

University of Szeged  
Faculty of Science and Informatics  
Department of Computational Optimization

**Methods for the description and analysis of  
processes in real-life networks**

Theses of PhD Dissertation

**András Bóta**

*Supervisors:*

Miklós Krész, Dr.  
András Pluhár, Dr.

Szeged, 2014.



## Abstract

The roots of graph theory lead back to the puzzle of Königsberg's bridges. In 1736 Leonhardt Euler published a paper detailing this problem, and also proposed a solution for it. Since then much has been learned about the mathematical properties of graphs, and the field has a long history of applications including sociology, biology or operations research and optimization. Things changed when computers became accessible and affordable to most researchers, allowing them to collect, store, share and study large amounts of data on graphs observed in real-life. This gave rise to the interdisciplinary field of network science, which is dedicated to the description and analysis of real-life graphs borrowing methods from mathematics, physics, computer science and sociology. The topics of network science include the study of the organization principles of networks, for example their degree distribution and the formation of groups, as well as several processes taking place on the networks themselves.

The goal of the dissertation is to provide an overview on the authors' works focusing on three major topics of network science: overlapping community detection, dynamic community detection and the study of infection processes. Based on our experiences with known community detection algorithms and the works of Csizmadia et al. we have developed an overlapping community detection method, the hub percolation method, that is capable of handling various types of networks or discover different layers of the community structure of single network. We have tested our method on several benchmarks including Newman's networks and Lancichinetti's community based graph generator. We have given two real-life case-studies: one of them examines the properties of an ownership network of Hungarian companies. The other one compares the community structures of a Hungarian and an English word association graph.

Motivation for examining another aspect of community detection – dynamic community detection – came from applications on other social and economic networks. Based on the works of Palla et al. we have developed an algorithm to efficiently match the communities of two neighboring networks in a time series of graphs. Our method distinguishes eleven community events, and its mechanism does not depend on the specifics of the used community detection algorithm. We have evaluated our method on two real-life networks.

The last topic of this dissertation is the study of infection processes in networks with a focus on economic applications. A common problem of these applications is, that the edge infection probabilities required to compute these processes are often not available, missing values are either estimated or constants are used. In order to provide a more systematic approach we have proposed the Inverse Infection Problem, its task being the computation of the edge infection probabilities with the help of observations on the beginning and the end of the process. We have proposed the Generalized Cascade model to help with the computations required for this model, as well as several heuristics to further improve performance. We have given a learning method based on Particle Swarm Optimization as a solution. Finally we have tested the performance of our solution on benchmark networks, and we have also published a case-study dealing with the application of our method in the estimation of credit

default in banking.

## 1 Basic concepts and preliminaries

The algorithms and methods in this work are defined on *graphs*, with *vertex* (*node*) and *edge* sets  $V(G)$  and  $E(G)$  for a graph  $G$ . In almost all cases the graph is considered to be *undirected*. The graph may contain numerical information stored as *weights* on the edges, or *attributes* on both edges and vertices. In all cases the graph is considered to be *connected*, although this might be the result of some filtering process not mentioned in the dissertation.

All of the graphs in the dissertation are *complex networks*. This term was introduced to describe graphs coming from diverse real-life examples, and these networks often contained non-trivial topological features unlike the ones that occur in the more simple graph classes like trees or lattices. Complex networks can be divided into several categories based on the source of the network: social networks, information networks, technological networks, economic networks, etc. One of the most important findings of network science is, that even though the source of them is different, the structure and behavior of the networks are very similar.

### Community detection

A behavior observed in social interactions is, that people tend to form groups according to their lines of interest, occupation, etc. In networks, this behavior presents itself as a tendency for nodes forming into sets, so that the nodes inside the sets are densely connected and the links between the sets are relatively sparse. This feature is called *community structure*, and the discovery of this structure is called *community detection*. This property is not only present in social networks, but many other complex networks as well.

While the phenomenon of communities is well observed, an exact definition is difficult to find. Traditional community detection methods consider communities to be disjoint vertex sets, and adopt the following intuition: they are looking for a partitioning of the nodes, which maximizes the number of edges between the nodes inside the sets, and minimizes them between the sets. It is also a goal to find meaningful communities, i.e. they discard trivial solutions of the problem (like a single community containing all of the vertices). A comprehensive review of community detection can be found in [14].

