UNIVERSITAT ROVIRA i VIRGILI

Prediction of interactions between pairs and triplets of genes in *Saccharomyces cerevisiae* using Mixed-Membership Stochastic Block Models

Aleix Mariné-Tena

BIOTECHNOLOGY FINAL PROJECT

Tutor and supervisor: Marta Sales-Pardo, ETSEQ,
(marta.sales@urv.cat)
In cooperation with: SEES:lab

September 2020

Contents

1 Abstract

In this project we implement a mathematical model based in Mixed-Membership Stochastic Block Models to be able to make predictions of the strength and type of genetic interaction between two or three genes from the human model organism *Saccharomyces cerevisiae*.

We use a data-set of 501510 entries obtained from the supplementary materials of the article *"Systematic Analysis of Complex Genetic Interactions"* [1]. This data-set contains the fitness data from yeast triple and double *knock-out* mutants, each with a different combination of mutated genes.

After validating the predictions of the model using different metrics, we compare how genes are related according to Mixed-Membership Stochastic Block Models are related in Gene Ontology terms.

Keywords: Genetic interaction, *Saccharomyces cerevisiae*, Gene Ontology terms, Mixed-Membership Stochastic Block Models, machine learning.

2 Introduction

The increasing amount of available information, the cheapening of computation power and the need for tools that are capable of digesting large amounts of data, have made Machine Learning (ML) one of the most efficient ways to do predictions from incomplete data.

Previous studies have shown the potential of Mixed-Membership Stochastic Block Models (MMSBM) to make accurate predictions of non-observed data with a scalable algorithm [2] [3]. MMSBM are generative models, that is models that assign probabilities to observed events. As such, they are amenable to Bayesian inference techniques, so that model parameters can be inferred from observed data (in our case known gene-gene or gene-gene-gene interactions).

In Biology and related fields, available data in public databases is growing every year specially due to the advance in multiplexing techniques such as ELISA, DNA arrays or Luminex. That points out the increasing need of bioinformatics and ML in biotechnology research in order to extract conclusion from these large amounts of data.

In this study we put the focus on genetic interactions in the human model organism *Saccharomyces cerevisiae* (*S. cerevisiae*). There are big data-sets available of genetic interaction between two or three genes in this organism, but they just represent 1% of the possible genetic interactions in (*S. cerevisiae*). This fact makes this problem ideal for a ML solution because even though this 1% is in absolute therms thousands of rows of data, it only represents a small fraction of what can actually be tested *in vitro*. Therefore, the algorithm is able to train efficiently due to the high amount of available information, but also a lot of new data can be obtained from its predictions.

3 Objective

- Develop an algorithm that is able to train from our data-set to make accurate predictions of the genetic interaction strength between genes.
- Compute different models with different parameters and obtain their performance metrics in order to select the best model.
- Compare the clustering of the genes in our model with the distance between their respective gene ontology therms.

4 Theoretical Background

4.1 Genetic Interaction

The central dogma of biochemistry states that the information in most living beings flows from genes, encoded in deoxyribonucleic acid (DNA) molecules, which are the fundamental elements of the genetic material, to proteins and ribonucleic acids (RNA), which are the responsible for many heterogeneous functions in the cell: structural, reaction catalysis, signaling, transcription factors...

We can understand genetics following a ”static” approach in which each gene is studied individually to determine its function and to determine what are the major chemical and physical conditions needed in the cell for that particular gene to activate and give its final product.

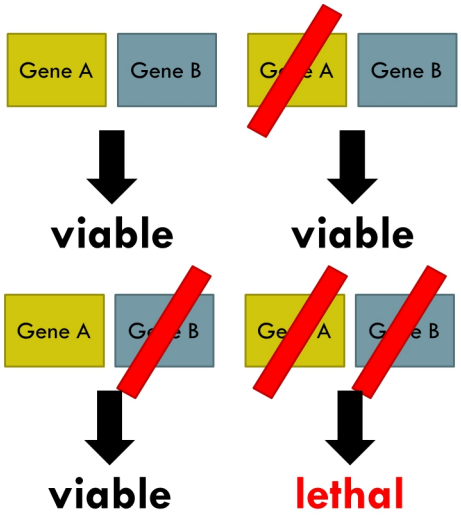
But that is just a simplification from reality since Genetics has a dynamic and very complex behaviour, specially in superior organisms: Genes and their products are interacting with each other in many ways. Usually, this interactions are very subtle and have functions that have not been discovered yet. Because of that, life at cellular level it is not a process that flows in one direction; is the result of a very complex network of interaction between the metabolites of the cell and its environment. [4] [5].

