagship conferences, leading venues in a discipline area; *A* for excellent conferences and highly respected in a discipline area; *B* for good conferences, and well regarded in a discipline area; and *C* for other ranked conference venues that meet minimum standards. The first edition of the CORE Ranking was published in 2008 and there have been three subsequent editions in 2010, 2013 and 2014. In this work we used the CORE conference ranking list of 2014[6].

The DBLP computer science bibliography[7], jointly operated by Schloss Dagstuhl and the University of Trier, is a comprehensive open data collection

[5] A more detailed description about how the classification is defined can be found at `http://www.core.edu.au/documents/RankingDescriptions2014.pdf`

[6] Available at `http://portal.core.edu.au/conf-ranks` as CSV file.

[7] `http://dblp.uni-trier.de`

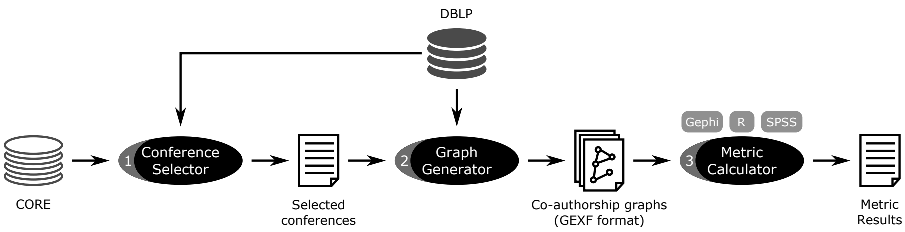

Fig. 1 Data collection process.

on bibliographic meta data in computer science, hosting more than three million publication records ranging from conference papers to technical reports and theses. This data is provided in two formats: (1) as a XML file[8] and (2) as a relational database[9]. In this paper we used the snapshot of the relational database version of DBLP from February, 13^{th} 2016, which is around 220 MB big. In a nutshell, the database includes a set of relational tables to store papers and authors. Papers details include, among others, title, venue where it was published, pages and year. On the other hand, authors data allows keeping track of paper authors and their possible name aliases.

To build our dataset, we defined an extraction process composed of three phases: (1) conference selector, (2) graph generator and (3) metric calculator. Figure 1 illustrates these phases and their corresponding inputs/outputs. The implementation of the process can be found at [17].

The first phase receives the full CORE ranking conference list and filters a subset of its entries to keep only international conferences in the software domain. In particular, the selection process takes all the conferences tagged as *computer software* in the *field of research* property in the CORE list and removes (1) workshops, (2) Australasian/Australian conferences, (3) conferences not included in DBLP, (4) conferences with less than five editions including one in 2014, and (5) conferences not reporting the length of their papers.

The condition checking the editions allows us to select conferences that are alive and have had at least five editions which is the minimum data period we will use in our analysis. Also, even though the DBLP snapshot we are using is from February 2016, we decided to set the last edition to 2014 given that some conferences from 2015 could not have been added to DBLP yet (e.g., conferences taking place in the end of 2015). The condition on the number of pages is due to the fact that the DBLP database does not include the type of a publication (i.e., full paper, short paper, demo, etc.) in their records and therefore we had to resort to the page length to guess what papers were in fact full papers, which are the ones that are taken into account in the CORE ranking to evaluate the conference.

In total, the full CORE ranking conference list for the computer software field includes 271 conferences. The selection process removed 72 workshops,

[8] Available at `http://dblp.uni-trier.de/xml`
[9] Provided by `http://dblp.l3s.de`

Table 1 Summary of the data collection process for the 271 conferences included in the CORE list.

Discarded conferences (reasons)		**Selected conferences (ranks)**	
Reason	**Num.**	**CORE rank**	**Num.**
Workshops	72	A*	12
Australasian/Australian conferences	12	A	29
Not included in DBLP	17	B	43
Number of editions < 5 or last edition < 2014	62	C	18
Not reporting page numbers	4		
Total	167	Total	102

Table 2 Summary of the number of unique authors and papers collected in the last five editions considered.

	Last five editions				
	-1	**-2**	**-3**	**-4**	**-5**
Num. Authors	12,269	12,665	12,352	11,511	12,055
Num. Papers	3,750	3,690	3,833	3,537	3,507

12 Australasian/Australian conferences , 17 conferences not present in DBLP (no name, alias or acronym match was possible), 62 conferences with less than five editions or that had not been hold in 2014, and 4 conferences not reporting the number of pages in DBLP. Additionally, two pair of conferences were considered the same as they were merged along the considered editions (i.e., *MODELS*/*UML*, *CSEE*/*CSEE&T* conferences).

The resulting list is composed of 102 conferences. This includes 12 conferences ranked as A*, 29 conferences ranked as A, 43 conferences ranked as B and 18 conferences ranked as C. Table 1 summarizes the result of this process and Table 2 shows the total number of unique authors and papers processed per edition. Appendix A lists the set of selected CORE-ranked conferences.

The second phase takes the list of selected conferences and builds the corresponding co-authorship graphs. In this kind of graphs, authors who have published a paper in the conference are represented as nodes while co-authorship is represented as an edge between the involved author nodes. Furthermore, the weight of a node represents the number of papers accepted in the conference for an author while the weight of an edge indicates the number of times those author nodes have coauthored a paper in the conference. As commented before, due to the fact that DBLP does not specify the type of publication in a conference, co-authorship graphs will be built for the set of authors having papers with more than 4 pages in the conference, which is a conservative length typically set for full research papers (and therefore we remove short papers, demos, etc.).

As we analyze conferences with at least five editions, we build two types of co-authorship graphs per conference: (1) a complete co-authorship graph representing the last five editions, and (2) a graph for each individual edition. Metrics will use one or the other depending on what they are measuring as explained in the next section. Figure 2 shows an example of a co-authorship graph for the last five editions of the conference *Automated Software Engineer-*

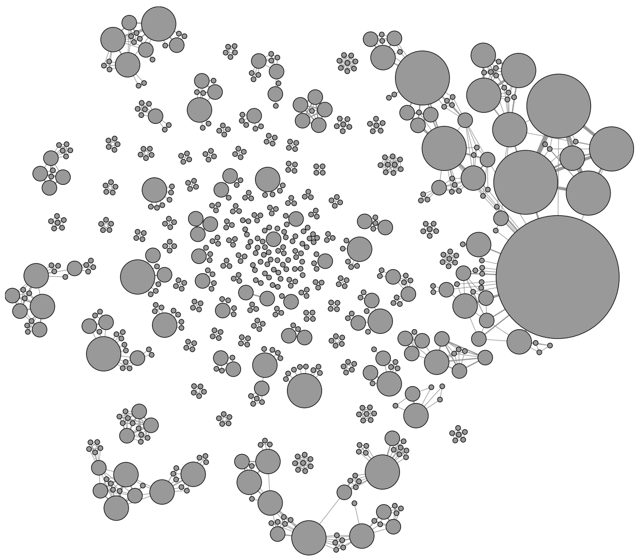

Fig. 2 Example of a co-authorship graph (generated for the last five editions of the conference *Automated Software Engineering*). Nodes represent authors and edges represent co-authorships. The bigger the node, the more papers such author has published in the conference. The thicker the edge, the more papers the involved authors have co-authored. Only papers with more than 4 pages are considered.

ing. In total, we generated 612 graphs, processing 213,286 nodes and 416,740 edges. Graphs are stored in GEXF format, a standard graph exchange format, to promote its reuse and the replicability of our study[10].

The third phase is in charge of calculating the metrics we have considered for the analysis of the conferences. To this end, we rely on graph (i.e., Gephi) and statistical tools (i.e., SPSS and R) to collect the results of the metrics and perform our study[11]. The execution time of the extraction process for all conferences took 6:13 minutes[12]. Next section will describe the metrics included in our study.

3 Metrics

Table 3 shows the metrics we have defined in this study along with a brief description for each of them. These metrics are derived from well-known metrics in graph theory, after selecting a subset we believed were the most relevant in our context and adapting their meaning to the specific case of co-authorship graphs. In total, we have defined 14 metrics organized in three sets according to the period of time they consider: metrics that focus on single editions

[10] The set of generated graphs can be downloaded from [17].

