

# Temporal Development of Overlapping Communities in Co-Authorship Networks

Alexa Schlegel

Freie Universität Berlin, Institute of Computer Science  
alexandra.schlegel@gmail.com

**Abstract.** Detection of overlapping communities plays an important role when it comes to analysing scientific collaboration using social networks. This paper focuses on describing a method to analyze the temporal development of overlapping communities within co-authorship networks. The paper mainly relies on the work of Palla et al. using the clique percolation method for community extraction. The main goal was to find and understand an appropriate algorithm, which can be applied to an existing dataset of scientific publications. Implications regarding further actions and time complexity are discussed.

**Keywords:** co-authorship networks, scientific collaboration networks, overlapping community detection, temporal analysis, clique percolation, dynamic communities

## 1 Introduction and Motivation

A *social network* is a representation of a social structure consisting of actors, e.g. individuals, affiliations, and their social interactions. The network model conceptualizes e.g. social, economic or political structures as lasting pattern of interactions between actors. [13]

A *co-authorship network* is a scientific collaboration network, where actors are scientists and the social interaction is collaboration work resulting in publishing papers together. In mathematical terms networks are graphs, therefore they consist of nodes (vertex) and links (edges). In a co-authorship network nodes are individual authors who are considered connected if they co-authored a paper. [6]

Usually those networks are used to study patterns of scientific collaboration in different fields, their development over time, identify key people regarding different network measurements, detection and evolution of communities and many more. See [1], [6] and [7] for more details.

This seminar paper is motivated by a dataset consisting of 2.277 publications related to geochemistry focusing on papers talking about stable metal isotopes (e.g. potassium, magnesium, lithium, silicon). This data was used to build an cumulated co-authorship network using an very simple approach to model collaboration.

The dataset was collected using the bibliographic database scopus<sup>1</sup> and includes publications starting from 1997/98 until 2014. The dataset consists of authors, co-authors, title, year, abstract and affiliations. A basic network analysis had already been conducted for example centrality measurement, community detection, degree distribution and average degree.

As the data set includes temporal information (publishing year), the following question arises: How do communities within this type of co-authorship network evolve over time? In geochemistry people often tend to focus on one isotope system, and stick to this topic for a long time. Do researchers in geochemistry really focus on one specific topic or do they change the field over time?

The goal of this seminar paper therefore is to find and explain an already established and well understood method for analysing the temporal development of communities within co-authorship networks. The main point is to understand the underlying algorithm in detail and to find out implications and challenges when applying this method to the geochemistry dataset.

## 2 Related Work

The related work focuses on algorithms for community detection especially for co-authorship networks and how to track those detected communities over time. "Most real world networks contain parts in which nodes are more highly connected to each other than to the rest of the network, those sets are usually called clusters, communities, cohesive groups or modules, they have no widely accepted, unique definition." [9] The way "more highly connected" is precisely defined makes the difference between community detection algorithms. It depends on the heuristic which is employed to identify communities. [10]

Fortunato [3] gives an extended overview about various types of community detection algorithms. To explain any would exceed the scope of this work. For example traditional methods include algorithms based on graph partitioning, hierarchical clustering and spectral clustering. There are also divisive and agglomerative algorithms. The best known one from this category is Girvan and Newman [4].

Most algorithms determine distinctive sets of communities, so a node can only belong to a single community. Most real world networks consists of overlapping and nested communities e.g. friendship network of a person [9]. Fortunato [3] gives also an overview about algorithms detecting overlapping communities. The best known and widely cited is the *clique percolation method (CPM)* by Palla et al. [9]. There is an implementation provided by the authors, called *CFinder*<sup>2</sup>, and an implementation in the R igraph library<sup>3</sup>.

<sup>1</sup> <http://www.scopus.com/>

<sup>2</sup> CFinder is a tool which implements the clique percolation method by Palla and finding and visualizing overlapping dense groups of nodes in networks. <http://www.cfinder.org/>, 05.12.2015, 22:43

<sup>3</sup> <http://igraph.wikidot.com/community-detection-in-r#toc0>, 05.12.2015, 22:45

Not only overlapping communities exist in networks, but also communities are structured in a hierarchical way. Recent work on algorithms detecting hierarchical and overlapping structures include [5], [2] and [12].