The traditional definition of community allows disjoint vertex sets only. Based on the observation that in real-life networks, nodes can belong to multiple communities, Palla et al. introduced the concept of *overlapping community detection* and proposed the clique percolation method [29]. The idea of finding maximal cliques and joining them according to some criteria is the basis of several overlapping community detection algorithms, but there are many other approaches as well [17,18,25].

Another aspect of community detection we have discussed in the dissertation is *dynamic community detection*, the changing of communities in time. Dynamic community detection deals with a time series of graphs  $\{G_i\}_{i \in \mathcal{T}}$ , where  $\mathcal{T} = \{1, \dots, \ell\}$

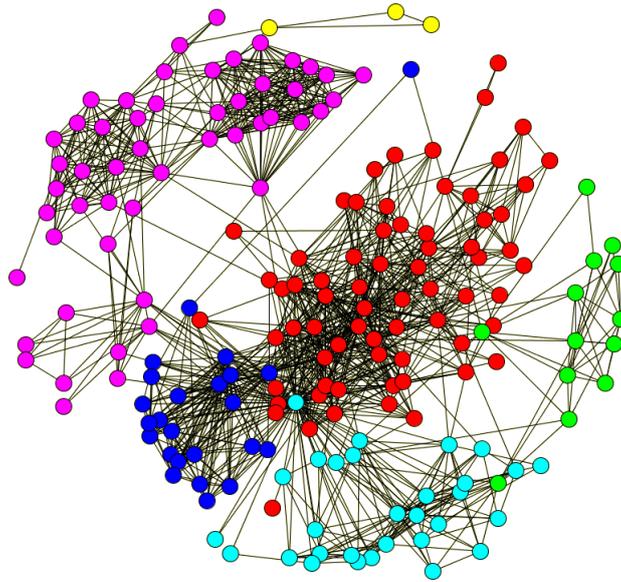


Figure 1: The community structure of a fraction of a social networking site. The communities were created with the method of Girvan and Newman [26].

represents a discrete time set, and the task is discovering how the community structure of one graph relates to the community structures of the others in the series. Algorithms for dynamic community detection usually consider two neighboring time instances, and most authors begin with the definition of *events* that may occur to communities between the instances. After the definition, the communities of the two graphs are matched with some algorithm according to the defined community events. The most cited algorithms can be found in [1, 28].

### Infection models

Graphs serve as the basis of many processes. A significant part of this dissertation deals with the spread of economic events, which falls into the category of *infection models*. These models deal with the diffusion or spread of a wide array of things: behavior, information, influence, diseases, etc. These methods were studied multiple times by many different branches of science. The Independent Cascade model was proposed in [12, 19], and forms the backbone of many results in our works.

Infection models assign states to the nodes of the network corresponding to the different phases of infection. The states of the Independent Cascade model, are the following. A node is *susceptible* if it is not yet infected, a node is *infected*, if it is infected and may infect others, and a node is *removed*, if it is infected but not infectious. Sometimes we will call these states as *inactive*, *newly activated* and *active*.

Any infection model can be described as a process, that has two inputs: the first one is a weighted graph, where the edge weights are real values between 0 and 1:  $\forall e \in E(G), 0 \leq w_{u,v} \leq 1$ , these are called as edge infection values. The second input is the set of initial infectors  $A_0 \subset V(G)$ . These nodes are considered as infected at

the beginning of the process. The infection process happens in discrete time steps. The process terminates at iteration  $t$  if  $A_t = \emptyset$ , and results in the set of infected nodes  $A = \bigcup_{i=0}^t A_i$ . The original formulation of the model was for directed graphs, but it is easy to generalize them for undirected ones by assuming, that the edge infection values are symmetrical  $w_{u,v} = w_{v,u}$ .

The specific way one vertex infects another varies depending on the model is the following: let  $A_i \subseteq V(G)$  be the set of nodes newly activated in iteration  $i$ . In the next iteration  $i + 1$ , each node  $u \in A_i$  tries to activate its inactive neighbors  $v \in V \setminus \bigcup_{0 \leq j \leq i} A_j$  according to the edge infection probability  $w_{u,v}$ , and  $v$  becomes active in iteration  $i + 1$ , if the attempt is successful. If more than one node is trying to activate  $v$  in the same iteration, the attempts are made independently of each other in an arbitrary order within iteration  $i + 1$ . If  $A_t = \emptyset$ , the process terminates in iteration  $t$ . It is easy to see, that the process always terminates in a finite number of iterations.