[11] The set of results obtained are available at [17].

[12] Calculated as the average value of time executions of three consecutive executions in an Intel Core i7 machine, with 8 GB of RAM and Windows 7.

Table 3 Metrics used in our study.

	Metric	Definition
1-edition	*Num_Papers*	Number of papers
	Num_Authors	Number of authors
	Papers_per_Author	Ratio of papers per author
	Authors_per_Paper	Ratio of authors per paper
Growth	*Newcomers*	Percentage of new authors (i.e., they have never published in the conference before)
	Survivors	Percentage of authors who publish in two consecutive conferences
	Newcomer_Papers	Percentage of papers whose authors are new to the conference
	Community_Papers	Percentage of papers whose authors are part of the conference community (i.e., they have published at least a paper before)
5-edition	*Avg_Degree*	Average degree of the complete co-authorshipg graph
	Connected_Components	Number of connected components in the complete co-authorship graph
	Graph_Density	Density of the complete co-authorship graph
	Graph_Modularity	Number of modular classes in the complete co-authorship graph
	Community_Size	Number of authors in the complete co-authorship graph
	Prominent_Figures	Percentage of authors with 5+ papers in the complete co-authorship graph

of the conference, metrics that look at the evolution of a conference in two consecutive editions and metrics that look at the global co-authorship graph aggregating the full studied period for the conference. Some other metrics could be defined by combining these ones.

1-edition metrics. This group of metrics characterizes essential information for each conference edition. To calculate them for a conference, we compute separately the value of the metric for each of the five co-authorship graphs of the considered editions and then we calculate the average value. We define the following 1-edition metrics:

– The metric *Num_Papers* is the number of papers in a conference edition.
– The metric *Num_Authors* is the number of unique authors in a conference edition.
– The metric *Papers_per_Author* is the ratio of papers per author in a conference edition.
– The metric *Authors_per_Paper* is the ratio of authors per papers in a conference edition.

Growth metrics. This set of metrics measures how the co-authorship graphs representing individual editions for the same conference evolve along the considered time span. In particular we are interested in assessing the flow of authors and paper authorships across consecutive editions of the conferences. To calculate them, we first obtain the growth value between each pair of editions (as a percentage) and then we compute the average value. We define four metrics:

– The metric *Newcomers* is the percentage of authors who are new to the conference, that is, people that have never published a paper in there before (where by *before* we mean the first considered edition as starting point).

Therefore, instead of considering the full history of each conference we count all authors as newcomers in the first edition and then we calculate how these value decreases in subsequent editions. Once an author has published once in a conference, we consider she becomes part of the conference community.
- The metric *Survivors* is the percentage of authors who repeat between two consecutive editions of the conference (i.e., they have at least one paper published in both editions).
- The metric *Community_Papers* is the percentage of papers for which all authors are part of the community.
- The metric *Newcomers_Papers* is the percentage of papers for which all authors are new to the conference.

5-edition metrics. This group of metrics assesses a set of properties in the complete co-authorship graph of a conference (the one aggregating all data for the period of time considered in our study). In particular, we consider the following metrics:

- The metric *Avg_Degree* calculates the average degree of the nodes of the graph and can help us to individually assess the collaboration degree of the authors of a conference.
- The metric *Connected_Components* determines the number of subgraphs in the co-authorship graph. It helps us to identify sets of authors that are mutually reachable through chains of co-authorships.
- The metric *Graph_Modularity* calculates the number of modular classes in the graph (applying [2]). It measures how well a network decomposes into modular communities. This structure, often called a community structure, describes how the network is compartmentalized into sub-networks. When applied to co-authorship graphs, it may help us to detect clusters of nodes where authors commonly collaborating together. Note that the number of modular classes is equal or greater than the number of connected components. When it is greater, it reveals that there are collaboration clusters inside the connected components. For instance, figures 3a and 3b show the connected components and modular classes identified in the complete co-authorship graph used as example in Figure 2, respectively. In the example, the number of connected components is less than the number of modular classes (note that there are several modular classes in the same connected component located on the top right of the graphs). Together with the previous metric, they give an indication of the collaboration level of the community, showing how many subcommunities we may find and whether they collaborate or not with each other.
- The metric *Graph_Density* is the relative fraction of edges in the graph, that is, the ratio between the actual number of edges and the maximum number of possible edges in the graph. This metric highlights how far is a community of the ideal goal of complete collaboration across all authors.

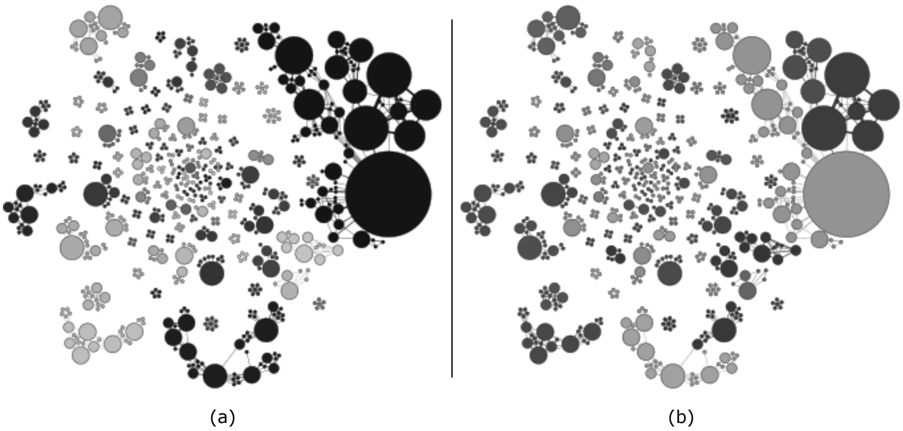

Fig. 3 (a) Connected components and (b) modular classes identified in the complete co-authorship graph for the *Automated Software Engineering* conference (shown in Figure 2).

- The metric *Community_Size* counts the number of unique authors in the complete co-authorship graph. This metric allows us to assess the size of the conference community.
- The metric *Prominent_Figures* is the percentage of authors with 5 or more papers in the complete co-authorship graph. We defined this metric to identify important authors in the conference community and its frequency.

4 Conference Analysis

We start our study by analyzing the metrics we have defined for the full set of conferences regardless of their CORE rank. We first show an overview of the metrics values and then we perform a correlation analysis among them.

4.1 Metric Analysis

For illustration purposes, Table 4 shows the metric values for some CORE-ranked conferences within the software engineering field. At first sight, the majority of the metric values do not seem to be significantly different among the conferences.

We calculated the main descriptive statistics of the metrics for all conferences. Due to the large number of conferences, Table 5 shows the aggregated results and Figure 4 the boxplot for each metric. The list of metric values for each individual conference can be found at [17]. The table also includes the trend in the last five editions for the sets of 1-edition and growth metrics. To measure metric trends we compute the Spearman correlation (ρ) between the value of the metric in each edition and the time axis. Spearman allows us to quantify monotone trends: as the time axis is monotonically increasing, strong

Table 4 Metric results for a subset of CORE-ranked conferences.

