Community Dynamics in Open Source Software Projects: Aging and Social Reshaping

Anna Hannemann, Ralf Klamma

Community Dynamics in Open Source Software Projects: Aging and Social Reshaping

Anna Hannemann and Ralf Klamma

Advanced Community Information Systems
RWTH Aachen University
Ahornstrasse 55, 52056 Aachen, Germany
{hannemann, klamma}@dbis.rwth-aachen.de

Abstract. An undeniable factor for an open source software (OSS) project success is a vital community built around it. An OSS community not only needs to be established, but also to be persisted. This is not guaranteed considering the voluntary nature of participation in OSS. The dynamic analysis of the OSS community evolution can be used to extract indicators to rate the current stability of a community and to predict its future development. Despite the great amount of studies on mining project communication and development repositories, the evolution of OSS communities is rarely addressed. This paper presents an approach to analyze the OSS community history. We combine adapted demography measures to study community aging and social analysis to investigate the dynamics of community structures. The approach is applied to the communication and development history of three bioinformatics OSS communities over eleven years. First, in all three projects a survival rate pattern is identified. This finding allows us to define the minimal number of newcomers required for the further positive community growth. Second, dynamic social analysis shows that the node betweenness in combination with the network diameter can be used as an indicator for significant changes in the community core and the quality of community recovery after these modifications.

1 Introduction

There are about 300,000 OSS projects registered in `sourceforge.net`, but only few of them succeed [6]. Most of the Open Source Software (OSS) projects are started by a very small group of people bound by a goal they want to approach with the project. Later on, successfully developing projects gain a community of peripheral developers, bug fixers, bug reporters and peripheral users. This project community needs to achieve a critical mass of people for the project breakthrough. The meaning of OSS community is multifold, e.g. community members bring new ideas to the project, present a kind of social reward for the developers effort and increase the "market shares" of the project by spreading the word [9], [13], [17]. Considering the voluntary nature of OSS development, the sustainability of an OSS project depends on the sustainability

of its community. The analysis of the OSS community evolution can be used to extract indicators to rate the current stability of community and predict its possible future development.

The study of the OSS movement in general and OSS development principles in particular have evolved to a separate research field of community-intensive socio-technical projects [12]. Plenty of those studies address the evolution of OSS systems [2]. However, the dynamic analysis of the OSS communities - a social component of OSS projects - is seldom. The existing research papers on OSS community dynamics either present a set of static measurements for a certain cut off of a project history [14], [5] or are restricted to the developer sub-communities only [10], [1]. In this paper, we combine the demographic analysis of an OSS project community as an aging population and dynamic analysis of an OSS as a social structure. We apply our approach on the whole communities of three bioinformatics OSS. The selected projects, BioJava, Biopython and Bioperl, are very similar in their goals, scientific communities and infrastructures (all three supported by "The Open Bioinformatics Foundation"). Thus, we hope to overcome the bias of results in case of too different communities in terms of policies, culture, lifetime, domain, organization of the OSS projects.

The rest of the paper is organized as follows. Section 2 presents an overview of the existing OSS community studies: statistical studies of community dynamics are described in Section 2.1 followed by an overview of OSS social evolution studies in Section 2.2. In Section 3 we present our approach to analyze OSS community evolution and the data used for validation. The results are described in Section 4. Section 5 discusses the results and concludes the paper. An outlook is given in Section 6.

2 Related Work

A large number of studies was executed upon publicly available communicational and development repositories of OSS projects during the last decade [12]. Many of those studies address the evolution of OSS system. In the systematic literature review Breivold et al. [2] identify four main research topics on OSS system evolution: software trends and patterns, evolution process support, evolvability characteristics addressed in OSS evolution, examining OSS at software architecture level. The researchers provide an overview of metrics that are used to analyze OSS evolution over time: Software growth metrics, system growth metrics, etc. address only the technical aspects of OSS projects. However, the success of an OSS project depends not only on technical quality of the developing system, but also on the social state of its community [17]. The attention of researchers is also attracted to the analysis of the OSS communities: motivation for voluntarism, participation and interaction patterns, social structure, etc. However, the dynamic analysis of the OSS communities is seldom. No metrics for measuring social quality of an OSS project are developed so far. In Section 2.1 we give an overview of the studies, which address evolution of the

community composition. While in Section 2.2 the existing investigations of the community restructuring in social terms are presented.

2.1 Population Evolution

In [17] Ye et al. present a conceptional framework of the OSS evolution. An OSS community is defined as an example of a community of practice (CoP) with the legitimated peripheral participation (LPP) [15]. According to the LPP concept, through continuous learning the newcomers become experienced community members, thus, they move from the periphery to the community center. Ye et al. call this process "role-transformation" and, thereby, extend the static onion-model of the OSS communities by time dimension. Role-transformation in open source leads to evolution of community social structure and composition, which in turn results in evolution of developer skills and organizational principles. The authors also define a term "second generation", which is achieved, when an OSS community core is evolved from a single project leader to a group of core members.

Von Krogh et al. in [14] study the early stage of community establishment in the Freenet project (year 2000). The researchers investigate which behavioral patterns (level of activity and specialization) increase the chances to be granted developer privileges (role-transformation). However, this study is restricted to one OSS project in the early stage.