So, understanding the way in which genes and their products interact with each other can be a difficult task, since many involved factors are still unknown. But, experimentally, observing a chosen phenotype, we can determine if two or more genes are actually interacting, following the next definition:

Genetic Interaction *Genetic interaction is a phenomenon that occur when two or more mutant alleles in a single individual combine to result in a phe-*

notype that is different from the expected phenotype when these alleles are tested separately in different individuals.

Figure 1: The yeast is not viable only when the two genes are missing.



For example 1 let A and B be the only two genes present in *S. cerevisiae* that code for an enzyme responsible of a step in a vital pathway. Also, let **fitness** be a numeric phenotype than can be calculated as the ratio between the real diameter of a given colony and the diameter of the *wild-type* colony.

When gene A is non-functional or missing, gene B can replace gene A's function and vice versa. This process allows *S. cerevisiae* to grow even when genes that provide a vital function are deleted. Consequently, the fitness phenotype when A or B are deleted in a individual should be 1 or **non-lethal**.

But if we delete gene A and B in the same individual, we will find that the fitness phenotype is 0 because *S. cerevisiae* will not be able to grow up, because there is no gene this time that can replace the function of the lost vital genes.

Knowing all the above and applying the definition we can say that genes A and B are interacting because when deleted in the same individual the obtained phenotype is different from the expected phenotype: Since genes A and B have **non-lethal** phenotype when deleted separately, we expect the same when combined in the same individual.

Note that this is an illustrative example and genetic interaction is not always so drastic and clear.

4.1.1 Types of genetic interaction

There are two basic types of fitness-based genetic interactions:

- **Negative genetic interaction:** Occurs when a combination of mutations leads to a fitness defect that is more exacerbated than expected.
  - **Synthetic lethality** occurs when two *non-lethal* mutations generate a *lethal* mutant when combined.
- **Positive genetic interaction:** Occurs when a combination of mutations leads to a fitness greater than expected.
  - **Genetic suppression**: Occurs when the mutations in the fitness defect of a query mutant is alleviated by a mutation in a second gene.

4.1.2 Quantification of a fitness-based genetic interaction

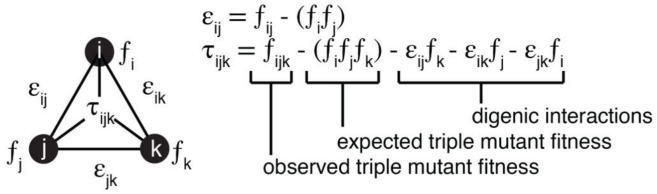
The study of the **fitness phenotype** allows to easily determine how lethal the suppression of genes can be, and also gives us the possibility of defining a formula to calculate a parameter that expresses the difference between the observed and the expected fitness of a mutant colony. Given a certain threshold of this parameter we can affirm that there is genetic

interaction and also we can determine what type of genetic interaction is happening.

To screen trigenic interactions, the Synthetic Genetic Array (SGA) analysis was adapted. The SGA analysis is a high-throughput genetic technique that allows systematic construction of double mutants using a combination of recombinant genetic techniques, mating and selection steps. This adaptation consisted in using a double-mutant as a query strain, and then crossing this mutant into an array of single mutants, generating triple mutants for trigenic interaction analysis.

Also, using the SGA score method 2 defined in previous studies [1] we can quantify the strength of a genetic interaction by computing a numeric parameter for each double or triple mutant yeast. As number of genes implied in a genetic interaction increases, complexity in the calculus of this parameter will increase too. In this project we will put our focus only in double and triple mutant yeasts, though.

Figure 2: This scheme shows how probabilities are calculated in the SGA score method, under the multiplicative model



The SGA score is calculated different depending on the number of implied gens in the genetic interaction:

Double mutant yeasts Let ε_{ij} be the digenic interaction score parameter under the probability multiplicative model. Let i and j be two different genes and let f be the observed fitness of a yeast. Then f_i and f_j represent the fitness of a yeast where the gene i or j, respectively, has been *knocked-out*. f_{ij} represents the observed fitness of a yeast where gene i and j have

been *knocked-out*. We can calculate ε_{ij} for each digenic mutated yeast to obtain its genetic interaction score using the next formula:

$$\varepsilon_{ij} = f_{ij} - f_i * f_j$$

We consider that genes i and j are interacting when $|\varepsilon_{ij}| > 0.08$ and $P_{value} < 0.05$. Otherwise, genes i and j are not interacting.