	CORE rank							
	A*		A		B		C	
Metric	**ICSE**[a]	**OOPSLA**[b]	**ASE**[c]	**IREC**[d]	**MoDELS**[e]	**SLE**[f]	**ICPC**[g]	**ICGSE**[h]
Num_Papers	97.600	56.200	52.200	39.200	48.600	21.000	21.000	27.200
Num_Authors	320.200	178.800	178.600	118.000	152.000	65.000	65.000	77.200
Papers_per_Author	1.087	1.081	1.062	1.067	1.108	1.098	1.100	1.122
Authors_per_Paper	3.583	3.439	3.567	3.223	3.491	3.402	3.404	3.177
Newcomers	70.910%	74.091%	80.417%	72.219%	68.685%	77.332%	77.311%	62.567%
Survivors	18.665%	15.690%	12.276%	18.287%	20.088%	14.591%	16.494%	28.551%
Newcomer_Papers	41.718%	38.676%	55.020%	49.003%	39.168%	52.168%	54.845%	43.799%
Community_Papers	9.889%	7.452%	5.411%	10.299%	8.173%	8.637%	7.242%	21.224%
Avg_Degree	4.168	3.986	3.913	3.737	4.053	4.985	3.654	3.549
Connected_Components	173.000	114.000	131.000	84.000	98.000	46.000	46.000	58.000
Graph_Density	0.003	0.006	0.005	0.008	0.007	0.019	0.014	0.013
Graph_Modularity	184.000	118.000	135.000	90.000	102.000	50.000	49.000	60.000
Community_Size	1211.000	715.000	732.000	456.000	570.000	264.000	260.000	284.000
Prominent_Figures	2.312%	1.818%	0.956%	1.974%	3.333%	1.894%	2.308%	4.225%

[a] International Conference on Software Engineering
[b] ACM Conference on Object Oriented Programming Systems Languages and Applications
[c] Automated Software Engineering Conference
[d] IEEE International Requirements Engineering Conference
[e] International Conference on Model Driven Engineering Languages and Systems
[f] International Conference on Software Language Engineering
[g] International Conference on Program Comprehension
[h] IEEE International Conference on Global Software Engineering

Table 5 Summary of the metric values.

Metric	**Min**	**1^{st} Qu.**	**Median**	**Mean**	**3^{rd} Qu.**	**Max.**	**SD (σ)**	**Trend ρ (σ)**
Num_Papers	9.800	21.600	30.000	35.920	40.850	130.600	22.246	0.063 (0.616)
Num_Authors	31.200	65.000	91.000	114.300	147.100	385.800	75.415	0.181 (0.563)
Papers_per_Author	1.011	1.042	1.066	1.072	1.095	1.315	0.043	0.020 (0.522)
Authors_per_Paper	2.100	3.027	3.304	3.361	3.542	5.417	0.547	0.364 (0.486)
Newcomers	56.100%	74.750%	79.450%	79.170%	83.470%	96.240%	7.309%	-0.680 (0.351)
Survivors	1.827%	9.330%	12.740%	13.030%	16.340%	31.940%	5.760%	0.005 (0.574)
Newcomer_Papers	28.200%	48.730%	56.360%	57.860%	66.880%	89.260%	13.382%	-0.680 (0.354)
Community_Papers	0.000%	4.082%	6.838%	7.573%	9.508%	25.660%	5.267%	0.317 (0.541)
Avg_Degree	1.904	3.023	3.436	3.685	3.921	16.320	1.535	N.A
Connected_Components	24.000	60.000	83.500	92.780	106.500	336.000	55.311	N.A
Graph_Density	0.002	0.007	0.009	0.010	0.013	0.028	0.005	N.A
Graph_Modularity	28.000	60.250	86.000	95.400	108.500	336.000	55.378	N.A
Community_Size	130.000	261.000	394.500	472.900	573.800	1602.000	308.663	N.A
Prominent_Figures	0.000%	0.470%	1.034%	1.353%	1.777%	6.154%	1.236%	N.A

correlation indicates presence of a trend in the metric. Furthermore, being a non-parametric test, Spearman is suitable when dealing with small samples as ours. This method has also been used in other works such as the study presented by Vasilescu et al. [20].

Interpretation of the results is performed in Section 6 but we highlight here some relevant data points. Regarding the set of 1-edition metrics, note the high standard deviation for the metric *Num_Authors*, mainly due to the presence of outliers (e.g., the conferences *Conference on Agile Software Development* and *International Conference on Software Engineering* have the values 38.6 and 320.2 for this metric, respectively). On the other hand, the values of the metrics *Papers_per_Author* and *Authors_per_Paper* seem to be similar for all the considered conferences according to the standard deviation. With regard to the metric *Authors_per_Paper*, the correlation analysis shows a remarkable positive trend in the last five editions.

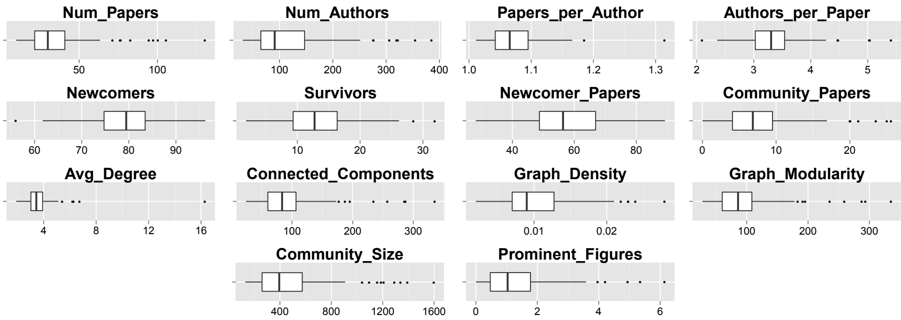

Fig. 4 Boxplots of each metric.

The main summary statistics regarding growth metrics shows high mean values for the metrics *Newcomers* and *Newcomer_Papers* as well as a high decreasing trend along the last five editions (see trends for metrics *Newcomers* and *Newcomer_Papers*). These results are influenced by the time window considered in the metric definition. Recall that we consider the last 5 editions of the conferences and all authors in the first considered edition are classified as newcomers and therefore the value of these metrics reflect the cumulative variation from that initial set in the following 4 editions.

Regarding the 5-edition metrics, note the extremely low values for the metric *Graph_Density* (as well as the low standard deviation). On the other hand, the values for the metrics *Graph_Modularity* and *Connected_Components* seem to be quite similar, which require further correlation analysis to be presented in the following section.

4.2 Correlation Analysis

The correlation analysis allows us to study the relationships between the metrics. We calculate the Pearson correlation coefficient between each single pair of metrics to detect the presence of linear trends.

Table 6 shows the results of the correlation study. The study shows two main groups of highly correlated metrics. On the one hand, there is a high positive correlation among the metrics *Num_Papers*, *Num_Authors*, *Connected_Components*, *Graph_Modularity* and *Community_Size*. This results may be expected for some pairs of metrics, for instance, between *Num_Authors* and *Num_Papers*, *Num_Authors* and *Community_Size*, or *Num_Authors Connected_Components*.

On the other hand, there is a very strong positive correlation (i.e., $\rho = 1$) between the metrics *Graph_Modularity* and *Connected_Components*, which means that each of the metrics is a perfect monotone function of the other. This fact will be also discussed further later on.

Table 6 Correlation Analysis.

	Num_Papers	*Num_Authors*	*Papers_per_Author*	*Authors_per_Paper*	*Newcomers*	*Survivors*	*Newcomer_Papers*	*Community_Papers*	*Avg_Degree*	*Connected_Components*	*Graph_Density*	*Graph_Modularity*	*Community_Size*	*Prominent_Figures*
Num_Papers	1.00													
Num_Authors	0.95	1.00												
Papers_per_Author	0.53	0.43	1.00											
Authors_per_Paper	0.28	0.52	0.10	1.00										
Newcomers	-0.13	-0.10	-0.50	-0.01	1.00									
Survivors	0.21	0.17	0.58	0.02	-0.66	1.00								
Newcomer_Papers	-0.24	-0.30	-0.45	-0.35	0.71	-0.78	1.00							
Community_Papers	0.08	-0.05	0.50	-0.36	-0.67	0.76	-0.51	1.00						
Avg_Degree	0.28	0.52	0.14	0.79	-0.15	0.14	-0.40	-0.11	1.00					
Connected_Components	0.87	0.77	0.33	0.06	0.14	-0.11	0.19	-0.10	0.01	1.00				
Graph_Density	-0.70	-0.66	-0.11	-0.13	-0.14	0.13	-0.07	0.19	-0.02	-0.73	1.00			
Graph_Modularity	0.90	0.81	0.35	0.10	0.11	-0.08	0.15	-0.10	0.05	1.00	-0.73	1.00		
Community_Size	0.94	0.99	0.39	0.51	-0.06	0.08	-0.20	-0.12	0.51	0.82	-0.68	0.85	1.00	
Prominent_Figures	0.23	0.18	0.72	0.03	-0.63	0.84	-0.69	0.70	0.14	-0.08	0.15	-0.05	0.10	1.00

5 CORE-ranked Conference Analysis

We now study the conferences by groups, which we build according to their CORE ranks. Table 7 and Figure 5 show the mean values (and standard deviation) and the bloxplots, respectively, of the metrics for each CORE rank. As can be noted, some metrics seem to behave differently according to the CORE rank (e.g., see the mean values for the metrics *Num_Authors* or *Newcomer_Papers* in the table) while others do not seem to present differences (e.g. see metrics *Papers_per_Authors* or *Prominent_Figures*).