In contrast to [14], in [5] Jensen et al. study the joining behavior across four different OSS projects. The projects are analyzed not at their early stage, but when they were already widely acknowledged and supported by a bigger community. The authors estimate a "survival rate" of newcomer in the mailing lists: 9.4% of those, who entered the project in three month period (643), were still participating in mailing lists after six month period. However, only 9 month of the projects' history are taken into consideration.

Robles et al. in [11] investigated the meaning of evolution within the Debian project. The finding, that if a package leader leaves a project, the package is very likely to be abandoned in the future, shows the importance to understand and even predict the community restructuring.

In [10] Robles et al. use the term "generation" to describe the projects, where the core developers change over time. The results show, that the core remains stable in very seldom cases (3 of 21 projects) and support the expected strong evolution of the leading group and constant need for the emerging gaps to be filled. However, the study is restricted not only to the developers, but even to the core group of them (the most active 20% of committers).

To summarize, the above described studies consider OSS communities as a population: concepts like "generation", "survival rate", "migration" are applied. Demographic methods and models present one possible basis for quantitative analysis of OSS community evolution.

2.2 Social Dynamics

Beside quantitative analysis of an OSS community, its social state can be estimated. Hereby, an OSS community is mapped to a graph. The nodes of the graph represent OSS project members and the edges between the nodes represent interaction between the project members. Plenty of social network analysis measures can be calculated using OSS community graph. Similarly, for each project member his/her social status can be estimated.

In [1] Bird et al. study the chances of migration from non-developers to full developer among others as a function of social status. Analysis of Apache, Postgres and Python communities shows, that the meaning of different aspects vary significantly from project to project. The evolution of newcomers is not considered.

Further, the dynamic of social characteristics is addressed in [4] by Howison et al. Using the data from `sourceforge.net` bug-tracker from 120 different projects, social networks based on direct interaction on submitted bugs are depicted. In order to analyze the evolution of outdegree centralization, the data is sampled in 90-day overlapping windows. Strong variation in community social structure is detected across different projects and within one project over time. The participation behavior proves to be distributed according to power-law: most of the project members join the project for a short period of time. However, the study considers only a relatively short period of project life time.

In [16] Wiggins et al. adapted the analysis methods from [4] and applied to investigate the centralization dynamics of Gaim and Fire. In this study, the significant evolution of communication centralization is showed. For example, a project management activity can reshape the community to a highly centralized network structure.

To summarize, there is a growing interest in OSS community evolution. Monitoring of community social state can be applied in order to detect important internal/external events and thus, to approach sustainability of OSS communities. Both demographic and social network analysis methods and concepts are applied to analyze OSS communities dynamically. However, most of the existing studies are mainly concentrating on the migration from non-developers to the developers. The whole community is rarely addressed. Often only a short cut off of the project history is used for analysis. To our knowledge, the only studies which combine the community statistical and social evolvement are [4] and [16].

3 Methods

In this study we approach an OSS community as an aging population on the one hand and as a social structure on the other hand. We adapt methods of population projection and dynamic social network analysis and apply them to three bioinformatics OSS projects.

3.1 Data

In this study, we use the data from three well-established bioinformatics OSS projects. Bioinformatics is an interdisciplinary research field, where innovative computer science techniques and algorithms are applied to answer emerging research questions of computational biology. There is a branch of commercial bioinformatics applications. However, according to [7]"most of them are not scientific for the level of data analysis required in bioinformatics research. It was partly the frustration with commercial suits that drove the foundation of the Bio* groups." All open source projects used in this study, BioPerl, BioJava, Biopython, belong to the Open Bioinformatics Foundation. The selected projects are very similar in the problems they address, the community they are intended for, policy and organizational issues they experience. The infrastructure used for the project management all cases is composed of a wikipage, mailing lists and a code repository.

The communication data from the project mailing lists and the development history from the project code repositories for the period of eleven years ($2000-2010$) is crawled, filtered from spam and stored in a local database [3]. Multiple aliases of same individuals are semi-automatically detected and consolidated. We detected 5507 distinct users 3259 of them had written more then one posting and had got at least one reply. The mailing list aliases are mapped to the developer aliases. Further insights in the project history, we collected from the project wikipages and project participants via private emails.

3.2 Analysis Procedures

For our study, we monitored the population evolution over time in combination with the changes in social structure of OSS communities. The data was divided in equal one-year-long periods $\{01.01.x - 31.12.x \| \forall x \in (2000, 2010)\}$.

Population Ecology In order to study the evolution of OSS communities, we defined the population characteristics in the following way:

Year of Birth is a time point $t_{0_{p_i}}$ when a participant p_i entered an OSS project. In the context of this study it is a time point, when a user has written his/her first posting in a project mailing list.

Age Group $(x; x+1)$ at time t consists of all active project members, who participate in a project at the given time point for at least x and at most $x+1$ years. In context of this study, a user is defined to be currently active in a project, if he/she has written at least one posting in a mailing list of a project in the current year.

Survival Rate $(x; x+1) \rightarrow (x+1; x+2)$ is a percentage of active users in the last year in the *age group* $(x; x+1)$, who are still active in the current year.