## 2 Topics of the dissertation

As we have mentioned in the abstract, the topics of the dissertation can be divided into three major topics:

1. The development of a new high-resolution clique-based overlapping community detection algorithm with customizable parameters and its applications.
2. The development of a dynamic community detection algorithm able to handle large real-life networks.
3. The development of a new infection model and a methodology capable of estimating the edge infection probabilities on bank transaction and other networks.

We will discuss each of these in this section.

### 2.1 The hub percolation method

We have created the *hub percolation method* with the aim to develop a versatile tool for community detection, that is flexible enough to handle multiple applications with different requirements. It is a high-resolution clique-based overlapping community detection algorithm for complex networks. The parameters of this method govern the various aspects of the resulting communities, like the number and size of the overlaps between them. This allows us to discover different kinds of communities from large, loosely overlapping groups to ones with a dense, highly overlapping structure. The development of the hub percolation method was heavily influenced by our works with the clique percolation method [29] and the  $N^{++}$  method in [3] as well as several well known benchmark networks like Zachary's karate club [30] and Newman's works [15, 27]. Much of the details of this algorithm came from experiences gained during test runs on these networks.

The hub percolation method has two simple ideas at its core: *cliques* and *hubs*. Cliques are fully connected subgraphs, with a given number of nodes: a  $k$ -clique is a fully connected subgraph with  $k$  nodes. A clique is maximal if it is not a subset of any other cliques. Finding all maximal cliques is difficult in arbitrary graphs, but in complex networks the variants of the Bron-Kerbosch method perform well enough [10,13]. A concept common in all overlapping community detection methods is, that cliques are natural communities, we will also use this intuition. An important observation on real-life networks is, that inside a community some members are more important than others with respect to the role of the nodes in connecting different communities. We will denote these nodes as hubs, and we will use them to extend and join the cliques of a given graph. The identification of hubs is the task of the *hub selection strategy*, we have suggested several of these in the dissertation.

The hub-percolation method can be divided into the following steps:

1. Find the set of all maximal cliques of size greater or equal than 3.
2. Select the set of hubs according to a hub selection strategy. The strategy may have a parameter  $q$ .
3. Identify the  $k$ -cliques formed by hubs and extend them with a limited percolation rule.
4. Join communities with the same hubs.

The hub-percolation method [4] provides several ways to fine-tune its results. One of them is the filtering parameter  $k$  of phase 3, another is the hub selection strategy and its possible parameter  $q$ . This allows our method to handle diverse real-life networks. We have measured the effect these parameters have on the results of the method in several ways: the number of communities, the community size distribution, the average overlap between the communities and the number of nodes left without communities. We have used the benchmark networks of Newman for this purpose [15,27].

## Evaluation and case-studies

We have evaluated the performance of our method in several ways. We have used the community based graph generator of Lancichinetti and Fortunato [23] to compare the results of our method to the clique percolation method of Palla et al. [29]. We have used the mutual information metric [24] to compare the results of these methods with the natural community structure of the generated networks<sup>1</sup>. We have shown, that our method has better performance if the network has a high number of overlapping nodes.

We have also presented two case-studies. We examined the communities of an economic network constructed from the Hungarian company register. We examined three aspects of the companies: the geographical location of them, the industrial sector they belong to and the age of the companies. The observed companies were

---

<sup>1</sup>These were given by the graph generator.



to us. In the end we chose the  $N^{++}$  method [3].

The last difference was the kinds of *events* communities go through during the temporal dynamics of the network. We have also found that the basic events Palla et al. used are not enough to explain the events that occur in the networks we were observing.

Our approach is based on the one proposed by Palla et al., but we made several modifications, that improve the usefulness of the method. We extended the number of community events, and we gave a different matching algorithm, that does not depend on the specific properties of the used community detection method.

Our dynamic community detection algorithm can be divided into the following steps. It works by comparing two subsequent graphs in a time series,  $G_1$  and  $G_2$ .