For illustration purposes and to take a look from a different perspective, Figure 6 shows the evolution of a subset of metrics according to the CORE rank along the last five editions. As can be seen, *Num_Authors*, *Newcomers* and *Avg_Degree* metrics seem to behave differently among CORE ranks, in particular, with regard to A* conferences. On the other hand, *Papers_per_Author* metric does not reveal any remarkable difference.

To verify this behavior, we perform a further analysis to study our metrics on a CORE-ranked basis. We first perform an analysis of variance to assess whether individual metrics present different behavior according to the CORE rank. We then conduct a regression analysis to study the set (or different subsets) of metrics which play a fundamental role to discern between CORE ranks.

5.1 Analysis of Variance

To analyze the set of CORE-ranked conferences we want to assess whether the distributions of a given metric are different for the distinct CORE rank values. This assessment is usually performed in two steps: (1) a global null hypothesis is tested, which checks if there are significant differences in the distributions; and then, since there are four different rank values, (2) a pairwise comparison is used to test sub-hypotheses for each pair of groups. To accomplish the first step we apply ANOVA while in the second step we apply the Tukey's range test.

Table 7 Summary of the metrics for each CORE rank.

	CORE rank							
	A*		**A**		**B**		**C**	
Metric	**Mean**	**SD (σ)**	**Mean**	**SD (σ)**	**Mean**	**SD (σ)**	**Mean**	**SD (σ)**
Num_Papers	51.250	20.742	32.186	12.540	36.256	26.653	30.889	20.765
Num_Authors	188.550	68.488	102.069	43.090	109.842	87.491	95.144	64.805
Papers_per_Author	1.074	0.029	1.061	0.037	1.074	0.051	1.085	0.041
Authors_per_Paper	4.027	0.693	3.344	0.447	3.195	0.507	3.343	0.345
Newcomers	67.558%	5.080	72.909%	11.318	76.083%	7.846	74.709%	13.818
Survivors	16.142%	2.830	12.889%	5.221	12.316%	5.286	12.905%	8.356
Newcomer_Papers	41.615%	7.310	56.662%	11.207	61.100%	10.888	62.858%	16.968
Community_Papers	7.095%	3.829	7.303%	5.072	7.701%	4.959	8.023%	7.187
Avg_Degree	4.949	1.115	3.553	0.636	3.491	2.097	3.520	0.616
Connected_Components	90.583	37.430	82.276	29.956	100.349	63.498	93.111	74.117
Graph_Density	0.007	0.002	0.010	0.004	0.010	0.005	0.012	0.007
Graph_Modularity	99.083	36.946	84.552	29.599	101.930	64.084	94.833	73.681
Community_Size	734.167	258.164	419.862	176.667	459.372	356.683	416.444	316.802
Prominent_Figures	1.824%	0.878	1.191%	1.289	1.174%	0.987	1.729%	1.732

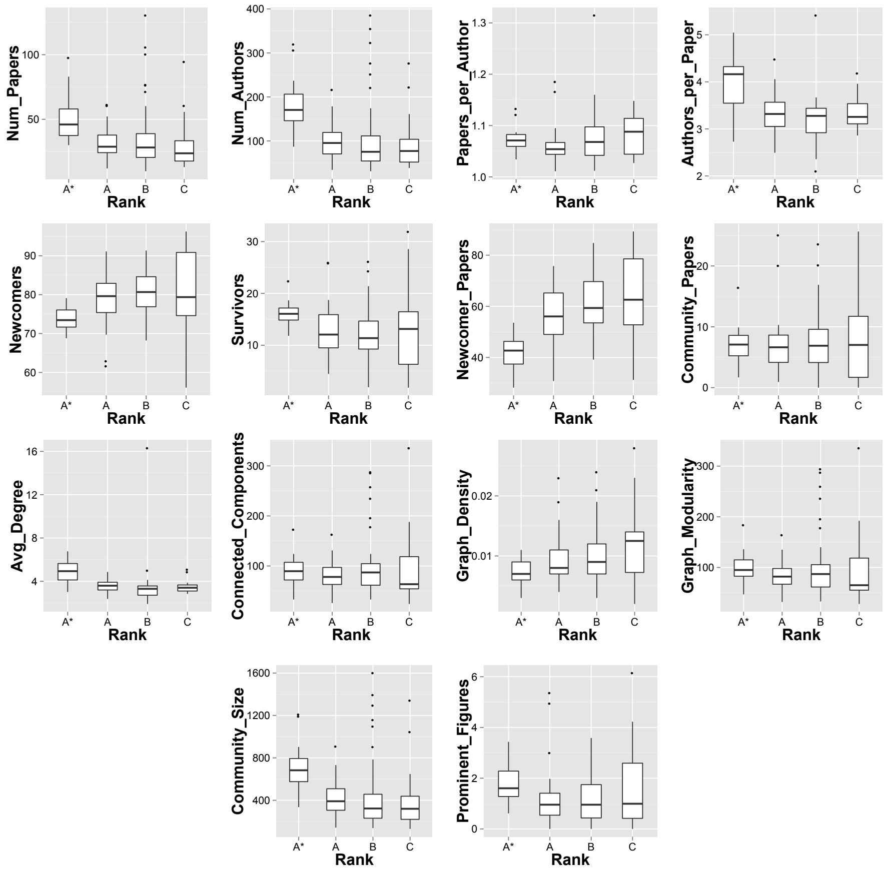

Fig. 5 Boxplots of the metrics for each CORE rank.

ANOVA assumptions require the data to both follow a normal distribution and equality (or "homogeneity") of variances. To check these assumptions we

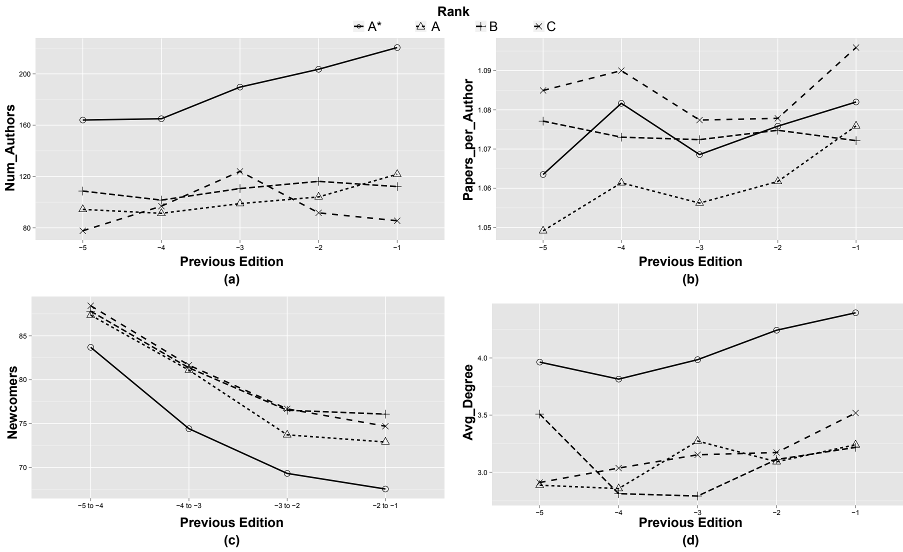

Fig. 6 Summary of the metrics (a) *Num_Authors*, (b) *Papers_per_Author*, (c) *Newcomers* and (d) *Avg_Degree* per CORE rank for the last five editions.

apply the Saphiro-Wilk normality test for the former and the Barlett test for the latter. As the null hypotheses in these tests check that the data follows a normal distribution and that variances are homogeneous, respectively, we expect the p-value to be higher than a confidence level. To avoid clutter when reporting p-values, we superscript the results using the following convention: no superscript corresponds to p-value ≥ 0.05 , * corresponds to $0.01 \leq$ p-value < 0.05, ** corresponds to $0.001 \leq$ p-value < 0.01 and *** corresponds to p-value < 0.001. Table 8 shows the ANOVA analysis for the metrics considered in our study. As can be seen no metric passes the assumption tests and therefore we cannot trust the ANOVA results, even if for some metrics, the p-value is significant (as it happens, for instance, with *Num_Authors* or *Newcomers*).