For example, in 2006 all project participants who posted their first post in the project mailing list not earlier than 01.01.2006 belong to the $(0, 1)$ age group. In turn, those, who have posted their first post before 01.01.2006, but not earlier than 01.01.2005 and at least one post in the current (2006) year form the $(1, 2)$ age group. As the earliest data set we consider originates from 2000 for all three projects BioJava, Biopython and BioPerl, the "oldest" possible project participants in year 2006 present $(4; 5)$ age group. In year 2007 the survival rate $(0, 1) \rightarrow (1, 2)$ presents a percentage of users from $(0, 1)$ age group in 2006, who are still active.

In order to visualize the population age structure, the population pyramids are applied. The population pyramids present an effective graphical way to visualize the population development and to detect some tends and outliers, which can lead to some environmental and historical events. It can be also help to indicate the likelihood of continuation of population under study. The X-axis of a population pyramid represents age or age-groups, while the numbers of people in each age group is plotted along the Y-axis.

Social Network Analysis (SNA) is applied to study the social characteristics of an OSS community modeled as a graph. Individual participants of a community are modeled as nodes of the graph and their relationship (friendship, family relatedness, etc.) is reflected by network ties. The BioJava, Biopython and BioPerl participants are mapped to the nodes V of the project networks. If at least one thread exists, to which both participants v_i and $v_j \in V$ have submitted at least one posting, the link $(v_i; v_j)$ is added to the graph. The edges are binary: either there is a link or not. To analyze the project networks we applied the following SNA measures[1] [8]:

- *Shortest Path* σ_{st} is the minimal length of the path between two nodes s, t
- *Diameter* is the length of the longest shortest path $d = \max_{s,v \in V} \sigma_{st}$
- *Node Betweenness* is the fraction of shortest paths between two nodes s and t that contain node v_i

$$g(v_i) = \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

- *Largest Connected* identifies the maximal connected components of a graph
- *Density* is the ratio of the number of edges to the number of possible edges
- *Transitivity* (=Clustering Coefficient) measures the probability that the adjacent nodes of a node are connected

[1] `http://www.r-project.org/` is used for calculations

– *Edge Betweenness Clustering* is a method to detect dense interconnected nodes subsets (communities) within social networks with sparse connection to outside of the cluster.

Dynamic network analysis (DNA) extends SNA with the time domain. To analyze the development of social characteristics of the OSS projects over time, we generated the project networks for each year.

4 Results

The following sections present the results of the previously described dynamic analysis procedures applied to three bioinformatics OSS.

4.1 Demographic Forecast

For each project under study, a number of distinct users in each age group $(x; x+1)$ per year was calculated. Starting from the year 2006, we estimated the survival rates for users of each age group. For example, for 2006 the following survival rates were calculated $(0,1) \rightarrow (1,2), (1,2) \rightarrow (2,3) \ldots (3,4) \rightarrow (4,5)$. Accordingly, for year 2010 the survival rates of the oldest project members are represented in $(7,8) \rightarrow (8,9)$. On average, our investigations showed a pattern of the survival rates for certain age groups in all three bioinformatics OSS (cf. Figure 1). A ratio P_x of the project participants aged x to $x+1$ at time t being still active in the age group $x+1$ to $x+2$ at time $t+1$ follows certain rules:

– $P_0 = [(0,1) \rightarrow (1,2)] \approx 20\%$

– $P_1 = [(1,2) \rightarrow (2,3)] \approx 40\%$

– $P_m = [(x, x+1) \rightarrow (x+1; x+2)] \approx 90\%$, $\forall x > 1$

The results showed, that 20% of people, who were newcomers in the year n, remain active in the year $n+1$. Out of those, who remained with a project already for one year ($(1,2)$ age group) in the year n, there remain only about 40% in the year $n+1$. Other age groups have a survival rate of about 90%. The population pyramids of BioJava, Biopython, BioPerl in year 2010 in Figure 1 provide a visual representations of the identified pattern. To summarize, the distribution of survival rates in the investigated OSS projects follows the power law. Additionally, a phenomena of *"rebirth"* can be observed in the OSS projects. Some project participants leave the project for several years and after some period of time come back to the community. In the years of their absence, these users do not appear in our measurements. However, when they reactivate their participation, we still consider the date of the first posting as the date of the entrance into the project. These users represent no newcomers anymore, as they already have some experience with the project. Therefore, it

can happen, that the age group $(x+1; x+2)$ contains more people, than the age group $(x; x+1)$ contained a year before. Thus, especially for older age groups a survival rate higher than 100% is possible.

This finding leads to the conclusion, that if a user was actively participating in a project for more than two years, he/she will probably get "attached" to it on long-term bases. In turn, the percentage of those who survive over two years is very low. Based on the identified pattern, a minimal number of newcomers required to support the same level of participation and, thus, the continuation of project population in the next year can be estimated as follows:

$$|newcomer|_{t+1} >= |(0;1)_t| * 0.2 + |(1;2)_t| * 0.4 + \cdots + |(x;x+1)_t| * 0.9 \quad (1)$$