Triple mutant yeasts Let τ_{ijk} be the trigenic interaction score parameter under the probability multiplicative model. Let i, j and k be three different genes and let f be the observed fitness of a yeast. Then f_i, f_j and f_k represent the fitness of a yeast where the gene i, j or k, respectively, has been *knocked-out*. f_{ijk} represents the observed fitness of a yeast where gene i, j and k have been *knocked-out*. We can calculate τ_{ijk} for each trigenic mutated yeast to obtain its genetic interaction score using the next formula:

$$\tau_{ijk} = f_{ijk} - f_i * f_j * f_k - \varepsilon_{ij} * f_k - \varepsilon_{ik} * f_j - \varepsilon_{jk} * f_i$$

We consider that genes i, j and k are interacting when $\tau_{ijk} < -0.08$ and $P_{value} < 0.05$. Otherwise, genes i, j and k are not interacting.

4.2 *S. cerevisiae* as a human model organism

S. cerevisiae is a single-celled fungus, a type of yeast used industrially in the manufacture of bread, beer, and wine. It is one of the most suitable models for the study of biological problems because is an eukaryotic system with a biological complexity slightly higher than the bacteria's; but shares many of its technical advantages with it: In addition to its rapid growth, the dispersion of cells and the ease with which cultures replicate and isolate mutants, it stands out for a simple and versatile DNA transfor-

mation system. Also, the absence of pathogenicity allows its manipulation with the minimum precautions.

An additional advantage of this microorganism is that the complete sequence of its genome is known and it is kept under constant review. This has allowed the genetic manipulation of the nearly 6600 genes that the yeast genome encodes, the extensive use of DNA microarrays to investigate the transcriptome and genomic-scale studies of gene expression, protein localization and functional organization of the genome and proteome.

The molecular machinery of many eukaryotic cellular processes is conserved in yeasts. This allows *S. cerevisiae* to be one of the most widely used eukaryotic model organisms. It has been used as a model to study aging [6], regulation of gene expression [7], signal transduction [8], cell cycle [9], metabolism [10] [11], apoptosis [12], neurodegenerative disorders [13], and many other biological processes.

Also, yeasts and humans share a significant fraction of their functional pathways that control key aspects of eukaryotic cell biology such as protein folding, quality control and degradation [14], vesicular transport [15]... In the majority of cases, yeast has been the model organism in which these pathways were originally identified and studied. These conserved biochemical pathways drive cellular growth, division, trafficking, stress-response, and secretion, among others, all of which are known to be associated with various human pathologies. This explains the significant role for yeast as a model organism for human disorders [16]. For example, up to 30% of genes implicated in human disease may have orthologs in the yeast proteome [17].

For all these reasons, *S. cerevisiae* has become an important large-scale functional genomics analysis tool, providing a starting point for the analysis of more complex eukaryotic organisms, such as humans. Being a single-celled organism with a rapid growth rate and having many biochemical processes in common with other, *S. cerevisiae* can be used for cell

studies that would be very complicated or expensive in multicellular organisms.

4.3 Machine Learning

ML is a subset of artificial intelligence that studies the use of computer algorithms that improve automatically through experience. ML algorithms build a mathematical model based on sample data, known as ”training data”, in order to make predictions or decisions without being explicitly programmed to do so. ML algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

ML is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical models and mathematical optimization delivers methods, theory and application domains to the field of machine learning.

4.3.1 Types of Machine Learning

ML approaches are traditionally divided into three broad categories, depending on the nature of the ”signal” or ”feedback” available to the learning system:

- **Supervised Learning:** The computer is presented with example inputs and their desired outputs, given by a ”teacher”, and the goal is to learn a general rule that maps inputs to outputs.
- **Unsupervised Learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

- **Reinforcement Learning:** A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game). As it navigates in the problem space, the program is provided with a reward that the algorithm tries to maximize.

