CHARACTERIZING CONDITIONAL INDEPENDENCE IN GENE CO-EXPRESSION NETWORKS

Hugo Sales Corrêa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Valmir Carneiro Barbosa

Rio de Janeiro
Agosto de 2023

CHARACTERIZING CONDITIONAL INDEPENDENCE IN GENE CO-EXPRESSION NETWORKS

Hugo Sales Corrêa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Valmir Carneiro Barbosa

Examinada por: Prof. Valmir Carneiro Barbosa
Prof. Daniel Ratton Figueiredo
Prof. Aline Marins Paes Carvalho

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2023

Sales Corrêa, Hugo

Characterizing Conditional Independence in Gene Co-Expression Networks/Hugo Sales Corrêa. – Rio de Janeiro: UFRJ/COPPE, 2023.

X, 56 p.: il.; 29, 7cm.

Orientador: Valmir Carneiro Barbosa

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2023.

Referências Bibliográficas: p. 53 – 56.

1. Gene Regulatory Networks. 2. Gaussian Graphical Models. 3. Single-Cell RNA seq. I. Carneiro Barbosa, Valmir. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Ao Renato.

Agradecimentos

Agradeço aos meus pais, Cláudia e Ricardo, pelas minhas origens e formação. Agradeço ao meu orientador, Valmir, em toda a sua paciência, pela jornada de crescimento, coragem, e curiosidade intelectual que ele pôde proporcionar. Agradeço à minha namorada, Janaína, que ao encontrá-la, ajudou a me reencontrar também. Após uma dura pandemia, ainda que muitos tenham ficado para trás, agradeço a todos os que, com sua vontade de viver e esperança num mundo melhor, tiveram a força de reconstruir este Nosso Lugar. Que haja cerveja para comemorar!

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CHARACTERIZING CONDITIONAL INDEPENDENCE IN GENE CO-EXPRESSION NETWORKS

Hugo Sales Corrêa

Agosto/2023

Orientador: Valmir Carneiro Barbosa

Programa: Engenharia de Sistemas e Computação

Apresentamos, nesta tese, um estudo, por métodos computacionais e estatísticos, de dados biológicos do transcriptoma do organismo *Saccharomyces cerevisiae*. Esse estudo de biologia está relacionado a avanços recentes no entendimento acerca de como os genes de um organismo interagem para produzir fenótipos complexos, onde passamos a atribuir uma maior importância à rede regulatória de genes como um todo, em vez de somente atribuir fenótipos a uma quantidade pequena de genes. O nosso trabalho se dá por uma ótica de otimização, modelos gráficos probabilísticos e redes complexas, com o objetivo de ilustrar o potencial dessa mudança de perspectiva ao corroboramos parcialmente a Hipótese Omnigênica.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CHARACTERIZING CONDITIONAL INDEPENDENCE IN GENE CO-EXPRESSION NETWORKS

Hugo Sales Corrêa

August/2023

Advisor: Valmir Carneiro Barbosa

Department: Systems Engineering and Computer Science

In this work, we present a computational and statistical study of transcriptomic data, in the context of the model organism *Saccharomyces cerevisiae*. This study is related to recent advances in the understanding of how genes interact in a cell to produce complex phenotypes, where we start to give higher importance to the Gene Regulatory Network as a whole, instead of attributing phenotypic variation to mutations in a small group of genes. Hopefully, our work brings a fresh perspective, combining optimization, probabilistic graphical models, and complex networks, with the objective of showcasing the promise of this shift in understanding, by partially supporting the Omnigenic Hypothesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this dissertation, we task ourselves with exploring questions of how genetic variation leads to different phenotypes in a population. As one can imagine, such questions are long-standing, with the first models of inheritance, such as Mendel's, arising in the 19th century, and focusing on single-gene phenotypes. With technological advances in the 20th century, such as DNA sequencing technology, we now know that the majority of traits with a hereditary factor do not follow a monogenic model, i.e., we cannot explain the trait's observed variability in a population, with mutations to limited genomic regions, even when accounting for environmental factors. Curiously, the history of polygenic models starts as early as 1918, with Fisher's publication of his Infinitesimal model [23], an additive model under which it can be shown that if many genes affect a (quantitative) phenotype, then the random mutation of alleles at each gene converges to a continuous, normally-distributed phenotype in the population. Perhaps the best example of a trait consistent with this model is adult height [28], which is historically known to follow a normal distribution.

This model, however, assumes that each gene produces an additive effect which is independent of other genes, thus completely disregarding all the regulatory ways in which genes interact with one another. The existence of what we call epistasis, denoting gene mutations whose effect is dependent on the presence or absence of mutations in other genes, coined by Florence Margaret Durham and Muriel Wheldale, was pioneerly evidenced in their investigations in the early 1900's [27] and would eventually become central in the understanding of most complex traits, evidencing the limitations of the infinitesimal model for explaining variability in a population. And thus, we turn our attention to more recent efforts, which have focused on the interaction between genes, and more specifically, attempt to specify and model the emergence of complex traits as a consequence of properties of Gene Regulatory Networks (GRNs) [20]. Such networks abstractly represent the collection of epistatic phenomena of a given organism, and given that a GRN is a model for the expression of mRNA and consequently of proteins, it ultimately determines the function of a

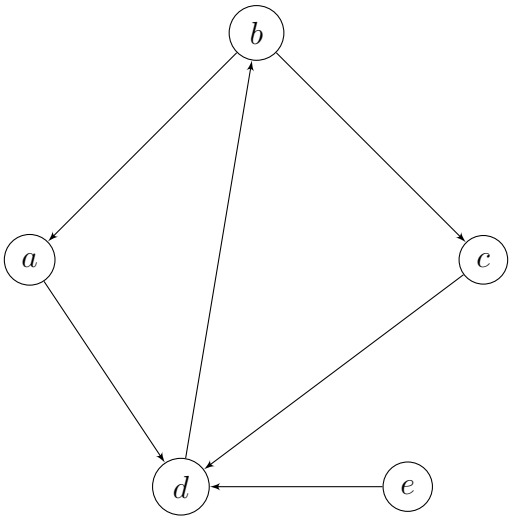


Figure 1.1: Graph representation of a GRN with genes $\{a, b, c, d, e\}$. The directed edges represent a direct influence of one gene's expression level on another's. Notably, we have two loops: $\{d, b, a\}$, $\{b, c, d\}$, and if we consider the sum of a gene's outdegree and indegree as a measure of centrality, we would have one hub gene: d, and one peripheral gene: e.

cell, it being a unicellular organism or even a member of a more complex organism.

More mathematically, a GRN is a dynamical system that can be summarized as a directed network, where all the incoming (directed) edges into a gene indicate what variables (excluding environmental variables) can influence its expression levels, as exemplified in Figure 1.1. Fully uncovering and specifying the underlying GRN of any given biological strain is currently computationally or experimentally infeasible, as it would require an understanding of how every possible protein and mRNA molecule interact with each other under each of many environmental conditions. This challenge is closely related to the field of protein-protein interaction prediction [26]. This fact, alongside an expansion in biological data availability due to innovations in sequencing technology, has led to many recent efforts taking a more statistical approach to uncovering gene regulatory networks.

An example of such approaches are Gene Co-Expression Networks (GCNs)[37][5], used mainly within a context of analyzing DNA Microarray or bulk RNA-Seq datasets, both yielding the averaged expression level of each gene in a given tissue or population of cells. These GCNs are typically based on measures of pairwise "relatedness" between gene expressions, commonly adopting metrics based on correlation or mutual information of the expression levels. When one determines a cutoff for the significance of these metrics, genes with significantly "related" expression levels produce and edge in the network. These networks were observed to give rise to small-world properties, such as the presence of hub genes and small average distances between genes.

These properties are arguably plausible from an evolutionary perspective, given

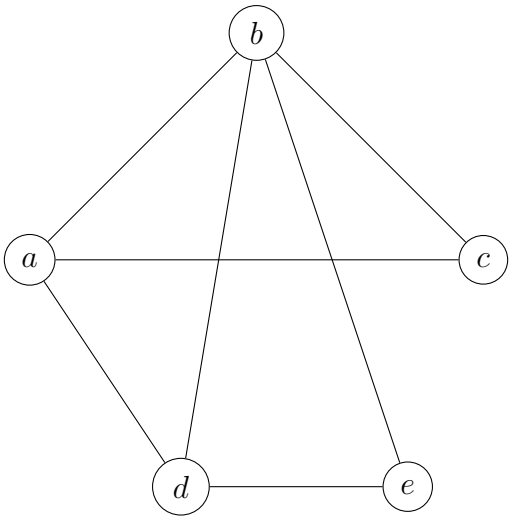


Figure 1.2: A GCN built from data generated from Figure 1.1

that scale-free networks are robust under random perturbations. Whether these networks are actually scale-free, however, has been a topic of debate since the early 2000's, with [16] stating that biological networks better fit truncated power-law degree distributions, rather than pure power-laws.

Nevertheless, the small-world and "approximate" scale-free properties of gene networks have yielded insight into the regulatory mechanisms of gene expression [36].

There are two limitations to the use of GCNs. First, it only produces undirected edges, so directionality is not preserved. Second, it does not attempt to control for correlations that are not a product of direct causation: given a gene pair, there could also be indirect causation, or even no causal relationship at all, in a situation where both genes are affected simultaneously by a separate gene or set of genes.

The graph in Figure 1.2 is an illustration of how we might expect a GCN-based method to work: we lose edge directionality and may have to deal with false positives (inferred edges that do not exist) and false negatives (missing edges), with false positives tending to be more prevalent. In particular, we illustrate how e could be missed as a peripheral gene, since it has strong correlation with d, a hub gene, which in turn has correlations with other genes. Indeed, the appearance of cluster regions may, in part, be exacerbated as a mathematical consequence of the method, given that correlations obey a weak form of transitivity.

As illustrated by Tao [31], if, for a certain gene v, its correlation with other genes u_i exceeds a certain threshold ε, meaning $\langle v, u_i \rangle \geq \varepsilon$ for at least εn values of $i = 1, \ldots, n$, then $\langle u_i, u_j \rangle \geq \varepsilon^4/2$ for at least $\varepsilon^4 n^2/2$ pairs (i, j), i.e., many genes u_i will also be correlated among themselves. With this mathematical necessity, we will likely see higher average degrees and graphs that are more "triangular".

In general, past results had several limitations both in terms of methodology

and in the quality/availability of data. This notwithstanding, they were already pointing toward a shift to considering most relevant phenotypes to arise from many interacting genes.

1.1 Previous Work

Besides GCNs, there have been a number of methods attempting to model gene regulatory networks.

Some through the lens of dynamical systems, by simplifying the gene regulatory process as Boolean Networks, where each variable's truth value is determined by a Boolean expression containing its incoming neighbours. Although seemingly simplistic, this approach provides the possibility of analyzing dynamics, instead of only producing descriptive results on structure. Such networks can exhibit vastly different regimes, depending on their degree distribution, and a number o studies suggest that they do capture biologically plausible features and processes [29]. Having been explored since the late 60's, our theoretical understanding of them is advanced, and most recent work has been focused on inference of these networks from biological data.

Other works focus on clustering and dimensionality reduction to uncover different "modes" of cell behaviour within a given tissue [1] [18]. These methods are more closely related to the general area of Manifold Learning, such that one assumes that the high dimensional data produced by genes reside in a lower dimensional manifold. Some recent examples even adapt popular Neural Network self-supervised models, for instance, auto-encoders [32] to try and capture such a hypothesized manifold. And within this captured structure, one would be able to observe different "regions" of cellular behaviour, therefore, such methods are not intended to directly indicate regulatory connections or pathways between genes, but rather help the understanding of intratissue or cell culture heterogeneity and its relationship to traits or diseases.

Finally, Gaussian Graphical Models (GGMs), the main method explored in this dissertation, which will be extensively explained in later chapters, have been used at least since the publication of the optimization method called Graphical Lasso, enabling the inference of such models in problems scaling to thousands of nodes, thus becoming a viable approach to handling gene expression data, in which each gene represents a node. However, the initial version of the graphical lasso is no longer widely adopted, as it was shown repeatedly to have issues with false discoveries. One such instance, in a study of gene expression in cancer cells, the authors even judged necessary to remove edges with weights below 0.2 as an attempt to reduce false positives from the method [39]. Others tried to deal with this problem by

incorporating previous knowledge about the network into the learning procedure [35].

1.2 The Omnigenic Hypothesis

Recently, evidence has been accumulating that GRNs are highly connected, with Genome-Wide Association Studies (GWAS) failing to attribute the appearance of a phenotype to variations in only a few genes [22]. A GWAS of schizophrenia (Purcel et. al, 2014) [25], for example, failed to demonstrate the relation of individual gene expressions to the appearance of the phenotype, which as the authors mention, was already known for autism. Such mounting evidence has motivated the formulation of the omnigenic hypothesis [4], which we summarize into two claims:

1. (pleiotropy) The GRN of complex organisms is sufficiently connected for most complex traits to be affected by thousands of genes, and consequently, pleiotropic genes (those affecting different, seemingly unrelated traits) are prevalent.

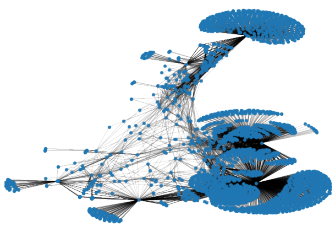
2. We can characterize a GRN as comprising both core genes and peripheral genes for a given phenotype, with the expression of core genes making them influential, while that of peripheral genes only allows them to influence the phenotype indirectly through the core genes.

Claim (2) is consistent with the "robust yet fragile" effect in complex networks [7], since mutations in core genes could indeed give rise to significant variation, while failing to explain the full spectrum of a phenotype in a population.

