Worksheet 0

Setting up a Research Project

Introduction to Programming – Research Project

Florian Klimm & Eoin Malins

Michaelmas 2018

This project is aimed at students that are already familiar with the material covered in the lectures. It is a project similar to a 'real' research project. As in the real world, things can go wrong and the plan for the project might change along the way. So please see the exercises below as a guideline from that you might deviate, if there are good reasons to do so.

In this project, you will construct and analyse *protein–protein interaction networks* (PPINs). Specifically, you will calculate their *topological fractal dimension* (TFD). Both terms are (most likely) new for you but you learn about them along the way. In general, you will learn most mathematical and biological concepts that you need from the assigned readings. Occasionally, there might be terms that you are not familiar with — in this case, just google them.

We strongly recommend using `Python` as the main programming language for this tasks. When you realise that there are computational bottlenecks in this research project you can decide whether you want to use `C` as a replacement of these computationally intensive tasks.[1]

Exercise 0.1 Read about your Research Topic

Before you start code you have to familiarise yourself with the research topic. As you are undertaking an interdisciplinary DPhil this involves reading about biological concepts and understanding mathematical tools.

Subexercise 0.1.1 A Brief Introduction to Networks

Networks (more specifically *graphs*) are abstract mathematical representations of real-world systems. They are used to study many different systems including social networks (e.g., online and offline friendships), infrastructure networks (e.g., roads and the internet connections), and biological networks (e.g., food webs and biological pathways)

To get an insight into the basic mathematical concepts, please read section 'I. Motivation' and subsections A and B in section 'IV. Basic structural properties of networks' in the lecture notes of the course 'C5.4 Networks' for 4th year undergraduate students at Oxford. This is available online under `https://courses.maths.ox.ac.uk/node/view_material/34560` .

For a comprehensive overview over networks and their study, see Newman (2018). An ebook version of this book is available online in the Bodleian Library.

After reading this material you should be able to answer the following questions:

- What is the mathematical definition of a network/graph?
- How can one represent a network in a matrix?
- What is the degree k_i of a node i? How does one compute it from a matrix-representation of a network?

[1] In a real research project, this is also often a fruitful procedure. One would not want to spend a lot of time coding a new algorithm in `C` to then figure out that it doesn't work. Thus, one first prototypes in `Python` (or other high-level programming languages) and then decides which parts to speed up by implementing in computationally-efficient languages.

Subexercise 0.1.2 A Brief Introduction to Protein–Protein Interaction Networks

For this project, we are interested in a specific type of networks that represent the interaction between proteins. Proteins are active biomolecule that are present in all organisms. In the human body there are approximately 20,000 different proteins that have a large variety of different biological functions.

For a brief introduction, see `https://www.ebi.ac.uk/training/online/course/network-analysis-protein-interaction-data-introduction/protein-protein-interaction-networks`

Please read this tutorial. You might skip over the last two sections 'Clustering analysis' and 'Annotation enrichment analysis'. After reading this material you should be able to answer the following questions:

- What are three properties that are often associated with PPINs?
- How can one create a PPIN?
- Are PPINs the 'real' biological networks? Are there any caveats?

Subexercise 0.1.3 NetworkX

For this project, we are using the `NetworkX` library for `Python`. Please read the tutorial (available under `https://networkx.github.io/documentation/stable/tutorial.html`; you can skip the section about 'Multigraphs') and test some of the commands on your computer.

Exercise 0.2 Aim of this Project

After reading the material, you should now be be familiar with the basic concepts of network science and PPINs. As you learned, there are many different tools and approaches to analyse them. While different empirical networks (i.e., networks derived from real-world data) can have very different properties, most of them are said to be *complex*. There are different mathematical definitions of complexity but one description is, that complex networks have a structure that is neither very regular nor completely random. One measure for the complexity of networks is the *topological fractal dimension* (TFD) by Song et al. (2005). Colloquially speaking, it describes the dimension d of the network. A line, for example, has dimension $d = 1$ and a planar lattice has $d = 2$.