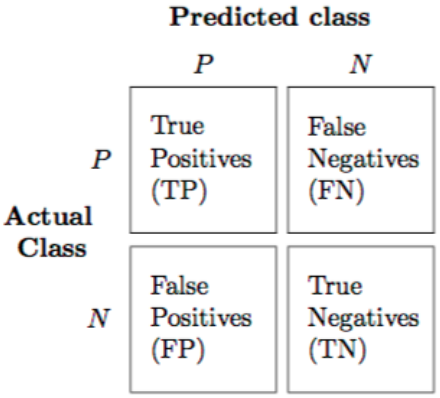
Supervised Learning will be the selected approach of ML for our learning algorithm because is the only approach that fits with the nature of our data and problem.

4.3.2 Metrics

Evaluating algorithms is a basic part of ML. There are many evaluating metrics available but we will explain the ones that we used:

Confusion Matrix Confusion Matrix gives as a matrix 3 that describes the complete performance of the model. To implement it, we need to assume a binary classification in two classes for our problem: In our case, if genes are interacting (positive) or not (negative). Also, we need to have N samples tagged with its true membership to one of the two classes. We will evaluate these samples with our algorithm, obtaining the predicted membership. Then, using a certain threshold to binarize the predictions, we can build the next table:

Figure 3: Generic confusion matrix.



In order to binarize the results, the first step consists in calculating the **threshold value** to determine the minimum value of a prediction to be considered true. To do so, we will assume equal distribution of positives between our model data and our test data.

First, we will calculate the **fraction of positives** in our model data. Second, we sort upwards in a list the samples of the test data by its predicted probability. Then, starting from the beginning of the list, we select the sample that leaves behind a number of samples equals to the fraction of positives times the number of samples in the test data. The probability that our algorithm has given to that sample is our **threshold value**.

With this value we can compute different derived metrics:

- Recall or True Positive Rate $TPR = \frac{TP}{TP+FN}$
- Fallout or False Positive Rate $FPR = \frac{FP}{FP+TN}$
- Precision or Positive Predict Value $PPV = \frac{TP}{TP+FP}$

The Area Under the Receiver Operating Characteristic Curve metric The Area Under the Receiver Operating Characteristic Curve (AUC) is another metric to asses the algorithm. To obtain it, we need to compute C by iterating over the actual positive (AP) samples and the actual negative (AN) samples in our test data to count how many times the predicted probability of an actual positive sample is bigger than the predicted probability of an actual negative sample. Then we obtain the AUC metric with the following expression:

$$AUC = \frac{C}{AP * AN}$$

4.4 Stochastic Block Model

The Stochastic Block Model (SBM) is a generative model for **random graphs**. This model tends to produce graphs containing communities, subsets characterized by being connected with one another with particular edge densities. For example, edges may be more common within communities than between communities.

The SBM is important in statistics, ML, and network science, where it serves as a useful benchmark for the task of recovering group structure in graph data.

The SBM takes the following parameters:

- Data that can be expressed as a non-directed graph G with P vertices and E number of edges. The number of edges related at the same time by the data is called degree D. If data relates each vertex with more than one other vertex at the same time, edges will be **multiple** (connecting three or more nodes) instead of **simple** (connecting two nodes).
- The number of groups K. Each vertex p can be classified into a certain group $p_{1,...,K}$. Vertex in the same group are called **communities**

$C_{1,...,K}$. This allows the full graph G to be expressed as the union of K disjoint subsets of communities $G = C_1 \cup C_2 \cup ... \cup C_K$.

- A symmetric matrix $p = \times_1^D K$. For example, $p = K \times K$ if $D = 2$ and $p = K \times K \times K$ if $D = 3$, and so on. Each cell in these matrix expresses the probability P of group i, group j and group k (when degree = 3) of being related: $P_{ijk} = p_{ijk}$.

4.4.1 Mixed-Membership Stochastic Block Models

Unlike "Single-Membership" SBM, MMSBM associate each vertex p with multiple groups rather than a single group. This is achieved via associating each vertex p with a membership probability-like vector with K length.

To do so, we define θ, a matrix that gives the probability P of a certain gene p belonging to a certain group α.

$$P = \theta_{p\alpha}$$

Then, we can define our model by defining the probability that the genes i, j, j interact:

$$p_{ijk}^{\xi} = \sum_{\alpha\beta\gamma} \theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi}$$

We also need to define our **likelihood function**. This function measures the goodness of fit of a statistical model to a sample of data for certain values of the model parameters.