1.2.1 The Gene Regulatory Network, seen as a dynamical system

More formally, RNA transcription can be modelled as a stochastic dynamical system (ignoring environmental inputs) with an underlying undirected G.

$$\frac{\partial p(x_k)}{\partial t} = f_G^k(x_1, x_2, \ldots, x_p)$$



,

where G is such that If a variable pair ij is not in $E(G)$ then both $\frac{\partial f_G^i}{\partial x_j} = 0$ and

$\frac{\partial f_G^j}{\partial x_i} = 0$, meaning that x_i holds no direct influence on x_j and vice-versa.

1.2.2 The Omnigenic Hypothesis - Dynamical System Version

Given the definition of a GRN as a dynamical system, let us translate what the Omnigenic Hypothesis actually postulates regarding the dynamical system. Let $\Gamma \subset \Delta$ be the subset of genes directly related to a phenotype ϕ, i.e., Γ is the set of core genes of ϕ. The manifestation of the phenotype will be defined as function $\phi(x_\Gamma)$ of the expression levels of its core genes.

We can decompose the variations in $\phi(x_\Gamma)$ into three causes:

1. Random perturbations on x_i for $i \in \Gamma$
2. Inside "influence" of genes x_i for $i \in \Gamma$ among themselves
3. Outside "influence" of genes x_j for $j \notin \Gamma$.

In items 2 and 3, what we mean by "influence" is how an intervention on a certain gene can propagate to another, as time passes. If an intervention on the expression levels x_i of gene i causes a change in x_j of gene j that would otherwise not happen, we say that gene i is influential to j. On such terms, Claim (2) translates to the statement that: For most complex phenotypes, outside "influence" is the most dominant on determining variation of phenotypes, and this influence of peripheral genes is long-tailed, spreading to thousands of genes.

1.3 Connecting the Omnigenic Hypothesis with the Selected Modelling Method

In this study, we attempt to support the existence of a mechanism which is consistent with Claim 2. If an organism's GRN does indeed have approximate scale-free properties, like hub genes and small distances, and if core genes, which are a priori known to be "essential", also tend to be hub nodes in the network, then core genes will be highly connected (by a small distance) to thousands of peripheral nodes, and it will be plausible that these peripheral genes' influence accumulate and significantly impact the expression of the core genes, and thus, impact the development of a given phenotype. If we were to find that the GRN is not scale-free or that the hub genes do not intersect with known "essential" genes, then the support to Claim 2 would be much weaker.

1.3.1 The Omnigenic Hypothesis - Graph Version

More formally, let us again take a set of genes Γ directly related to a phenotype ϕ, representing its core genes. We can now derive a weaker version of the hypothesis that only states facts about the underlying graph G, instead of the dynamical system. The hypothesis can then be broken into three statements:

1. Genes $i \in \Gamma$ are contained in large connected components.
2. G has low degrees of separation (low average distance).
3. Genes $i \in \Gamma$ have higher degrees.

These three conditions are closely linked to graphs called "small-world". Therefore, the hypothesis basically says that G should be "small-world" and that, for the majority of complex phenotypes ϕ, their respective Γs should be composed of "central"/hub nodes.

It is important to note, however, that we will deal with non-directional connections in our methods and will not attempt to make causal claims. So in a (Platonic) sense, we will not study the GRN directly, but only its "shadow" cast in the form of correlations and partial correlations, and thus we will not be able to differentiate the case of peripheral genes impacting core genes versus core genes impacting peripheral genes, or anything in between, although it makes more evolutionary sense that core genes would be highly regulated as to maximize robustness of an organism.

Through this lens, the Omnigenic hypothesis, as stated above, can be related to GCN-based studies in the sense that one can use a GCN to identify hub-genes and relevant biomarkers (clusters of genes related to similar biological functions). As previously mentioned, however, GCNs are typically based on measures of pairwise "relatedness" between genes and can suffer from exacerbated transitivity. In this study, therefore, we aim to explore an alternative definition of a GCN, by attempting to infer conditional independences, instead of direct correlations, as an attempt to reduce false positives, when compared to the GRN. As such, we will attempt to model gene expression data through a Probabilistic Graphical Model (PGM). This formulation is more appropriate to isolate direct effects between pairs of genes, and it would be interesting to know whether the network continues to have approximate scale-free properties. Our choice of Graphical Model, however, still yields undirected networks, and as previously discussed, could ever only partially support the plausibility of Claim 2.

1.4 Single-Cell RNA-Seq

The first studies applying GCNs used Microarray data or bulk RNA-Seq data. These experimental methods provide gene expression measurements for whole tissues or cell cultures simultaneously, and thus provide an expression profile for the "average" cell at different timestamps, therefore it enables the observer to assess how the average expression level of each gene evolves over time.

Such methods come with two main drawbacks. First, different cell types within the same tissue can have distinct roles in multicellular organisms, i.e., a tissue or cell culture can have subpopulations with unique transcriptional profiles. Naturally, significant correlations that only appear in certain subpopulations are likely to be missed when only average expression profiles are considered. Second, bulk assays fail to recognize whether a change in the expression profile is due to a change in regulation or in composition (e.g., if one cell type arises to dominate the population). These drawbacks become specially problematic when one attempts to study a cell's developmental stage, because it is highly variable in duration. So if one takes many cells with the same chronological age, they will wildly vary in their "maturity", or biological age, and thus, we are likely to obscure many intricate regulatory relationships between genes, if we only consider average expression data [15].

Single-Cell RNA-Seq (scRNA-Seq), which overcomes this limitation, has now become the standard, as it allows to individually measure the RNA expression levels of each cell in the population. Nevertheless, it has drawbacks of its own: when a measurement of a cell is performed, the cell is destroyed, i.e., we cannot use scRNA-Seq to observe the expression profile of a single cell through time; furthermore, scRNA-Seq only yields partial readings, as the technology only has enough sensitivity to capture around 10% of a cell's RNA molecules [11]. The latter point makes special treatment of the data necessary, as we have an aggressive amount of missing data, we need more elaborate and domain-consistent imputation methods, which will be detailed later on, to make analyses feasible.

1.5 Which PGM?

As mentioned above, although scRNA-Seq is a large step toward the observation of intra-tissue/population heterogeneity, however, most experiments only output a few hundred measurements, this means that the ratio of samples to genes is very small, putting us in the high-dimensional/small data domain. In a context such as this, we must select a PGM that is inherently parsimonious in its learning procedure. Thus, a reasonable choice is to model gene expression through a Gaussian Graphical Model (GGM), which is a linear model, therefore not data-hungry, especially given that

regularization methods are much better known in the linear context. Moreover, a GGM's conditional independence structure, as we will see in the following chapters, can be conveniently represented through the inverse of its covariance matrix [33]. It is important to note, however, that this choice resides in the extreme left of the spectrum between statistical significance and the capacity of modelling non-linear relationships, whose existence is extensively proven and documented in gene-gene interactions. We, however, do not currently have the resources to create big datasets, and we will also ultimately benefit from the mathematical tractability of linear models.

1.6 Contributions and Roadmap

In the following chapters, we will first contextualize the reader with some mathematical background on Gaussian graphical models and conditional independence testing. In the third chapter, we will then delve into the mathematical and algorithmic core of the work; given that our methodology entails generating a set of competing GGMs, we will explain our approach to evaluating a GGM's performance and selecting the most appropriate model out of multiple options. In chapter 4, we will provide a review of our methodology as whole, which can be seen as a comprehensive data processing pipeline, from data treatment to model inference and evaluation. We will go into more detail on relevant contributions in the efficient implementation of the GGM Maximum Likelihood Estimator. Finally, in the fifth chapter we study the results, in terms of model performance, algorithmic convergence, and perhaps more importantly, how the results relate to current biological knowledge, and if we do indeed obtain a positive outlook in terms of supporting the Omnigenic hypothesis. We also understand that we ended up constructing a useful software package (`https://github.com/hugosc/ggm_scrna_seq`) that can be expanded upon and even applied to other domains outside of biology.

Chapter 2

Background

In this chapter, we aim to provide some theoretical context and intuition on multivariate Gaussian distributions, explain the instances where such distributions can be seen as graphical models, and finally give an overview of methods for finding the conditional independence graphs under the assumptions of Gaussianity.

2.1 The Multivariate Gaussian

A multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$, takes two parameters: a mean vector μ, and a covariance matrix Σ (or alternatively, its inverse, $\Omega = \Sigma^{-1}$), which is always positive semi-definite.

It is an appropriate choice for modelling multivariate data if

- each individual variable has a well behaved marginal distribution, having no fat tails and being approximately symmetric.
- Higher order relationships are not relevant to the problem, and one is content with only modelling the linear relationships between variables.

Although these are "simplifying" assumptions in most settings, they come with the benefit of mathematical and computational tractability, and often result in data efficient methods, especially useful when not many data are available.

In our setting, however, we are dealing with count data, which are usually modelled with a Negative Binomial Distribution or a Poisson distribution, which in theory are more appropriate to capturing the skewness prevalent in count data. Nonetheless, the overall number of RNA molecules in a cell is in the hundreds of thousands, with an average count per gene in the thousands, a large enough number for many genes to exhibit an approximately normal distribution. In the ideal setting, most or all marginal distributions would "look" Gaussian, but a typical distribution of a gene on our dataset can be seen in Figure 2.1, clearly displaying some skewness.

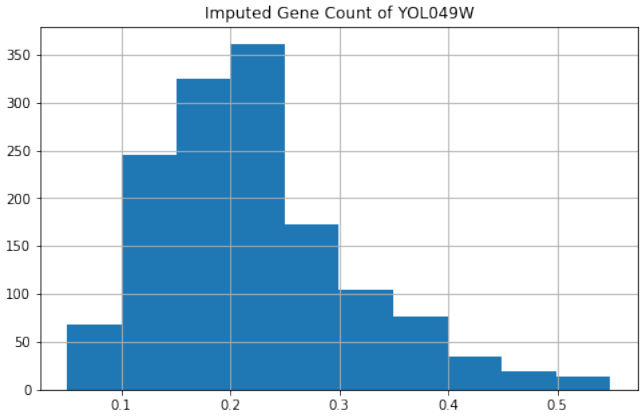


Figure 2.1: Histogram of YOL049W's expression levels

We recognize that this is an indication that using Gaussians models might not be ideal, but in most genes, such as the one shown above, the skewness is not large enough to the point of the empirical covariances being problematic or ill-defined, which would be the case, were the expression levels fat-tailed.

The formula for its pdf is as follows:

$$f(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\Big\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big\}. \tag{2.1}$$

This formula may seem daunting at first, but it is at bottom quite simple, which is better identifiable if we exclude the normalizing constants, and observe that the main part of the formula is simply the exponential

$$\exp\Big\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big\}.$$

This way, one might recognize that the exponent resembles some kind of norm on $\boldsymbol{x}-\boldsymbol{\mu}$, that is, it could relate to a distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}$, in a certain space not yet obvious. Indeed, $d_M(\boldsymbol{x},\boldsymbol{\mu},\Sigma) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$ is a well-defined distance between $\boldsymbol{\mu}$ and $\boldsymbol{x}$ in a space defined by Σ, and it is called the Mahalanobis distance. Such a space is linear, and there actually exists some matrix L, related to Σ, such that

$$(\boldsymbol{x}-\boldsymbol{\mu})^{T}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = \|L\boldsymbol{x}-L\boldsymbol{\mu}\|^2.$$

This means that we can interpret the inner term $(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ as the squared distance between $\boldsymbol{x}$ and $\boldsymbol{u}$, transformed to the linear space given by L. Furthermore, L is actually what underlies the definition of Σ, which can be found by $\Sigma = (LL^{\mathsf{T}})^{-1}$. Conversely, considering that Σ is, by definition, positive semidefinite, by the spectral theorem, and using the Cholesky decomposition, one can obtain a (non-unique) L

from Σ. By properties of the trace operator, we can express this distance term as

$$\|L\boldsymbol{x} - L\boldsymbol{\mu}\|^2 = \langle LL^\mathsf{T}, (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T} \rangle = \langle \Sigma^{-1}, (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T} \rangle$$

Where $\langle A, B \rangle$ is the summation of the element-wise product of A and B and $(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T}$ is the outer product of vector $\boldsymbol{x} - \boldsymbol{\mu}$. Therefore, we get the much more intuitive formula:

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \Omega^{-1}) \propto \exp\Big\{ -\frac{1}{2} \langle \Omega, (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\mathsf{T} \rangle \Big\}, \tag{2.2}$$

$$\propto \prod_{ij} \exp\Big\{ -\frac{1}{2} \omega_{ij} (x_i - \mu_i)(x_j - \mu_j) \Big\}, \tag{2.3}$$

where, as we mentioned previously, $\Omega = \Sigma^{-1}$ represents the precision matrix of the distribution.

2.1.1 How to Assess Conditional Independence

Given the formula above, we can make an interesting observation about the matrix Ω related to the influence of "cross terms" in the multivariate gaussian.

Let's take $z = (x_i - \mu_i)(x_j - \mu_j)$ as the product of the i and j elements of $\boldsymbol{x} - \boldsymbol{\mu}$. Then, the derivate of $f(\boldsymbol{x}; \boldsymbol{\mu}, \Omega^{-1})$ with respect to z is

$$\frac{\partial f(\boldsymbol{x}; \boldsymbol{\mu}, \Omega^{-1})}{\partial z} = -C\omega_{ij} \exp\Big\{ -\frac{1}{2} \omega_{ij} z \Big\}$$

From this derivative, we can see that the only term in Ω which interacts with the element z is ω_{ij}. We can also see that, if ω_{ij} is zero, then the derivative with respect to z is also zero. Therefore, the probability of $\boldsymbol{x}$ would not change directly with the product $(x_i - \mu_i)(x_j - \mu_j)$. Also, if ω_{ij} is not zero, the influence of z in the pdf is directly controlled by ω_{ij}. Although this may seem a convoluted way to state a tangential, albeit curious fact, it is of high importance if one notices that z looks a lot like the covariance between variables X_i and X_j. Additionally, in general, $\omega_{ij} = 0$ does not mean $\sigma_{ij} = 0$. So even if $\omega_{ij} = 0$, variables X_i and X_j may still be correlated.