For these cases a non-parametric test is usually applied to confirm or discard the reults. Thus, the global null hypothesis is normally tested by applying the Kruskall-Wallis one-way analysis of variance by ranks [9] while the pairwise comparison is checked with the rank-based Wilcoxon-Mann-Whitney test [23] with Bonferroni correction [3]. However, simulation studies suggest that the Wilcoxon-Mann-Whitney test is not robust to unequal population variances, especially in the unequal sample size case [26], which is our case. Therefore we propose to employ the multiple contrast test procedure $\tilde{\text{T}}$ [11], which is robust against unequal population variances. $\tilde{\text{T}}$ procedure takes as input a type of contrast and the threshold for the family-wise error rate. We use the Tukey-type contrast as it allows us to check all pairwise comparisons, while we use the traditional threshold of 5%. Among its outputs, the procedure returns an overall p-value for the test as well as the p-values for each pair.

Table 8 ANOVA analysis of the metrics according to the CORE rank.

Metric	Norm. test	Var. Homog.	ANOVA
Num_Papers	$\times^{***}$	$\times^{**}$	$\times$
Num_Authors	$\times^{***}$	$\times^{**}$	$\checkmark^{**}$
Papers_per_Author	$\times^{***}$	$\checkmark$	$\times$
Authors_per_Paper	$\times^{***}$	$\checkmark$	$\checkmark^{***}$
Newcomers	$\checkmark$	$\times^{***}$	$\checkmark^{*}$
Survivors	$\times^{*}$	$\times^{**}$	$\times$
Newcomer_Papers	$\checkmark$	$\times^{*}$	$\checkmark^{***}$
Community_Papers	$\times^{***}$	$\checkmark$	$\times$
Avg_Degree	$\times^{***}$	$\times^{***}$	$\checkmark^{*}$
Connected_Components	$\times^{***}$	$\times^{***}$	$\times$
Graph_Density	$\times^{***}$	$\times^{**}$	$\times$
Graph_Modularity	$\times^{***}$	$\times^{***}$	$\times$
Community_Size	$\times^{***}$	$\times^{**}$	$\checkmark^{*}$
Prominent_Figures	$\times^{***}$	$\times^{*}$	$\times$

Table 9 shows the $\tilde{\mathrm{T}}$ analysis results. As can be seen, we obtain significant differences for the distributions of the metrics *Num_Papers*, *Num_Authors*, *Authors_per_Paper*, *Newcomers*, *Survivors*, *Newcomer_papers*, *Avg_Degree*, *Community_Size* and *Prominent_Figures*. When studying the results for each pair of conferences, we observe that there are significant differences in the distributions of the metrics *Num_Papers*, *Num_Authors*, *Authors_per_Paper*, *Newcomers*, *Avg_Degree* and *Community_Size* for all pairs involving conferences ranked as A*. The metrics *Survivors* and *Prominent_Figures* shows that there are significant differences for the pairs involving conferences ranked as A*-A and A*-B. No significant differences are observed for A-B, A-C or B-C pairs that could suggest these conferences behave differently.

5.2 Regression Analysis

Previous analysis assesses how each metric individually behaves with regard to the CORE rank. Since the individual analysis only shows significant results to distinguish A* conferences form the rest we now study whether a set of metrics (if any) as a group may help to discern more precisely the CORE rank of a conference.

To this aim we perform an ordinal regression analysis where we study how one or more independent variables (our metrics) influence to derive our dependent variable (i.e., the CORE rank). Building an ordinal regression model can also help us to predict the CORE rank given a set of values for the metrics.

As this kind of regression requires meeting the multicollinearity assumption, we first apply factor analysis. Furthermore, factor analysis allows us to reduce the data and identify underlying variables, or factors, that explain the pattern of correlations studied before. Factor analysis also help us to uncover potential latent structures within the variables.

Table 9 $\tilde{\mathrm{T}}$ and pairwise analysis of the metrics according to the Rank factor.

Metric	**$\tilde{\mathrm{T}}$ p-value**	**Tukey's Range Test**		
Num_Papers	$\checkmark^{***}$	A* vs. A $\rightarrow \checkmark^{***}$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \checkmark^{**}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Num_Authors	$\checkmark^{***}$	A* vs. A $\rightarrow \checkmark^{***}$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \checkmark^{**}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Papers_per_Author	$\times$	$\times$		
Authors_per_Paper	$\checkmark^{**}$	A* vs. A $\rightarrow \checkmark^{*}$ A* vs. B $\rightarrow \checkmark^{**}$ A* vs. C $\rightarrow \checkmark^{*}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Newcomers	$\checkmark^{***}$	A* vs. A $\rightarrow \checkmark^{***}$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \checkmark^{**}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Survivors	$\checkmark^{**}$	A* vs. A $\rightarrow \checkmark^{*}$ A* vs. B $\rightarrow \checkmark^{**}$ A* vs. C $\rightarrow \times$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Newcomer_papers	$\checkmark^{***}$	A* vs. A $\rightarrow \times$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \times$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Community_papers	$\times$	$\times$		
Avg_Degree	$\checkmark^{***}$	A* vs. A $\rightarrow \checkmark^{***}$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \checkmark^{***}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Connected_Components	$\times$	$\times$		
Graph_Density	$\times$	$\times$		
Graph_Modularity	$\times$	$\times$		
Connected_Components	$\times$	$\times$		
Community_Size	$\checkmark^{***}$	A* vs. A $\rightarrow \checkmark^{***}$ A* vs. B $\rightarrow \checkmark^{***}$ A* vs. C $\rightarrow \checkmark^{**}$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$
Prominent_Figures	$\checkmark^{*}$	A* vs. A $\rightarrow \checkmark^{*}$ A* vs. B $\rightarrow \checkmark^{**}$ A* vs. C $\rightarrow \times$	A vs. B $\rightarrow \times$ A vs. C $\rightarrow \times$	B vs. C $\rightarrow \times$

5.2.1 Factor Analysis

Factor analysis assumptions require the observations to be independent, thus we remove the metrics *Authors_per_Paper* and *Papers_per_Author* (as they depend on both *Num_Authors* and *Num_Papers*). We therefore use the remainder 12 metrics for our analysis. Factor analysis performs first an extraction process to obtain a set of components and then a rotation method is applied to facilitate the interpretation of the results.

As extraction method we use principal component analysis. To decide the number of components to extract we draw a scree plot, which helps us to visually assess the number of components explaining most of the variability in the data. A scree plot displays the eigenvalues (i.e., amount of variance in the metrics accounted for by each component) associated with a component in descending order versus the number of the component. Figure 7 shows the scree plot for our principal component analysis. Generally, we extract the components on the steep slope as the components on the shallow slope contribute little to the solution. In our case the last big drop occurs between the third and fourth components, so we decide to extract three components.

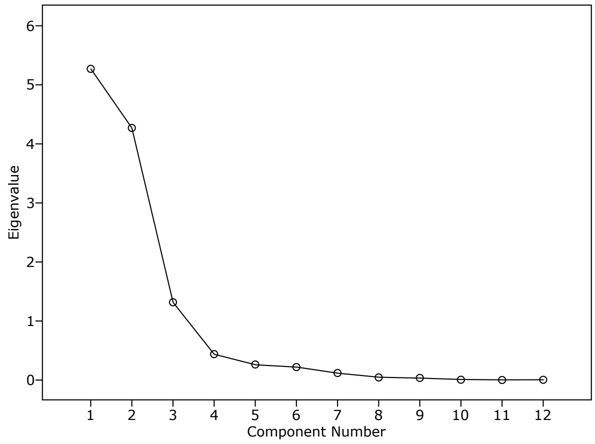

Fig. 7 Scree Plot displaying the eigenvalues associated with a component in descending order versus the number of the component.