The likelihood function describes a hyper-surface whose peak, if it exists, represents the combination of model parameter values that maximize the probability of classifying the data correctly. Our objective when training the model is to maximize the result of the likelihood function. To achieve that, we will use the following equations to update the values of matrix p

and matrix θ from our model. We will call **iteration** to each time that we apply the definition of the model to update these values in order to rise the result of the likelihood function.

The likelihood function can be defined using a variational approach and an expectation maximization algorithm [2] as:

$$L = \prod_{ijk}(\sum_{\alpha\beta\gamma} \theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk})$$

Introducing logarithms:

$$\mathcal{L} = log(L) = \sum_{ijk} log \sum_{\alpha\beta\gamma} \theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk}$$

And the auxiliary variable ω:

$$\mathcal{L} = \sum_{ijk} log \sum_{\alpha\beta\gamma} \frac{\theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk}}{\omega_{\alpha\beta\gamma}(ijk)} * \omega_{\alpha\beta\gamma}(ijk)$$

Knowing that:

$$\sum_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma}(ijk) = 1$$

$$log\bar{x} \geq log\bar{x}$$

$$\sum_{\alpha} \theta_{i\alpha} = 1$$

$$\sum_{\xi} p_{\alpha\beta\gamma}^{\xi} = 1$$

Then:

$$\geq \sum_{ijk} log \sum_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma}(ijk) * log \frac{\theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk}}{\omega_{\alpha\beta\gamma}(ijk)}$$

We can define ω as:

$$\omega_{\alpha\beta\gamma}(ijk) = \frac{\theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk}}{\sum_{\alpha\beta\gamma} \theta_{i\alpha} * \theta_{j\beta} * \theta_{k\gamma} * p_{ijk}^{\xi ijk}}$$

Because of the properties of the likelihood, it can be used in the training phase to know when the algorithm is trained enough. But also, it can be used in the test phase as a **metric**. We will call the likelihood of the test data **held-out likelihood**.

5 Methodologies

5.1 Replication of the experiment

The first step of the project was to replicate the methodologies to obtain the same results obtained by [1] in the selection of the triplets of genes that have genetic interaction to validate our primary algorithm of the application of the SGA score in the data. In their experiment that we want to replicate, they used two data-sets: data-set S1, which had all the raw results from all the evaluated mutants, and data-set S2, in which they only conserved those experiments with mutants of duplets and triplets of genes that are interacting according to the SGA score method and have an accepted significance level $P < 0.05$.

This first step also involved the design of the data structures that will contain all the needed data and algorithms able to digest and work with these data structures.

5.2 Implementation of MMSBM

The second step was to implement our model along with the data of our problem. Basically two algorithms were implemented:

- The computation of the likelihood of our model regarding our data.
- The **iteration** algorithm.

5.3 Validation of the MMSBM

To validate the results of the algorithm and to find which K parameter gives better results, a **five-fold crossed-validation** method was applied to be

able to compute metrics of the algorithm for each obtained sample.

Groups of 100 samples were computed for each K parameter in a list ranging from ”2” to ”5” for each of the 5 different folds of data (100 * 4 * 5 = 2000 samples).

Precision, recall, fallout and AUC metrics were computed for each sample. Each group of 100 samples was summarized by calculating the AUC metric with the average of the prediction result of every triplet, obtaining the **average AUC** metric. Also, the average and standard deviation of the held-out likelihood was computed for each group of 100 samples.