Now, one should wonder, what does it mean, when two variables X_i and X_j can be correlated, but $(x_i - \mu_i)(x_j - \mu_j)$ not influence the pdf? Curiously, what it means is that X_i and X_j, although dependent and correlated, are conditionally independent, given all other variables of the distribution.

Indeed, as is formally proved in the appendix, the element ij of the precision matrix Ω is 0 if, and only if, variables X_i and X_j are conditionally independent, given all other variables. So, as we see in the next section, most methods of determining conditional independence in the Gaussian setting basically consist of trying

to identify which entries of Ω are zero. And finally, once one determines the sparsity pattern of the distribution's precision matrix, one obtains a Gaussian Graphical Model (GGM), which, when represented as a graph, its adjacency matrix is given by the precision matrix.

2.2 Finding CI Graphs

Given what we've previously discussed, the task of modelling a dataset through a GGM can therefore be described as attempting to obtain a sparse, positive semidefinite matrix, such that when interpreted as a precision matrix, the resulting Gaussian distribution maintains some level of likelihood with respect to the data. By far, the most challenging aspect of this task is actually finding the conditional independence structure (graph) of the distribution, i.e., the sparsity pattern of the precision matrix, because, as we will see in the next chapter, finding the actual values of the matrix after a certain structure is imposed is actually a straightforward maximum likelihood estimation problem. There are two main types of approaches when attempting to find the CI graph.

The first type of method consists of performing global optimization on the precision matrix, which attempts to simultaneously find a sparse solution while fitting the actual values of the matrix. So when the optimization is done, one already obtains a sparse precision matrix, optimized for some likelihood-ish objective function, without having previously specified the sparsity structure. To illustrate this general idea, we start with the already mentioned Graphical Lasso [9]: One of the first widespread methods for finding conditional independence in the Gaussian setting. With similar interpretation to Lasso regularization for linear regression, it is a regularized maximum likelihood method, where one adds an $l1$-norm penalty term to the log-likelihood function as an attempt to impose a sparse solution to the problem. Its most basic version consists of solving the optimization problem below:

$$\max_{\Omega \succeq 0} \log|\Omega| - \operatorname{tr}(S\Omega) - \lambda\|\Omega\|_1, \tag{2.4}$$

where $\log|\Omega| - \operatorname{tr}(S\Omega)$ is related the multivariate Gaussian's log-likelihood, as we will show in Chapter 3. To see why this would lead to sparse solutions, we can interpret the optimization problem above as the Lagrangian form of

$$\begin{aligned} &\max_{\Omega \succeq 0} \log|\Omega| - \operatorname{tr}(S\Omega) \\ &\text{s.t } \|\Omega\|_1 \leq t. \end{aligned}$$

The constraint $\|\Omega\|_1 \leq t$ produces a "tilted" hypercube solution space, with edges occurring when a subset of variables are zero, somewhat like multidimensional version of a diamond. The method's popularity was due to having the first computationally efficient implementation, enabling instances of 1000 variables to be solved. Many newer methods built upon the ideas developed and attempted to address their shortcomings; such examples are the Bayesian Graphical Lasso [35], the Graphical ElasticNet [17], and the de-sparsified Graphical Lasso [13].

The second class of methods take a more classical statistical approach, where we derive a statistic for the entries of the precision matrix, the idea being that ω_{ij} is 0 under the null hypothesis, indicating conditional independence of the variables X_i and X_j, given all other variables of the distribution:

$$H_{0ij} : \omega_{ij} = 0 \text{ versus } H_{1ij} : \omega_{ij} \neq 0 \tag{2.5}$$

Therefore, the problem of finding the CI graph of the distribution becomes a problem of multiple hypothesis testing. Such methods will commonly utilize linear regressions as an intermediate step to deriving these statistics. For instance, to check if X_i and X_j are conditionally independent, one can produce two regressions with variable set Z which excludes both X_i and X_j for finding the residuals ϵ_i and ϵ_j. It can be shown that if ϵ_i and ϵ_j are uncorrelated, then X_i and X_j are conditionally independent, which implies fitting two regressions for each variable pair ij. Alternatively, one can also make use of a node-wise Lasso Regression:

$$\underset{\beta_i,\sigma}{\arg\min} \left\{ \frac{\|X_i - \boldsymbol{X}_{i^c}\beta_i\|}{2n\sigma} + \frac{\sigma}{2} + \lambda \sum_{k \epsilon i^c} \frac{a}{a}|\beta_{ik}| \right\}, \tag{2.6}$$

where one attempts to predict the value of variable X_i from the rest of the distribution, $\boldsymbol{X}_{i^c}$. One then is able to combine β_{ij}, obtained when attempting to predict X_i and β_{ji}, obtained when attempting to predict X_j and derive a statistic for ω_{ij}. Unlike the Graphical lasso or similar methods, these approaches do not necessarily produce the Ω matrix, but only provide the matrices sparsity structure through the H_{0ij}s which failed to be rejected.

In general, these methods tend to be faster to execute, given that we skip a convex optimization problem with millions of variables (p^2 if the matrix is $p \times p$), and instead perform individual linear regressions. We will go into detail on the specific methods utilized in the Methodology chapter, but they generally follow the outline illustrated here. There is an additional component we've omitted in this chapter, for pedagogical reasons, but it will also be detailed later in chapter 4, which is a formula developed for conditional independence testing that enables one to control the false discovery rate of the results.

Ultimately, we did indeed observe that the statistical approaches were more

promising and computationally efficient, when compared to the Graphical Lasso-like methods we've briefly discussed. There are, of course high quality implementations for the Graphical Lasso, which claim to be able to solve problems with p up to one million, in a single machine [10], but we failed to apply these solutions to the same claimed efficiency, and finally, due to time and resource restrictions, we decided, after some preliminary testing, to not further explore the Graphical Lasso or its descendants, and they will not be present in the Results chapter.

Chapter 3

Selecting Candidate CI graphs

In the previous chapter, we discussed various approaches for determining the Conditional Independence (CI) graph of a distribution assuming Gaussianity. These methods included the Graphical Lasso and test-oriented approaches. In this chapter, we will focus on evaluating different and competing CI structures for the same data. This evaluation becomes essential when we are uncertain about the best method for various scenarios.

Our final objective is to find a plausible Gaussian Graphical Model (GGM) for the data, which is represented by a precision matrix denoted as Ω. Since a GGM is essentially a multivariate Gaussian distribution, we can assess the quality of each candidate CI graph based on how well the resulting Ω and Gaussian Distribution fit the data. In this case, the most straightforward measure for evaluating goodness of fit is the negative log-likelihood (NLL) of the distribution with respect to the data. The CI graph that leads to the smallest NLL is the preferred choice.

To illustrate the soundness of using the NLL for goodness of fit in distributions, we will provide an information theoretic interpretation of the NLL, showing why minimizing the NLL with respect to the data in a space of solutions results in the minimum KL divergence to the true distribution. Given a single data point $X \sim f$, sampled from a distribution f, the expected NLL of f is:

$$\mathbb{E}_X[\mathrm{NLL}(X; f)] = \mathbb{E}_X[-\log f(X)], \tag{3.1}$$

which coincides with the definition of Shannon Entropy, $\mathrm{H}(X)$. However, in the scenario one does not know the true distribution f, one could then bravely attempt to model and approximate f by means of another distribution g. In this case, regardless of the bravery one might hold, it would be reasonable to anticipate a

worse NLL, whose expected value would be given by

$$\mathbb{E}_X[\mathrm{NLL}(X; g)] = \mathbb{E}_X[-\log g(X)] \tag{3.2}$$

$$= \mathbb{E}_X\Big[-\log \frac{g(X)}{f(X)}\Big] + \mathbb{E}_X[-\log f(X)] \tag{3.3}$$

$$= \mathrm{KL}(f||g) + \mathrm{H}(X). \tag{3.4}$$

It is in fact not only reasonable, but mathematically necessary, to expect a worse NLL, given that $\mathrm{KL}(f||g)$ is a non-negative number. Moreover, since $\mathrm{H}(X)$ is constant with respect to g, the NLL of g can be seen as proportional to the KL divergence from the underlying distribution f to the modelling candidate g. The KL divergence is not a proper distance, it does, however, tend to be continuous and differentiable, and $\mathrm{KL}(f||g) = 0$ if, and only if, $f = g$. Therefore, if one searches a space G of distributions and uses the NLL for selecting $g \in G$, it would provide us with the closest distribution to the underlying f, in terms of the KL divergence.

Now, we have three pending methodological questions:

1. How do we find the Gaussian Graphical Model of a given CI graph and dataset?
2. How do we provide an unbiased estimate of this model's NLL?
3. How do we know what is a "good" model, given some NLL value?

To answer question 1, we uncover the GGM from the CI graph as the solution to an optimization problem, which we call Structured Maximum Likelihood Estimation. This problem is very similar to Multivariate Gaussian Maximum Likelihood Estimation, however, it adds the restriction of a predetermined sparsity structure to the precision matrix. This restriction causes the problem to no longer have a closed formula solution in the general case. So we need to approach it with an optimization algorithm. In any case, obtaining the maximum likelihood solution induces bias in the NLL on the training data (used for the optimization procedure). So, to answer question 2, we could either find additional data sampled from the distribution and use it exclusively for quantifying the NLL of a previously obtained model or derive a penalty term that compensates for the bias of the MLE. Given that we already have a small dataset, reserving a fraction of it for validation would compromise the quality of the models found through MLE. Therefore, in this dissertation, we decided to approximate penalty terms for the MLE bias. Such methods (Information Criteria) are detailed in the end of the chapter.

As for question 3, we propose a baseline model to compare against. The baseline for a GGM would be a model with all variables being independent and conditionally independent, i.e, a CI graph with no edges. There are two reasons for this graph

to be an appropriate baseline: First, as we saw in the previous chapter, in Section 2.5, the null hypothesis H_0 when assessing whether two variables are conditionally independent is that they are, in fact, conditionally independent. So, in some, sense, the joint null hypothesis would be that all variables are conditionally independent. The second reason is that, when all variances of the distribution (the diagonal of the covariance matrix) are fixed, the multivariate Gaussian distribution with the highest possible entropy, i.e., the highest degree of uncertainty, is the one where all non-diagonal entries of the covariance and precision matrices are zero.

3.1 Structured Maximum Likelihood Estimation

3.1.1 Multivariate Gaussian MLE

From here onward, we assume that the mean vector μ is 0 in order to simplify notation. We first review the formula for the classical MLE of a multivariate Gaussian so we can build upon it to show how we can add the conditional independence restrictions. Assuming $\mu = 0$, we can reduce the pdf in Equation 2.1 to

$$f_\Sigma(\boldsymbol{x}) = (2\pi)^{-p/2}|\Omega|^{1/2}\exp\Big\{-\frac{1}{2}\boldsymbol{x}^T\Omega\boldsymbol{x}\Big\}$$

where $\Omega = \Sigma^{-1}$.

Let $\mathcal{D}_{[1:n]}$ be the dataset of p-dimensional random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. We calculate the NLL in regard to the precision matrix Ω as

$$\begin{aligned}\mathrm{NLL}(\mathcal{D}_{[1:n]};\Omega) &= -\sum_{i=1}^{n}\log f(\boldsymbol{x}_i) && (3.5)\\ &= -\frac{np}{2}\log 2\pi - \frac{n}{2}\log|\Omega| + \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{X}_i^T\Omega\boldsymbol{X}_i && (3.6)\end{aligned}$$

As shown in the appendix Section A.1.1, we can simplify the minimization of the NLL to the following problem:

$$\max_{\Omega\succeq 0}\log|\Omega| - \mathrm{tr}(S\Omega),$$

where $S = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i^T$, is the empirical covariance matrix. In general, if $n > p$, then the problem is well defined, and the closed form solution is demonstrated to be $\hat{\Omega} = S^{-1}$.

3.1.2 Conditional Independence

As illustrated in the previous chapter, given a precision matrix Ω, we can easily know whether two variables X_i, and X_j, are conditionally independent:

Proposition 1. *Let $A = \{i, j\}$, $B = N \setminus A$, then $X_i \perp\!\!\!\perp X_j | B$ if, and only if $\omega_{ij} = 0$.*

Proof. The proof can be read in the appendix, Section A.1.2. □

From proposition 1, we can conveniently codify the structure of conditional dependence between our variables as a set E of edges, containing self-loops, codifying which entries of Ω are allowed to be non-zero. And thus, the MLE becomes

$$\begin{aligned} &\max_{\Omega \succeq 0} \log |\Omega| - \operatorname{tr}(S\Omega) \\ &\text{s.t. } \omega_{ij} = 0 \ \ \forall ij \notin E. \end{aligned} \tag{3.7}$$

Additionally, we can reformulate the problem in Eq. (3.7) as the dual:

$$\begin{aligned} &\max_{\Sigma \succeq 0} \log |\Sigma| \\ &\text{s.t. } \sigma_{ij} = s_{ij} \ \ \forall ij \in E, \end{aligned}$$

where s_{ij} are the entries of S, the empirical covariance matrix. The derivation of the dual is not obvious, but the reader can refer to [33].

3.1.3 Algorithm for the MLE

To solve the optimization problem shown in the previous section, there is no known closed-form solution for general graphs, but it is, at least, a convex problem to solve. So, although it has many variables, we can obtain a simple algorithm that converges to the optimum.

We ultimately decided on a method that is memory-efficient, and is most widely used for this problem in the literature: coordinate descent, which does not require us to calculate Hessians or Jacobians, or any other German sounding matrix involving derivatives (which scale quadratically with the number of variables in the problem).

Although both the primal and dual formulations lead to valid implementations, we implement the solution on the primal, given that we believe the instances in our domain to be edge-sparse. As such, we would cycle through edges, and update their respective entry in the precision matrix iteratively. There is, however, a way to speed-up the cycling by grouping edges together and optimizing them all at the same time. Below we show what are the conditions for a set of edges to be grouped together and have a closed-form solution in their simultaneous optimization step.