1. The communities of  $G_1$  and  $G_2$  are computed by a community detection method.
2. The union of the two graphs  $G_U$  is computed, and the community structure of this new network is computed as well.
3. The communities of  $G_1$  and  $G_2$  are matched to the ones of  $G_U$ .
4. The results of these matchings are summarized resulting in the community events, that happen between  $G_1$  and  $G_2$ .

We have used our method to evaluate the dynamics of the community structures of two real-life networks. Our findings indicate, that there is a significant difference between these datasets. One of the networks is more stable, but loses communities steadily, the other network is more dynamical, but maintains its community number. We have also examined the changes in community structure in relation with the sizes of the involved communities.

## 2.3 Inverse infection

The idea of the *Inverse Infection Problem* comes from the previous work of Csernenszky et al. [11] published in 2009. This work focused on the possible applications of infection models on bank transaction networks. A conclusion of this work was, that the precision of current methods for the prediction of credit default can be improved by taking into account the effect members of the network have on each other. This is the so-called network effect: the probability of a node influencing each of its neighbors. This led to idea, that these influence values should be estimated somehow, and over time, to the formulation of the Inverse Infection Problem [8].

Our approach to the prediction of the influence (or infection) probabilities between nodes is different from the ones existing in literature [16,22], although the field is scarcely studied. Compared to these methods, ours does not require information on the individual steps of the infection process. Instead it builds on the experiences gained from the previously mentioned work on bank transaction networks, such as the use of estimations of the default probabilities for individual companies and additional information characterizing the connections between them.

The development of the Inverse Infection Problem required the creation of additional methods and algorithms, starting with the generalized infection framework, the *Generalized Cascade Model* [7]. Based on the Independent Cascade Model, this generalized model uses a probabilistic framework to describe the inputs and outputs of an infection process.

### The Generalized Cascade Model

In the Generalized Cascade (GC) model each vertex is assigned a real value  $p_v$  between zero and one, that represents the probability of infection before the beginning of the process. We refer to these values as the *a priori distribution*. Vertices may become infected independently from each other before the beginning of the process according to their a priori infection probability  $p_v$ . The a priori distribution will be the equivalent of the initial infectors of the Independent Cascade model. The GC model is capable of summarizing the effect of the a priori infections and the effect of these infections transmitted through the network. Similarly to the a priori distribution, the output of the model is given as an *a posteriori distribution*, where values  $p'_v$  indicate the probability of being infected during the process for all  $v \in V$ . The actual way a vertex infects another is the same as in the IC model, although it is possible to use other infection models in the terms of the GC model.

We can define the Generalized Cascade model in the following way:

*The Generalized Cascade Model:* Given an appropriately weighted graph  $G$  and the a priori infection distribution  $p_v$ , the model computes the a posteriori distribution  $p'_v$  for all  $v \in V(G)$ .

The computation of the GC model is difficult, so we have proposed four heuristics methods of it in [5].

- Complete Simulation is the direct adaptation of the method in [20] for the GC model.
- The Edge Simulation method is a combination of both simulation and exact computations that decreases the standard deviation appearing in other simulations.
- If the infection probabilities are small, then the infections typically do not travel far from the source of infection. Neighborhood Bound Heuristics exploits this property.
- The Independent Cascade model itself can be substituted for a similar, but a computationally more tractable model, the ALE model.

### The Inverse Infection Problem

The definition of the Inverse Infection Problem [8] is very similar to the GC model:

*Inverse Infection Problem:* Given an unweighted graph  $G$ , the a priori and the a posteriori probability distributions  $p_v$  and  $p'_v$ , compute the edge infection probabilities  $w_e$  for all  $e \in E(G)$ .

Independently estimating all edge weights of a network is both underdetermined and computationally unfeasible, even if the number of edges is small. Instead, we assume the edge probabilities can be expressed as (normalized) functions of some properties of the edges or nodes that are available in the form of attributes. This way we only have to estimate the coefficients of these functions, and since the number of attributes and coefficients is limited, the problem becomes tractable.

We can define a learning method based on the formulation above.

- The problem definition states, that the a posteriori distribution is required as an input of any algorithm. In the case of a learning algorithm, it is considered to be a test or reference dataset.
- Then the initial coefficients for the edge attribute functions are chosen from reasonable bounds.
- Given these attribute functions, the coefficients and the a priori distribution we can compute an a posteriori distribution corresponding to the chosen coefficients.
- Finally, an error function calculates the difference between the reference set and the newly calculated infection values. The process aims to minimize this error function by repeatedly adjusting the coefficients.