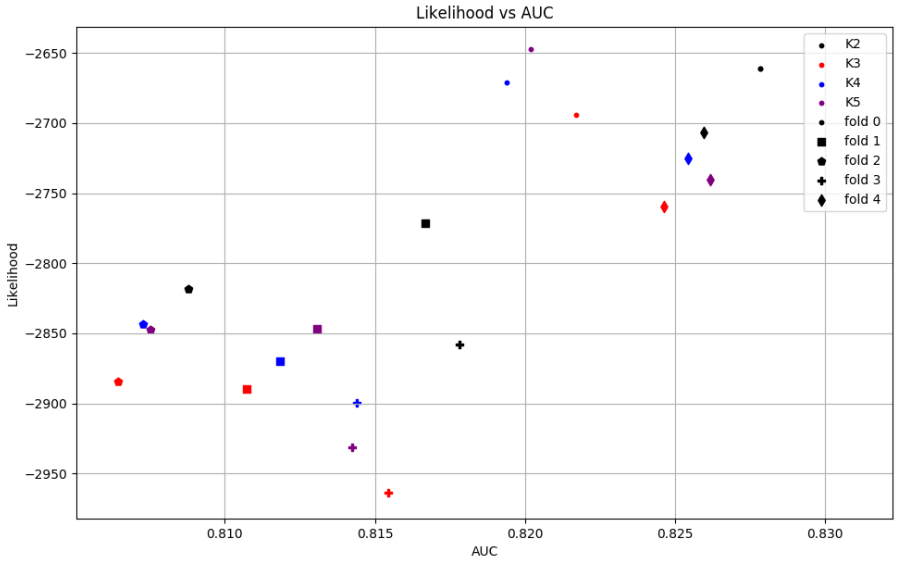
Also, the average AUC metric was computed from 150 samples with $K = 1$ to compare our metrics with a baseline reference.

6 Results and Discussion

The replication of the experiment was done successfully and the results obtained were exactly the same: trigenic mutants in the data-set S2 were exactly the same as the ones that we obtained after applying the SGA score method in mutants in data-set S1.

Metrics from each group of 100 samples was summarized in the next figure:

Figure 4: Average AUC metric vs average held-out likelihood for each group of 100 samples belonging to 5 of the different folds and $K = \{2, 3, 4, 5\}$.



This figure clearly shows that $K = 2$ is the best parameter for our model, since it shows the best metrics for every different fold. Probably, when increasing the K parameter there is **over-fitting** of the model for the training data, giving worse metrics and results.

Also, the average of the AUC metric in 150 samples with the baseline $K = 1$ was 0.15. Therefore, our model is effective in prediction since it gives better predictions than choosing randomly.

7 Conclusions

The fact that we developed a model that makes better predictions than choosing randomly by assuming that there are groups of genes behaving similarly confirms that MMSBM can be used as an effective method to do predictions in complex biological networks. Thus, there are patterns in the way genes interact and relate to each other.

This model may not be very accurate at local level, but can be used to discover patterns at general level that sometimes are not seen, specially when working with large amounts of data. This is specially important in fields like Biology, because finding new biological patterns will help to interpret unknown data and will allow to discover new biological mechanisms.

Also, a better inference of the model parameters can be achieved by extending the model to interactions between pairs of genes.

Finally, further studies are needed to be done in order to compare how genes are related according to our model parameters with the actual ”grouping” of the genes: its placement, function and process. To do so, we will compare the relation between the associated GO therms of the genes and its model parameters. This could reveal patterns in some groups of genes that may lead to the discovery or better characterization of some biological processes and the discovery of new therapeutic targets.

8 References

[1] Kuzmin, E., VanderSluis, B., Wang, W., Tan, G., Deshpande, R., Chen, Y., ... Myers, C. L. (2018). Systematic analysis of complex genetic interactions. Science (New York, N.Y.), 360(6386), eaao1729. https://doi.org/10.1126/science.aao1729

[2] Antonia, G. L., Guimera, R., Moore, C., & Sales-Pardo, M. (2016). Accurate and scalable social recommendation using mixed-membership stochastic block models. Proceedings of the National Academy of Sciences of the United States of America, 113(50), 14207–14212. https://doi.org/10.1073/pnas.1606316113

[3] Guimerà, R., & Sales-Pardo, M. (2013). A Network Inference Method for Large-Scale Unsupervised Identification of Novel Drug-Drug Interactions. PLoS Computational Biology, 9(12). https://doi.org/10.1371/journal.pcbi.1003374

[4] Serrano Moral, M., Sales Pardo, M., Alarcón, T., Guimerà Manrique, R., & Sagués Mestre, F. (2015). Xarxes complexes en biologia cel·lular. Revista de Física, 5(2), 23–28. https://doi.org/10.2436/rev.fís..vi.140429

[5] Guimerà, R. & Sales-Pardo M. (2016) La promesa de las redes metabòlicas. Sebbm, Ene2016, 1-3, papers3://publication/uuid/F620AFC0-1257-4F0F-910A-45A7A3D53525

[6] Murakami, C., & Kaeberlein, M. (2009). Quantifying yeast chronological life span by outgrowth of aged cells. Journal of Visualized Experiments, 27, 1156. https://doi.org/10.3791/1156