For a given set $M \subseteq \{\{i,i\} \mid \{i,i\} \in E\}$ and a current solution Ω^0, we need to solve

$$\begin{aligned} \max_{\Omega} \quad & \log|\Omega| - \mathrm{tr}(S\Omega) \\ \text{s.t} \quad & \omega_{ij} = \omega^0_{ij} && \forall ij \notin M \\ & \omega_{ij} = 0 && \forall ij \notin E, \end{aligned}$$

in order to perform a coordinate step on M. If M defines a block sub-matrix of Σ, we can apply the Schur Complement to isolate a subproblem related to M to solve. The complete derivation can be found in the appendix.

Proposition 2. *Let $M \subseteq \{\{i,i\} \mid \{i,i\} \in E\}$ and $A = \bigcup_{ij \in M}\{i,j\}$ Fixing all variables outside of M is equivalent to the following optimization problem:*

$$\begin{aligned} \max_{\Omega' \succeq 0} \quad & \log|\Omega'| - tr(S_{A,A}\Omega') \\ s.t \quad & \omega_{ij} = 0 && \forall ij \notin E \\ & \Omega' = \Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}. \end{aligned}$$

Proof. The proof can be read in the appendix, Section A.1.2. □

If we select the edge set M as a clique, then we know that Proposition 2 has a closed solution.

Corollary 1. *Let $B = V \setminus A$ be the vertex complement of A. If M is a clique, then the solution $\hat{\Omega}$ to 2 is*

$$\begin{aligned} \hat{\Omega}_{A,A} &= S^{-1}_{A,A} + \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}, \\ \hat{\Omega}_{A,B} &= \Omega^0_{A,B}, \\ \hat{\Omega}_{B,B} &= \Omega^0_{B,B}. \end{aligned}$$

Proof. The proof can be read in the appendix, Section A.1.2. □

It is proved by Speed and Kiiveri that if one selects a sequence of such sets that cover all the edges, an algorithm that sequentially solves the problem in Proposition 2 will converge to the solution of Equation 3.7 [30]. Therefore, a full algorithm for this optimization problem could be described as first finding a clique cover

$M_1, M_2, \ldots, M_k$ of all the edges, and then iteratively applying the formula in Corollary 1 to obtain solutions $\Omega^0 = I, \Omega^1, \ldots, \Omega^{lk+i}$, where one goes through each of the k cliques at least l times until one reaches some convergence criterion. It is important to note that there are some special cases of graphs with which we are guaranteed to reach the solution within a short number of iterations. The best known class is the triangular, or chordal graphs, where one reaches convergence with $l = 1$, i.e., after only traversing each clique once, given an appropriate choice of clique sequence. Trivially, the complete graph is chordal, and the sequence of $M_1 = E$ reduces to the original MLE of the multivariate Gaussian, where we assume all variables are conditionally dependent, and has closed-form solution.

Algorithm 1 Strucutured MLE Algorithm

```
1: procedure MLE(S, G, ϵ)
2:     Ω_1 ← I
3:     s ← ∞
4:     C ← CliqueEdgeCover(G)
5:     while s > ϵ do                          ▷ Check for convergence
6:         Ω_0 ← Ω_1
7:         for A ∈ C do
8:             B ← V − A
9:             Ω^1_{A,A} ← (S_{A,A})^{-1} + Ω^0_{A,B}(Ω^0_{B,B})^{-1}Ω^0_{B,A}
10:        end for
11:        s ← distMetric(Ω_0, Ω_1)        ▷ distance metric for convergence criterion
12:    end while
13:    return Ω_1
14: end procedure
```

The function call CliqueEdgeCover(G) in the algorithm represents a procedure that finds sets of clique-inducing vertices that cover all edges of G, i.e., if uv is an edge of G, then at least one set returned by CliqueEdgeCover(G) must contain both u and v. Although not necessary, this cover can also be a partition, and we discuss best choices of such an algorithm in the Methodology chapter. Also worthy of note is that we must initialize the algorithm with a viable solution, and such a solution, for any graph, will always be the identity matrix, because it is positive definite and trivially, all non-edges of G appear as a 0 in the I matrix.

3.2 Information Criteria

Suppose we have a dataset $\mathcal{D}_{[1:n]}$ and a known CI structure codified by graph G. Furthermore, suppose we obtain the solution Ω^* to the Structured Maximum Likelihood with respect to$\mathcal{D}_{[1:n]}$ and G. Let $l = \text{NLL}(X_n, \Sigma^*)$ be the NLL obtained by the Structured MLE algorithm. As we previously mentioned, l comes with the bias of

being calculated on $\mathcal{D}_{[1:n]}$ given that we already utilized $\mathcal{D}_{[1:n]}$ for the optimization procedure.

The **Akaike Information Criterion** estimates that this bias is proportional to the number of free parameters of the model. More specifically, in our context:

$$\mathrm{AIC}(G, N) = 2l + 2m(G), \tag{3.8}$$

where $m(G)$ represents the number of edges in the graphical model codified by G. This formula basic states that, for models with the same log-likelihood, models with fewer parameters would be preferred [2]. Intuitively, one could think of this as the mathematical version of Occam's razor, where an explanation with fewer assumptions is preferred. Indeed, more model parameters can be seen as more assumptions, especially in this context, where an edge ij of G implies the rejection of the null hypothesis of i and j being conditionally independent. Additionally, we can interpret that, if one chooses a model with few parameters, then the NLL calculated in the training data, has very little bias, and is a reliable metric of model fit, however the quality of the fit itself may be poor, due to the model being overly simplistic. Therefore, it makes sense to continue to add parameters to the model as long as the ensuing decreases in the NLL are significant.

This criterion, however, due to its strong assumptions, can often lead to underestimating the needed compensation from the number of parameters, especially in the high-dimensional but small sample size setting [6]. This led to the development of criteria that involve incorporating some prior knowledge about the distribution of solutions, in this case, graphs, giving preference to more sparsity or vice-versa. They are called the **Extended Bayesian Information Criteria** [8], and are defined as follows.

$$\mathrm{BIC}_\gamma(G, N) = 2l + \log(N)m(G) + 2\gamma\binom{(n(G)-1)n(G)/2}{m(G)}, 0 \leq \gamma \leq 1. \tag{3.9}$$

In the formula, the parameter γ allows one to control the prior of the solution in terms of how prevalent solutions with the same number of parameters are, given the maximum possible number of parameters. Larger values of γ more strongly penalize models with a large number of parameters. In the special case where one takes the parameter γ to be zero, the EBIC simplifies into the original **Bayesian Information Criterion**.

$$\mathrm{BIC}(G, N) = 2l + \log(N)m(G). \tag{3.10}$$

In [8], however, the authors report that their simulation works best with $\gamma \geq 0$, in the

context of Gaussian Graphical Models. We can see that when the number of data points (N) is larger than 7, then the penalty of $m(G)$ exceeds the AIC's penalty.

In general, assuming that any of the above criteria are suitable, if one selects the candidate model $\hat{\Omega}$ that minimizes a criterion, this same model should also be the one with the smallest NLL with respect to true underlying distribution f, not the sample $\mathcal{D}_{[1:n]}$. Now one might ask, how do we know which are suitable to this context? Luckily, as we will see in chapter 5, all criteria mostly agree with one another on which model is best.

Chapter 4

Methodology

In this chapter, we will turn our focus to implementation specifics, given that we already provided some level of mathematical detail or, at least, intuitive explanations on the major points.

4.1 Data Pipeline Overview

In order to produce GGMs from our data, we needed to implement a pipeline of data processing steps. This consists of three main blocks: data preprocessing, model inference, and model evaluation.

In data pre-processing, we both do some preliminary data cleaning, such as removing columns that have no information, we then perform data imputation through MAGIC [34], a manifold Learning technique that aims to recover the original relationships between data points, when count data suffer from dropout.

In the model inference block, we perform three steps. We calculate the empirical covariance matrix from the imputed data. Here, we apply a Shrinkage Covariance estimation method. After that, we run the SILGGM [38] package to uncover the conditional independence structure of the data. And finally, using the estimated empirical covariance matrix and the learned independence structure, we run the Structured MLE algorithm to find the GGM.

In the last block, we run Information Criteria to perform model selection, given that we generate more than one candidate Conditional Independence structure.

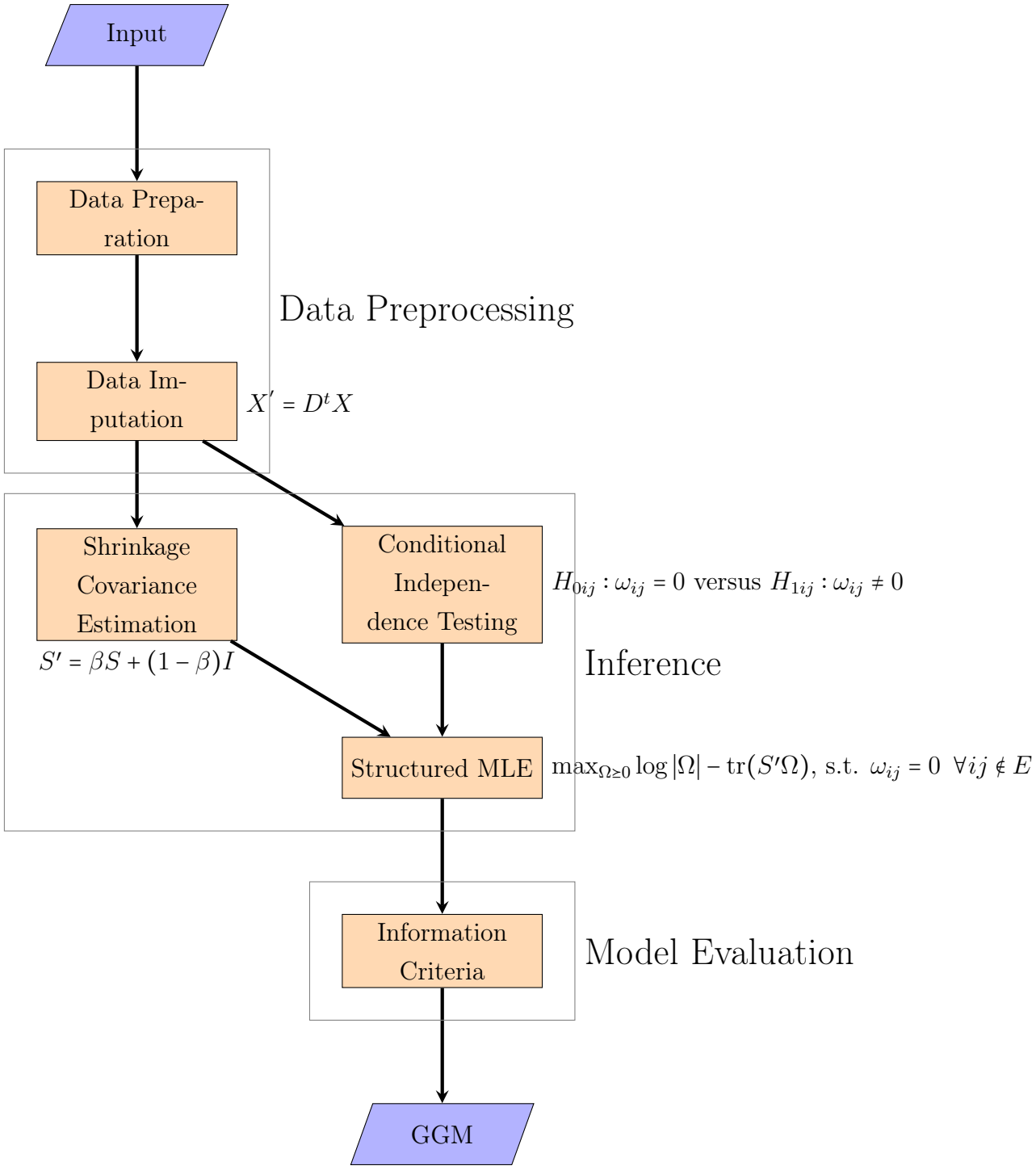


Figure 4.1: The execution steps of the pipeline

4.2 Data Imputation

In the context of Single Cell RNA-Seq data, we have high dimensional count data (thousands of genes) and a high dropout rate. Most SCRNA-Seq technologies are reported to only account to 5-10% of RNA molecules found in the cell. This creates issues when attempting to study correlations between genes, especially given that

many genes represent less than 5% of the total RNA count, and are likely to have 0 counts in some SCRNA-Seq readings. Therefore, we are faced with the need for some imputation method prior to learning the GGM or performing analyses. We present the algorithm we selected below.

4.2.1 MAGIC algorithm

The MAGIC algorithm can be classified as a manifold learning algorithm, where high dimensional data exist embedded in a lower dimensional manifold, and the manifold's structure can be somewhat reconstructed when looking at small regions at a time and observing neighboring data. The MAGIC method (as is characteristic of manifold learning methods) represents this manifold using a nearest neighbor (NN) graph, and in the context of expression data, each node represents a cell, and edges connect most similar cells, based on gene expression [34]. Finally, it leverages this NN graph to perform and physics inspired imputation procedure.

In the article, MAGIC was evaluated on four different scRNA-seq datasets, and reportedly recovered fine phenotypic structure in the data, including well-separated clusters, bifurcating developmental trajectories as well as heterogeneous state transitions.

The method can be summarized as follows:

1. Given K, produce a K nearest-neighbour graph,
2. From the graph, create a diffusion operator D (a $p \times p$ matrix),
3. Calculate the imputed data $X' = D^t X$.

Both the values of K and t are adjustable parameters that the model itself does not provide a way of determining optimally. Thus the need for a validation method that evaluates the overall qualify of the imputation method, so we can test different options for MAGIC's parameters.

