

# Coding Together at Scale: GitHub as a Collaborative Social Network

Antonio Lima, Luca Rossi and Mirco Musolesi

School of Computer Science  
University of Birmingham, UK  
{a.lima,l.rossi,m.musolesi}@cs.bham.ac.uk

## Abstract

GitHub is the most popular repository for open source code (Finley 2011). It has more than 3.5 million users, as the company declared in April 2013, and more than 10 million repositories, as of December 2013. It has a publicly accessible API and, since March 2012, it also publishes a stream of all the events occurring on public projects. Interactions among GitHub users are of a complex nature and take place in different forms. Developers create and fork repositories, push code, approve code pushed by others, bookmark their favorite projects and follow other developers to keep track of their activities.

In this paper we present a characterization of GitHub, as both a social network and a collaborative platform. To the best of our knowledge, this is the first quantitative study about the interactions happening on GitHub. We analyze the logs from the service over 18 months (between March 11, 2012 and September 11, 2013), describing 183.54 million events and we obtain information about 2.19 million users and 5.68 million repositories, both growing linearly in time. We show that the distributions of the number of contributors per project, watchers per project and followers per user show a power-law-like shape. We analyze social ties and repository-mediated collaboration patterns, and we observe a remarkably low level of reciprocity of the social connections. We also measure the activity of each user in terms of authored events and we observe that very active users do not necessarily have a large number of followers. Finally, we provide a geographic characterization of the centers of activity and we investigate how distance influences collaboration.

## 1 Introduction

In recent years, GitHub<sup>1</sup>, a hosting platform for software projects, has gained much popularity among a large number of software developers around the world. This platform offers version control hosting, as other platforms have done in the past (e.g., SourceForge<sup>2</sup>, Assembla<sup>3</sup>, BitBucket<sup>4</sup>). How-

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup><https://github.com/>

<sup>2</sup><https://sourceforge.net/>

<sup>3</sup><https://www.assembla.com/>

<sup>4</sup><https://bitbucket.org/>

ever, this service has much emphasis on its social features, as summarized in its motto “GitHub: social coding”. In fact, GitHub is not simply offering a code hosting service, like its competitors had been doing for a long time, but also an easy-to-use and cheap (or even free in its basic version) online tool for collaborative software development and many features supporting the community of developers. For all these reasons, GitHub has successfully lowered the barrier to collaboration in open source. The importance of this collaboration platform seems to be increasing, as its founder has plans to extend the use cases beyond software development (Lunden 2013). At the same time, most of the data concerning collaboration on public<sup>5</sup> software repositories can be accessed and analyzed. This represents a unique opportunity to study aspects of human behavior related to collaboration at scale.

GitHub is based on the Git revision control system<sup>6</sup>. In GitHub a user can *create* code repositories and *push* code to them. Every repository has a list of *collaborators*; they can make changes to the content of the repository and they review the contributions that are submitted to the repository, accepting or discarding them. In this sense, they are not the only people collaborating on the project. In fact, every person that wishes to contribute to a project, without being a collaborator, can *fork* it<sup>7</sup>. This action makes a duplicate of the repository, allowing developers to work independently, *committing* changes only to their own fork. When developers complete a certain task (e.g., a new feature or a bug fix), they can send the changes to the original repository, through a so-called *pull request*. Then, a collaborator of the original repository reviews the changes contained in the pull request and decides whether to accept it in the original repository (in the Git jargon, *merge* it to the parent repository), or refuse it, optionally motivating his or her choice. Once the new code is accepted in the original repository, its author becomes one

<sup>5</sup>GitHub also offers fee-based private repositories. Since it is not possible to access any information about private repositories, our analysis will focus on public repositories.

<sup>6</sup><http://git-scm.com/>

<sup>7</sup>In the open-source context, the term *forking* had a negative connotation in the past, i.e., it has often been used to refer to groups of developers separating for different views on a project and splitting their forces on independent projects. Instead, in GitHub, forking is a normal part of the process of contributing to a project.

of the *contributors* of the project. In addition to that, GitHub users can *follow* other users, to be notified of their actions. The website is not used only for collaboration, but also as a resource to find quality software. Users can *star* interesting repositories that they want to bookmark for later reference. Other features are also available (e.g., issue tracking, downloads, gists, and so on) but we will not consider them in this work.

In this paper, to the best of our knowledge, we present the first in-depth quantitative analysis of GitHub, as a unique example of large-scale real-world collaboration platform mainly used for software projects. The contributions we make in this paper can be summarized as follows:

- We conduct basic structural analyses and we show that the distributions of the number of contributors and watchers per project and followers per user show a power-law-like shape.
- We analyze social ties and repository-mediated collaboration patterns. We find a very low reciprocity of the social ties, which is remarkably different from the findings of studies of other types of social networks.
- We study the depth and width of the trees corresponding to forked repositories and we observe that in GitHub collaboration on forks seems to happen on a limited number of key projects.
- We investigate the correlation between the activity of users and their popularity in the network and we observe that very active users do not necessarily have a large number of followers.
- We provide a geographic characterization of activities and collaborations. We find that users tend to interact with people they are close to and that repositories with a low number of collaborators tend to have them concentrated around a few specific geographic locations, rather than scattered around the world. Finally, we observe a similarity between the geographical distributions of following and contributions ties.

The paper is organized as follows. We first discuss related work in this area. Then, we present the data collection methodology and we describe the characteristics of the dataset. We study the networks representing the interactions between entities, extracted from the dataset. We conclude the article by discussing our key findings and outlining our future work.

## 2 Related Work

Several researchers from different communities have been interested in analyzing behavior on websites and online tools that enable large-scale collaboration, most notably Wikipedia. Indeed, a large body of research has focused on understanding how people coordinate their collaboration efforts in the constant update and expansion of the crowd-sourced online encyclopedia through a variety of methodologies (see for example (Kittur et al. 2007; Vuong et al. 2008)). A relevant approach in relation to the topic of this work is the network analysis of the collaboration structure in Wikipedia presented in (Brandes et al. 2009). More in

general, open-source projects have been the subject of several studies specifically aimed at uncovering the social structure that emerges from the interactions between developers (Valverde and Solé 2007; Bird et al. 2008) and at analyzing the individual contributions to specific projects (Hindle, German, and Holt 2008).