[7] Biddick, R., & Young, E. T. (2009). The disorderly study of ordered recruitment. In Yeast (Vol. 26, Issue 4, pp. 205–220). Yeast. https://doi.org/10.1002/yea.1660

[8] Hohmann, S., Krantz, M., & Nordlander, B. (2007). Yeast Osmoregulation. In Methods in Enzymology (Vol. 428, pp. 29–45). Academic Press Inc. https://doi.org/10.1016/S0076-6879(07)28002-4

[9] Nasheuer, H. P., Smith, R., Bauerschmidt, C., Grosse, F., & Weisshart, K. (2002). Initiation of eukaryotic DNA replication: Regulation and mechanisms. Progress in Nucleic Acid Research and Molecular Biology, 72. https://doi.org/10.1016/s0079-6603(02)72067-9

[10] Brocard-Masson, C., & Dumas, B. (2006). The fascinating world of steroids: S. cerevisiae as a model organism for the study of hydrocortisone biosynthesis. Biotechnology and Genetic Engineering Reviews, 22(1), 213–252. https://doi.org/10.1080/02648725.2006.10648072

[11] López-Mirabal, H. R., & Winther, J. R. (2008). Redox characteristics of the eukaryotic cytosol. In Biochimica et Biophysica Acta - Molecular Cell Research (Vol. 1783, Issue 4, pp. 629–640). Biochim Biophys Acta. https://doi.org/10.1016/j.bbamcr.2007.10.013

[12] Owsianowski, E., Walter, D., & Fahrenkrog, B. (2008). Negative regulation of apoptosis in yeast. In Biochimica et Biophysica Acta - Molecular Cell Research (Vol. 1783, Issue 7, pp. 1303–1310). Biochim Biophys Acta. https://doi.org/10.1016/j.bbamcr.2008.03.006

[13] Miller-Fleming, L., Giorgini, F., & Outeiro, T. F. (2008). Yeast as a model for studying human neurodegenerative disorders. In Biotechnology Journal (Vol. 3, Issue 3, pp. 325–338). Biotechnol J. https://doi.org/10.1002/biot.200700217

[14] Brodsky, J. L., & Skach, W. R. (2011). Protein folding and quality control in the endoplasmic reticulum: Recent lessons from yeast and mammalian cell systems. In Current Opinion in Cell Biology (Vol. 23, Issue 4, pp. 464–475). Elsevier Current Trends. https://doi.org/10.1016/j.ceb.2011.05.004

[15] Bonifacino, J. S., & Glick, B. S. (2004). The Mechanisms of Vesicle Budding and Fusion. In Cell (Vol. 116, Issue 2, pp. 153–166). Cell Press. https://doi.org/10.1016/S0092-8674(03)01079-1

[16] Petranovic, D., & Nielsen, J. (2008). Can yeast systems biology contribute to the understanding of human disease? Trends in Biotechnology, 26(11), 584–590. https://doi.org/10.1016/j.tibtech.2008.07.008

[17] Foury, F. (1997). Human genetic diseases: A cross-talk between man and yeast. Gene, 195(1), 1–10. https://doi.org/10.1016/S0378-1119(97)00140-6

9 Auto-evaluation

When working in collaboration with SEES:lab I understood better the roll of a bioinformatician: As a bioinformatician, I understand the nature of biological problems and data, and I can implement the algorithm that solves a certain problem, but I need to obtain a specification of that problem: Bioinformatics and computational biology feeds on problems and data from biology. My main goal as a Biotechnologist and Computer Scienctist is to link these two fields by using my acquired knowledge from my studies. My expectations with this project were fullfilled because I worked in an interdisciplinary environment, in which computation in used to solve problems from other fields of the science, such as physics, biology or chemistry.

Also, I did some of the tasks expected from a worker of a laboratory: I attended research group meetings and I learnt which are the available resources for my work and how I can access them.

I improved many skills over the course of this project such as my programming level in python and bash, my ability to manage and work with scientific articles or my knowledge of scientific databases. But specially, I acquired new skills and ability to work with new different environments: I learnt LaTeX, Python, bash, machine-learning using MMSBM and to use new tools that I will use in the future in further projects.

I consider that my collaboration with SEES:lab has been completely enriching and and productive for me.