4.2.2 Molecular Cross-Validation

Molecular Cross-Validation (MCV) is an extension of the basic cross-validation method for the case of count data. It is the approach we used for evaluating methods for denoising/imputing SCRNA-Seq data. It provides an unbiased way to both calibrate a given imputation method and to compare its performance to that of other methods [3].

The key feature is that it directly estimates the quantity of interest: the similarity of the denoised data to the full set of mRNA present in the original cell. It proposes a specific way to split count data from a matrix X into two matrices of the same

dimension, X_{train} and X_{val}, where the number of counts in X_{val} is α times the number of counts in X_{train}. If we perform such as split, then the MCV loss for imputation method f and dataset X is

$$\text{loss}(f, X) = \mathbb{E}\|\alpha f(X_{\text{train}}) - X_{\text{val}}\|^2 \tag{4.1}$$

One can then use this splitting procedure and the accompanying loss function to find the best hyper-parameter set for a given method.

4.3 Shrinkage Covariance Estimation

Shrinkage methods for covariance matrices, in general, are tools for handling the estimation of a covariance matrix in the setting of high-dimensional small data sets. In this scenario, the pure sample covariance matrix is typically not well-conditioned and can even be not invertible, which is a problem in our context, given that we are very often computing things in terms of the precision matrix, the inverse of the covariance matrix.

These methods usually follow the idea of performing a convex combination of the sample covariance matrix and the identity matrix

$$S' = \beta S + (1 - \beta)I, \tag{4.2}$$

and the specifics of each method involve determining the value β as a function of the data set (usually its size). Independent of the value of β, we can see that performing this convex combination guarantees that the resulting S' will be positive definite, a necessary property for us to execute the Structured MLE problem across many possible CI graphs.

We ultimately chose to apply the Ledoit-Wolf estimator [19] to our imputed data, given that this estimator has a proper implementation in Scikit-Learn [24], is probability distribution-free, and is an asymptotically optimal shrinkage method.

4.4 Conditional Independence Testing

For learning the conditional independence structure from the imputed data, we used the R package SILGGM [38]. This package provides options for an array of algorithms for determining conditional independence structures in the Gaussian setting, it is however more geared towards the more computationally efficient methods which do not involve global optimization, and instead perform individual inference of the precision entries. We gave preference to utilizing the de-sparsified nodewise scaled Lasso (D-S_NW_SL) [14]. We, found of particular interest their implementation

of a False Discovery Rate (FDR) control framework, taking the work of Liu [21] on his method and generalizing it to other available methods in the package.

Liu's framework works with any statistic T_{ij} that, under the null hypothesis $w_{ij} = 0$, asymptotically follows a standard normal distribution. I.e., $T_{ij} \xrightarrow{n} \mathcal{N}(0,1)$.

Given some true value of Ω with underlying CI graph G, and given a threshold t for the statistics T_{ij}, the proportion of false discoveries in data set $\mathcal{D}_n$ sampled from $\mathcal{N}(0,\Omega)$ will be

$$\mathrm{FDP}(\mathcal{D}_n, t) = \frac{\sum_{ij \notin E(G)} I\{|T_{ij}| \geq t\}}{\max\{\sum_{i,j \in V(G)} I\{|T_{ij}| \geq t\}, 1\}}, \tag{4.3}$$

the ratio of non-edges to node pairs whose respective $|T_{ij}|$ exceeded t in absolute value. The maximum in the denominator is there in the case of no discoveries, to make the FDP = 0. Additionally, we can define the FDR as the expected value

$$\mathrm{FDR}(t) = \mathbb{E}_{\mathcal{D}_n}[\mathrm{FDP}(\mathcal{D}_n, t)]. \tag{4.4}$$

Liu demonstrates two things: that $\mathrm{FDP}(\mathcal{D}_n, t)$ converges with high probability to $\mathrm{FDR}(t)$ as n grows, and that, despite not knowing the distribution's true precision Ω and its respective CI graph G, we can approximate the enumerator in Equation 4.3 by

$$\sum_{ij \notin E(G)} I\{|T_{ij}| \geq t\} \approx (2 - 2\Phi(t))(p^2 - p)/2, \tag{4.5}$$

where p is the number of variables in the distribution and Φ is the inverse CDF of the standard normal. Therefore, our FDR estimator becomes

$$\widehat{\mathrm{FDR}}(t) = \frac{(2 - 2\Phi(t))(p^2 - p)/2}{\max\{\sum_{i,j \in V(G)} I\{|T_{ij}| \geq t\}, 1\}} \tag{4.6}$$

Now, if we want to keep the FDR below certain level α, we just have to choose a t such that $\widehat{\mathrm{FDR}}(t) \leq \alpha$. The smaller the t, the better statistical power, so the recommended method is to get the infimum defined by the previous inequality. In the end, we settled on 4 different graphs, all generated using the D-S_NW_SL algorithm. We generated the graphs by varying the p-value threshold for the graph edges and by enabling and disabling FDR control.

(G0): with p-value threshold 0.05,

(G1): with p-value threshold 0.01,

(G2): with p-value threshold 0.05 and FDR control enabled,

(G3): with p-value threshold 0.01 and FDR control enabled.

4.5 GGM Maximum Likelihood Estimation

The algorithm, as described in chapter 3, was implemented in Python, despite Python being known for its inefficient execution, since it has a wide array of efficiently implemented libraries. And indeed, the algorithm itself has a very simple logic, most if its complexities lying in two parts: the numerical computations related to line 9 of algorithm 1, and the calculation of the clique edge cover.

For the numerical parts, we used Numpy and Scipy, whose linear algebra functions are basically wrappers to C code and BLAS calls.

4.5.1 Clique Algorithms

For the clique edge cover algorithms, the first one implemented, the basic partition algorithm, we used C++ and called the function in Python by using the Google-funded pybind11 library. The basic partition algorithm consists of finding an arbitrary partition of the graph edges into cliques. This is done in the most straightforward way possible: We select an arbitrary edge uv, find a maximal clique c which contains uv and remove the clique (edges) from the graph. This procedure is repeated until the graph has no more edges. The algorithm is guaranteed to terminante because at every step, we remove at least one edge.

Algorithm 2 Basic Partition Algorithm

```
1:  procedure PARTITION(G)
2:      G_0 ← G
3:      P = ∅
4:      while m(G_0) > 0 do
5:          uv ← selectEdge(G_0)
6:          C ← FindMaximalClique(uv, G_0)
7:          P ← P ∪ {C}
8:          E(G_0) ← E(G_0) ∖ E(C)
9:      end while
10:     return P
11: end procedure
```

The FindMaximalClique procedure also has a straightforward implementation, we start the clique with two nodes, u and v, and add one node at a time. If C is the current clique, we take any vertex $w \in \bigcap_{u \in V(C)} N(u)$, that is neighbours with all clique vertices, and add it the the clique.

This next algorithm was implemented using the Python library networkx for graphs and networks. We decided to explore the case where we allowed edge repetition in our cover, and thus no longer obtained a partition. This required a different approach to the basic partition algorithm, where instead of finding one clique at

a time and removing it from the graph, we instead decided to first enumerate all maximal cliques in the graph and then greedily and sequentially selected the cliques until all the edges are covered. The criterion for greedy selection is that the most "valuable" clique is the one containing the most edges that have not yet been covered by previously selected cliques.

Algorithm 3 Clique Edge Cover Algorithm

```
1: procedure CliqueEdgeCover(S, G, ϵ)
2:     MC = AllMaximalCliques(G)
3:     S_mc ← {m(C) | ∀C ∈ MC}
4:     Q ← PriorityQueue(MC, S_mc)
5:     P ← ∅
6:     E_c ← ∅
7:     while Σ_{C∈P} m(C) < m(G) do
8:         C ← Q.pop()
9:         P ← P ∪ {C}
10:        E_c ← E_c ∪ E(C)
11:        for C′ ∈ Q do
12:            Q.updatePriority(C′, |E(C′) − E_c|)
13:        end for
14:    end while
15:    return P
16: end procedure
```

We first initialize the priority queue with all maximal cliques, where each clique's priority is its number of edges. Therefore the first clique to be selected will be the graph's largest clique. After a clique is selected, we then downgrade the priority of each clique by the number of edges that they share with the selected clique (which were not yet covered). This guarantees that, at each step, we are able to greedily select the clique the covers the most number of edges by simply popping the queue.

4.5.2 Conditional Decorrelation

there are some relevant implementation details to highlight here. As shown in Algorithm 1, in Chapter 3, there is a step that computes $S_{A,A}^{-1} + \Omega_{A,B}^0 (\Omega_{B,B}^0)^{-1} \Omega_{B,A}^0$, for all cliques A. But given that we know all these cliques ahead of time, we precompute all inverses $S_{A,A}^{-1}$ at the beginning of the algorithm as a preprocessing step. Additionally, we initially experienced numerical instability when computing the inverse $(\Omega_{B,B}^0)^{-1}$ for calculating the term $\Omega_{A,B}^0 (\Omega_{B,B}^0)^{-1} \Omega_{B,A}^0$. So we decided to apply a trick of skipping the calculation of that inverse and instead solve the linear system

$$\Omega_{B,B}^0 \boldsymbol{x} = \Omega_{B,A}^0$$

, which has solution $\boldsymbol{x} = (\Omega^0_{B,B})^{-1}\Omega^0_{B,A}$, so we arrive at

$$\Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A} = \Omega^0_{A,B}\boldsymbol{x}.$$

These optimizations led to faster execution and better numerical instability.

4.5.3 Convergence Criteria

For this kind of algorithm, usually one utilizes the Δ in the solution as the criterion for stopping. In this case, it would be whether $\Delta = \Omega^1 - \Omega^0 \approx 0$. Given that the omegas are matrices, one common approach would be to check that the maximum element $\max\{\delta_{ij} \mid i, j \leq p\}$ has reached a certain very small threshold ϵ. In practice, however, we would not recommend this approach, because, as we will see later, due to the large size of the matrices, there is still some numerical instability (even after the optimizations shown in the previous section), and the solution may always fluctuate, even if there is a theoretical guarantee of convergence. We recommend instead using a δ in the objective function, and maintain the execution of the algorithm while the difference in log-likelihood is positive, and above a certain ϵ.

Chapter 5

Results

In this chapter, we delve into the results of the pipeline as a whole, so we look into what the results of each important step were, such as data imputation, the resulting networks inferred by the Condition Independence testing algorithms, the execution profile of the structured MLE and finally, how the obtained GGM maintains biological plausibility.

5.1 The Selected Data set

We selected, for our experiments, a scRNA-seq data set of the model organism *Saccharomyces cerevisiae* [12]. The full data set contains readings for combinations of different strains (lab-generated by gene deletions) and environmental conditions. We selected a "rich" environment, Yeast Extract + Peptone + Dextrose (YPD), assuming it would provide us with "typical" expression patterns. As for the strain, we simply selected the one which had the most data points in combination with the YPD environment. This selection resulted in a subgroup of the data that contained 1419 datapoints. Out of all 6827 columns (genes), we selected 5697 gene columns which had a non-zero count in at least one of the datapoints.

5.2 Impact of Data Imputation

We applied the Molecular Cross-Validation procedure to the MAGIC imputation algorithm, testing different values of k (number of nearest neighbors to use when constructing the similarity graph), and t (the intensity of the diffusion) and obtained the results shown in Table 5.1.

Table 5.1: Table showing the Molecular Cross-Validation loss for each hyper-parameter configuration

	mcv_loss	k-nn	t
0	0.086977	5	1
1	**0.080744**	**5**	**3**
2	0.080875	5	5
3	0.083197	10	1
4	0.080844	10	3
5	0.081049	10	5
6	0.082036	15	1
7	0.080937	15	3
8	0.081164	15	5

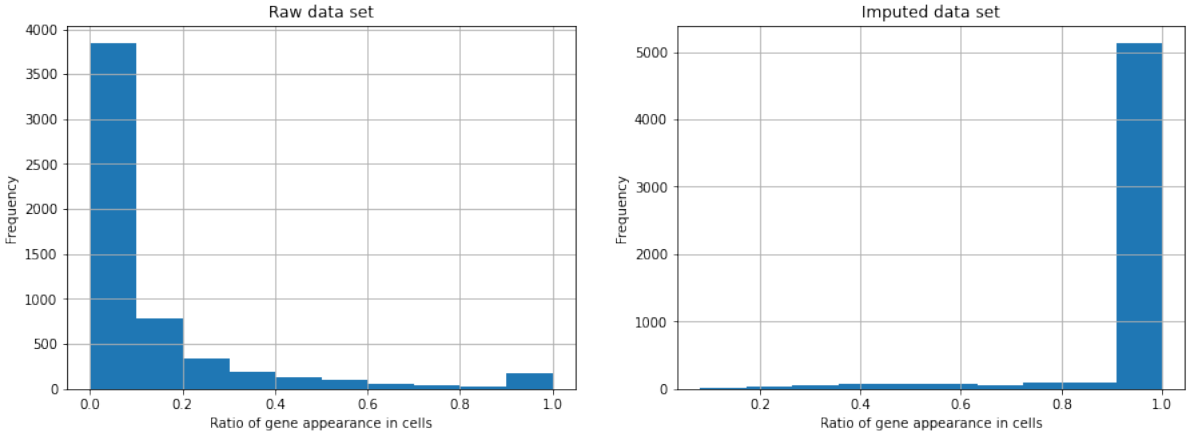
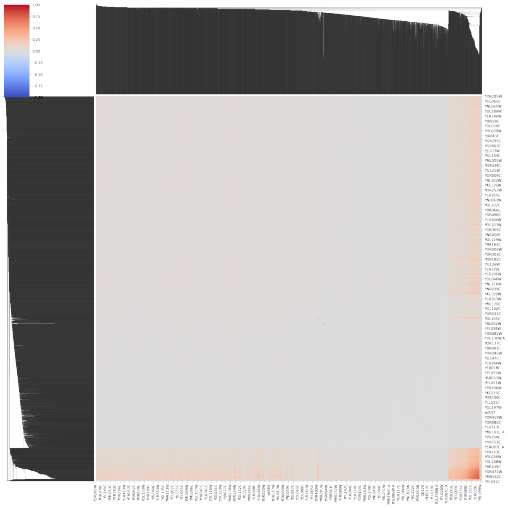


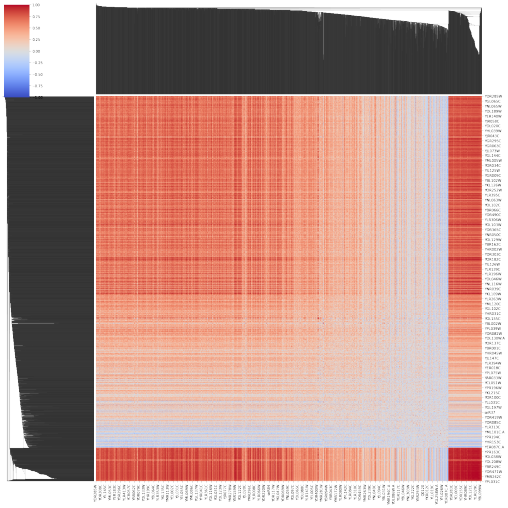
Figure 5.1: Histograms illustrating the frequency or zero counts before and after the imputation procedure

In these results, we see that out of all values of k, $t = 3$ was the optimal option, as 1 did not perform a strong enough imputation, and 5 arguably makes too strong an effect in the data. Overall, the parameter values found effected a somewhat "light" amount of imputation to the data. It however, made a big difference in terms of eliminating zero values from the dataset. As we can see in Figure 5.1, in the raw data, most of the genes very rarely were expressed in the cells ($<$10%), and after the imputation procedure, almost all genes were present in all cells. It is important to note that the imputation procedure creates continuous values. In the raw data set, a gene was expressed in terms of a non-negative integer. In the imputed data, a gene can now have an expression of, let's say, 0.3, in a given cell. The application of the diffusion operator pulled expression levels of each gene closer to their average expression value, as the average standard deviation for each gene reduced from 0.5 to 0.21.