Recently, given its increasing popularity, there has been a surge in interest in GitHub and its underlying social dynamics. Some projects are currently undergoing with the specific aim of providing easy-to-use and efficient tools for accessing data from GitHub, especially in real-time. For example, (Gousios and Spinellis 2012) discusses a system to gather streams and data from GitHub in a scalable fashion to overcome the limitations imposed by the GitHub API, specifically directed at researchers.

In (Dabbish et al. 2012) an in-depth qualitative user study is conducted on a small group of GitHub users, aimed at understanding the motivations that are the basis of online collaboration and the consequences of using a transparent large-scale tool on the practice of software development. They find that people use GitHub for several reasons: to learn how to code better, to follow popular developers, to find new interesting projects, and to promote themselves and their work. They also find that actions in GitHub, such as following a user, committing changes and accepting/rejecting code, are heavily influenced by specific characteristics of the interactions happening in the system. Some other studies have investigated the geographical distribution of collaborations (Heller et al. 2011; Shrestha, Zhu, and Miller 2013). An example is that of Heller et al. (Heller et al. 2011), who use visualization techniques to identify patterns of collaboration, including geographic characteristics of the interactions between cities and influence among them.

With respect to this body of work, to the best of our knowledge, our paper presents the first systematic quantitative analysis of the interactions in GitHub. We believe that our quantitative methodology complements the existing qualitative work by providing insights about global patterns of interactions that are not possible to obtain by means of small-scale and interview-based studies.

## 3 Description of the Dataset

The full list of public events that have happened on GitHub is available on the GitHub Archive website<sup>8</sup>. In this paper, we analyze events that happened on GitHub over a period of 18 months, between March 11, 2012 and September 11, 2013, retrieved from that archive. Our dataset includes various types of events performed by users on public repositories or following events between users (i.e., when a user starts following another user). The total number of retrieved events is 183,540,210 and they fall into 18 categories<sup>9</sup>. Each event, regardless of its kind, usually includes some metadata about the entities involved (e.g., the profile information of a

<sup>8</sup><http://www.githubarchive.org>

<sup>9</sup><http://developer.github.com/v3/activity/events/types/>

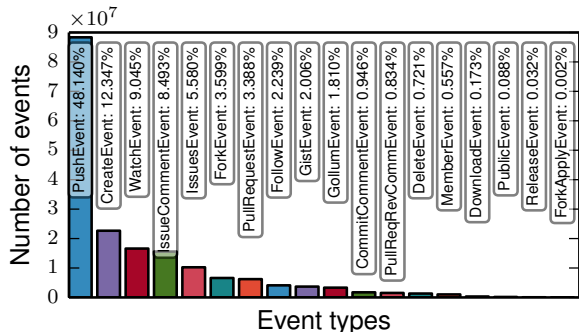


Figure 1: Number of events detected in the GitHub stream.

user, his or her number of followers, the language of a repository, etc.). Fig. 1 shows how events are distributed among the various categories. One outlier user under the name of Try-Git shows an uncommonly high number of collaborations. As it is a learning tool that pushes code automatically to other users’ repositories, we discarded it from the dataset.

In order to explore the geographic features of users, we investigate the location information that can be found in the user profiles. In our dataset, 345,625 users have a non-empty location field. As the field is optional, there is little incentive to fill it with fake information. Therefore, we can reasonably assume that most of the non-empty entries are truthful. In order to convert the text field to an unambiguous location, we use the MapQuest Open Geocoding API<sup>10</sup>. We evaluate the validity of the geocoder by considering a sample of 1,000 users in the population of users with non-empty location field and assessing the fraction of correctly geocoded elements by manually labeling them. We find that 106 elements are incorrectly geocoded. From the analysis of this sample, therefore, we can say that the geocoder fails to correctly convert to coordinates in  $10.6 \pm 1.91\%$  cases of the original population, with 95% confidence level. Incorrectly geocoded entries in the sample fail mostly for the following reasons: because they describe multiple locations (for example "London and Nottingham"), because they have no geographic meaning (e.g., "localhost", "emacs") because they are ambiguous (e.g., "San Jose", rather than "San Jose, CA").

It is important to be aware that this data source suffers a time bias, since the archive does not include events happened before March 2011. In Fig. 2 we show the number of unique users and public repositories seen in the event stream since March 11, 2012. As previously discussed, we are able to retrieve metadata when entities are involved in an *event*. In other words, we do not have information about dormant entities that were created before March 11, 2012 and do no longer generate *any* event during the subsequent 18 months (e.g., an inactive user, an abandoned repository). We are

<sup>10</sup>Data: © OpenStreetMap contributors, available under the Open Database License. Geocoding: courtesy of MapQuest (<http://www.mapquest.com>).

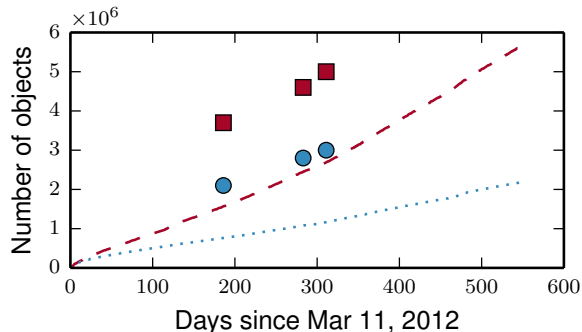


Figure 2: Number of unique repositories and unique users detected from the stream since March 11, 2012. The dashed and dotted blue lines show the number of repositories and the number of users detected from the event stream, respectively. The three squares and circles indicate the number of repositories and users in three specific dates as advertised by GitHub on its website.

also not able to extract pre-existing following relations from the stream. After a short transitory period, which is present because of the temporal bias of our data collection process based on events, both curves show linear growth with different coefficients, with the ratio describing the number of repositories over the number of users reaching a steady value of approximately 2.59. The figure also reports the number of repositories and users (indicated using squares and circles, respectively) publicly declared by GitHub. In our dataset, we observe a lower number of users and repositories for two reasons. Firstly, the official numbers include *all* the users and repositories created since the launch of the website in 2008, whereas our dataset contains only the *active* users and repositories in the period taken into consideration. Secondly, the official statistics probably include private repositories, that do not produce events on the public timeline we are able to access. For these reasons, we can conclude that a large number of users do not actively use the website (i.e., do not generate events) or they act exclusively on private repositories. These figures also suggest that a large number of repositories are either abandoned or private.