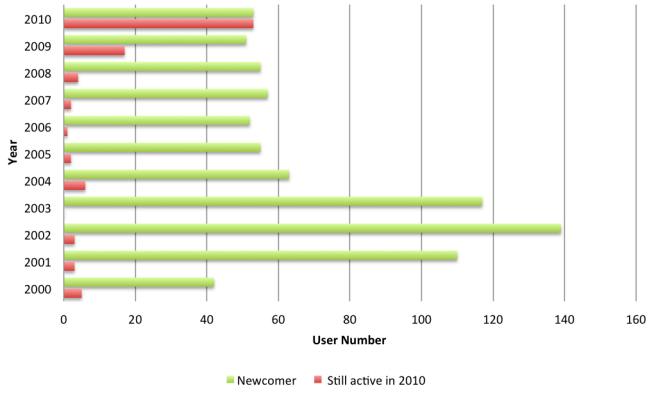
The history of newcomer numbers can be investigated in order to uncover the events, which influenced the rate of user inflow in a project. In Figure 1 the newbie numbers in all three projects under study are illustrated. The highest inflow of new users in all thee projects can be indicated during $2001 - 2004$. This observation can be linked to the fact, that in those years bioinformatics won a lot of attention due to the announcement of sequencing of the human genome on June 26, 2000[2]. Also the "Bioinformatics Open Source Conference" started in year 2000, attracting the attention of scientists to the open source software for computational biology. We can conclude, that the newcomer rates depend among others on the events within the project domain outside the project community.

Despite the rise of newbie numbers in all three projects during the mentioned time period, the absolute numbers differentiate considerably in BioJava (over 100), Biopython (less than 100) and BioPerl (over 250). In turn, different newbie numbers result in different absolute numbers of users, who get involved in the project on long-term bases, even if the percentage of survival is almost the same (cf. Figure 1). This occurrence can be linked to the different development stages of three projects at the mentioned time period. While BioJava and Biopython were started at around 1999, the BioPerl has been already developed since 1995. Hence, in early years of 2000, the BioPerl was the most-established project in comparison to the other two and could attract more people, even that the topics addressed by all three are quite similar. Thus, we observe interplay among similar OSS project: "the rich get richer" effect.

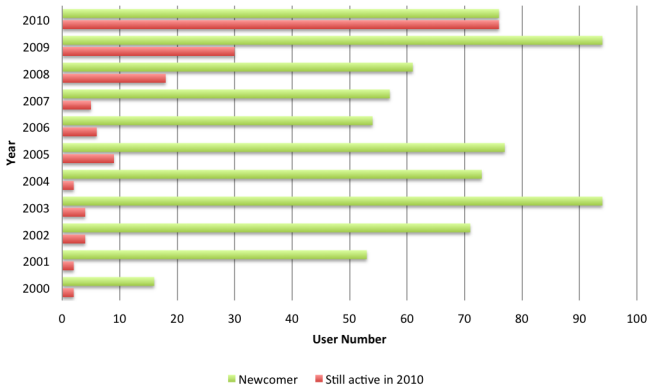
4.2 Social Evolution

For every year and for each project we generated a network of the currently active project participants. For every of these networks six SNA measures were calculated: diameter, average path length, maximal betweenness, size of the largest connected component, density, transitivity. Some remarkable outlier values in diameter and maximal betweenness series are identified in Biopython and BioJava projects (cf. Figure 2).

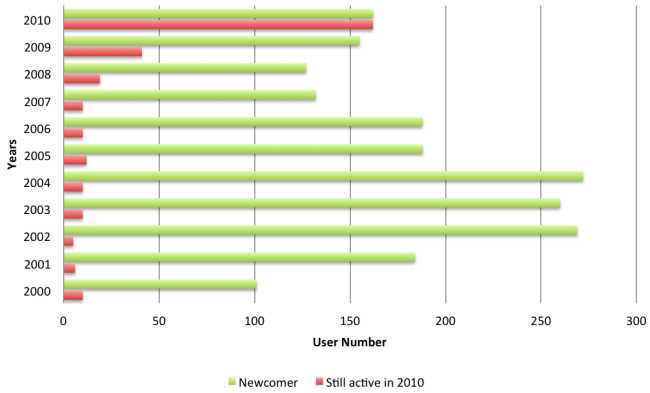
[2] `http://www.ornl.gov/sci/techresources/Human_Genome/project/clinton1.shtml`