Table 10 Eigenvalues, variance accounted for by each component to the total variance and cumulative variance for each component in the principal component analysis.

Component	Eigenvalue	% of Variance	Cumulative %
1	5.084	42.367	42.367
2	4.162	34.684	77.051
3	1.611	13.422	90.473

Table 10 shows the eigenvalue of each component (*Eigenvalue* column), the variance accounted for by each component to the total variance in all of the metrics (*% of Variance* column) and the percentage of variance accounted for by the first n components (*Cumulative %* column). As can be seen, the usage of three components explain nearly 90% of the variability in the original 12 metrics, so we can considerably reduce the complexity of the data set by using these three components, with only a 10% loss of information.

The previous analysis produces a component matrix where a rotation method is usually applied to clarify and simplify the results. We apply a Varimax rotation, which minimizes the number of variables that have high loadings on each component and therefore simplifies the interpretation of the components by highlighting the main variables in each one. Table 11 shows the rotated component matrix. Boxes have been added to facilitate the identification of the main variables loading each component.

We extract three components:

– The first component is most highly correlated with *Graph_Modularity*, *Connected_Components*, *Num_Papers*, *Community_Size*, *Num_Authors*, and *Graph_Density*. These results confirm our previous correlation study.
– The second component is most highly correlated with *Survivors*, *Prominent_Figures*, *Community_Papers*, *Newcomers* and *Newcomer_Papers*.
– The third component is most highly correlated with *Avg_Degree* but also includes a number of other variables with a relative weight.

Table 11 Rotated Component Matrix produced by the principal component analysis.

	Component		
Metric	**1**	**2**	**3**
Graph_Modularity	0.976	-0.100	-0.081
Connected_Components	0.955	-0.126	-0.127
Num_Papers	0.953	0.205	0.172
Community_Size	0.903	0.039	0.415
Num_Authors	0.881	0.136	0.439
Graph_Density	-0.820	0.161	0.050
Newcomers	0.097	-0.960	-0.112
Survivors	0.017	0.953	0.065
Prominent_Figures	0.036	0.894	0.066
Community_Papers	-0.046	0.865	-0.281
Newcomer_Papers	0.006	-0.816	-0.438
Avg_Degree	0.123	0.077	0.948

Table 12 Ordinal regression model results.

				95% Confidence Interval	
Component	Coef.	Std. Error	Sig.	Lower Bound	Upper Bound
1	0.104	0.113	0.357	-0.118	0.327
2	0.223	0.114	0.050	0.000	0.446
3	0.701	0.180	0.000	0.348	1.054

5.2.2 Ordinal Regression

Once identified the components, we apply ordinal regression using them as covariates and the CORE rank as our ordinal dependent variable. Table 12 shows the effect of our components in the regression model[13]. For each component, we show the coefficient in the model, the standard error, the significance of the test and the bounds of the confidence interval.

The signs of the coefficients for components can give important insights into the effects of such components in the model. Positive (negative) coefficients indicate positive (inverse) relationships between components and the outcome category (i.e., the CORE rank). An increasing value of a component with a positive coefficient corresponds to an increasing probability of being in one of the higher cumulative outcome categories. In our context, an increasing value of a component with a positive coefficient corresponds to an increasing probability of having higher CORE ranks.

The significance of the test for the second and third components suggests that its observed effect is not due to chance. Since the coefficients are positive, as the corresponding component increases, so does the probability of being in one of the higher CORE ranks. To interpret this result, it is important to note that the components we are using in the ordinal regression model are correlated with our metrics (recall Table 11). The second component is highly correlated with the metrics *Survivors*, *Prominent_Figures*, *Community_Papers*, *Newcomers* and *Newcomer_Papers*. In particular, the correlation is negative with regard to the metrics *Newcomers* and *Newcomer_Papers*, and positive with regard to the metric *Survivors*, *Prominent_Figures*, *Community_Papers*. This means that when the metrics *Newcomers* and *Newcomer_Papers* increases,

[13] We obtained a R^2 value of 0.243 (Nagelkerke index)

the probability of being low ranked conferences increases, and when the metric *Community_Papers* increases, so does the probability of being high ranked conferences. Similar reasoning can be applied for the third component. The third component is positively correlated with the metrics *Avg_Degree*, thus meaning that when they increase, the probability of being high ranked conferences increases.

The results from the ordinal regression also allow us to identify the most influential component to infer the CORE rank. Thus, as the value for the coefficient obtained for the third factor is higher than for the second factor (recall 0.701 vs 0.223), the former has bigger impact in the model (i.e., it is more helpful to infer the CORE rank). Finally, while the regression model may also help to classify other conferences not included in the CORE rank and study how they differ each other, it is important to be aware of the loss of information in the factor analysis (i.e., around a 10%) as well as the correlation of the components in the rotated component matrix.

6 Discussion

Previous section has provided detailed information on the way the software research community publishes its work in conference venues. In this section we emphasize and interpret some of those findings. We first comment on the results obtained for the whole set of selected conferences (cf. Section 4) and then we comment on the results when considering the CORE rank (cf. Section 5).

6.1 Analysis of CORE-Ranked Conferences

Community size and papers per author remain steady. The trend values for the metrics *Num_Papers* and *Num_Authors* show that the size of the community in terms of number of authors and papers does not tend to vary in time. We also find interesting to remark that the ratio *Papers_per_author* is steady along the editions considered for the full set of CORE-ranked conferences (see trend values for such metric), having around 3 authors per paper. Instad, the trend in the last five editions for the metric *Authors_per_Paper* is not negligible, which suggests an increase in the effort required to produce the paper (e.g., nowadays most conferences require strong validation processes and/or providing tool support which typically implies involving more people in the research project). This behavior can also be observed in the trend value for the metric *Num_Author*, which is slightly positive.

Conferences seem to be stabilized. The results from the set of growth metrics reveal that conference communities are reaching a stable point where newcomers are quickly decreasing[14] and papers coming from community authors tend to monopolize the conference (recall metric results for *Newcomers* and *Community_Papers*). The strong negative trend for the metric *Newcomer_Papers* is not compensated by a high positive trend for the metric *Community_Papers*. This may mean that there is an increasing number of papers written by a mix of newcomers and community members suggesting that perhaps it is easiest for newcomers to enter the community together with an established author.

Collaboration among authors is scarce. The analysis of the set of 5-edition metrics reveals a low collaboration among full set of authors in the conference community according to the metric *Graph_Density*. Note also the relatively high values for the metrics *Connected_Components* and *Graph_Modularity*. This result together with the value for the metric *Avg_Degree* may reveal that, despite the low global collaboration, there exist mature collaboration links but within the sub-communities formed by the components.

Island-based grow behavior. Linked to the previous result, one interesting finding has to do with the high correlation between the metrics *Graph_Modularity* and *Connected_Components*, which may reveal an interesting behavior in the conference communities. We believe that this result denotes a community growth based on islands, where groups of authors grow in the conference community as isolated graphs: the component may grow but it does not grow by connecting with other components, it grows by attracting new authors to its own component.

Prominent figures traction. The correlation analysis shows several high correlations involving the metric *Prominent_Figures*: (1) there is a high positive correlation between such metric and the metric *Community_Papers*, which may show that prominent figures mainly publish together people with co-authors from within the community; and (2) the metric *Prominent_Figures* is correlated with the metric *Survivors*, which may reveal that prominent figures usually publish in consecutive editions of a conference.