## 4 Structural Analysis

In this section we define, extract and analyze several networks, generated from the event stream, which describe interactions between users and repositories.

- We represent users’ following relations by means of a directed graph  $G_F$ , which we call *followers graph*. We are able to reconstruct this network by looking at *follow events* in the stream.
- We represent the collaborations of users on repositories as a bipartite graph  $G_C$ , the *collaborators graph*, where repository nodes are connected to their collaborators nodes. We are able to infer this network by extracting from *push events* information about who uses write per-

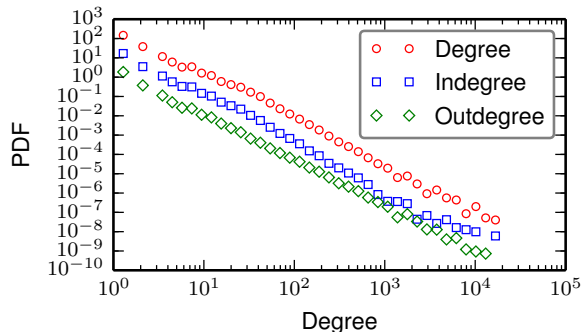


Figure 3: Distribution of degree, in-degree and out-degree of the social graph. The distributions were shifted along the y-axis to put in evidence their structure. The three distributions exhibit a power-law scaling behavior, with different exponents, for values in the range from 20 to 1000.

mission and on which repositories. We refer to  $G_C^\perp$ , the *projected collaborators graph*, as to the graph obtained by projecting the collaborators graph onto the set of users. In this projected graph users who collaborate in at least one repository are connected to each other.

- We represent users assigning a star to a repository as a bipartite graph  $G_S$ , the *stargazers graph*. This network can be generated using the information found in *watch events*.
- Finally, we build the *contributors graph*  $G_N$  by analyzing the content of every *push event*, which includes authorship information of the pushed commits.

For our static analyses we consider these networks as they appear on the final day of the time window we take into consideration.

### Followers and Collaborators Networks

As previously explained, a user follows other users in order to be regularly updated about events regarding them (e.g., forks, created repositories, starred repositories, and so on). The followers graph  $G_F$  we obtain has a total of 671,751 nodes and 2,027,564 edges, with a resulting graph density of 4.4932e-06 and an average degree of 3.019. The low graph density and average degree indicate that on GitHub the follow action is associated with a high cost, as following many developers results in receiving many notifications from them. This result also reflects the fact that following links in GitHub do not play the same important role they have in other social networks, such as Facebook or Twitter.

Fig. 3 shows the distributions of the in-degree, out-degree and total degree of the users in  $G_F$ . All the three distributions show a power-law scaling behavior, characterized by different regimes. We also note that for degrees smaller than  $k \approx 20$ , in all the three cases the scaling relation is not satisfied. Interestingly, we also find that the degree distributions of  $G_F$  and of  $G_C^\perp$  follow the same power-law regime, as

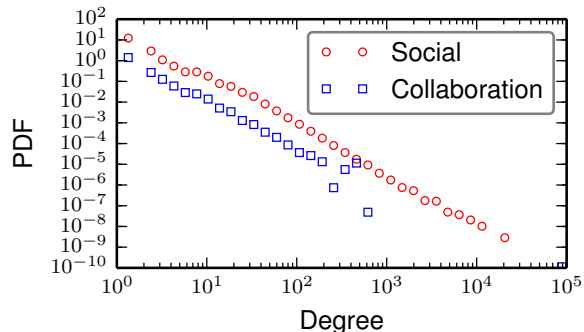


Figure 4: Distribution of the number of followers per users (red) and the number of total collaborators per user, (blue) which corresponds to the degree distribution of the users projection of the collaborators bipartite graph.

shown in Fig. 4. However, the node degree in the followers graph grows considerably larger than in  $G_C^\perp$ .

The followers network is also characterized by low reciprocity: only 9.6% of the pairs of users have a reciprocal relation between them, while the remaining 90.4% are one-way. Other studies on social networks reported considerably higher levels of reciprocity, such as 22.1% for Twitter (Kwak et al. 2010), 68% for Flickr (Cha, Mislove, and Gummadi 2009) and 84% for Yahoo! 360 (Kumar, Novak, and Tomkins 2010). The consistently lower reciprocity in Twitter is partially motivated by the presence of a few popular programmers, the so-called “rockstar programmers”, who exhibit high in-degrees and low out-degrees. However, we believe the profoundly different nature of GitHub, compared to other social networks, might also play a role in this. In fact, social networks are mostly used for leisure and they thrive on distractions coming from noisy timelines; on the contrary, the productivity of GitHub developers might be critically disrupted by non-relevant notifications, which are hence kept to a minimum. In other words, establishing links has high cost in GitHub, as people do not “follow-back” unless they are professionally interested in the activity of their followers.

In order to uncover the presence of node degree correlations, we first measure the degree assortativity. We say that a network shows an assortative mixing (Newman 2002) if nodes with a large number of links tend to share edges with high degree nodes. Similarly, if nodes with a small number of links tend to share edges with low degree nodes we say that a network shows a disassortative mixing. In our case we find a value of -0.0386, which suggests a tendency to a disassortative mixing of users. We also evaluate the rich-club coefficient  $\phi$  (Zhou and Mondragón 2004), which measures the tendency of high degree nodes to form tightly interconnected communities. Although apparently similar to the concept of assortative mixing, the rich-club phenomenon is not necessarily associated with the latter, as one can define a disassortative network that still shows evidence of a