There are different ways to calculate the TFD of a network. The simplest one is the so-called *box-covering algorithm* Song et al. (2007). In this project, we use the box-covering algorithm to compute the TFD of PPINs.

While the TFD of PPINs has been measured before for some organisms (e.g., *Homo sapiens* and *Escherichia coli* each to $d = 2.3$), we do not know the TFD of PPINs of many other organisms. Furthermore, this analysis is from 2005. Since then, the amount of available data on protein–protein interactions increased dramatically.

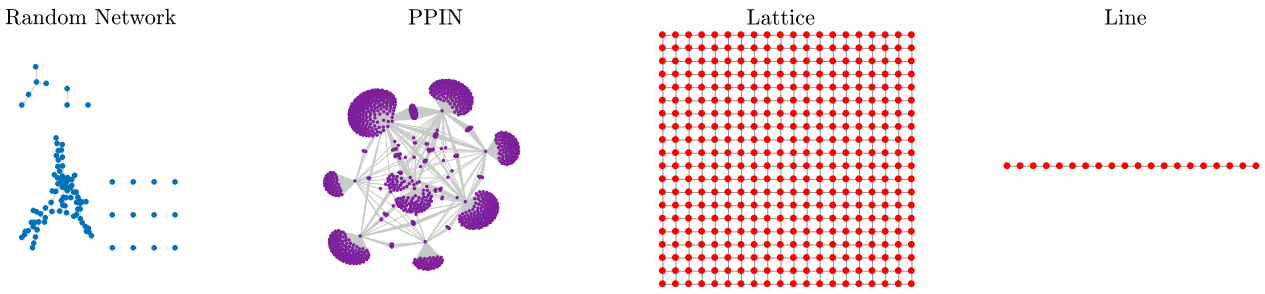

Figure 1: **Networks with different complexity:** A line graph and a lattice graph are regular and thus not complex. A random network has no regular structure. Real networks are often in between these two extremes. Here, we show a PPIN of the HIV virus.

This leads to two questions that we would like to answer in this research project:

1. Do the values reported in 2005 still hold for PPINs in 2018?
2. Do different organisms have PPINs with different TFD? Can we associate some trend for these (e.g., 'higher' organisms have more complex PPINs or organisms with smaller PPINs have less complex PPINs?)

Exercise 0.3 Organise your Research Project and your Team

Often, research projects are too complex to be finished on your own in the allocated amount of time. Therefore it can be fruitful to collaborate with other researchers. While this had advantages (the project will be finished faster and the researchers might have different expertises), there are also disadvantages: for example, you have to communicate with each other, you have to share your program code, and you have to document your code so your partners can understand how to use them.

In this research project (in which we want to compute the TFD of PPINs), it is possible to separate the tasks into four distinct steps:

1. create a program that constructs PPINs from protein-interaction data,
2. create a program that computes the TFD of a network,
3. use (1) with (2) to calculate the TFD of PPINs, and
4. write a report in LaTeX.

You will realise that tasks (1) and (2) are independent of each other and can be undertaken in parallel. Therefore, you can do this research project in groups of two students, each of which attempts one of those two steps. Afterwards, you combine these programs and do tasks (3) and (4) together (and can organise among yourselves how to divide these tasks most efficiently).

Subexercise 0.3.1 Find a Research Partner

As we need teams of size two for this project, organise yourself in groups of size two. At best, each group consists of one student that is more interested in writing a program that implements an existing algorithm (task 1) and another student that is more interested in writing a program that parses large text files (task 2). Both of you will work together in tasks (3) and (4) and you also should write only one report in the end.

For task (2) there is more guidance in this descriptions. Thus task (1) is more challenging. Both tasks, however, are crucial for a successful research project.

Subexercise 0.3.2 Set up GitHub

As we discussed earlier, working in teams comes with disadvantages. Luckily, many researchers and programmers experienced these problems before and there exist practices and tools that reduce the negative impact of these problems. One problem is that you have to share code with each other when you collaborate in a project. There a multiple ways to do this. For example, you could just write each your own code, finish it and then email it to each other. (But whenever you update your code after you fixed a bug you have to email it again.) You could use a shared dropbox folder. (But whenever your partner looks at your code and changes something there will be a problem with synchronising it and you might override each other's changes). A very good solution is to use GitHub.

