A Network Science Summer Course for High School Students

Florian Klimm[1,2] and Benjamin F. Maier[3,4]

[1] Mathematical Institute, University of Oxford, [2] Department of Statistics, University of Oxford, [3] Department of Physics, Humboldt Universität zu Berlin, [4] Robert Koch-Institute, Berlin

Motivation

Teaching network science is usually omitted in high school curricula. We organised a two-week summer school for German high school pupils with the topic 'Networks and Complex Systems'. Fifteen students took part and it was integrated into an established summer school program called *Deutsche Schülerakademie*.

Course Preperation

Participant Questionnaire To assess the knowledge of the pupils we conducted a non-anonymous online survey. We mainly asked questions about their mathematical knowledge, for example, 'Do you know what a matrix is?'.

Reader Based on the results from the questionnaire we provided the participants with a so-called *Reader*. It mainly consisted of an introduction into linear algebra and a short online course into programming with Python.

Course Structure

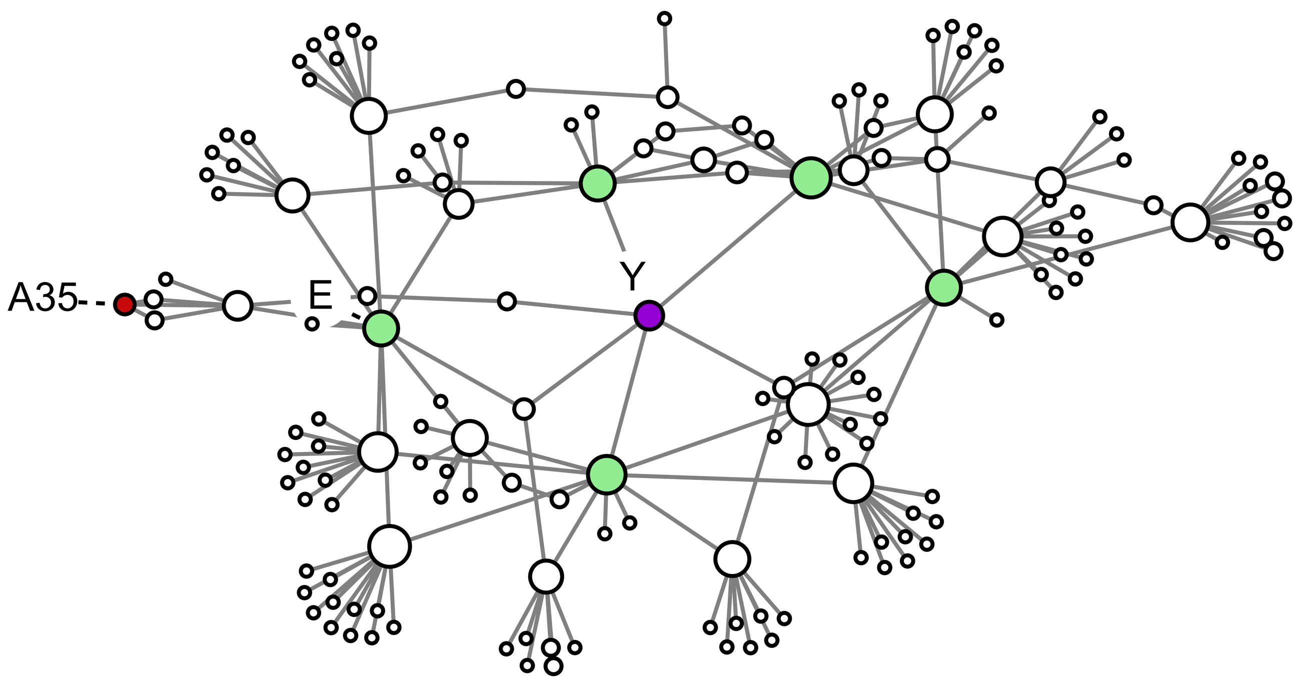
Student Presentations Each of the participants gave a presentation during the course. In preparation of the course we suggested a topic to each of them based on their interests as deduced from the survey and provided them with reading material. The talks were supposed to be 10–15 min long and we advised the participants to approach the topics from a phenomenological side. The topics ranged from the 'Seven bridges of Königsberg' to 'Turing patterns'.

Lectures Lectures were a fundamental part of the course. Usually, they covered concepts that were introduced by the pupils earlier in the presentations. We subsequently formalised the mathematical description and went into more detail.

Teaching Modules In addition to longer lectures we also used a couple of smaller modules, which had a stronger component of pupil involvement and interactive discussion. We introduce two of them on this poster.

Problem Sheets We created six problem sheets that gave the pupils the chance to apply some of the learned concepts. The tasks consisted of a mixture of pen-and-paper analytical questions and programming exercises. All problem sheets are available online in the original German and an English version.