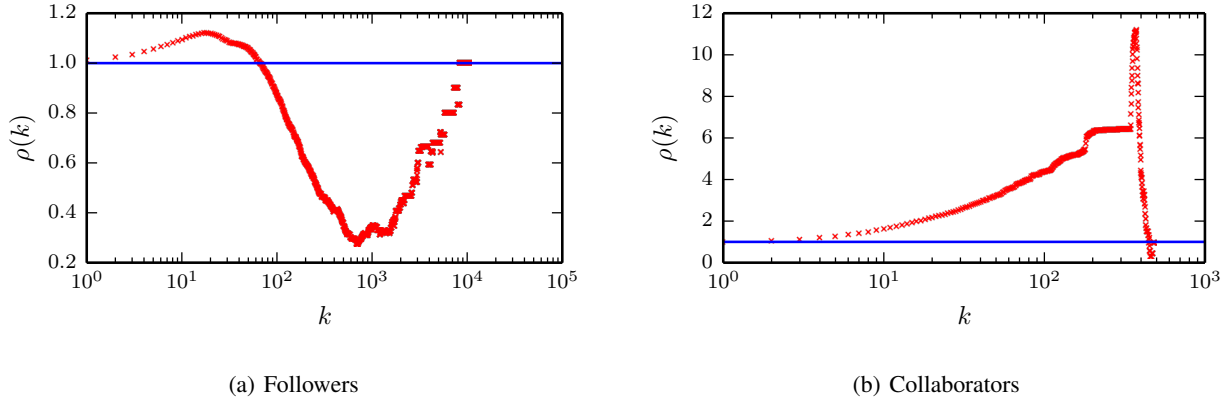


Figure 5: The normalized rich-club coefficient  $\rho(k)$  as a function of the node degree. The blue horizontal line represents the value of  $\rho(k)$  on a maximally random graph. Values of  $\rho(k)$  above (below) 1 correspond to the presence (absence) of the rich-club phenomenon with respect to the random case. The two networks show remarkably different rich-club behaviors, due to their distinct nature.

rich-club phenomenon. Let  $E_k$  denote the number of edges among the  $N_k$  nodes having a degree higher than  $k$ . The rich-club coefficient  $\phi(k)$  is defined as follows:

$$\phi(k) = \frac{2E_k}{N_k(N_k - 1)} \quad (1)$$

It represents the fraction of edges connecting nodes in  $N_k$  out of the maximum possible amount they can share, i.e.,  $\frac{N_k(N_k-1)}{2}$ . More specifically, here we use the normalized rich-club coefficient proposed by (Colizza et al. 2006), where the normalization is introduced to account for the fact that high degree nodes have a higher probability of sharing edges than low degree ones. Fig. 5(a) shows the rich-club index of the followers graph, for increasing degree  $k$ . We use the definition of the rich-club index for  $G_F$  considering it as an undirected graph. Interestingly, we see that low degree nodes show a less accentuated rich-club phenomenon, while high degree nodes do not. In other words, the plot indicates that hubs, i.e., popular developers, tend to share links with lower degree nodes rather than being tightly interconnected among them.

Compared to the followers graph, the collaborators graph  $G_C^\perp$  also shows disassortative mixing of the nodes, with a value of  $-0.0518$ . However, the characteristics of the rich-club phenomenon are remarkably different. Fig. 5(b) shows the rich-club index of  $G_C^\perp$ , for increasing values  $k$  of the degree. We observe that up to  $k \approx 30$  the nodes show a strong rich-club phenomenon, with a pronounced increase followed by a sudden drop around  $k \approx 40$ . This effect is amplified by the projection operation itself, as each group of collaborators forms a clique in  $G_C^\perp$ .

We also measure the clustering coefficient (Watts and Strogatz 1998) of  $G_C^\perp$  and we compare it with that of the followers graph. Again, we expect the average clustering coefficient of the network to be high due to the way in which  $G_C^\perp$  is constructed. Indeed, we find a value of  $0.395$  for  $G_C^\perp$  and of  $0.012$  for  $G_F$ . Note, however, that this implies that

users contributing to the same repositories do not necessarily follow each other, as in that case we would expect the average clustering coefficients of the two networks to be similar. Once again, this underlines the fact that the social interactions captured by the two structures are rather different.

We now investigate the relation between the number of followers of a user and his/her contributions to GitHub. We would expect popular users in terms of contributions to be followed by a higher number of people. In order to evaluate this, we measure the Spearman correlation coefficient (Lehmann 2006) between the number of followers and the number of contributions per user, and we find a value of  $0.2568$ , with  $p\text{-value} < 0.01$ , indicating the lack of a clear correlation between the two dimensions. This result is unexpected, as it would seem reasonable to assume for *active* users, i.e., users that contribute to a large number of repositories, to be more popular in terms of followers.

## Interactions on Repositories

Despite the large number of repositories hosted at GitHub, developers work only on a consistently smaller fraction of them. Only 62.90% of the total number of repositories we obtain information for experience at least one code commit during the 18 months taken into consideration. Only 74.22% of these repositories have at least two contributors, meaning that one active repository out of four is exclusively authored by a single individual. This might happen for a variety of reasons: the project might not look promising to other users or the owners of the repository might reject contributions. This fraction includes activity both from one-time and habitual collaborators. Commonly, long-term contributors are turned into collaborators, so that they can help developing big projects. However, this kind of collaboration is quite rare, as only 9.61% of the repository has at least 2 of them. This is not surprising: collaborators need to be trusted individuals who have full understanding of the project goals and structure, as they have write access on the repository

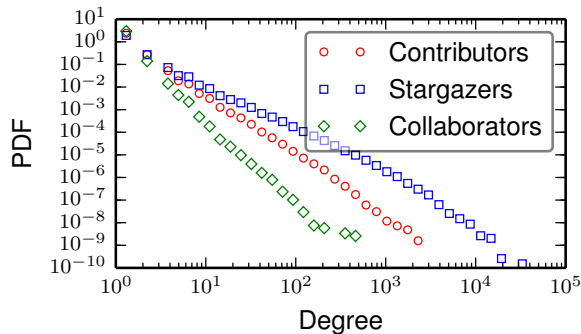


Figure 6: Distribution of the number of contributors, collaborators and stargazers per repository. The contributors and stargazers distributions are best fitted by a power law distribution with exponential cut-off and exponent  $\alpha$  equal to 2.34 and 1.77, respectively. The collaborators distribution is consistent with a power law with  $\alpha = 3.39$ . The distributions parameters are estimated using the approach of (Clauset, Shalizi, and Newman 2009).

and they determine which contributions should be accepted. Fig. 6 reports the distribution of number of contributors, stargazers and collaborators per repository.