GitHub is an online hosting service that is mainly used for computer code. It uses the version control Git that allows the tracking of changes in files that are written

by different people. GitHub is free to use for all personal users. It has advantages, however, to create an academic account, which is also free for students.

To set up an account and familiarise yourself with GitHub and Git, follow this tutorial: https://services.github.com/on-demand/intro-to-github/ Afterwards, update your Free account (which has only unlimited public repositories) to a Developer account for free. The procedure is explained here: https://help.github.com/articles/applying-for-an-academic-research-discount/

Note: If you know how to use GitHub, guide your partner through the tutorial and set-up process and make sure they know what they are doing. If both of you are familiar with GitHub, skip the tutorial but set up a shared, private repository.

After you both are set up with GitHub, one of you should create a private repository for this research project. After creating it, invite your partner to collaborate on this repository.

Subexercise 0.3.3 Discuss in your Research Team how to Split the Workload

Now you are all set up for this research project, you should decide on how to split up the work and a time plan for completing the tasks. The following two exercise sheets give you an overview of tasks (1) and (2). Exercise sheet 3 gives you instructions on how to complete the project.

Bibliography

Newman, M. (2018). *Networks*. Oxford University Press.

Song, C., Gallos, L. K., Havlin, S., and Makse, H. A. (2007). How to calculate the fractal dimension of a complex network: the box covering algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(03):P03006.

Song, C., Havlin, S., and Makse, H. A. (2005). Self-similarity of complex networks. *Nature*, 433(7024):392.

Worksheet 1

Constructing Protein–Protein Interaction Networks

Introduction to Programming – Research Project

Florian Klimm & Eoin Malins

Michaelmas 2018

The aim of this research project is to calculate the *topological fractal dimension* (TFD) of protein–protein interaction networks (PPINs). For this, we have to construct PPINs from data that is available in *protein interaction databases*.

Exercise 1.1 Accessing Protein Interaction Data

There exists many different protein interaction databases that store information about the pairwise interaction between protein. For this project, we use the Biological General Repository for Interaction Datasets (BioGRID; see Stark et al. (2006)). It is available under `https://thebiogrid.org/`.

BioGRID stores information about experimentally measured pairwise interactions between proteins in ~ 70 different species.

Subexercise 1.1.1 Downloading Protein Interaction Data

Download the file `BIOGRID-ORGANISM-3.5.165.tab2.zip` from the database and unpack the zip file. You now have 66 files, each contains the protein interaction information of a different organism. Sort the files by size; which organisms have the most interaction data available?

Subexercise 1.1.2 Investigate Protein Interaction Data

Before we create a program that reads these files to create a PPIN, we want to investigate the files and their structure. The files are in a tab-separated format, which is very very common for data files. It is basically a large table with the first line as header that gives a brief description about the contents contained in each column. Each of the following lines contains one entry, i.e., an interaction between two proteins. Each field value of an entry is separated from the next by a tab character. Open the file `BIOGRID-ORGANISM-Human_Herpesvirus_6B-3.5.165.tab2.txt` with OpenOffice Calc and with a Text Editor.

Question: How many interactions are in this file?

Each of these files contains a lot of information on the interactions. Most of these informations are not important for our project. Have a look at the description of this file format under `https://wiki.thebiogrid.org/doku.php/biogrid_tab_version_2.0`.

Question: The information of which columns are necessary to create a PPIN? There are different choices possible. Read the column definitions to identify which columns are best suited.

Have a look at one of the larger files. Is it still feasible to open them in OpenOffice Calc or with a Text Editor?

Exercise 1.2 Construct PPINs from BioGRID data

Now we will create PPINs from these files. First, we create a small PPIN manually to get an understanding of what our program is supposed to do. Second, we write a program that constructs a PPIN from a file.

Subexercise 1.2.1 Create a PPIN maually

To get an understanding of what the program is supposed to do automated, we do it first manually ourselves. Open the file `BIOGRID-ORGANISM-Human_Herpesvirus_6B-3.5.165.tab2.txt`. Draw a PPIN with pen and paper.

