Bioinformatics III

Prof. Dr. Volkhard Helms, Dr. Tihamér Geyer
Nadine Schaadt, Christian Spaniol
Winter Semester 2011/2012

Saarland University
Chair of Computational Biology

Exercise Sheet 1

Due: October 28, 2011 13:15

1 Introduction to Python and some Network Properties

In order to keep the complexity down for this assignment, we focus on the random network and provide several hints and code stubs to ease your the work in respect of the upcoming exercises. We want to avoid to write code for the same stuff over and over, hence we split this assignment into separate modules which can (and have to) be reused in later assignments. Next week we will proceed constructing scale-free networks and learn how to visualize networks.

General remarks

Since the first couple of assignments are based on each other, it is strongly recommend you visit the tutorials if there are problems with the first sheets. Otherwise you **will** encounter difficulties when working on the next assignment sheets.

Python

For the programming tasks of the assignments we use the "*Python*" scripting language, Version 3.1. The aim of this assignment is to learn basic Python concepts and to get familiar with network properties.

You can find a lot of documentation on Python online in the "documentation" section of `www.python.org`. The most valuable documents are the classic tutorial by Guido van Rossum and the "Python Library Reference". not random.
(typed or handwritten) or electronically (as a single printable PDF file — don't expect me to have the latest MSWord or OpenOffice, or the same version as you) ;-). abgegeben werden! each.

Submission

(a) You are advised to work in groups of *two* people. If necessary, we will suggest teammates. You may submit your solutions in english or german language.

(b) Submit your solutions on paper, hand-written or printed at the *beginning* of the lecture in the lecture hall or in Room 3.01, both E2 1. Alternatively you may send an email with a **single** PDF attachment. Late submissions will not be considered.

(c) Include source code listings into the submitted document, we will **not** merge and layout your source code. If relevant sources are missing in the exercise sheet, they will not be graded.

(d) Feel free to use the LaTeX-template provided at our homepage

(e) If appropriate, **additionally** hand in your source code files via mail to christian.spaniol@bioinformatik.uni-saarland.de.

(f) Do not forget to mention your names/matriculation numbers. :)

Discussion of this exercise will be on Wednesday, November 2nd at 12:00 in the lecture room (E2 1 007).

Exercise 1.1: Constructing The Network (50 Points)

We start building a simple data structure that represents a network in general. Therefore we (a) construct a "`Node`"-class that represents a single node in the network and stores its links to other nodes, (b) define an abstract superclass, which can be used to derive our different network types from, and (c) implement a subclass that actually specifies how to build a random network.

We provide class stubs that you may use as guideline to help those getting started who are not familiar with the Python syntax at all. Most method implementations are supposed to not exceed 1-2 lines. Feel free to extend your interface as you need.

(a) Implement the missing methods of the Node-class that is provided in the supplementary material.

(b) We need a network class that stores all nodes for a network. Use a Python-Dictionary to store all nodes in a `key` → `node` fashion within a class variable called `self.nodes`.

Your task is to write an abstract network class using the API given in the supplementary material.

Hint:

- An example how to use python dictionaries:

```
# init dictionary
nodes = {}

# create node with id 0
node = Node(0)

# add entry to dictionary
nodes[node.id] = node
```

each line then lists the (index of the) two nodes. Take care not to list the same link twice (only print a link when $i < k$).

(c) Implement the algorithm given in the lecture to set up a random graph. Start from a given number of nodes and add unique links stepwise. Extend the AbstractNetwork from (b) and save it to "`RandomNetwork.py`":

```
from AbstractNetwork import AbstractNetwork
from Node import Node

class RandomNetwork(AbstractNetwork):
    """
    This is the specific random network implementation,
    derived from the AbstractNetwork
    """
    def __createNetwork__(self, amount_nodes, amount_links):
        """
        Create a random network
        1. ...
        2. ...
        """
        ...
```

Hints:

- When setting a link for two nodes i and k, do not forget the entry for $k \rightarrow i$.

Exercise 1.2: Degree Distribution (25 Points)

(a) Write a class that determines and prints the degree distribution for a network created above. Again, a stub is given in the supplement.

Hints:

- First determine the biggest number of links that occurs in your network and initialize a list of that size with zero values. An example to do this with 10 entries is:

```
histogram = [0] * 10
```

  This array then holds the degree distribution.
- Now loop over the network list and increment the cells indexed with the corresponding degree.
- Normalization?

(b) Verify that the degree distribution obeys the Poisson distribution $P(k)$ with a mean value of λ:

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

Calculate λ from the numbers of nodes and links and determine the Poisson distribution for a sufficiently large range of k.

Hint: In case you encounter numerical problems calculating the factorial, consider an iterative solution, e.g.

$$P(0) = e^{-\lambda}, P(1) = \frac{\lambda}{1} \cdot P(0), \ldots, P(n) = \frac{\lambda}{n} \cdot P(n-1)$$

(c) Use the provided script (createNeworks.py) that sets up the following networks (each given in the form nodes/links) and stores the degree distribution for each network in a separate file.

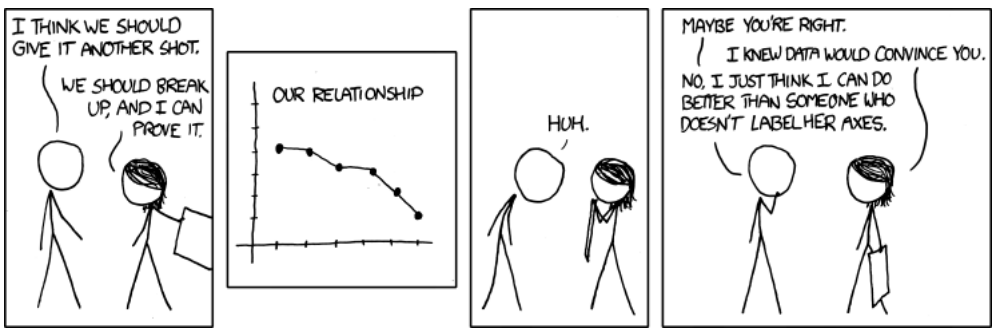
Plot 1:	50/100	500/1000	5000/10000	50000/100000
Plot 2:	20000/5000	20000/17000	20000/40000	20000/70000

random network plots]../src/createRandomNetworks.py Plot $P(k)$ together with the degree distributions of the random networks.

Describe both plots and *explain* the difference (or the trend) between the different parameter sets.

Hints:

- On a Mac or Linux box the easiest way to plot those files is to use "*Gnuplot*".
- You may adapt the Gnuplot script from our homepage.
- Do not forget to label the axes! (This hint applies to the entire lecture.)



Exercise 1.3: Classify Real-World Network Examples (25 Points)

Characterize (with a short explanation) the following examples of networks into the following categories: random, scale-free, hierarchic, and clustered. Are they directed, undirected? Some of the examples might fit into more than one category. If so, explain your choice.

(a) the German and the French railway systems

(b) the physical backbone of the internet (cables, routers, computers, ...)

(c) the world wide web

(d) american airports connected by direct flights

(e) your own social network ...

- ... when you were in school (living at home)
- ... now as a student

(f) online network services such as ...

- ... Twitter
- ... Facebook

Have fun!