Looking at the temporal development of communities “the main phenomena occurring in the lifetime of a community are: birth, growth, contraction, merger with other communities, split, death” [3] (see figure 1). The detection of dynamic communities has been studied by Palla et al. [8] using CPM for each snapshot of the graph. Further information on dynamic communities can be found in section 13 of [3].

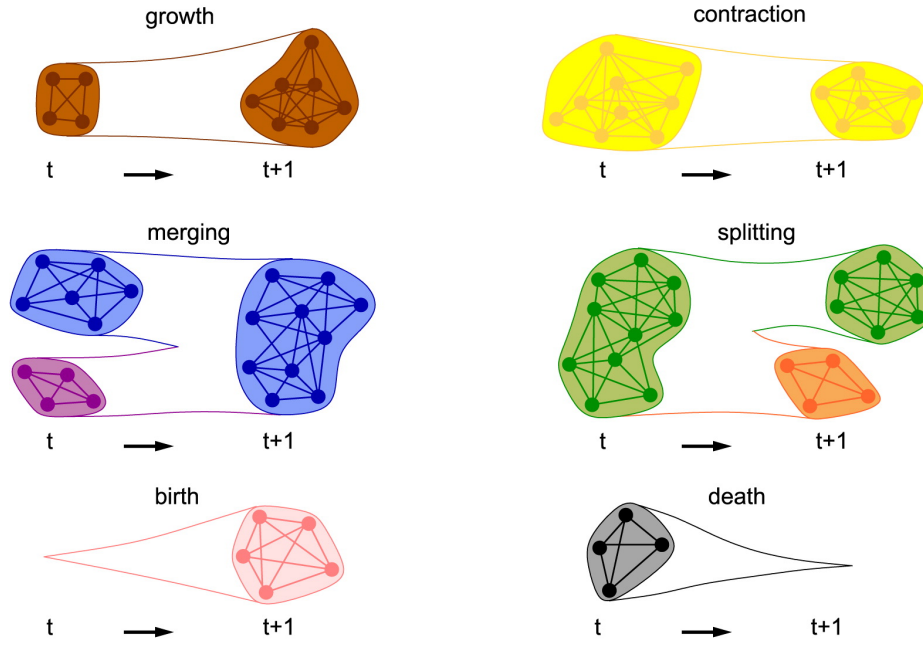


Fig. 1: lifetime of a community

As CPM is a widely used method for detecting overlapping communities I will explain this method in more detail. The following sections describe the CPM and also the temporal algorithm mainly based on the two papers of Palla et al. [9] and [8].

### 3 Clique Percolation Method

The clique percolation method (CPM) developed by Palla et al. [8] is used to identify overlapping communities in unweighted and undirected networks. The authors tested and evaluated their method using a co-authorship network, protein-protein interaction network, word associations network and a network constructed from variables

within the source code of an open source ftp programm. The following section explains the algorithm and summarizes the main findings of the paper regarding co-authorship networks.

The underlying community definition relies on the fact that a community consists of fully connected subgraphs, called *cliques*.  $k$ -cliques are fully connected subgraphs with  $k$  nodes. In figure 3 an example of  $\{3, 4\}$ -cliques are given. A community, in this context, is called a  $k$ -clique-community, which is defined as a union of all  $k$ -cliques, which can be reached from each other through a number of *adjacent  $k$ -cliques*. Two  $k$ -cliques are called adjacent if they share  $k - 1$  nodes. In figure 4 the so called  $k$ -clique template rolling is visualized for a 3-clique. A  $k$ -clique-community is best described by relocating one node until all nodes of the community are discovered.