Question: How many nodes and edges are in this network? What is the size of the largest connected component[1]?

Subexercise 1.2.2 Write a program to Construct PPINs from a BioGRID file

Write a program that reads the file `BIOGRID-ORGANISM-Human_Herpesvirus_6B-3.5.165.tab2.txt` and saves the PPIN as an edgelist and the 'official symbols' of each node/protein in a separate file.

Subexercise 1.2.3 Use NetworkX to Illustrate Graphs

NetworkX is a library for Python that allows the investigation and illustration of networks. You can find an introduction under `https://networkx.github.io/documentation/latest/reference/introduction.html`.

Create a new program that draws the PPIN of Herpesvirus 6B by using NetworkX.

Question: Is this network the same as the one you drew? If not, check for errors!

Subexercise 1.2.4 Create PPINs for other Organisms

Thus far, we wrote a program that constructs a PPIN for a single organism. Create a new version of your program that takes a file name as an input and so is able to create a PIN of any of the 66 organisms. The output files should be named the same as the input file but with the file extension '.edgeList' and '.proteinSymbols', respectively. Save the output file automatically in a separate folder. If the file is not available display a warning and return an empty edge list file.

Subexercise 1.2.5 Cleaning PPINs

Adapt you program such that it removes parallel edges (multiple edges connecting the same pair of nodes) and self-loops (edges connecting a node to itself).

Subexercise 1.2.6 Optional: Automatically Create PPINs for All Organisms

Find a way to create automatically the PPINs for all 66 organisms. Hint: Use the Python library os.

Exercise 1.3 Analyse PPINs

We created edge lists of PPINs of different organisms. We now examine these networks with the help of NetworkX.

Subexercise 1.3.1 Read an edge list into NetworkX

Use NetworkX to read the edge list. Compute the number N of nodes and number M of edges of the PPIN for at least 5 different organisms, among them *H. sapiens*.

Question: Check these numbers with other students. Do the networks you created have the same N and M? If not, discuss potential discrepancies and check your code for bugs.

[1] If you do not know what a 'largest connected component' is, google it or check in Newman (2018)

Subexercise 1.3.2 Centrality in PPINs

Node-centrality measures help to quantify the importance of nodes in a network. NetworkX has approximately 20 different centrality values available. For the PPIN of *H. sapiens*, compute the top five central proteins for three different centrality measures.

Question: Do a literature research on some of these proteins. What is their biological function? Are the same proteins identified as important by different measures?

Subexercise 1.3.3 Degree Distributions of PPINs

Plot degree distributions for two different PPINs, among them one of *H. sapiens*.

Exercise 1.4 Optional Exercises

If you should be finished with the exercises above but your partner is still in the process of creating a program that computes the TFD of a network, here are some questions that could be interesting to investigate. The exercises are not in any particular order.

Subexercise 1.4.1 Computational Complexity

We have protein interaction data for many different organisms available. As we discussed earlier, they vary in size. Use the Python module `time` to stop the time T needed to create the PPIN for each of these files.

Question: What is the relationship between T and the N and T and M? Use linear regression and log–log plots to obtain an estimate about the *time complexity* of this algorithm. Are there ways to speed the algorithm up?

Subexercise 1.4.2 Create Illustrations of PINs

Use NetworkX to create illustrations of a PIN of HIV.

Subexercise 1.4.3 The Development of PPINs

BIOGRID has a release archive of the databases from June 2006 onwards. Download the data for one organism (e.g., human) of one version per year. Create a PPIN for each of these files (and store them for a later task). Create an illustration that shows the development of number N of nodes and number M of edges in the PPIN of this organism over time. In the same plot, show number N' of nodes and number M' of edges in the largest connected component. Optional: Investigate the development of other network measures (e.g., global clustering coefficient, mean degree, ...).

Question: Do the most central proteins change over time?

Bibliography

Newman, M. (2018). *Networks*. Oxford University Press.

