Theoretical Computer Sciences Project

Deadline 15 January

Blerina Sinaimeri and Sergio Peignier

The homework is 1/3 of the **total grade**. Exercise (1) – (2) are **obligatory**, whereas you should **choose one** from the points (3) and (4). The project should be carried out in **groups of 2 or 3 students**. Send a **compressed file** with the **code** and the **report** to sergio.peignier@insa-lyon.fr and blerina.sinaimeri@gmail.com **before the 15 of January 2019**.

1 Introduction

In this project we will apply graph algorithms to study the gene regulatory network (GRN) of *Saccharomyces cerevisiae.*

This species of yeast, it is a small single-cell eukaryote, with a short generation time, and two possible forms: an haploid one and a diploid one. Moreover, this organism can be easily cultured, and it has an important economic impact since it is extensively used for instance, in winemaking, baking, and brewing. Due to these characteristics, *Saccharomyces cerevisiae* is studied as an important model organism.

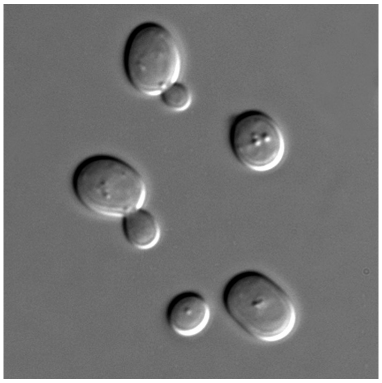

Figure 1: *Saccharomyces cerevisiae*

In this work we will study the gene regulatory network of *Saccharomyces cerevisiae*, using graph theory algorithms. The files that are provided for this project have been used in [MCK^{+}12], as gold-standards to assess gene regulatory network inference algorithms, and they are the result of biological experiments based on ChIP binding data [MWG^{+}06], and systematic transcription factor deletions [HKI07]. Hereafter we describe each dataset in details:

- GRN_edges_S_cerevisiae.txt: contains the edges of the *S. cervisiae* regulatory network (from transcription factors to target genes). The intended meaning is that if there is an edge between transcription factor X and the target gene A, then X regulates the transcription of A.

- net4_transcription_factors.tsv: Is a file containing in a single column the identifiers of the transcription factors of *S. cervisiae* that were studied.

- net4_gene_ids.tsv: The two previous files, use specific identifiers to denote genes, and this file contains the gene name associated to each gene identifier.

- go_slim_mapping.tab.txt: Only columns 0 and 5 will be used in this work. Column 0 contains the gene name, and column 5 contains its Gene Ontology (GO) annotation (`http://www.geneontology.org/`). Notice that two different rows may give for the same gene different Gene Ontology annotations.

2 Exercises

Exercise 1. Exploration and characterization of the gene regulatory network

- Load the dataset and create a NetworkX graph instance.

- Plot the gene regulatory network, the plot should be readable, understandable, and informative. Which information did you decide to convey in your plot? Why?

- Describe the network by computing pertinent local and global metrics, explain your choices, represent the results graphically if necessary, and interpret the results.

- Implement and apply the k-shell decomposition algorithm.

- For at least 4 of the metrics that you have used: what is the time complexity of the algorithm that calculates it (explain)?

Exercise 2. Community detection

- You can choose between the Girvan Newman method and the Louvain algorithm to find communities in the graph.

- Describe both algorithms, and their time complexities (explain).

- Which algorithm did you choose, why?

- For the the Girvan Newman method, the user should select one of the output partitions, explain the criterion that could be used to make this choice, and its complexity.

- Study the GO composition of each community. To do this you can produce a counting matrix M, such that $M_{i,j}$ is the number of genes from community j that have GO annotation i[1].
- Is there a relationship between graph communities and particular cell functions?

Exercise 3. Let G be the undirected graph representing the *S. cervisiae* regulatory network. Write the pseudocode of an algorithm that determines the length of the smallest cycle in G (if the graph has no cycles then the algorithm should output no cycle otherwise it must output the length). Describe the running time of your algorithm.

Exercise 4. Given an undirected graph G consider the problem of determining whether G has a cycle that goes through at *least half* of the vertices of G. Describe a polynomial time algorithm to solve this problem, or prove that the problem is NP-complete.

References

[HKI07] Zhanzhi Hu, Patrick J Killion, and Vishwanath R Iyer. Genetic reconstruction of a functional transcriptional regulatory network. *Nature genetics*, 39(5):683, 2007.

[MCK+12] Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Andrej Aderhold, Richard Bonneau, Yukun Chen, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796, 2012.

[MWG+06] Kenzie D MacIsaac, Ting Wang, D Benjamin Gordon, David K Gifford, Gary D Stormo, and Ernest Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC bioinformatics*, 7(1):113, 2006.

[1] For the sake of clarity you may need to exclude the rows associated to GO having too many counts, since they denote too general features