### Forking and Repository Tree Structure

The fork operation is intended to let users actively contribute to a project. This action produces a copy of the parent repository and essentially generates a simple tree structure. Further forks on the leaves of the tree increase its depth, while forking an internal node results in an increased width of the set of its children. We interpret the repository tree as a directed acyclic graph, where the fork operation generates a directed edge from the parent repository to its child. In the following we refer to the *depth* of the tree as the longest path from the root to its leaves, and to its *width* as the maximum number of children over the internal nodes or 1 if the root has no children.

For a few repositories the maximum depth goes up to 12. However, these few structures are hardly the result of collaboration, in our opinion. In fact, user accounts involved in their creation do not exist anymore. For this reason, we suppose these accounts have been removed due to abnormal or suspicious activity. We also find that the average depth is 3.0695, but the mode is 0, indicating that the majority of repositories has a low number of contributions. The width, on the other hand, goes up to 10,256, which is normal considering that many people fork to contribute to popular packages, such as `mxcl/homebrew`. Top repositories include `heroku/node-js-sample`, `YOU-LOST/THE-GAME` (apparently, a ludic non-software repository) and `facebook-tornado`. The overall average width is very low (1.0653), showing that just a few popular repositories get forked, while the vast majority of them (93.91%) have a width of just 1. This, together with the observation

that the majority of the repositories has depth equal to 0 and width equal to 1, seems to suggest that forks on GitHub happen on a limited number of key projects.

## 5 Activity, Social Presence and Indirect Rewards

Human activities are commonly driven by reward mechanisms of some kind: people work to earn money and achieve a social status, they play games because they have fun, they travel because they enjoy seeing new places. A recent work has found that areas of brain connected to rewards are activated during the use of social networks websites (Meshi, Morawetz, and Heekeren 2013). One of the aspects that drives activity in GitHub, among others, is self-promotion (Dabbish et al. 2012). We hypothesize that for a hybrid service like GitHub, both a social network and a collaboration network, some kind of indirect reward mechanism might and potentially underpin user activity. Even if it is not possible to provide definitive evidence about that, in the following we will show some interesting correlations between the activity of a user and some indirect rewards in terms of “social prestige” in GitHub.

In social networks, a common measure of user popularity and influence is given by the in-degree (Wasserman and Faust 1994). Therefore, it is reasonable to consider new connections as sort of rewards for users receiving them, as they increase their popularity. In order to investigate this aspect in GitHub we will search for correlation between user activity and degrees in the several graphs we have introduced. In Fig. 7(a) we plot the number of authored events (i.e., for which the user *actively* executes an action) for each user against the number of followers. We firstly note that people with a higher number of followers are commonly more active and people with lower levels of activity generally have fewer followers. However, we also observe that many users with a very high number of events have a very low number of followers: a higher level of activity does not directly translate into a larger number of followers. A similar phenomenon is also visible in Fig. 7(b), where we plot the number of authored events against the number of repositories for which a user is a collaborator or the repository owner. Being the collaborator can also be seen as a kind of indirect reward, as it is more important and prestigious than being a contributor. Collaborators receive permissions to modify the repository, whereas contributors only contribute their code through pull requests.

We are also interested to see whether a higher out-degree on the social graph is an indicator of a higher activity. However, in Fig. 7(c) it is possible to note that a much weaker correlation between these two quantities is present. A similar behavior can be observed in Fig. 7(d), where we plot activity versus the number of starred (i.e., bookmarked) repositories. In other words, users who follow many other users or bookmark many repositories are not much more active than those who do not.

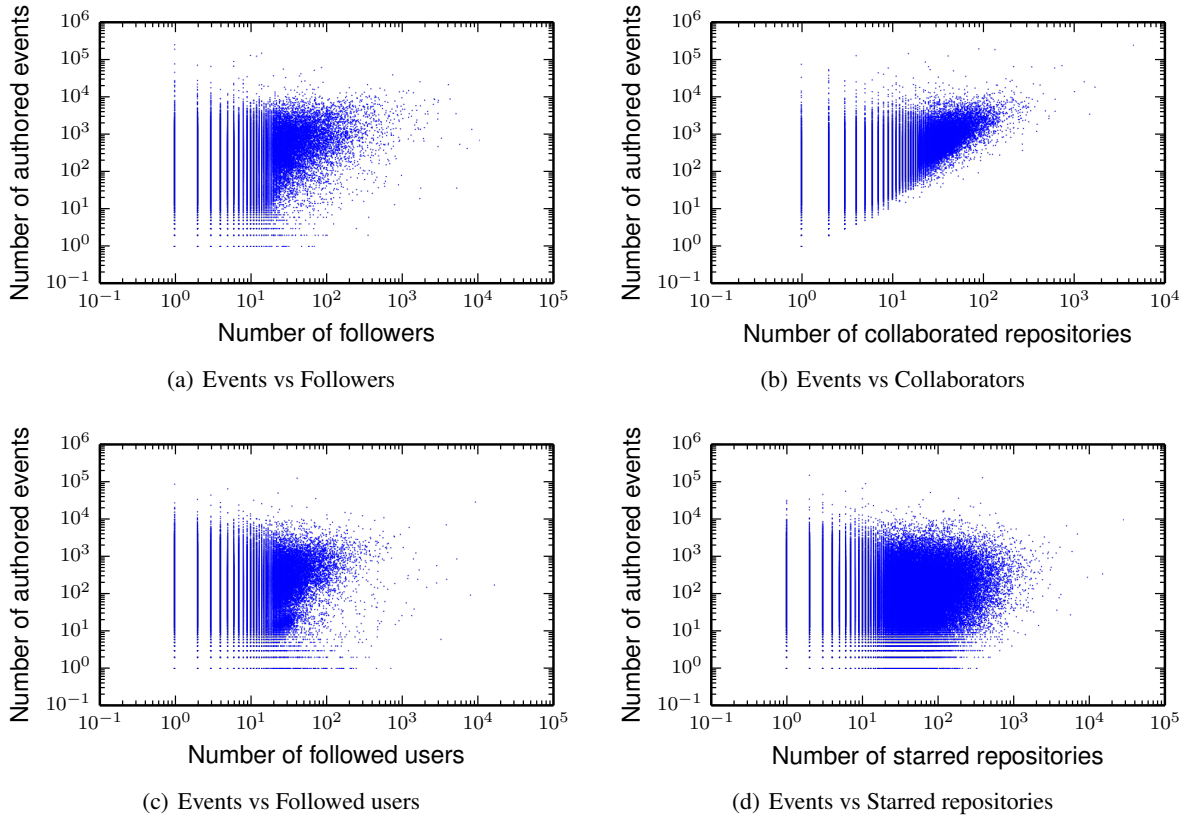


Figure 7: Number of actions executed by a user against (a) user followers, (b) number of repositories with write privileges, (c) followed users, (d) starred repositories.