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). BIOGRID: A general repository for interaction datasets. *Nucleic Acids Research*, 34(suppl_1):D535–D539.

Worksheet 2

Computing the Topological Fractal Dimension of Networks

Introduction to Programming – Research Project

Florian Klimm & Eoin Malins

Michaelmas 2018

The aim of this research project is to calculate the *topological fractal dimension* (TFD) of protein–protein interaction networks (PPINs). For this, we have to create a computer program that calculates the TFD for a network.

Exercise 2.1 The Box-counting Method

We are using the box-counting method to compute the TFD of networks (see Song et al. (2007)).

Subexercise 2.1.1 Understand the Method

This box-counting method consists in covering the entire network with the minimum number of boxes N_B of linear size l_B. If the number of boxes scales with the linear size l_B and follows a power law then d is the TFD of the network. This power law is then

$$N_B(l_B) \sim l_B^{-d}, \tag{1}$$

and $\sim$ indicates an identical asymptotic behaviour in the limit $N \to \infty$ (do not worry about this at the moment, treat it as meaning 'proportional to').

Subexercise 2.1.2 Write a Pseudocode for this Method

While there is a description of the box-covering method in Song et al. (2007), the description in Concas et al. (2006) is easier to follow. Read the paragraph 'Fractal dimension of software systems' on page 1223 that describes the method.

Use this description to design an efficient method to implement the method. For this, write a *pseudocode*. Roughly, the algorithm should consists of the following steps:

1. Calculate the number N_B of boxes necessary to cover the network with boxes of size $l_B = 1$.
2. Increase the box size l_B and calculate $N_B(l_B)$.
3. *repeat step 2* until $N_B = 1$.
4. Fit a power law to the function $N_B(l_B)$ to obtain the TFD d.

Subexercise 2.1.3 Implement the Method

Use NetworkX to implement the box-counting method. In the beginning, use small synthetic network (e.g., a lattice graph with $N = 20$ nodes) to test your method while developing it.

Write it in a way that it takes a network as input and returns the TFD d.

Subexercise 2.1.4 Test the Method

Once you created a function that computes the TFD, you want to test it on synthetic networks. For path graphs we expect $d = 1$ and for lattice graphs $d = 2$. For Erdős–Rényi (ER) graphs we expect a much larger TFD.

Test whether the TFD changes for ER graphs with varying number N of nodes.

Exercise 2.2 Optional Exercises

If you are finished with exercises above but your partner is still in the process of creating a program that constructs PPINs, here are some questions that could be interesting to investigate. The exercises are not in any particular order

Subexercise 2.2.1 Computational Complexity

Create synthetic networks of different size N, number M of nodes, and topologies. Use the Python module `time` to stop the time T needed to compute the TFD of these networks.

Question: What is the relationship between T and the N and T and M? Use linear regression and log–log plots to obtain an estimate about the *time complexity* of this algorithm. Are there ways to speed the algorithm up?

Subexercise 2.2.2 Variability of Results under Repetition

The box-counting method is a non-deterministic algorithm. Thus, different realisations might return different results. For different networks, run the algorithm repeatedly and compare the results. Illustrate the results in a violin plot (e.g., from the `seaborn` library).

Subexercise 2.2.3 Influence of network density on TFD

Construct random ER networks of varying network density $\rho = (2M)/(N(N-1)) \in [0,1]$. A network with density $\rho = 1$ has all possible edges and a network with $\rho = 0$ has no edges. Investigate whether the network density influences the TDF. Can you explain the results for the extreme cases $\rho = 0$ and $\rho = 1$?

Subexercise 2.2.4 Other Algorithms

There exist other algorithms to compute the TFD of a network, see Song et al. (2007). Implement one of them. How do the results and performance compare with the box counting method.

Subexercise 2.2.5 Implement the Box-counting method in C.

For a speed-up of the method you can implement it in `C`. Compare the performance of both methods for large methods. Evaluate the performance improvement.

Subexercise 2.2.6 Analytical Expressions of the TFD of Synthetical Networks

For some synthetical networks it is possible to derive analytical approximations of the TFD. Try this for a *path graph*, *lattice graph*, and others.