We can also observe, in Figure 5.2 the difference in the correlation matrix caused by the imputation procedure, yielding much stronger correlation signals.

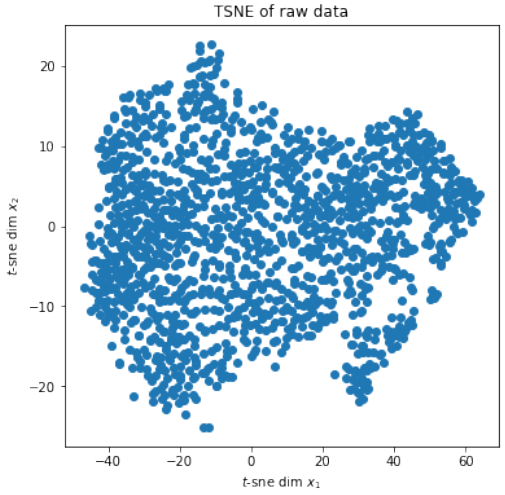


(a) Correlation matrix before imputation

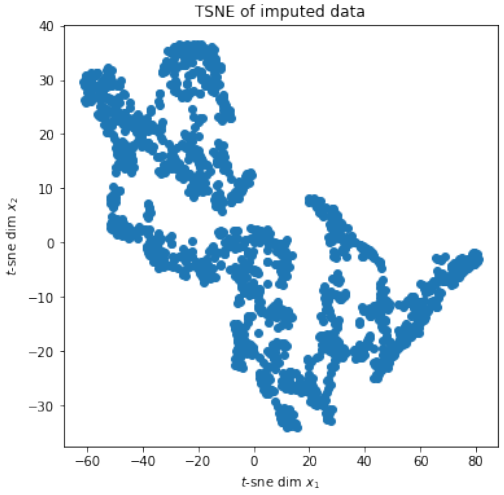


(b) Correlation matrix after imputation

Figure 5.2: Correlations of gene expression levels before and after imputation



(a) Correlation matrix before imputation



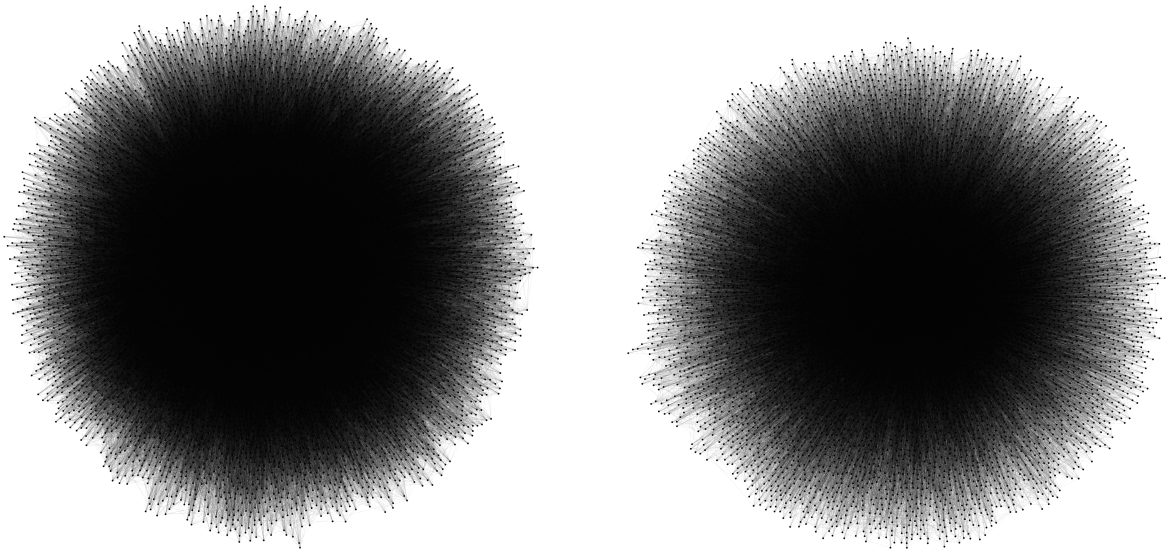
(b) Correlation matrix after imputation

Figure 5.3: Correlations of gene expression levels before and after imputation

One the could wonder if this increase in correlation signals comes at the expense of the fine structure of the data, but as we see in Figure 5.3, it is unlikely to be the case, since the TSNE plots both before and after imputation still indicate a latent subspace.

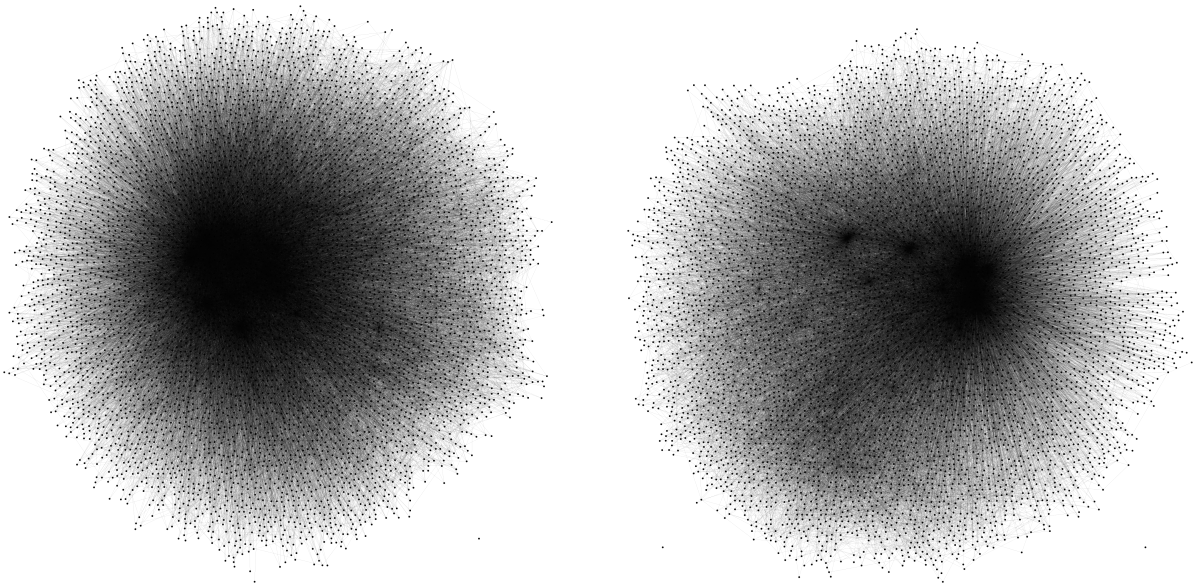
5.3 Network Analysis of the CI Networks

Let us first delve into the basic network statistics for our graph instances. In table 5.2, we see that all graphs have a low density between 0.2% and .9%, but they are still mostly connected, all having a giant component and at most two isolated nodes, even though the sparsest graph has an average degree more than four times lower



(a) G0 (b) G1

Figure 5.4: Instances without FDR control



(a) G2 (b) G3

Figure 5.5: Instances with FDR control

Table 5.2: Descriptive table of basic network statistics for the graph instances.

	G2	G0	G3	G1
FDR Control	Yes	No	Yes	No
Edge p-value	0.05	0.05	0.01	0.01
n(G)	5697	5697	5697	5697
m(G)	44472	153210	35002	102132
average_degree	15.6124	53.7862	12.2878	35.854
edge_density	0.00274	0.00944	0.00215	0.006295
largest_component	5696	5697	5695	5697
#_isolated_nodes	1	0	2	0
avg_shortest_path_length	2.57309	2.1950	2.65633	2.31923

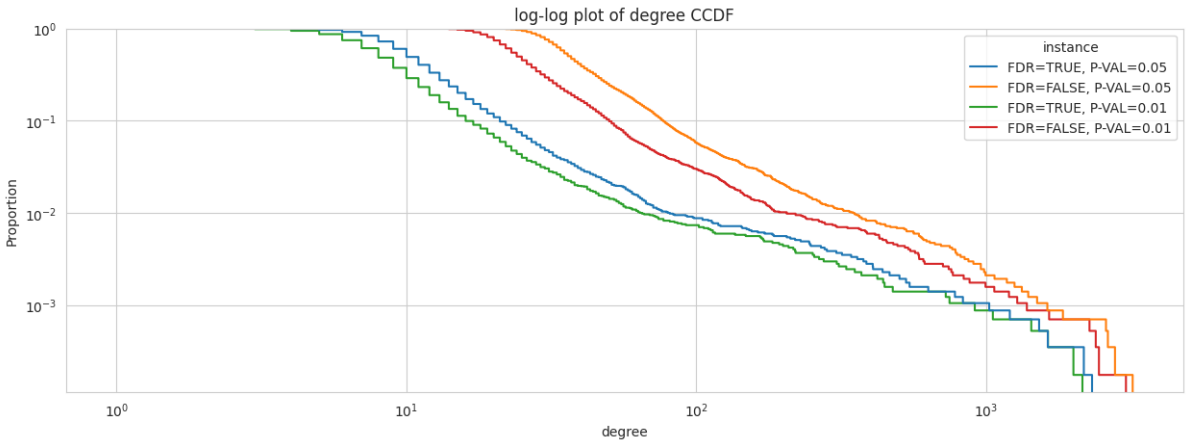


Figure 5.6: CCDF plot the graph instances' degree distribution.

than the densest graph.

We also observe that the average shortest path length is very small, smaller that 3 in all instances, despite the graphs having low densities. These observations are consistent with the graph version of the Omnigenic Hypothesis. Also, this is highly indicative of the existence of hubs in the networks and of heavy-tailedness in the degree distribution. Indeed, as we investigate the degree distribution for all graphs, we see that all of the instances exhibit fat-tailed behaviour. This behaviour can be observed by inspecting the log-log plot of the empirical complementary cumulative distribution function (CCDF), as shown in Figure 5.6:

Usually, in such a plot, fat-tailed distributions yield an almost linear or with a very slowly changing derivative. And we can see that this is indeed the case for all instances, as shown by Figure 5.6. Upon further inspection, one can observe that all the empirical CCDFs have a very similar shape, and they all display an unusual curve, which is not quite power law, and could be interpreted to have two different regimes. The first, with a stronger decay, and a second, towards higher values, that decreases more slowly. Indeed, if we attempt to fit a power law to each instance, we

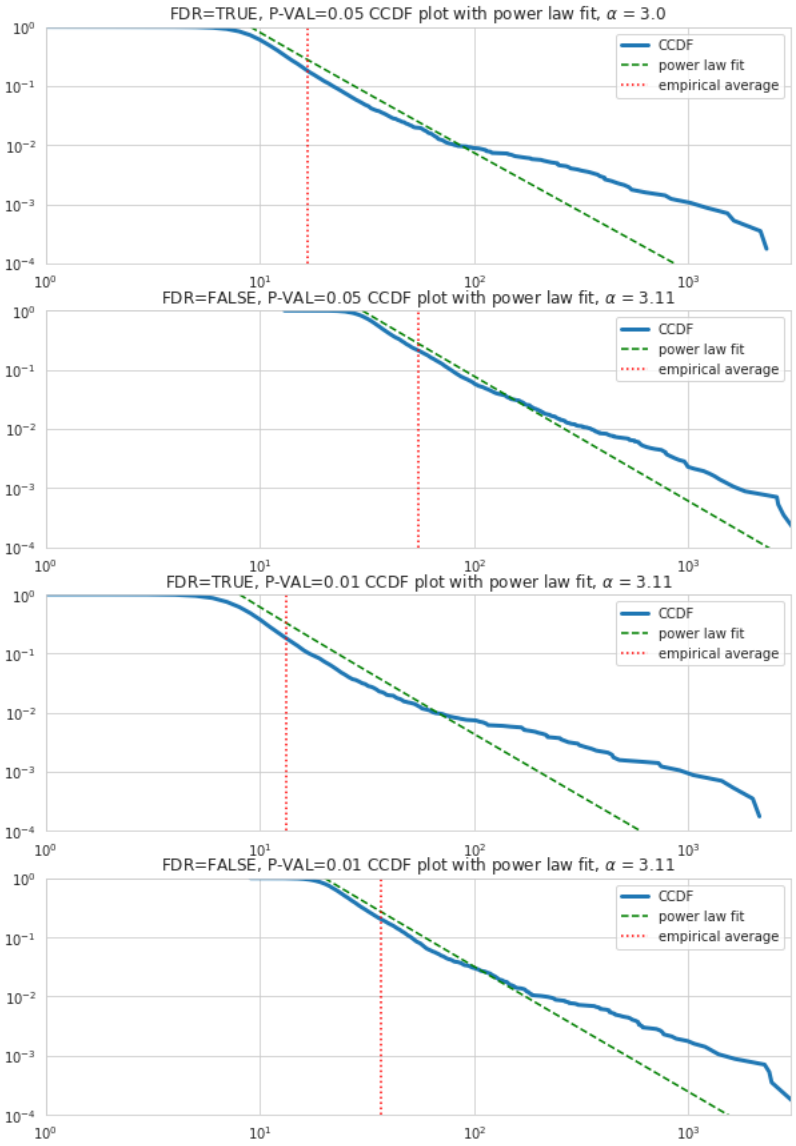


Figure 5.7: Power-law fits to the degree distributions.

see that higher end of the curve always exceeds the fit. Additionally, all αs are very similar (with the exponent of the curves being $-\alpha$), of around 3 and the way that the curve exceeds the power-law fit is very consistent among the instances. The fits were found using the Python library powerlaw, and in this case, in addition to α, we also allowed the fit to adjust the minimum values of the curves.