(a) BioJava

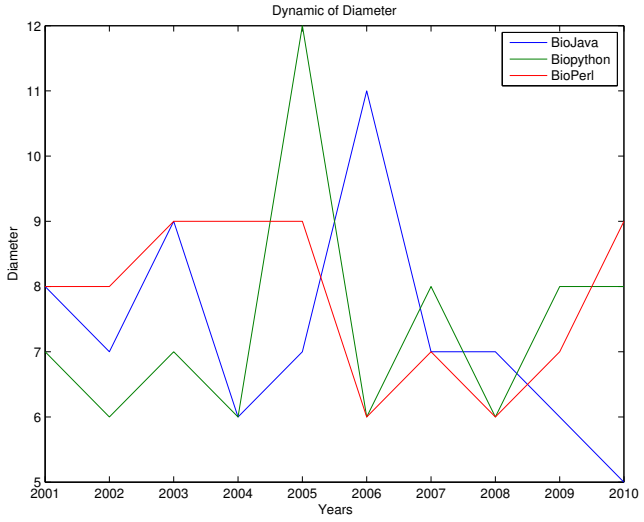


(b) Biopython

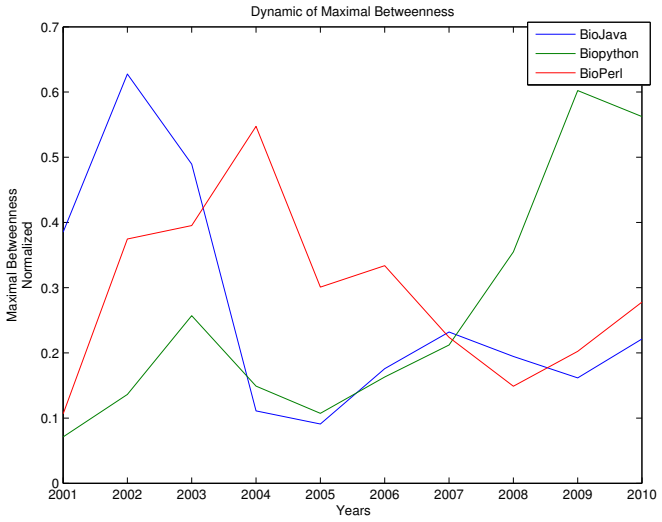


(c) BioPerl

Fig. 1. Newcomers vs. Survived Users

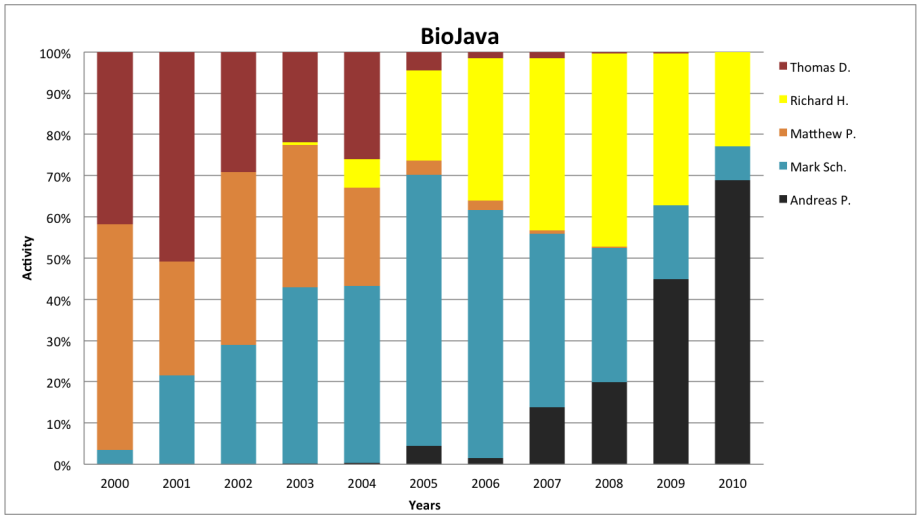


(a) Dynamic of Diameter

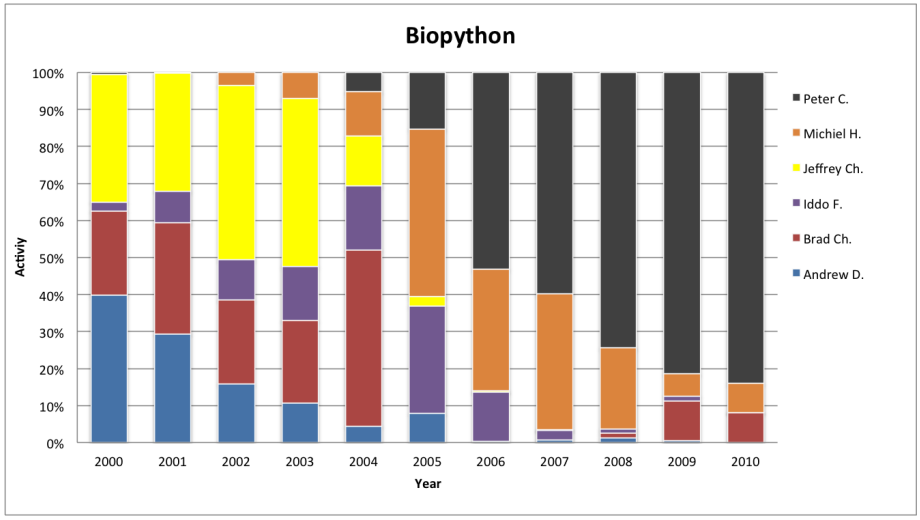


(b) Dynamic of Max Betweenness

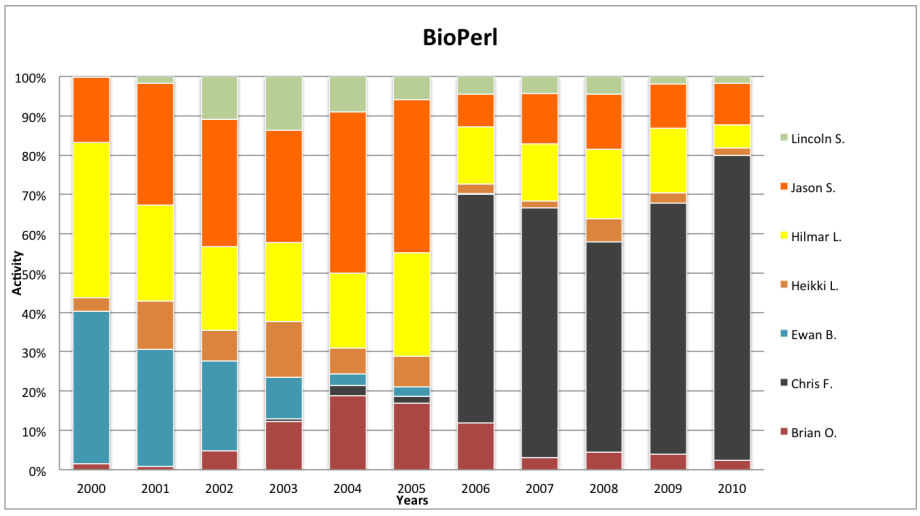
Fig. 2. Evolution of Structural Parameters of the Project Social Networks