This is a typical task for global optimization, and after several tries we have selected the Fully Informed Particle Swarm method of Kennedy and Mendes [21].

We have used artificial infection scenarios to explore the boundaries of our method. We have tested the stability and accuracy of the optimization, we have given a general approach to choose the correct attribute functions, we have examined the implications of choosing between the heuristics of the GC model and we have tested our method in low-quality inputs as well. The learning method is able to accurately predict the edge infection probabilities in a small number of iterations while the number of attributes and the shape of the attribute functions have only a small effect on this even if the quality of the inputs is low.

Finally, we have presented a case-study of the Inverse Infection Problem on a bank transaction network [2]. The goal of this application was to improve the efficiency of existing models for the prediction of short-time credit default events. Since the creation of IIP was heavily influenced by banking applications, it was quite suited to handle this task. Our model has better predictive power than traditional methods: it can identify the companies where default is most probable better than the previously used models of the bank. Our model was implemented in August 2013 into the OTP Bank of Hungary's credit monitoring process.

### 3 Summary of the results of the dissertation

The results presented in the dissertation are the following.

### 3.1 Overlapping community detection

Our first goal was *the development of a high-resolution clique-based overlapping community detection algorithm with customizable parameters with several applications.*

1. We have given a detailed description of the hub percolation method and several hub selection strategies including one that is able to handle weighted networks in a natural way. We have examined how the hub selection strategy and the filtering parameter influences the community structure found by the algorithm, and we have demonstrated this effect on several of Newman's benchmark networks [15, 27].
2. We have used the community based graph generator of Lancichinetti et al. [25] to compare the performance of our method to a very popular clique-based method, the Clique Percolation Method [29].
3. We have used our method to examine the community structure of an economic ownership network constructed from the Hungarian company register.
4. We have provided a case-study on the community structure of an English and a Hungarian word association graph.

The hub percolation method itself, its evaluation on benchmark networks and the economic case-study (1-3) was submitted for publication in an international journal [4] and is yet to appear. The case-study on word association graphs (4) was accepted for publication in the proceedings of an international conference [9].

### 3.2 Dynamic community detection

The second topic was *the development of a dynamic community detection algorithm able to handle large real-life networks.*

1. We have introduced the eleven community events our method can detect. We provided a detailed description of the dynamic community detection method, and elaborated on its time complexity.
2. We have used our method to evaluate the dynamics of the community structures of two real-life networks.

All of our works in this topic was published in a journal paper [6].

### 3.3 Inverse Infection

The final topic of the dissertation was *the development of a new infection model and a methodology capable of estimating the edge infection probabilities on bank transaction and other networks.*

1. The Generalized Cascade model is an infection model and a generalization of the Independent Cascade Model. The Inverse Infection Problem is based on this model.

2. The computation of the GC model is #P-hard, therefore we have proposed four heuristic methods for it: Complete Simulation, Edge Simulation, Neighborhood Bound Heuristics and the Aggregated Linear Effect model.
3. We have proposed the Inverse Infection Problem: Given an unweighted graph  $G$ , the a priori and the a posteriori probability distributions  $p_v$  and  $p'_v$ , compute the edge infection probabilities  $w_e$  for all  $e \in E(G)$ .
4. We can define a learning method based on the formulation above which reduces the problem to global optimization.
5. We have used artificial infection scenarios to explore the boundaries of our method. We have tested the stability and accuracy of the optimization, we have given a general approach to choose the correct attribute functions, we have examined the implications of choosing between the heuristics of the GC model and we have tested our method in low-quality inputs as well.
6. Finally, we have presented a case-study of the Inverse Infection Problem on a bank transaction network.

Our preliminary works (1, 3) on inverse infection and the GC model was published as an extended abstract in an international journal [7]. After this, we have given a thorough examination of the GC model and proposed four heuristics for it (1, 2) in another journal publication [5]. The inverse infection problem itself, the learning method and the artificial infection scenarios (3-5) appeared in the proceeding of an international conference [8]. The banking application (6) was accepted for publication in a well-known international journal [2].