6.2 Analysis of Conferences according to their CORE Rank

A* conferences separate from the pack. Our analysis of variance of the metrics according to the CORE rank has revealed significant differences in the distribution of a large subset of metrics involving conferences ranked as A*. While this reveals (and confirms) a specific behavior of A* conferences when dealing with metrics individually, it also unveils that A, B and C ranked conferences behave similarly according to these metrics and therefore the CORE rank is not helping to differ them. These findings support the classification

[14] This is true even when we keep in mind that the way this variable is defined (i.e., all are considered newcomers in the first evaluated edition) makes it obvious to observe at least a slight decrease in the *Newcomers* value.

of conferences in a ranking but it does not support the need for having four different levels in such ranking.

Latent structure in our set of metrics. The factor analysis allowed us to reduce the number of metrics down to three new components which are representative of, and can be used in place of, the 12 original variables with only a 10% loss of information. The first component is mainly loaded by the variables representing the size and structure of the community behind conferences (recall *Graph_Modularity*, *Connected_Components*, *Num_Papers*, *Community_Size*, *Num_Authors*, and *Graph_Density*). The second component represents the stability of conferences (including the metrics *Survivors*, *Prominent_Figures*, *Community_Papers*, *Newcomers* and *Newcomer_Papers*). Finally the third component was only highly correlated with *Avg_Degree*.

Only the first and second components allow us to make explicit latent structures in our set of metrics, as the third component is mainly loaded by only one variable (and had some "pollution" coming from other variables with a small but not neglectable load). The discovery of these latent structures highlight the existence of the same evolution patterns in all conferences: variables in the same latent structure tend to evolve according to the same behavior in all conferences.

Key role of components in the CORE rank. The results from the regression model help us to understand which components are more relevant to discern the CORE rank of conferences. Thus, the first component does not seems to be relevant in the model. As such component is mainly related with metrics measuring the size and structure of the community, we can conclude that such metrics are actually not useful to derive the CORE rank of a conference (for instance, because their values represent a common behaviour of all software conferences). On the other hand, variables related with the second and third components (e.g., the set of growth metrics) seem to play an important role to set the CORE rank.

7 Threats to Validity

Our work is subjected to a number of threats to validity which we classify into: (1) internal validity, which is related to the inferences made based on the application of our research methodology; and (2) external validity, which discusses the generalization of our findings.

Regarding the internal validity, our data collection process relied on the DBLP dataset where 17 ranked conferences were not present (there was not matching conference names nor acronym) and 4 more were also rejected since DBLP did not include the page numbers, required as a filter in our process. On the positive side, being DBLP a curated dataset where authors, publications and venues are well-identified, there was no need to apply identity matching and merging techniques. Our rule of thumb for restricting the analysis to full papers is also a thread to validity since different conferences propose different lengths for full papers (also depending on the paper template they use) so our analysis may include papers that were in fact short papers which may slightly pollute the results.

As for the external validity, note that our results cover the CORE conference ranking list of 2014 and therefore our results may not be applicable for past or future editions. Similarly, results should not be generalized to other areas of research.

8 Related Work

Co-authorship graphs and, in general, collaboration networks have been largely studied using the network science approach (e.g., [13, 14, 25, 12]). Several works have focused on the study of co-authorship graphs in the field of computer science (e.g., [8, 4, 6, 10, 16, 1, 24]). However, to the best of our knowledge, ours is the first work focusing on the study of the full set of SE ranked conferences including their grouped behavior according to their rank value. In what follows we describe in more detail these related publications.

The work presented in [8] focuses the analysis on the co-authorship graph from only one conference (i.e., Working Conference on Reverse Engineering) and checks whether the community can be categorized as a *small world* (a special kind of graph with particular values for its clustering coefficient and characteristic path length). The works presented in [6, 10] study several metrics on the co-authorship graph representing the full set of computer science conferences (the former using DBLP and the latter Citeseer datasets), however, they do not include any metric similar to our set of growth metrics nor any rank analysis. In [1] authors provide an study of communities in DBLP where individual researcher behavior is analyzed (e.g., productivity or collaboration trends). Instead, we focus on the conference community behavior.

Other works have studied specific metrics in collaboration networks. In [4] authors analyze the co-authorship graph built out of a subset of conferences included in DBLP to assess several metrics with the aim of checking the so-called six degrees of separation phenomenon. The work presented in [16] studies the relation between the centrality of authors in the co-authorship network and the future success of their publications. An alternative perspective for measuring author impact by applying PageRank algorithm to a co-authorship network is presented in [24]. The work presented in [20] measures 11 software engineering conferences over a period of more than 10 years by means of a suite of metrics assessing dimensions such as community, openness to new authors or introversion (all similar to those included in our study)

Additional works have studied the differences between publishing in conferences versus journals in the field of computer science [5, 18, 7, 15] assessing, for instance, the impact of the publications.

Besides research publications, several datasets have been recently made available with the aim to facilitate analyzing conference data. Apart from existing online databases like DBLP, Microsoft Research, IEEE Explore, ACM

Digital Library and others, authors have proposed curated datasets such as [19], which complements the publication data from a reduced set of conferences with information on their program committee members; or [21], which provides a dataset merging information from DBLP and the ACM digital library. In this work we used the DBLP dataset as it offers the most updated information of the papers and authors we need to analyze.

Finally, there are also tools specially tailored to the analysis of individual conferences. For instance, the work presented in [21] describes ReaSoN, a comprehensive set of tools for visualizing and exploring social networks resulting from academic research. MetaScience[15] is a website which provides some metrics for conferences, authors and journals. Part of this work has reused the backend of this MetaScience infrastructure.

9 Conclusion and Future Work

In this paper we have studied the co-authorship graphs of all software conferences included in the CORE ranking list of 2014 from different dimensions. To build the graphs we have used the DBLP dataset. For replicability purposes, the extraction process, co-authorship graphs and metrics results are made available at [17].

We have analyzed each conference individually and then grouped according to their CORE ranking. We believe our results raise some interesting findings related to how the SE conference communities grow (e.g., island-based and scarce collaboration) and behave (e.g., latent structures), and that only conferences with the highest CORE rank behave significantly differently from the rest; which can be useful to rediscuss the factors that CORE takes into account when evaluating conferences. Also, these results may help to rethink the number of ranks in the CORE system. Thus, the CORE classification in four separate quality categories is not justified based on the structure and patterns found in the co-authorship graphs and that, instead, two categories would be enough (separating really good conferences from the rest) which would simplify the whole conference evaluation process.

Our results can be beneficial to both the CORE rank committee as a suggestion to rethink/complement their current CORE classification process with some of our metrics and to conference organizers and steering committees as a way to plan actions that could help improve the classification for their venues (e.g., stimulating the growth of their graph towards the values shown in A* conferences).

As further work, we would like to replicate our analysis on different CS fields and also using other ranking mechanisms such as MAS, SHINE or GII-GRIN (still work in progress but aiming at aggregating the results of CORE, MAS and SHINE), thus allowing us to better understand what community factors are most relevant to achieve high-quality conferences. We also plan

[15] http://som-research.uoc.edu/tools/metaScience/

to enrich our graph with additional information coming from other sources (e.g., program committee membership, author affiliation,...) to explore other community-based factors of good conferences. Finally, we are interested in applying our co-authorship graph analysis to conferences outside computer science where conferences may play a very different role in the publication process.

Acknowledgements We would like to thank Claudia Malpica for providing useful feedback on earlier versions of this paper.