This observation of consistency, alongside the fact that all the instances can be sorted into a subgraph sequence, motivates the hypothesis that all instances are structurally very similar, with all nodes (degreewise) being affected very uniformly as we cut edges due to more conservative p-values. This hypothesis is further corroborated by the fact that node degrees are highly correlated: the degree increase or decrease from one instance to another is very linear, as can be seen in Figure 5.8.

From this heatmap, we see that indeed, all correlations are larger than 0.93. I.e.,

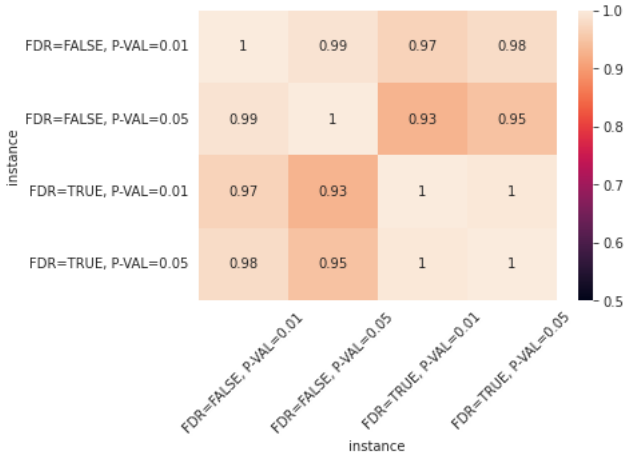


Figure 5.8: Gene degree correlations across graph instances.

Table 5.3: Converge information for all instances G1, G2, G3, and G4, comparing the two different cover algorithms.

FDR	p-val.	algo.	ll	nodes / clique	n_cliques	s / clique	h / it
yes	0.05	COVER	1.412e+07	2.587165	31165.0	1.922	16.646
yes	0.05	PART	1.414e+07	2.063467	39359.0	1.917	20.964
no	0.05	COVER	1.476e+07	3.323004	79841.0	1.132	25.114
no	0.05	PART	1.478e+07	2.182667	107124.0	1.153	34.327
yes	0.01	COVER	1.396e+07	2.521616	25028.0	2.271	15.794
yes	0.01	PART	1.395e+07	2.049594	31778.0	2.270	20.040
no	0.01	COVER	1.446e+07	2.971808	58562.0	2.179	35.449
no	0.01	PART	1.450e+07	2.128923	76255.0	2.267	48.032

there exists some α and β, such that increasing the p-value from 0.01 to 0.05 causes the degree $d_G(u)$ of node u to increase to $d_{G'}(u) \approx \alpha + \beta d_G(u)$, for any u, with very low error (an R^2 of at least 0.86).

Overall, it does indeed appear that in all tested instances, our results we approximately scale-free, therefore we are, at least initially, consistent with our interpretation of the Omnigenic hypothesis.

5.4 Convergence of the MLE

We ran our computational experiments on a computer having an Intel(R) Xeon(R) E-2146G CPU 3.50GHz processor, which contains 6 physical cores, split into 12 virtual cores and 62GiB RAM memory. Overall, the execution of the MLE algorithm is the most costly step of the whole pipeline, with the execution having taken up to a week to ensure full convergence (when all instances were running in parallel).

Naturally, as we saw previously, the instances without FDR control are more dense, have more parameters to control, and thus are able to provide a more exact fit

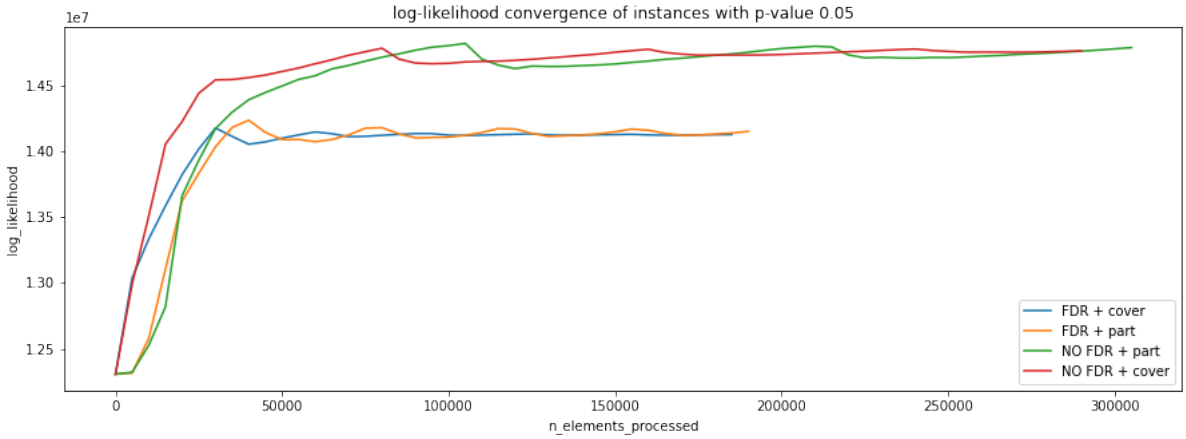


Figure 5.9: Convergence plot of the 0.05 instances, comparing the two different cover algorithms and the use of FDR control.

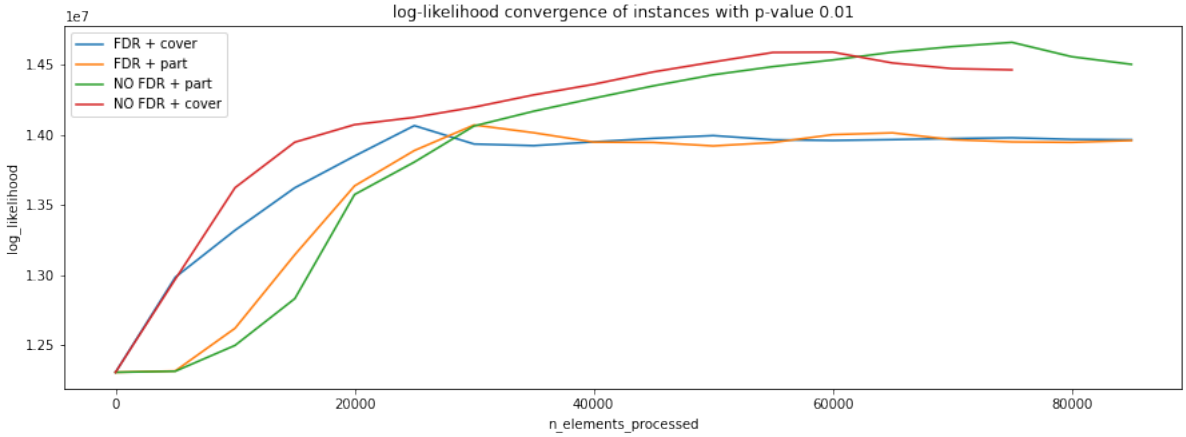


Figure 5.10: Convergence plot of the 0.01 instances, comparing the two different cover algorithms and the use of FDR control.

in the training data. This is evidenced in Table 5.3, where the instances without FDR achieve a higher log-likelihood (ll) than their counterparts. As we've extensively commented before, however, the pure log-likelihood value is not a direct indication of model performance, and we still need to apply the information criteria to obtain a, hopefully, unbiased view of performance. The log-likelihood is however useful for checking the convergence of the algorithm. Additionally, we can see the average execution time per clique is independent of average clique size, therefore, we conclude that cover algorithms that provide larger average cliques could provide an advantage over other cover algorithms.

5.4.1 Comparison of the Edge Cover Algorithms

Furthermore, when comparing the log-likelihoods using the partition algorithm versus the cover algorithm, we see that they are able to achieve approximately the same

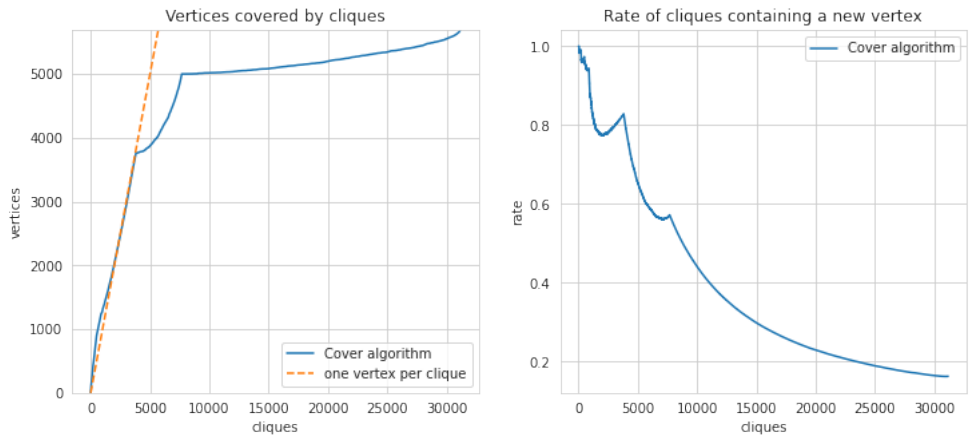


Figure 5.11: Plots showing how close the clique edge cover is to a detachable clique sequence for the cover algorithm

level of log-likelihood, but the cover algorithm reaches convergence faster in both instances, as seen in Figures 5.9 and 5.10. The main factor that causes this, is that the algorithm was able to cover all edges with a smaller number of cliques, and the processing time for each clique is appears to be independent of clique size. There is also a gain in the fact that edges can be in more that one clique, so they are optimized potentially multiple times per sweep. However, we hypothesize that the most gain is in the fact that we used a greedy approach to selecting cliques, instead of just iteratively finding arbitrary maximal cliques.

A way that we might explain why the cover algorithm provides a faster convergence is by evaluating the quality of the clique sequence in terms of how close it approaches an ideal sequence that would be constructible in a chordal graph. As mentioned in the previous chapter, if the graph is chordal, one can find a sequence of cliques with which the algorithm converges with only one sweep. This sequence $C_1, C_2, \ldots, C_l$ is such that

$$V(C_i) - \bigcup_{j=1}^{i-1} V(C_j) \neq \varnothing,$$

i.e., the next clique of the sequence always introduces a new vertex that had not been seen previously. We can see in Figure 5.11 that the cliques found by the cover algorithm approximately follow this trend, of introducing one new vertex per clique, until we reach about 4000 vertices, 70% of the total.

The partition algorithm, in turn, only follows this trend until less than 3000 vertices, which represents about 50% of the vertex total (Figure 5.12). A percentage significantly lower than the cover algorithm.

It is curious that these graphs obtained by conditional independence testing are still somewhat triangular, in the sense that we are able to find a clique sequence that approximates a sequence of simplicial vertices. If the reader recalls, triangularity

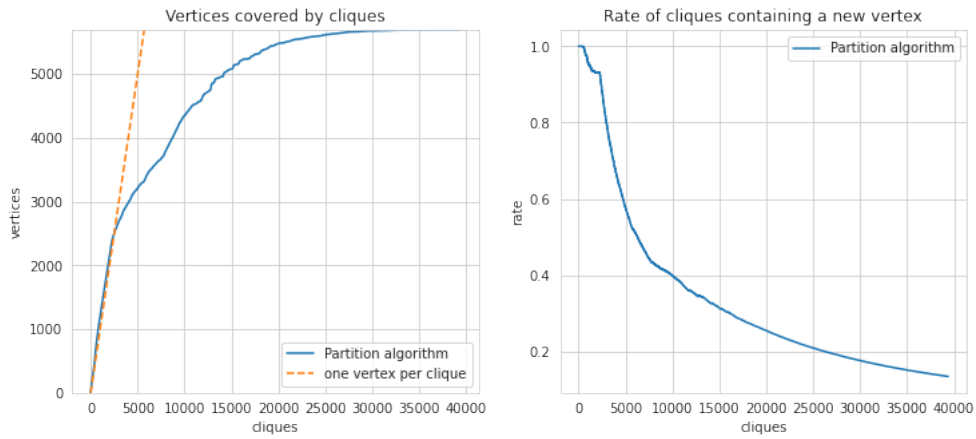


Figure 5.12: Plots showing how close the clique edge cover is to a detachable clique sequence for the partition algorithm

Table 5.4: Table containing the information criteria results for all instances.