(a) BioJava



(b) Biopython



(c) BioPerl

Fig. 3. Participation Activity over Years for Selected OSS Members

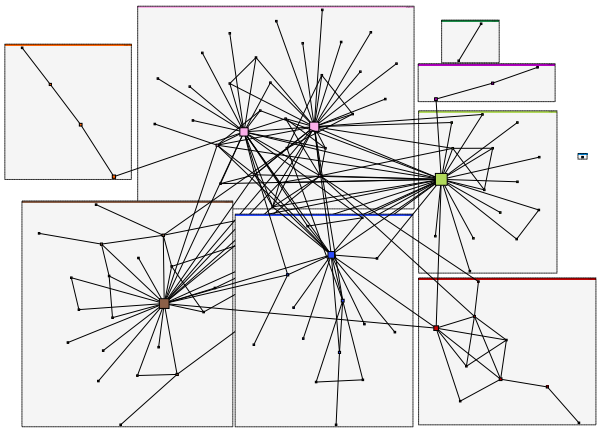
In most cases, the diameter value within a network was 6, which is consistent with the well-known small-world phenomena of the social networks [8]. In 2005 in the Biopython network and in 2006 in the BioJava network, the diameter values reached 12 and 11 respectively. In these periods, the networks do not show previous compactness. Noteworthy are the low values of the maximal node betweenness at the same period of time in both projects. Figure 4 shows social networks of the BioJava, Biopython and BioPerl communities in 2006, 2005 and 2004 respectively. The presented networks are clustered using edge betweenness clustering provided by `www.yworks.com`. The diameter of the node representations reflects its social importance in the network.

Node betweenness is a centrality measure, which determines dominance of a node within a network. Assuming that the information flow takes the shortest path, the node betweenness let us estimate the fraction of information going through a node. However, the information does not always flow along the shortest path and, therefore, the assumption presents only an approximation. Nevertheless, this approximation allows us to estimate quite well the substantial influence of the network nodes. Nodes with high betweenness values often present an interlink between network clusters (community subgroups). Thus, the low value of maximal betweenness can be an indicator for that a central node looses its influences or leave the network. Therefore, we identified the most central and active project members in BioJava, Biopython and BioPerl for each year.

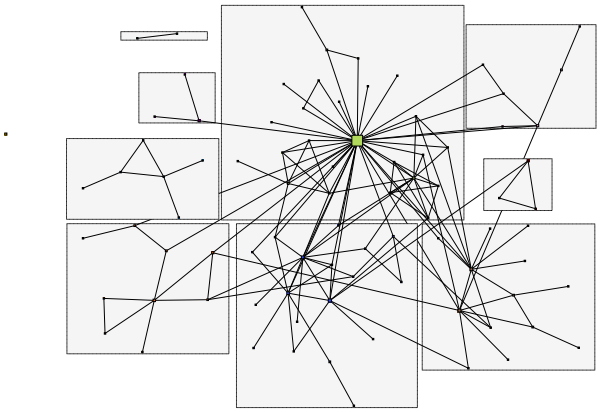
We detected a change of "main actors" within the Biopython community in 2005: Jeffrey C., Andrew D. and Brad C. got "substituted" by Peter C. and Michiel H. However, this takeover was not very smooth. Figure 3(b) presents contribution level of each central member in each year. First after Jeffrey C. and Brad C. had already reduced remarkably their input to the project, Andrew D. and Michiel H. brought the project to its previous progress state. This could be a reason for low maximum betweenness and diameter values. The Biopython network was almost breaking apart, when its core members left the community (cf. Figure 4(a)).

In the BioJava community there were three central members until 2004: Thomas D., Matthew P. and Mark Sh. In 2005, two of them, Thomas D. and Matthew P., left the project (cf. Figure 3(a) and Figure 4(b)). This period is marked by low maximal betweenness and low value of transitivity, but by almost "normal" diameter value of 7. Hence, a community shrunk due to user "retirement", but it remained joined by the third central member Mark Sh. (who remained in the project from the beginning until present). In2006, many new active "actors" entered the BioJava community. Hereby, the community got less centralized, resulting in a higher diameter value. Later (around 2007), Andreas P. and Richard H. gained the central role in the project. The community again presented a hierarchical, very centralized structure with small diameter and high maximum betweenness values.