# Bibliography

- [1] S. Asur, S. Parthasarathy and D. Ucar, An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data* **3**(4), 2009.
- [2] A. Bóta, A. Csernenszky, L. Gyórfy, Gy. Kovács, M. Krész, A. Pluhár, Applications of the Inverse Infection Problem on bank transaction networks. Accepted for publication in: *Central European Journal of Operations Research*.
- [3] A. Bóta, L. Csizmadia, A. Pluhár, Community detection and its use in Real Graphs. *Proceedings of the 2010 Mini-Conference on Applied Theoretical Computer Science*, 95–99, 2010.
- [4] A. Bóta, M. Krész, A high resolution clique-based overlapping community detection algorithm for small-world networks. Submitted.
- [5] A. Bóta, M. Krész and A. Pluhár, Approximations of the Generalized Cascade Model. *Acta Cybernetica* **21** 37–51, 2013.
- [6] A. Bóta, M. Krész, A. Pluhár, Dynamic Communities and their Detection. *Acta Cybernetica* **20** 35–52, 2011.
- [7] A. Bóta, M. Krész and A. Pluhár, Systematic learning of edge probabilities in the Domingos-Richardson model. *Int. J. Complex Systems in Science*, **1**(2) (2011) 115–118.
- [8] A. Bóta, M. Krész and A. Pluhár, The inverse infection problem. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 75–83, IEEE, 2014. <http://dx.doi.org/10.15439/978-83-60810-58-3>.
- [9] A. Bóta, L. Kovács, The community structure of word association graphs. Accepted for publication in: *The Proceedings of the 9th International Conference on Applied Informatics*, Eger, 2014.
- [10] C. Bron, J. Kerbosch, Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM* **16** (9): 575–577, 1973. doi:10.1145/362342.362367.
- [11] A. Csernenszky, Gy. Kovács, M. Krész, A. Pluhár, T. Tóth, The use of infection models in accounting and crediting. *Challenges for Analysis of the Economy, the Businesses, and Social Progress Szeged* (2009) pp. 617–623.

- [12] P. Domingos, M. Richardson, Mining the Network Value of Costumers. *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, ACM (2001) 57–66.
- [13] D. Eppstein, D. Strash, Listing all maximal cliques in large sparse real-world graphs. *Experimental Algorithms*, Springer Berlin Heidelberg, 364–375, 2011.
- [14] S. Fortunato, Community detection in graphs. *Physics Report*, **486**(3):75–174, 2010.
- [15] M. Girvan, M. E. J. Newman, Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, **99**(12):7821–7826, 2002.
- [16] A. Goyal, F. Bonchi, L.V.S. Lakshmanan, Learning influence probabilities in social networks. *Proceedings of the third ACM International Conference on Web search and data mining*. ACM (2010) 241–250.
- [17] S. Gregory, Finding overlapping communities in networks by label propagation. *New J. Phys.*, **12**(10):103018, 2010.
- [18] E. Griechisch, A. Pluhár, Community Detection by using the Extended Modularity. *Acta Cybernetica*, **10**:69–85, 2011.
- [19] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the Spread of Influence though a Social Network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM (2003) 137–146.
- [20] D. Kempe, J. Kleinberg, E. Tardos, Influential Nodes in a Diffusion Model for Social Networks. *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, Springer-Verlag (2005) 1127–1138.
- [21] J. Kennedy, R. Mendes, Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. **36** (4) (2006) 515–519.
- [22] M. Kimura, K. Saito, Tractable models for information diffusion in social networks. *Knowledge Discovery in Databases*, Lecture Notes in Computer Science Springer Berlin / Heidelberg, (2006), 259–271.
- [23] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, **80**(1):016118, 2009.
- [24] A. Lancichinetti, S. Fortunato, J. Kertész Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, **11**(3):033015, 2009.
- [25] A. Lancichietti, F. Radicchi, J. J. Ramasco, S. Fortunato, Finding statistically significant communities in networks. *PLoS One*, **6**(4):e18961, 2011.

- [26] M. E. J. Newman, Detecting community structure in networks. *Eur. Phys. J. B* **38**, 321–330 (2004).
- [27] M. E. J. Newman, The structure of scientific collaboration networks. *PNAS*, **98**(2):404–409, 2001.
- [28] G. Palla, A.-L. Barabási and T. Vicsek, Quantifying social group evolution. *Nature* **446**, 664–667 (2007).
- [29] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**(7043):814–818, 2005.
- [30] W. W. Zachary, An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 452–473, 1977.