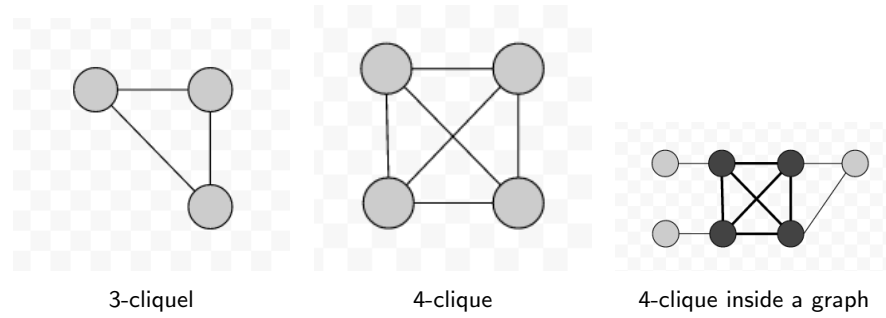


Fig. 3: Examlpe of cliques

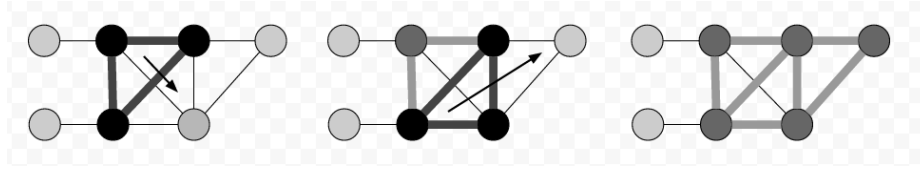


Fig. 4: Template rolling for a 3-clique

### 3.1 Algorithm

The starting point for the algorithm is an undirected and unweighted graph. In section 3.2 we talk about generating the unweighted co-authorship graph. Based on the explained community definition the algorithm consists of the following steps:

- (1) Find all maximal cliques.

- (2) Prepare clique-clique overlap matrix, or clique graph.
- (3) Threshold the matrix with a certain  $k$ .
- (4) Each connected component in the clique graph form a  $k$ -clique-community.
- (5) Post-processing.

**(1) Find all maximal cliques** Instead of finding all  $k$ -cliques, as a first step, the algorithm searches for all maximal cliques. Instead of finding all  $k$ -cliques, which is a polynomial problem, this can be done in exponential time the authors state. Maximal cliques cannot be subsets of larger cliques. That is why they are detected in decreasing order of their size. The largest possible clique size  $s_{max}$  is determined by the maximal degree  $d_{max}$  found in the network.

- (1) Determine  $s = s_{max}$ .
- (2) Repeatedly choose a node  $v$  from the graph:
  - (2.1) Extract all cliques of size  $s$  containing  $v$ .
  - (2.2) Delete the node and its edges.
- (3) When no nodes are left set  $s = s - 1$  and start with (2) on the original graph.

The set of already found cliques do influence the found cliques in later steps, as the later found cliques are smaller. The detailed algorithm for step (2.1) finding cliques of size  $s$  of  $v$  can be looked up in supplementary material to the paper on section 1.1.2, page 3. The result of this step is a set with size  $n_c$  of all maximal cliques, an example can be seen in figure 2a.

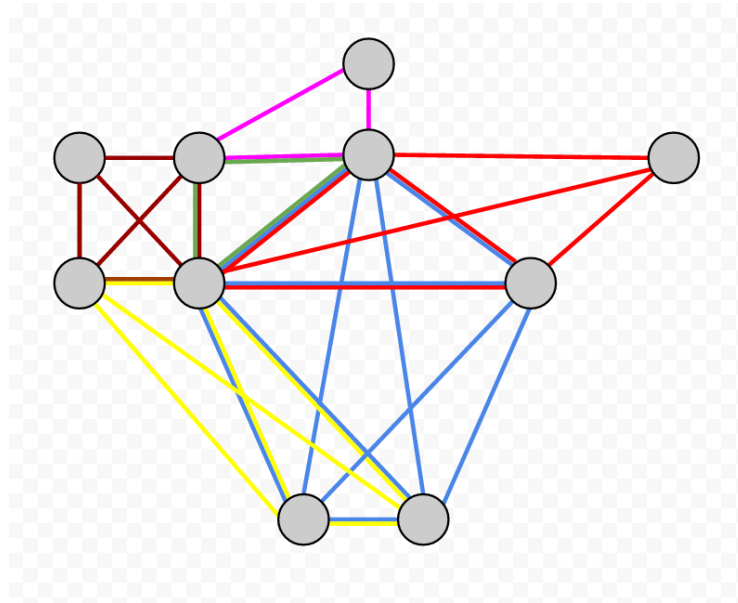


Fig. 5: All maximal cliques in a graph. Each color represents a maximal clique.