## 6 The Geography of Collaboration

In Fig. 8 we show the geographic distribution of users in our dataset around the world. The majority of users is located in Europe and North America while other geographic regions have a consistently smaller number of users. The Tab. 1, listing the 20 most common countries and cities indicated in GitHub user profiles, confirm this consideration. The popularity of GitHub among developers living in the USA is really prominent, as 3 users out of 10 are based there.

### Impact of Geographic Proximity

We also analyze the impact of physical proximity on the patterns of collaboration between different users. Are people more likely to follow people who are closer to them? In Fig. 9 we show the distribution of the distance covered by each pair of users connected by a directed link in the social network. The first part of the distribution shows that links decrease with distance, until  $x = 5000 \text{ km}$ : these are intra-continental links. The sudden drop at  $x = 5000 \text{ km}$  is due to the ocean separating North America and Europe, that are the two regions where GitHub is mostly popular. For larger distances, the distribution increases again, showing a big presence of intercontinental links. This analysis, however, considers all the links, without discriminating them on a per-user basis.

We now want to see how *local* or *global* is the neighborhood of a user, depending on how far his connections are located. In order to do that, we calculate for each user the average distance of their followers, their followed users and reciprocated links (i.e. users that are both followers and followed). In Fig. 10 we show the probability density function of the values of this measure. As expected, the distribution of these values decreases as the distance increases, indicating that users tend to interact with people that are close. We also notice that in the majority of the cases the average distances of the reciprocated connections of a user, usually considered as evidence of friendship or mutual acquaintance or collaboration, tend to be smaller compared to the other two types of links.

### Globality and Distant Collaboration

We now investigate if geographic proximity has an impact on the collaboration between users. In this case, we cannot compute the geographic distance between collaborators of a certain repository and the repository itself, as we cannot assign geographic coordinates to a repository. Project collaborators might be sparse around the globe or concentrated in a single city. In order to quantify how sparse they are, we define the *globality* of a set of users  $\mathcal{S}$  as follows:

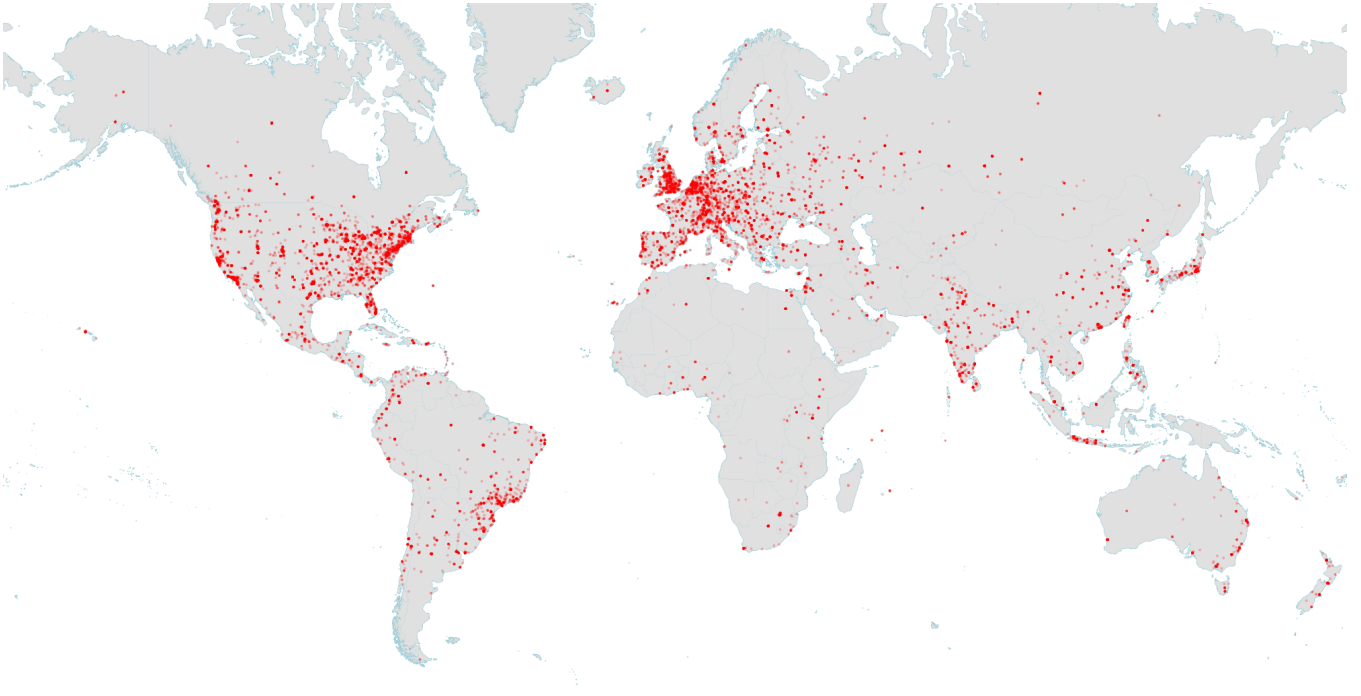


Figure 8: Distribution of GitHub users in the world. For each user, a partially transparent point is drawn on the map. The majority of users is located in North America and in Europe. The leading countries are the United States and the United Kingdom. A 15% random sample of the original distribution was used to make this figure.