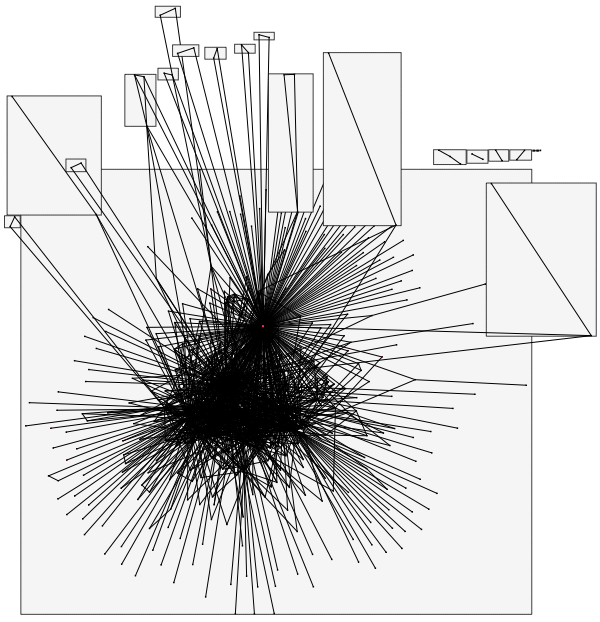
In BioPerl, in $2003 - 2005$, the diameter value raises only up to 9. The maximal betweenness values during the period are very high. This period overlaps



(a) Biopython, year 2005



(b) BioJava, year 2006



(c) BioPerl, year 2004

Fig. 4. Project Social Networks

with the highest newbies inflow in BioPerl (cf. Figure 1(c)), which resulted in the community expansion (and thus, in higher diameter values). The power within the community stayed in the hands of the same leading people. The community expansion just increased their "power" (resulting in increase of maximum betweenness). There was also a switch in roles among the "main actors" in BioPerl (cf. Figure 3(c)). Until 2006 the maximal node betweenness is gained by Jason S. In 2004, Chris F. enters the BioPerl and achieves the maximal betweenness in the 2006. However, Jason S. continued to contribute to the project actively.

The core of BioPerl is much bigger than in the other two projects. There are on average 24 active distinct developers in BioPerl, while Biopython and BioJava are supported on average by 7 and 11 respectively. A more detailed investigation of the BioPerl community shows, that in contrast to Biopython and BioJava, where only core (very active and socially central actors, experts) and periphery (very passive actors, lurkers) are present, an intermediate layer of "contributors" has been established. Although the project members of this layer put much less effort, than the core, they still provide some active contributions to the project. The edge betweenness clustering of BioPerl network in 2004 detects one very big cluster, which includes almost all project participants (cf. Figure 4(c)). The intermediate layer of active contributors can be a reason for the strong community interconnection and better resistance against "retirement' of core members

4.3 Social Evolution and Demographic Forecast

In Figure 1(b) an increase of the newcomer rate in the year 2009 in Biopython can be observed. At the same time, the rise in commits and releases number per year can be detected starting from 2007 (cf. Figure 5). More detailed investigations show that these changes in release- and effort-culture were introduced by the new leading people in the Biopython community (cf. Figure 3(b)). This organizational and development modifications made the Biopython project more attractive for the newcomers.

5 Discussion and Conclusions

In this paper, we adopted demographic concepts to analyze OSS communities as an aging population and applied several SNA measures to trace social evolution of OSS communities. A survival rate pattern $20-40-90\%$ was identified within the communities of three bioinformatics projects. Only 20% of the newbies "survive" over their first year in a project, the 40% out of them over the second year followed by about 90% of the previous amount to survive in the next years. This pattern leads to the following conclusions:

– The identified pattern allows to predict the minimal number of newbies required to support the same level of participants in the community.

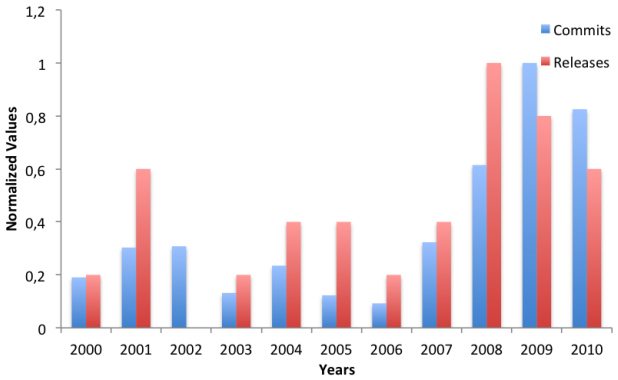


Fig. 5. Biopython: Release and Commit Numbers

- There is a very high probability, that a user who remained with an OSS project longer than two years, will remain with the community further.
- The fraction of the users, who "survives longer" than three years is only about $7,2\%$. The very low survival rate is conform to the results presented in [5].
- Within ten years of the project history no maximal possible participation duration was identified, causing the continuous community growth even with slightly decreasing newcomer rates.
- The core group of each OSS project evolves strongly (conform to the results from [10]).
- Retirement of a central community member(s) presents a danger for the project sustainability (conform to the results from [11]).
- There is a phenomena of "rebirth" within an OSS community. Especially those, who get involved deeply in the project for several years and then left it, tend to return to the project later on.
- The number of "oldies" gets continuously bigger. This can lead to seclusion of community against newcomers. The concept of "contribution barrier" from [14] should be extended by social aspects.

The SNA results show, that the combination of increasing diameter and falling maximal betweenness can be used as an indicator for the retirement of the central community member, with a risk of a community to break apart. In the history of all three projects there was a change of the central person within community in about $5-6$ years. In the BioPerl project the change seems to have no strong effect: the community participants remained strongly interconnected, due to the relatively big and well-developed hierarchical community center. On contrary, the Biopython and BioJava communities show a very loosely structure at the period of the change. BioJava project seems to execute the change more smoothly than Biopython, thanks to the overlap of the central user participation time. Many other active members left the community together with the central actors. The both Biopython and BioJava communities experienced the

great restructuring. In Biopython we also observe a complete modification of the development principals. The findings confirm the OSS problem, that the knowledge concentrated in the core of community bringing the danger of its total loss, if the core members leave the project. Especially, considering that the retirement of a central member can induce the further outflow of project members from an OSS project.