10 Annex

10.1 Materials

This section covers all the elements needed to do our experiment. Since our experiment is an algorithm its materials are basically hardware and software elements.

10.1.1 Hardware requirements

The training algorithm is the only algorithm that is strongly hardware-demanding. Using just one thread of execution in a last-generation user PC, the algorithm can delay at most two weeks to finish. That is why there is an extra layer of parallelization that allows to run the algorithm in parallel using the number of threads corresponding to the number of cores available in the machine running the algorithm. That allows to divide the execution time by the number of cores available, so better performances are expected in better machines with more cores.

10.1.2 Software Requirements

Due to having big data-sets, long scripts and the need for computing power; many software dependencies needed to be satisfied before working with the algorithm. Some are completely necessary and some others are just accessories for increased comfort when working in the project. Also, some tools can be substituted with similar ones.

Python3 The main algorithm is written in Python3 so it needs a Python3 interpreter in order to run the code. The algorithm also uses the following

Python3 packages that need to be installed:

- NumPy: Used for random picking function.
- codecs: To read from different files.
- matplotlib: To create all the graphics from the results.

PyPy3 PyPy3 is a Python3 interpreter that runs much faster in many cases. We strongly recommend to use this interpreter since it reduces many times the computation time of the training algorithm.

Linux Even though Python scripts are portable, we strongly recommend to use Linux to run the script. The last version of the code is tested in Ubuntu 20.04 LTS Focal.

GNU-parallel GNU-parallel is a Linux utility that allows to easily run batches of commands in parallel. It is not needed to run the main algorithm, but is needed to run the parallelization layer, which is strongly recommended to do.

Git Git is a software versioning tool to maintain and develop software. Git is needed at least to clone the repository where all the data-sets and scripts are stored.

Also, because of the need to store all the data-sets in the same repository, the complementary utility **Git-LFS** was installed in the repository. This utility allows to work with big data-sets in the repository without sacrificing speed when cloning and pulling.

A ***.gitignore*** file was defined for all the possible junk, temporals and product of compilation for our repository in order to ignore all those files when updating the repository.

LaTeX The documentation was written in LaTeX and compiled with PDFLaTeX. Due to that, at least a LaTeX compiler is needed to obtain the final document of the documentation. Also a LaTeX IDE is strongly recommended: We used **TexMaker** in local Ubuntu 20 and **Overleaf**, an online LaTeX editor.

Mendeley Mendeley was used to organize, maintain and format all the citations present in this documentation.

Virtual Box Virtual Box was used to run virtual machines with Ubuntu Linux. This allowed to run the project even if the available machine to do so did not have Linux installed.

Linux-Auto-Customizer Since there is a lot of software dependencies in our project and most of the work has been done in virtual environments that needed to be configured each time that are initialized, a side project called Linux-Auto-Customizer was created with its own repository in GitHub.

This repository contains software to automatize the configuration and the dependency installation of this project for Ubuntu/Debian machines. The installation script can apply some custom features (depending on the received arguments) to the current user console and to the Ubuntu-Linux environment, such as local functions, file templates and global variables. Also, third-party software can be installed too, including its dependencies.

An uninstall script is provided too, in order to uninstall previously in-

stalled software dependencies. Please, refer to the *README.md* in the repository to understand the usage of this software.

Libraries To run many of the previous software many libraries need to be installed, such as: gcc, perl-tk, pkg-config, libfreetype6-dev, libpng-dev, libffi-dev, curl... Using the Linux-Auto-Customizer dependencies are automatically solved.

10.2 Design

10.2.1 Data structures

Different data structures to contain data of our model and problem along with methods to work with them were implemented:

- p matrix was implemented as a 3D matrix with a fourth level of indirection for the ”rate”, that in our case is 0 for the probability of no interaction or 1 for the probability of interaction.
- θ was implemented as a 2D matrix in which you could index with the unique identifier of a certain gene, and then with the number of the group.
- Two dictionaries were implemented to allow the transformation between the name of a gene and its identifier and vice versa.
- Two dictionaries that relate a triplet with its rating, using ids or using the gene names.

10.2.2 Methods

Different methods were designed to perform the experiment: initialize the data structures of the algorithm, perform an iteration, calculate metrics, do prediction for a certain triplet, validate the SGA score method with our data, digest a folded or non-folded data-set...

Also a method to print all the data in the model and a help method was designed.