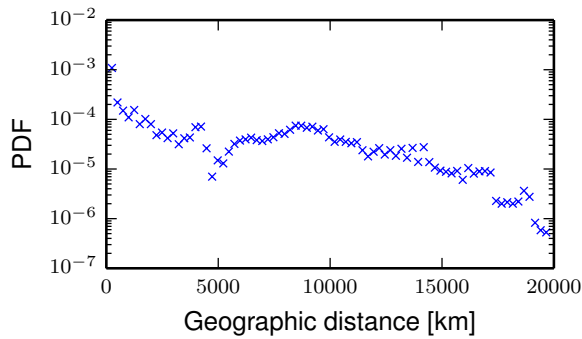


Figure 9: Distributions of the inter-users distance covered by each follow link. The distribution has a maximum at the lowest distance and gradually decreases for high distances.

$$G = \frac{1}{Nd_{max}} \sum_{i,j \in S} d_{ij} \quad (2)$$

where  $d_{max}$  is the maximum distance between two points on Earth where two generic users are localized and  $N$  is the number of users taken into consideration. This measure is the normalized average of distances between all the points in the set. When all points coincide the measure is 0, whereas when the points are evenly distributed at the antipodes the measure is 1.

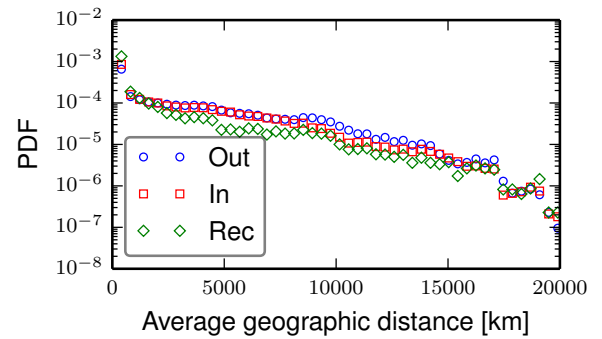


Figure 10: Distributions of the average geographic distance of a user's outgoing, incoming and reciprocated links, respectively represented by blue circles, red squares and green diamonds.

In Fig. 11 we plot the value of globality against the number of collaboration for all the repositories that have at least two collaborators with location information in their profile. Although points are quite dispersed in the plot we can make some considerations. For repository with a low number of collaborators, globality reaches values close to its maximum. As the number of collaborators goes up, the value of globality is found to be lower. This suggests that repositories with a low number of collaborators tend to have them



Rank	Country	%	City	%
1	USA	30.14	San Francisco, US	3.84
2	UK	6.43	London, GB	3.33
3	Germany	5.28	New York City, US	2.93
4	China	5.11	Beijing, CN	1.98
5	India	4.05	Paris, FR	1.80
6	France	3.87	Tokyo, JP	1.69
7	Canada	3.69	Seattle, US	1.59
8	Brazil	3.60	Berlin, DE	1.49
9	Russia	3.14	Chicago, US	1.39
10	Japan	2.83	Shanghai, CN	1.34
11	Australia	2.00	Bangalore, IN	1.32
12	Spain	1.92	Toronto, CA	1.23
13	Netherlands	1.84	Moscow, RU	1.17
14	Sweden	1.51	Austin, US	1.12
15	Ukraine	1.37	Boston, US	1.07
16	Italy	1.32	Los Angeles, US	1.01
17	Poland	1.02	Sydney, AU	0.94
18	Switzerland	0.86	Portland, US	0.88
19	Belgium	0.75	Melbourne, AU	0.85
20	Mexico	0.74	Stockholm, SE	0.81

Table 1: Top 20 countries and cities, ranked by absolute number of users.

concentrated around one or more key locations rather than scattered around the globe.

We now investigate how social and collaborations links are distributed among countries. In order to do that, we build two square matrices  $M_{G_F}$  and  $M_{G_C}$  describing the number of links between countries. An element  $m_{ij}$  of the matrices indicates the number of links from people in country  $i$  to people in country  $j$ . The rows are normalized to sum to unity. This matrix has a precise meaning: each row shows how links coming from the people in a given country are distributed geographically. For clarity, in Fig. 12 we show the normalized matrices only for the top 20 countries, although the following measures are calculated on the full matrices. We first note that both matrices have a strong diagonal component (on average 0.245 and 0.346 for followers and collaborators, respectively), in accordance with the fact that links are more likely to be directed to the same country of origin. The two matrices are also significantly similar, as confirmed by the low average cosine distance, amounting to 0.277.

## 7 Conclusions

In this paper we have analyzed the events happening on GitHub, the most popular repository for open source code, for 18 months between March 11, 2012 and September 11, 2013. We have obtained information about 2.19 million users and 5.68 million repositories. From this dataset we have derived four networks: a bipartite network describing the collaborations of users on repositories, a bipartite network describing the stars (bookmarks) assigned by users to repositories, a bipartite network describing the contributions of users on repositories and a directed social network describing the follow relations between users. We have showed that the distributions of the number of collaborators per project, contributors per project, stargazers per project and

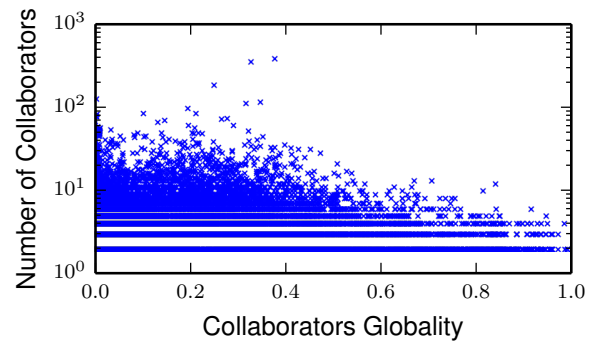


Figure 11: Scatter plot showing, for each repository, the number collaborators and the globality calculated over their geographic points. Intriguingly, repositories with a high number of collaborators exhibit smaller values of globality.

user followers show a power-law-like shape. We have found a very low reciprocity of the social ties, which is remarkably different from results of studies in other social networks; we have also observed that collaboration between users happens on a small fraction projects. We have found that very active users do not necessarily have a large number of followers. Finally, we have investigated the impact of geography on collaboration. Consistently to what happens in other social networks, users tend to interact with people that are close, as long-range links have a higher cost. A similar consideration can be made for repositories with a high number of collaborators, which tend to be managed by collaborators gravitating around specific locations.