Our findings indicate, that a combination of maximal betweenness and diameter values, can be used as metric for measuring social stability of an OSS community. Survival rate and newcomer inflow can be applied in order to detect the important internal/external events.

5.1 Threats to Validity

The presented findings may not be directly transformed to all OSS projects. The bioinformatics OSS projects are mostly driven by bioinformatics scientist, mainly PhD students working on their thesis. Once they finish their PhD, they may loose the interest or/and time for the contributing, which can be one reason for the observed survival pattern. Further, the quality of any dynamic analysis may be influenced by the selected step size. Until now, we performed the population analysis on BioJava communication data cut at the time point of each release. The achieved results are very similar with those presented in this paper. However, there is about one release per year in BioJava. The survival pattern in the projects with another release culture has to be investigated.

6 Future Work

Considering socio-technical nature of OSS projects, social evolution of OSS communities presents a big area for further studies. To validate the results of this study, the proposed measurements should be applied to other OSS projects. Moreover, there is a great deal of possibilities to extend the proposed methods for dynamic analysis of OSS communities by additional parameters. For example, the analysis of participation duration can be combined with the information about participant's activity. For each OSS project member we can define a time series: a sequence of contribution numbers within uniform time intervals (e.g. per month). Using statistical methods like Principle Component Analysis, we can detect different "activity-participation duration" patterns.

References

1. Christian Bird, Alex Gourley, Prem Devanbu, Anand Swaminathan, and Greta Hsu. Open borders? Immigration in open source projects. In *Proceedings of the Fourth International Workshop on Mining Software Repositories*, MSR '07, pages 6–13, Washington, DC, USA, 2007. IEEE Computer Society.

2. Hongyu Pei Breivold, Muhammad Aufeef Chauhan, and Muhammad Ali Babar. A systematic review of studies of open source software evolution. In *Proceedings of the 17th Asia PacificSoftware Engineering Conference (APSEC)*, pages 356–365, November 2010.
3. Anna Hannemann, Michael Hackstein, Ralf Klamma, and Matthias Jarke. Adaptive filter-framework for quality improvement of open source software analysis. In *Software Engineering 2013*, 2013. to appear.
4. James Howison, Keisuke Inoue, and Kevin Crowston. Social dynamics of free and open source team communications. In *Proceedings of the IFIP Second International Conference on Open Source Systems*, pages 319–330. Springer, Lake Como, Italy, 8-9 June, 2006.
5. Carlos Jensen, Scott King, and Victor Kuechler. Joining free/open source software communities: An analysis of newbies' first interactions on project mailing lists. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1 –10, January 2011.
6. Greg Madey, Vincent Freeh, and Renee Tynan. The open source software development phenomenon: An analysis based on social network theory. In *Americas Conference on Information Systems (AMCIS2002)*, pages 1806–1813, Dallas, TX, USA, 2002.
7. Harry Mangalam. The bio* toolkits–a brief overview. *Briefings in Bioinformatics*, 3(3):296–302, September 2002.
8. Mark I. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.
9. Eric S. Raymond. *The Cathedral and the Bazaar*. O'Reilly Media, 1999.
10. Gregorio Robles and Jesus M. Gonzalez-Barahona. Contributor turnover in libre software projects. In Ernesto Damiani, Brian Fitzgerald, Walt Scacchi, Marco Scotto, and Giancarlo Succi, editors, *Open Source Systems*, volume 203, pages 273–286. Boston: Springer, 2006.
11. Gregorio Robles, Jesus M. Gonzalez-Barahona, and Martin Michlmayr. Evolution of volunteer participation in libre software projects: Evidence from debian. In Marco Scotto and Giancarlo Succi, editors, *Open Source Systems*, pages 100–107, July 2005.
12. Walt Scacchi. The future research in free/open source software development. In *Proceedings of ACM Workshop on the Future of Software Engineering Research (FoSER)*, pages 315–319, Santa Fe, NM, November 2010.
13. Eric von Hippel and Georg von Krogh. Open source software and the "private-collective" innovation model: Issues for organization science. *Journal on Organization Science*, 14(2):208–223, March 2003.
14. Georg von Krogh, Sebastian Spaeth, and Karim R Lakhani. Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32(7):1217–1241, 7 2003.
15. Etienne Wenger. *Community of Practice: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge, 1998.
16. Andrea Wiggins, James Howison, and Kevin Crowston. Social dynamics of floss team communication across channels. In Barbara Russo, Ernesto Damiani, Scott Hissam, Björn Lundell, and Giancarlo Succi, editors, *Open Source Development, Communities and Quality*, volume 275 of *IFIP International Federation for Information Processing*, pages 131–142. Springer Boston, 2008.

17. Yunwen Ye, Kumiyo Nakakoji, Yasuhiro Yamamoto, and Kouichi Kishida. The co-evolution of systems and communities in free and open source software development. In Stefan Koch, editor, *Free/Open Source Software Development*, pages 59–82. Idea Group Publishing, Hershey, PA, 2004.