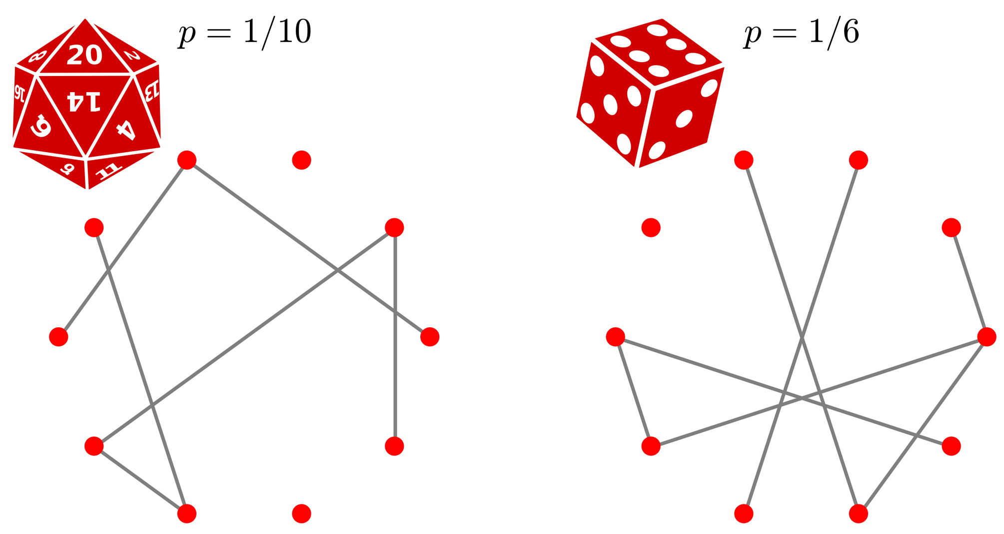
Creating a Floor Plan Network



We wanted the participants to experience a realistic experience of network creation. For this we gave them access to a printout of the floor plan of the *Klosterschule Roßeben*, which hosted the academy. They had the task to create a network in which nodes represent the $n = 161$ rooms and edges stand for direct connections between them. The students organised the splitting up of the network creation task themselves.

Then, the students investigated the degree distribution and compared different centrality measures. They found a positive correlation between degree and betweenness while some nodes contradicted that notion by having a much higher betweenness than expected from their degree. They identified these nodes to represent staircases that connect the hallways with each other.

Dicing ER Graphs



After introducing Erdős-Rényi random graphs $G(n, p)$ we gave the participants the task to create undirected random graphs with the help of six-sided and twenty-sided dice. We discussed how to generate graphs with different connection probabilities p and also identified an appropriate number n of nodes. We settled for $n = 10$ nodes given that 45 dice rolls per network seemed a reasonable time effort. We then split the pupils into four groups, each creating a graph of different connection probability $p \in \{1/20, 1/10, 1/6, 1/3\}$. Each group drew their final graph on the blackboard and we subsequently discussed their properties, e.g. the size of their largest connected component and their degree distributions $P(k)$, and compared them with theoretical expectations.

Course Material

All course material is available on GitHub: `https://github.com/floklimm/network-summer-school` Find more detailed information in our manuscript: `https://goo.gl/tVTJBU`

Scan the QR code for the course material:

Programming

```
# import necessary library
import numpy as np

# define a function
def path_graph( n ):

 #  Returns the (n x n) adjacency matrix
 #  of a path graph with n nodes.


    # empty adjacency matrix of shape (n x n)
    A = np.zeros((n,n))

    # loop over every node besides the last one
    for u in range(0, n-1):

        # connect this node to the subsequent one
        A[u,u+1] = 1

    # symmetrize matrix by adding its transpose
    # the original matrix
    # this ensures that that not only u is conn-
    # ected to v, but the other way around, too.
    A = A + np.transpose(A)

    # the matrix will now be returned
    return A

# call the function to obtain an adjacency matrix
# for a path graph of 10 nodes
path_10 = path_graph(10)

# put the matrix on screen in text
print(path_10)
```

We used Python as the main programming language because it is open source and relatively easy to learn and implement. In addition to the default Python functions we mainly used three libraries, NETWORKX, MATPLOTLIB, and NUMPY. We also used Gephi for the analysis of social networks and as an easy way to graphically access them. Main challenge for the programming task was a disparity in existing knowledge amongst the participants. To tackle this we split the class into a beginner and an advanced group.

Conclusion

We think that network science is a appropriate topic to give pupils an introduction to university-level science and mathematics.

A crucial part of our teaching was the flexibility in the covered topics. Sometimes the pupils had problems with concepts that we thought were taught at school, e.g. the binomial coefficient, while other new concepts took much less time to discuss than we anticipated.

Acknowledgements

FK is funded by the EPSRC and MRC under grant EP/L016044/1 through the Systems Approaches to Biomedical Science Doctoral Training Centre with contribution by *e-Therapeutics Plc*.