References

1. Biryukov M, Dong C (2010) Analysis of computer science communities based on DBLP. In: ECDL conf., vol 6273, pp 228–235
2. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (10):1000
3. Dunn O (1961) Multiple Comparisons among Means. J Am Stat Assoc 56(293):52–64
4. Elmacioglu E, Lee D (2005) On six degrees of separation in DBLP-DB and more. ACM SIGMOD Record 34(2):33
5. Franceschet M (2010) The role of conference publications in CS. Communications of the ACM 53(12):129
6. Franceschet M (2011) Collaboration in Computer Science: A Network Science Approach. Journal of the American Society for Information Science and Technology 62(10):1992–2012
7. Freyne J, Coyle L, Smyth B, Cunningham P (2010) Relative status of journal and conference publications in computer science. Communications of the ACM 53(11):124
8. Hassan AE, Holt RC (2004) The Small World of Software Reverse Engineering
9. Hollander M, Douglas W, Chicken E (1999) Nonparametric Statistical Methods, 2nd Edition. Wiley-Interscience
10. Huang J, Zhuang Z, Li J, Giles CL (2008) Collaboration over time: characterizing and modeling network evolution. In: WSDM conf., pp 107–116
11. Konietschke H, Hothorn L, Brunner E (2012) Rank-based Multiple Test Procedures and Simultaneous Confidence Intervals. Electron J Stat 6:738–759
12. McKerlich R, Ives C, McGreal R (2012) The large-scale structure of journal citation networks. Journal of the American Society for Information Science and Technology 63(4):837–842
13. Newman M (2001) Scientific collaboration networks. I. Network construction and fundamental results. Physical Review 64(1):1–8
14. Newman MEJ (2000) Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. Physical Review 64:17

15. Rahm E (2008) Comparing the scientific impact of conference and journal publications in computer science. Information Services and Use 28(August 2005):127–128
16. Sarigöl E, Pfitzner R, Scholtes I, Garas A, Schweitzer F (2014) Predicting scientific success based on coauthorship networks. EPJ Data Science 3(1):9
17. Tool, graphs and metric results used in the study (2016) `https://github.com/SOM-Research/metaScience-SoftwareConferences`
18. Vardi MY (2009) Conferences vs. journals in computing research. Communications of the ACM 52(5):5
19. Vasilescu B, Serebrenik A, Mens T (2013) A historical dataset of software engineering conferences. In: MSR conf., pp 373–376
20. Vasilescu B, Serebrenik A, Mens T, van den Brand MG, Pek E, Van Den Brand MGJ, Pek E (2014) How healthy are software engineering conferences? Science of Computer Programming 89(PART C):251–272
21. Veselin G, Zhaochen G, Serrano D, Tansey B, Barbosa D, Stroulia E (2009) An environment for building, exploring and querying academic social networks. In: MEDES conf., p 42
22. Wasserman S, Hollander M (1994) Social Network Analysis. Methods and Applications. Cambridge University Press
23. Wilcoxon F (1945) Individual Comparisons by Ranking Methods. Biom Bull 1(6):80–83
24. Yan E, Ding Y (2011) Discovering author impact: A PageRank perspective. Information Processing & Management 47(1):125–134
25. Yoshikane F, Kageura K (2004) Comparative analysis of coauthorship networks of different domains: The growth and change of networks. Scientometrics 60(3):433–444
26. Zimmerman D, Zumbo B (1992) Parametric Alternatives to the Student t-test under Violation of Normality and Homogeneity of Variance. Percept Mot Skills 74:835–844

A List of Selected CORE-ranked Conferences

Conference name	Rank	Eds.
ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication	A*	31
ACM Conference on Computer and Communications Security	A*	21
ACM Conference on Object Oriented Programming Systems Languages and Applications	A*	28
ACM International Symposium on Computer Architecture	A*	41
ACM Multimedia	A*	22
ACM SIGSOFT International Symposium on the Foundations of Software Engineering	A	21
ACM Symposium on Information, Computer and Communications Security	B	8
ACM/IEEE International Conference on Distributed Smart Cameras	B	8
ACM/IFIP/USENIX International Middleware Conference	A	14
ACM-SIGACT Symposium on Principles of Programming Languages	A*	41
Annual Computer Security Applications Conference	A	25
Architectural Support for Programming Languages and Operating Systems	A*	19
ASIAN Symposium on Programming Languages and Systems	B	15
Asia-Pacific Software Engineering Conference	B	21
Automated Software Engineering Conference	A	18
Computational Intelligence in Security for Information Systems	B	8
Conference on Agile Software Development	B	10
Conference on RFID Security	C	5
Conference on Security and Cryptography for Networks	B	7
Conference on Software Engineering Education (and Training)	C	26
Conference on the Quality of Software Architectures	B	8
Dynamic Languages Symposium	B	8
European Conference on Object-Oriented Programming	A	27
European Conference on Pattern Languages of Programs	B	12
European Conference on Software Architecture	A	6
European Symposium on Programming	A	23
European Symposium On Research In Computer Security	A	19
Eurosys Conference	A	9
Foundations of Aspect-Oriented Languages	C	7
Foundations of Software Science and Computational Structures	A	17
Fundamental Approaches to Software Engineering	B	17
GI International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment	C	11
IEEE Bioinformatics and Bioengineering	C	14
IEEE Computer Security Foundations Symposium	A	8
IEEE International Conference on Global Software Engineering	C	9
IEEE International On-Line Testing Symposium	C	12
IEEE International Requirements Engineering Conference	A	18
IEEE International Symposium on High Assurance Systems Engineering	B	15
IEEE International Symposium on Performance Analysis of Systems and Software	C	14
IEEE Symposium on Field Programmable Custom Computing Machines	A	20
IEEE Symposium on Rapid Prototyping	C	10
IEEE Symposium on Security and Privacy	A*	35
IEEE/IFIP International Conference on Dependable Systems	A	15
IEEE/IFIP Working Conference on Software Architecture	A	8
IFIP Joint International Conference on Formal Description Techniques and Protocol Specification, Testing, And Verification	A	22
IFIP WG 11.3 Working Conference on Data and Applications Security	A	28
Information Security	C	14
Information Security Practice and Experience Conference	B	10
International Computer Software and Applications Conference	B	21
International Conference on Agile Software Development	B	12
International Conference on Applied Cryptography and Network Security	B	12
International Conference on Availability, Reliability and Security	B	9
International Conference on Compiler Construction	A	22
International Conference on Computer Safety, Reliability and Security	B	17
International Conference on Coordination Models and Languages	A	16

International Conference on Cryptology and Network Security	B	10
International Conference on Dependable, Autonomic and Secure Computing	C	6
International Conference on Evaluation of Novel Approaches to Software Engineering	B	8
International Conference on Formal Engineering Methods	B	16
International Conference on Functional Programming	A*	19
International Conference on Generative Programming and Component Engineering	B	13
International Conference on Information and Communications Security	B	16
International Conference on Information Security and Assurance	C	7
International Conference on Information Systems Security	B	10
International Conference on Model Driven Engineering Languages and Systems	B	10
International Conference on Model Transformation	B	7
International Conference on network and System Security	B	6
International Conference on Principles and Practice of Constraint Programming	A	20
International Conference on Principles and Practice of Declarative Programming	B	16
International Conference on Program Comprehension	C	9
International Conference on Provable Security	B	8
International Conference on Quality Software	B	12
International Conference on Reliable Software Technologies	B	24
International Conference on Risks and Security of Internet and Systems	C	7
International Conference on Software Composition	B	12
International Conference on Software Engineering	A*	35
International Conference on Software Engineering and Formal Methods	B	12
International Conference on Software Engineering and Knowledge Engineering	B	22
International Conference on Software Language Engineering	B	7
International Conference on Software Methods and Tools	B	10
International Conference on Software Testing, Verification and Validation	C	7
International Conference on Tests and Proofs	B	8
International Conference on Trust, Privacy and Security in Digital Business	B	11
International Conference on Virtual Execution Environments	A	10
International Symposium Component-Based Software Engineering	B	11
International Symposium on Automated Technology for Verification and Analysis	A	11
International Symposium on Empirical Software Engineering and Measurement	A	7
International Symposium on High Performance Computer Architecture	A*	20
International Symposium on Memory Management	A	12
International Symposium on Microarchitecture	A	42
International Symposium on Software Reliability Engineering	A	24
International Symposium on Software Testing and Analysis	A	16
International Symposium on Theoretical Aspects of Software Engineering	C	8
Joint Working Conference on Secure Information Networks: Communications and Multimedia Security	C	12
Practical Aspects of Declarative Languages	B	16
Prague Stringology Conference	C	18
Principles and Practice of Programming in Java	C	10
Product Focused Software Process Improvement	B	14
Security and Privacy for Communication Networks	A	9
Symposium On Usable Privacy and Security	B	6
Tools and Algorithms for Construction and Analysis of Systems	A	20
Usenix Symposium on Operating Systems Design and Implementation	A*	10
Verification, Model Checking and Abstract Interpretation	B	13