**(2) Prepare clique-clique overlap matrix, or clique graph** The dimension of the overlap matrix is  $n_c \times n_c$ . Each row and column represent a clique. All non-diagonal matrix elements represent the number of nodes those cliques share. The diagonal entries represent the size of the corresponding clique. This matrix is only generated once. In figure 6 the clique-clique overlap matrix of the graph of figure 5 and the corresponding clique graph is shown.

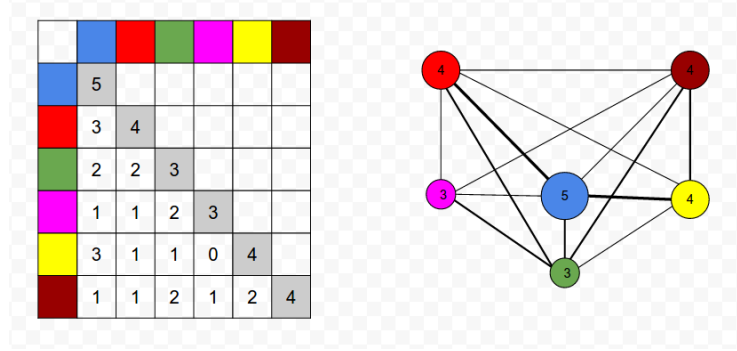


Fig. 6: The clique-clique overlap matrix and the corresponding clique graph. The number inside the node is the number of nodes belonging to that community, the width of the edge represents the shared nodes.

**(3) Threshold the matrix** All off-diagonal entries smaller than  $k - 1$  and diagonal entries smaller than  $k$  are set to 0. The remaining elements are set to 1, resulting in a binary matrix, representing a network of cliques. The thresholding can be done repeatedly without calculating a new matrix, see figure 7.

**(4) Each connected component in the clique graph form a  $k$ -clique-community.** In the resulting graph we need to look for connected components, those represent the  $k$ -clique-communities, see figure 8.

**(5) Post-processing.** The last setp of the algorithm assigns left over nodes, if possible, to communities. A node which is only connected to one certain community and is not part of a community already is associated with this community. The number of nodes belonging to a community can be increased.

### 3.2 Construction of the co-authorship network

The edge weights are calculated with

$$\frac{n}{n-1}$$

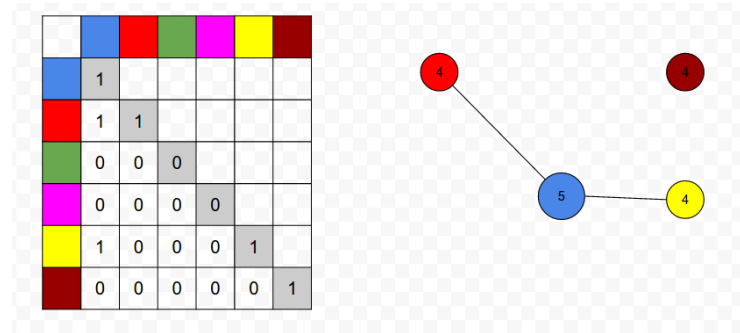


Fig. 7: This is the resulting clique graph after the thresholding with  $k = 4$ . All connected components do represent a  $k$ -clique-community. In this example are two communities.

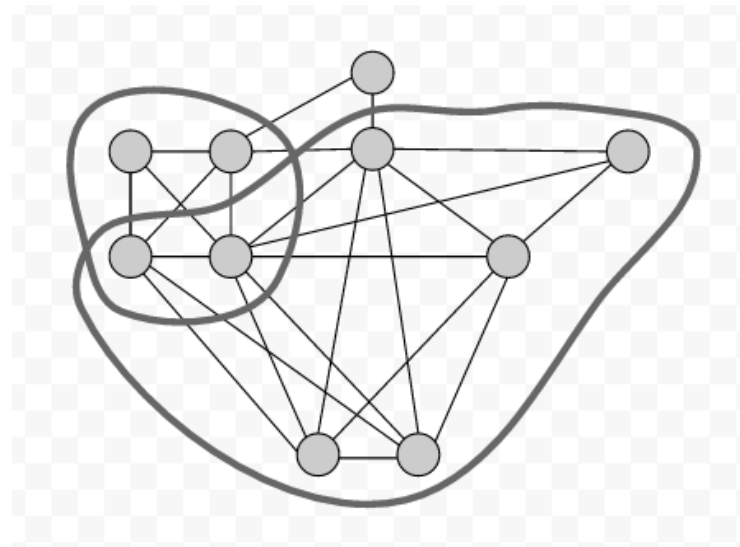


Fig. 8: Here the two resulting  $k$ -clique-communities are shown.