We believe that our work provides novel insights about the complex dynamics of collaboration on a planetary scale. Our future research agenda includes the investigation of the software engineering issues that emerge from our quantitative analysis, especially with respect to the flow of information (and knowledge) that is present in the network of users. We think that this might represent a starting point for the development of novel strategies and tools for supporting online collaboration more effectively and efficiently.

## Acknowledgments

This work was supported through the EPSRC Grant “The Uncertainty of Identity: Linking Spatiotemporal Information Between Virtual and Real Worlds” (EP/J005266/1).

## References

- Bird, C.; Pattison, D.; D’Souza, R.; Filkov, V.; and Devanbu, P. 2008. Latent social structure in open source projects. In *Proceedings of FSE’08*, 24–35. ACM.
- Brandes, U.; Kenis, P.; Lerner, J.; and van Raaij, D. 2009. Network Analysis of Collaboration Structure in Wikipedia. In *Proceedings of WWW’09*, 731–740. ACM.
- Cha, M.; Mislove, A.; and Gummadi, K. P. 2009. A measurement-driven analysis of information propagation in

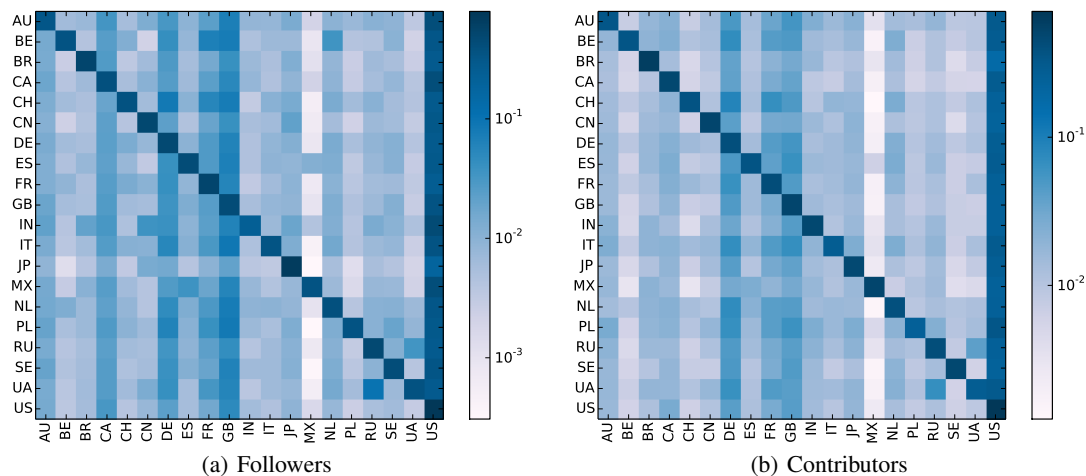


Figure 12: The ties among users of the top 20 countries in terms of number of users. The rows of the matrix are normalized to sum to unity. Note that the most followed users are in the United States, while the least followed country is Mexico. As expected, the matrix also shows a strong level of interaction between users from the same country. The left matrix is built from the followers graph, while the right matrix is built from the contributors graph.

the flickr social network. In *Proceedings of WWW'09*, 721–730. ACM.

Clauset, A.; Shalizi, C. R.; and Newman, M. E. 2009. Power-law distributions in empirical data. *SIAM review* 51(4):661–703.

Colizza, V.; Flammini, A.; Serrano, M. A.; and Vespignani, A. 2006. Detecting rich-club ordering in complex networks. *Nature Physics* 2(2):110–115.

Dabbish, L.; Stuart, C.; Tsay, J.; and Herbsleb, J. 2012. Social Coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of CSCW'12*, 1277–1286. ACM.

Finley, K. 2011. GitHub has surpassed Sourceforge and Google Code in popularity. <http://readwrite.com/2011/06/02/github-has-passed-sourceforge>.

Gousios, G., and Spinellis, D. 2012. GHTorrent: Github's data from a firehose. In *Proceedings of MSR'12*, 12–21.

Heller, B.; Marschner, E.; Rosenfeld, E.; and Heer, J. 2011. Visualizing collaboration and influence in the open-source software community. In *Proceedings of MSR'11*, 223–226.

Hindle, A.; German, D. M.; and Holt, R. 2008. What do large commits tell us?: a taxonomical study of large commits. In *Proceedings of MSR'08*. New York, NY, USA: ACM.

Kittur, A.; Chi, E.; Pendleton, B.; Suh, B.; and Mytkowicz, T. 2007. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *Proceedings of CHI'07*.

Kumar, R.; Novak, J.; and Tomkins, A. 2010. Structure and evolution of online social networks. In *Link Mining: Models, Algorithms, and Applications*. Springer. 337–357.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is twitter, a social network or a news media? In *Proceedings of WWW'10*, 591–600. ACM.

Lehmann, E. 2006. *Nonparametrics: Statistical Methods based on Ranks (POD)*. Prentice-Hall.

Lunden, I. 2013. GitHub hits the 4M user mark as it looks beyond developers for its next stage of growth. <http://techcrunch.com/2013/09/11/github-hits-the-4m-user-mark-as-it-looks-beyond-developers-for-its-next-stage-of-growth/>.

Meshi, D.; Morawetz, C.; and Heekeren, H. R. 2013. Nucleus accumbens response to gains in reputation for the self relative to gains for others predicts social media use. *Frontiers in Human Neuroscience* 7:439.

Newman, M. E. 2002. Assortative mixing in networks. *Physical Review Letters* 89(20):208701.

Shrestha, A.; Zhu, Y.; and Miller, B. 2013. Visualizing time and geography of open source software with storygraph. In *Proceedings of VISSOFT'13*, 1–4. IEEE.

Valverde, S., and Solé, R. V. 2007. Self-organization versus hierarchy in open-source social networks. *Physical Review E* 76(4):046118.

Vuong, B.-Q.; Lim, E.-P.; Sun, A.; Le, M.-T.; Lauw, H. W.; and Chang, K. 2008. On ranking controversies in wikipedia: Models and evaluation. In *Proceedings of WSDM'08*, 171182. New York, NY, USA: ACM.

Wasserman, S., and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1 edition.

Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of small-world networks. *Nature* 393(6684):440–442.

Zhou, S., and Mondragón, R. J. 2004. The rich-club phenomenon in the internet topology. *IEEE Communications Letters* 8(3):180–182.