Bibliography

Concas, G., Locci, M., Marchesi, M., Pinna, S., and Turnu, I. (2006). Fractal dimension in software networks. *Europhysics Letters*, 76(6):1221.

Song, C., Gallos, L. K., Havlin, S., and Makse, H. A. (2007). How to calculate the fractal dimension of a complex network: the box covering algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(03):P03006.

Worksheet 3

Topological Fractal Dimension of Networks of protein–protein interaction networks

Introduction to Programming – Research Project

Florian Klimm & Eoin Malins

Michaelmas 2018

The aim of this research project is to calculate the *topological fractal dimension* (TFD) of protein–protein interaction networks (PPINs). You already created a computer program that calculates the TFD for a network and a computer program that constructs PPINs from the BIOGRID database.

Exercise 3.1 Combining Your Code

Subexercise 3.1.1 Pull and Push the GITHUB repository

Use GITHUB to combine the code both of you created. Make sure that all of you have the most recent version of both parts of the computer progam.

Subexercise 3.1.2 Look at the other Code

Have a look at the computer program that your partner wrote. Is it well documented and easy to use? Have a look at each other's code and give each other constructive but critical feedback.

Question: Did you follow good coding practices?

Exercise 3.2 Computing the TFD in PPINs

Subexercise 3.2.1 Compare with Reported Results

Song et al. reported the TFD of the PPINs of two organisms. Compute the TFD for these organisms. Does your calculation yield similar results? Discuss potential discrepancies.

Subexercise 3.2.2 TFD of PPINs of More Organisms

Compute the TFD of PPINs of at least 10 out of 66 organisms (do it for all in an automated way, if possible). Plot the TFD vs the size of the PPINs. Is there any relationship observable? If yes, try to quantify it with a regression analysis. Discuss your findings.

Exercise 3.3 Writing a Report

Use LaTeX to write a report about this research project. Follow the style guide provided by the *Proceedings of the National Academy of Sciences of the United States of America* and use the associated LaTeX template (available under `http://www.pnas.org/page/authors/format` and `http://www.pnas.org/page/authors/latex`, respectively). Additionally, your writeup should include a section 'Feedback'. What did you like about this project, what did you not like? Did you learn something new? What would you do differently, if you would undertake the project again?

You can work together at the same LaTeX file and synchronise with GitHub[1]. Another option is to use https://www.overleaf.com/.

Note: Each team of two students submits one report.
The page limit is 5 pages (single-column). This includes abstract, references, tables, and figures.
Deadline for submission is 4pm Tuesday 23rd of October
Submit it by sending a pdf file to klimm@maths.ox.ac.uk

Some general comments about writing such a report:

- Start early with drafting a structure and with writing.
- It is good to first write a rough prototype of the paper to check whether your argument makes sense, and fill in the details later.
- Make sure that the figures and their captions are well-readable.
- Be consistent with naming and mathematical symbols.
- Use BibTeX for the bibliography.
- Create illustrations in a vector format (e.g., pdf).
- Proofread a couple of times before submitting (on computer and on printouts).

Exercise 3.4 Optional Exercises

Here are some further questions that you could explore, if you find the time. Feel also free to explore other directions, if you should find them more interesting.

Subexercise 3.4.1 Repeated Computation of TFD

Repeat the computation of the TFD for a PPINs. How much do the results vary?

Subexercise 3.4.2 TFD of one Organism over Time

If you created PPINs for one organisms over time (i.e., for different releases of the BioGRID database), investigate how the TFD of these PPINs changes over time.

Subexercise 3.4.3 Other Protein Interaction Database

There exist plenty of other protein interaction databases than BioGRID. Choose one (or multiple) other databases and compute the TFD for *H. sapiens*. Does this yield similar results?

Bibliography

Song, C., Havlin, S., and Makse, H. A. (2005). Self-similarity of complex networks. *Nature*, 433(7024):392.

[1] The command \input{filename} can be useful to work at the same time at different sections (e.g., introduction and conclusions) and to make sure the both of you are not overriding each other's changes.