with  $n$  being the number of authors per publication. This results in modeling the collaboration between fewer authors as more intensive than the collaboration between a higher number of authors. To generate an unweighted network all edges with an edgweight below a certain threshold  $w^*$  need to be removed, all others edgweights are replaced by weight 1. Increasing  $w^*$  results in fewer edges throughout the network, but stronger links and fewer detected communities.

In addition a good  $k$  needs to be chosen. When  $k$  is increased the detected  $k$ -clique-communities shrink, but at the same time become more cohesive (stronger connected). So a good balance between  $k$  and  $w^*$  needs to be found resulting in a community structure which is highly structured.

The authors state that  $k$  is usually between 3 and 6. If the number of links in a network are increased above some point a giant component evolves and covers the underlying community structure. So for each  $k$  the threshold  $w^*$  is lowered until the largest component becomes twice as big as the second largest component. For really choosing the best  $k$  some further characteristics describing the community structure need to be investigated, see Supplementary Information of [9] for more details.

### 3.3 Main Findings of the paper

Overlaps in networks are significant. The distributions introduced in the paper (community size, community degree, overlap size, membership number) reveal universal features of networks. The network of communities has non-trivial correlations and specific scaling properties. [9]

## 4 Community Evolution

Palla et al. [8] developed an algorithm based on clique percolation (see section 3) that allows the investigation of overlapping communities over time. They uncovered basic relationships characterizing community evolution within a co-authorship network and a phone-call network. The following section describes the main steps of the algorithm. A short summary of their main findings can be found in section 4.4.

### 4.1 Algorithm

The starting point of the algorithm is an undirected and unweighted graph for each timestep. The construction of those temporal graphs is explained in section 4.2. In general, the algorithm uses the clique percolation method to find communities in each temporal graph, and after that matches the communities of consecutive timesteps.

- (1) Extract communities with CPM for each graph  $g_t$  at time step  $t$ .
- (2) Match set of communities at consecutive time steps of graph  $g_t, g_{t+1}$ , as follows:
  - (2.1) Construct joint graph  $g_{\cup} = g_t \cup g_{t+1}$ .
  - (2.2) Extract communities  $V$  with CPM in joint graph  $g_{\cup}$ .
  - (2.3) For each extracted community  $V_i$ :
    - Extract communities in  $g_t$  and  $g_{t+1}$  that are contained in  $V_i$ .



- Calculate relative overlap for each pair.
  - Match communities in decending order.
- (3) Post-processing.

The following describes steps (2) and (3) in detail. Step (1) was already explained in section 3.

**Matching Communities** For matching communities, the *relative node overlap*  $C(A, B)$  between two nodes  $A$  and  $B$ , in a simple way, is defined as

$$C(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

As overlapping communities are allowed the matching from consecutive time steps in decending order of their relative node overlap can lead to missmatching. For example when small communities gain a lot of members or vice versa. An example for this problem is given in figure X [TODO-PIC]. As a solution, for each time steps  $t$  and  $t + 1$  a joint graph  $g_{\cup}$  is constructed, containing all links from both networks. Let  $D$  be the set of communities at time step  $t$  and  $E$  the set of communities at time step  $t + 1$ . The set of communities from the joint graph  $g_{\cup}$  are extracted using CPM and are called  $V$ . For any community  $D_i \in D$  or  $E_j \in E$  exactly one community  $V_k \in V$  can be found. For checking whether  $E_i$  or  $D_j$  is contained in  $V_k$  the links are compared instead of nodes. For each community  $V_i \in V$  the set of communities  $D_i^k \in D$  and  $E_j^k \in E$  contained in  $V_i$  are extracted. Now the relative node overlap between every possible pair can be calculated as with

$$C_{i,j}^k = \frac{|D_i^k \cap E_j^k|}{|D_i^k \cup E_j^k|}$$

and the pairs can be matched in decending order.

In figures X three examples are given: figXa is a simple matching of a propagating community, figXb showing two merging communities with one community dying and figXc showing the splitting of a community into two communities with one community is born.

**Post-processing** In some cases a community which was disintegrated at a certain time step suddenly reappear in a later timestep, due to low publishing rates for example. That means a newborn community includes a formerly dead community. This problem is overcome by just filling the gap with the last step of the almost disintegrated community, this is called gap-filling.

## 4.2 Construction of the temporal co-authorship networks

Events in the co-authorship network are publications. The social connection between people writing a paper together usually starts before the event and last for some time after the event. The higher the frequency the closer the relationship [11].

The edge weight resulting from one paper is  $n/(n-1)$  with  $n$  as the number of authors. The edge weight between nodes  $a$  and  $b$  at a certain time  $t$  is calculated as

$$w_{a,b}(t) = \sum_i w_i e^{\frac{-\lambda|t-t_i|}{w_i}}.$$

The summation runs over all collaboration events in which  $a$  and  $b$  are involved. The event  $i$  occurs at time  $t_i$  and the corresponding edge weight at this time is called  $w_i$ . This decay function models the strength of collaboration between authors over time considering all events ever occurred in time.

A threshold  $w^*$  is used to only include certain edges to the temporal graph. So for each timestep a graph can be constructed based on the collaboration strength per time step.

The authors of the paper used  $w^* = 1.0$  for the co-authorship network. No details about  $\lambda$  is given in the paper.

[TODO explain how  $w$  and  $k_a$  are chosen.]

The data set used in the paper contained 142 months of publications. Details about data aggregation are not given, but looking at the figures it looks like 50 timeslices.

### 4.3 Main Findings of the paper

The paper summarized differences between large and small communities and their development over time. Small communities live longer if their members stay the same over time. If members in small communities change frequently, they only live for a short time. Large communities live longer if members are changed permanently, if members stay the same they die quickly.

## 5 Implications regarding the geochemistry dataset

This part describes the steps I would need to undertake to carry out a temporal analysis of communities within the geochemistry dataset or with any other dataset, which forms a collaboration network.

- (1) Generate slices of the network for each year.
- (2) Find function for calculating edge weights.
- (3) Apply edge weights to each snapshot.
- (4) Find out  $k$ .
- (5) Adjust threshold  $w^*$  according to  $k$ .
- (6) Do CPM for each snapshot.
- (7) Conduct community mapping.

It is also important to look at basic statistics after (1) timeslicing. For example to look at how many publications per timeslice, degree distributions, nodes on average per timeslice to check if timeslicing was done right or needs to be adjusted. The function for calculating edge weights (2) could be used from the paper (see

section 4.2), but it would need to be checked if the resulting networks (3) are still valid. For finding the right  $k$  (4) and  $w^*$  (5) and extended analysis of network characteristics like coverage of community structure and distribution of community size (see 4.3) would need to be carried out. Also the quality of the detected communities would need evaluation, comparing density within the community to outside the community. Using CFinder or R for CMP (6) detection. It is open to me if there is an implementation for the community mapping (7), if not I would need to implement it. Time complexity will be no problem because my dataset is not very big. About 2.000 paper resulting in about 500 nodes.

## 6 Future Work

As this paper discusses only one method an extensive literature research should follow, finding out more methods for that problem. Also a closer look on recent work would be good. The results from this dataset should be compared to the results of Palla. Maybe some time should be invested for finding a good representation for edge weights calculation, related to collaboration definition. The visualization of the community evolution remains open and would be a broad topic, but exciting to investigate further.

## References

1. Albert-Laszlo Barabási, Hawoong Jeong, Zoltan Nédá, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3):590–614, 2002.
2. Yaozu Cui, Xingyuan Wang, and Junqiu Li. Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient. *Physica A: Statistical Mechanics and its Applications*, 405:85 – 91, 2014.
3. Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
4. Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
5. Andrea Lancichinetti, Santo Fortunato, and Jnos Kertsz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
6. Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.
7. Mark EJ Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the national academy of sciences*, 101(suppl 1):5200–5205, 2004.
8. Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.
9. Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
10. Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

11. José J Ramasco and Steven A Morris. Social inertia in collaboration networks. *Physical Review E*, 73(1):016122, 2006.
12. Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706 – 1712, 2009.
13. Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.