FDR	p-val	algo	AIC	BIC	$\text{EBIC}_{0.01}$	$\text{EBIC}_{0.1}$	$\text{EBIC}_{0.5}$	$\text{EBIC}_{1.5}$
-	-	-	-24602752	-24572835	-24571815	-24562634	-24521828	-24419815
no	1	-	280172	85483538	85483538	85483538	85483538	85483538
yes	0.05	PART	-28202615	-27950417	-27943865	-27884900	-27622833	**-26967667**
no	0.05	PART	-29314688	-28521386	-28504249	**-28350019**	-27664553	-25950887
yes	0.05	COVER	-28152501	-27900302	-27893750	-27834785	-27572719	-26917552
no	0.05	COVER	**-29284929**	**-28491626**	**-28474490**	-28320260	-27634793	-25921128
yes	0.01	PART	-27891323	-27688854	-27683425	-27634561	-27417387	-26874453
no	0.01	PART	-28765995	-28240921	-28228750	-28119212	-27632378	-26415293
yes	0.01	COVER	-27853123	-27650655	-27645226	-27596361	-27379188	-26836254
no	0.01	COVER	-28861500	-28336426	-28324255	-28214718	**-27727883**	-26510798

was also noted to be the case for analyses using Gene Co-expression networks based on correlation. However, we believe it arose as more of a mathematical consequence of the method, rather than necessarily an innate property of the studied biological systems. It is unclear if, in this scenario of conditional independence testing, this triangularity is purely a biological phenomenon or it was exacerbated by some step of our processing pipeline, such as the imputation method or even the conditional independence testing method itself. This question warrants further investigation, but we assume that, even if still present, a bias towards more triangular graphs is much weaker than in correlation-based methods.

5.5 Information Criteria Results

After we fit all our models, we assess their (hopefully) unbiased performance through Information Criteria. Before finding which instance has the best values, we want to know if they surpass the baseline model (as one might recall, the baseline model is defined by having all variables be conditionally independent). Indeed, as we can see in Table 5.4, all of them do across all criteria.

Additionally, we also add the instance given by the complete graph, equivalent to using the p-value threshold of edges equals 1. This instance also performed significantly worse than our four instances, indicating that the Information Criteria do indeed penalise overly complex. models. Nonetheless, ee see that the instance that best performed on the AIC and BIC was the densest, as can be seen in Table 5.4, with the p-value threshold of 0.05 and without FDR control. As we increase the γ parameter of the EBIC, we see that sparser instances obtain better results. We conclude from this that, in this setting, controlling for FDR does likely result in less Type 1 errors, but it is conservative to a point where it may hinder model fit, depending on what criterion is considered to be the most appropriate, but we know that the EBIC was constructed to function in the low-data setting. Furthermore, given that edge degrees are very highly correlated between the different instances, structural conclusions about the network are unlikely to change much. So, on the perspective of overall model quality, using FDR control appears superior.

5.6 Essential Genes

To further corroborate the plausibility of the model, we went into the analysis of essential genes: genes whose mutations/deletions are associated with (in)viability of the organism, in our case, *Saccharomyces cerevisiae*. We went into the Yeast-Mine platform, populated by the Saccharomyces Genome Database, and searched for genes associated with the inviability phenotype and found 2215 genes which can be considered essential. Of those 2215, 1230 were found in our data set. As is reported in the literature, essential genes should display some measure of centrality in the GRN, therefore, our method should be able to display such a property. Indeed, if we examine some centrality metrics computed on one of our instances, we see the the essential genes are more likely to have a higher value that a non-essential gene.

As for centrality metrics, we weighted each edge ij by the absolute value of the partial correlation $|\rho_{i,j|Z}|$ between i and i, given all other variables $Z = V(G) \setminus \{i, j\}$, where

$$\rho_{i,j|Z} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}. \tag{5.1}$$

The partial correlation measures the degree of association between i and j, when the effect of the controlling variables Z is removed. In the case of the multivariate Gaussian, when can calculate all partial correlations directly from the precision matrix. From these weighted edges, we calculated the weighted degree of each node. We also calculated the betweenness of each node, but instead of directly using $|\rho_{i,j|Z}|$ as the edge weight, we used $1-|\rho_{i,j|Z}|$ instead, as we wanted to have shorter distances between genes that had a high conditional dependence between each other.

Table 5.5: Centrality metrics of both essential and non-essential genes.

	G2	G3	G0	G1
Edge p-value	0.05	0.01	0.05	0.01
FDR Control	Yes	Yes	No	No
degree KS stat.	0.1554	0.1592	0.1590	**0.1596**
degree KS p-val.	8.667e-21	8.160e-22	9.386e-22	6.505e-22
between. KS stat.	0.1122	0.0964	0.0994	**0.1292**
between. KS p-val.	5.020e-11	2.960e-08	9.491e-09	1.721e-14
info_score KS stat.	**0.2227**	0.2073	0.2199	0.2219
info_score KS p-val.	1.968e-42	8.955e-37	2.281e-41	3.923e-42

The final centrality metric we computed was our specific development geared towards a GGM. We defined a metric that calculates the expected gain in information we obtain by discovering the value of a gene. Genes that are conditionally dependent on many other genes should significantly reduce the amount of uncertainty in the distribution. The new metric is

$$\text{info\_score}(i) = 2\,\mathbb{E}_y[H(\boldsymbol{X}) - H(\boldsymbol{X} \mid X_i = y)], \tag{5.2}$$

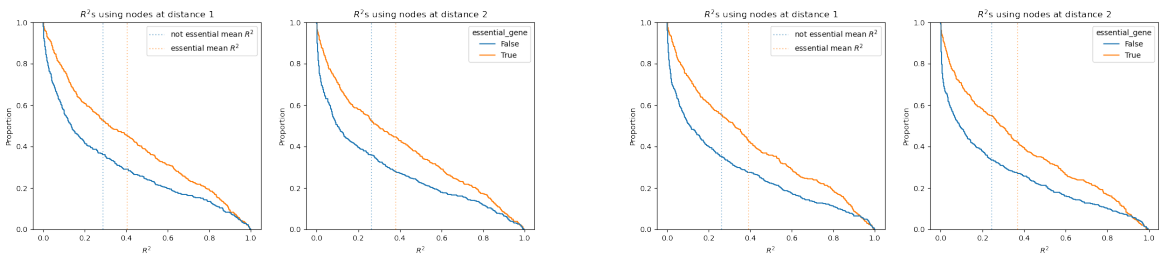
where we are taking expected value of the difference in entropy between the full distribution $\boldsymbol{X} \sim \mathcal{N}(0, \Omega)$, and the distribution conditional on variable X_i, across all possible values y of X_i, according to its marginal distribution given by $X_i \sim \mathcal{N}(0, \omega_{ii}^{-1})$. In the case of the Gaussian multivariate distribution, Equation 5.2 simplifies to

$$\text{info\_score}(i) = \log|\Sigma_{-i}| - \log|\Sigma| + \log \sigma_{ii} + C, \tag{5.3}$$

where we can disregard the constant C. This is a quite simple and computationally efficient formula, where $\log|\Sigma|$ only needs to be calculated once, and then, for every node i we would need to calculate $\log|\Sigma_{-i}|$, where Σ_{-i} represents the matrix Σ with the i-th row and column removed. Although the most popular determinant algorithms have $O(n^3)$ complexity, their numerical implementations are quite efficient in practice. The derivation of the formula can be seen in the appendix Section A.2. After computing the metrics for each node for all instances, we obtain Table 5.5:

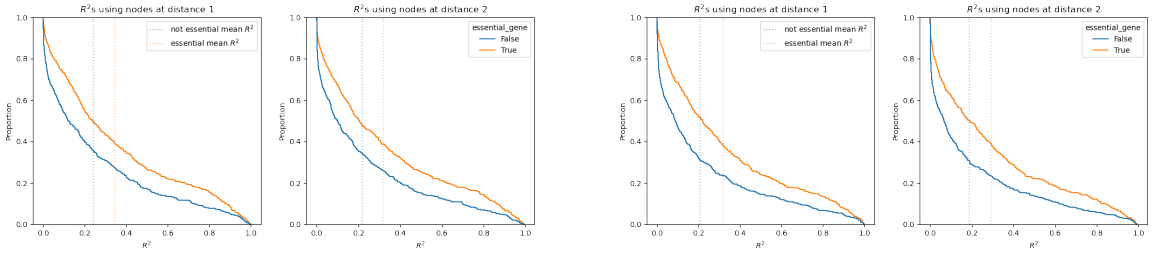
Overall, if we look at all quantiles displayed in Table 5.5, we can see that the essential genes always exceed the non-essential genes, furthermore the metric that we developed is the one that yields the largest distribution difference. Indeed, if we look at all our instances, every one of them has a significant difference in distribution for all centrality metrics.

We illustrate this claim by looking at the non-parametric Kolmogorov-Smirnov



(a) G0 (b) G1

Figure 5.13: Gene expression predictions R^2, for Instances without FDR control



(a) G2 (b) G3

Figure 5.14: Gene expression predictions R^2, for Instances with FDR control

(KS) statistic for detecting distribution differences. And we see that two instances displayed the more significant differences between gene and non-essential genes across the three metrics: instance G2, with p-value 0.05 and FDR control, and instance G1, with p-value 0.01 but without FDR control. Curiously, both these instances are sparse, but not the most sparse instance we tested. This result seemingly goes in a different direction compared to the Information criteria results, where the densest instance had the better performance. This divergence between the results above and the information criteria is likely due to their difference in purpose: information criteria attempt to evaluate a model from a predictive power standpoint, irrespective of whether the model's inner structure maintains biological plausibility. In this scenario, they ended up pointing towards denser solutions, and consequently, solutions with more Type 1 errors, which appear to have somewhat obfuscated structural properties of the network. We evaluate this phenomenon more as a limitation of using Information Criteria than an issue with GGM modelling in of itself. Ultimately, the information criteria showed that all models perform above baseline, and the essential gene analysis indicated that all models also display biologically plausible structures.

5.6.1 Influence Propagation

So far, we have be able to indicate that the graph structure of the GRN is likely consistent with a "small-world" regime. However, we have not yet given direct indication that core genes of phenotypes are more likely to be "hub" nodes of the graph, given that KS statistics between essential and not essential genes are not that large, only going up to 22, as seen in Table 5.5. So we turn to a more subtle structural question about the GGM. For a certain gene i, can expression levels of genes j at distance $d(i,j) = k$ be used to predict i?. If we take the precision matrix of the GGM, we can derive the coefficients of a linear regression to any gene i, and by doing so, we are able to calculate an R^2, defined by:

$$R^2 = 1 - \frac{Var(X_i - X_i^{\text{pred}})}{Var(X_i)}$$

We can see by Figure **??** that for all instances, essential genes had, on average, a much higher R^2 for both distances 1 and 2, and this difference is much more marked than the centrality metrics previously shown. This metric of prediction power of genes at distance k (especially distance 2) is much more indicative that peripheral genes tend to exert strong influence on core genes.

Chapter 6

Conclusion

Finally, given the information criteria results, specially in comparison to our baseline model, we can assert that all instances we tested did have an adequate level of fit. If we assume that this level of fit is enough to also make assertions on the structure of the underlying GRN, we cay say that

1. The GRN may indeed have scale-free properties, given that all instances we tested had scale-free properties, and

2. the essential (core) genes did indeed have a higher degree of centrality, across all centrality metrics we tested, than the average gene.

As we originally discussed in the introduction of both these facts, therefore, are supportive evidence for the Omnigenic hypothesis.

We also observed that the second algorithm for finding a clique edge cover worked better in terms of convergence speed compared to the first. We did not, however, spend time porting the Python code into an efficient C++ implementation, nor did we attempt theoretical improvements on it, given our insights into its possible connection to chordal graphs and simplicial vertices. Also on the topic of triangularity, we have not investigated, mathematically or empirically, whether it is an inherent property of biological networks, or if we are inducing it through some of the methods we applied in the pipeline. All these questions merit to be explored in further work.

Ultimately, we were able to build a full pipeline, from data processing to modelling, that yielded models with positive performance, evaluated under different Information Criteria. Furthermore, the models themselves appeared to have properties consistent with prevailing biological assumptions about genes, such as the fact that essential genes tend to have many regulatory connections in the GRN. Therefore, this approach shows promise for further development and tests on other domains in the high-dimensional/small data set regime. We recognize that many aspects of the pipeline were not fully explored, such as alternative imputation models, alternative

shrinkage methods, or more kinds of Conditional Independence tests, but the sheer scope of this project was quite overwhelming to this humble student. Nonetheless, we assess that the overall approach, combining its many components, is adequately validated in the transcriptomic data setting and there is ample opportunity to improve and generalize this software package.

Appendix A

Proofs

A.1 Chapter 3

A.1.1 Multivariate Gaussian MLE

The formula for the Multivariate Gaussian's log-likelihood is

$$l(\boldsymbol{X}_{[1:n]};\Omega) = \frac{np}{2}\log 2\pi + \frac{n}{2}\log|\Omega| - \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{X}_i^T\Omega\boldsymbol{X}_i$$

And given that $\boldsymbol{X}_i^T\Omega\boldsymbol{X}_i = \mathrm{tr}(\Omega\boldsymbol{X}_i\boldsymbol{X}_i^T)$, and by the linearity of the trace operator, we have

$$\sum_{i=1}^{n}\boldsymbol{X}_i^T\Omega\boldsymbol{X}_i = \sum_{i=1}^{n}\mathrm{tr}(\Omega\boldsymbol{X}_i\boldsymbol{X}_i^T) = \mathrm{tr}(\Omega\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i^T)$$

Therefore,

$$l(\boldsymbol{X}_{[1:n]};\Omega) \propto n\log|K| - \mathrm{tr}(\Omega\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i^T)$$

Finally, we can write the maximum Likelihood problem as

$$\max_{\geq 0}\log|\Omega| - \mathrm{tr}(S\Omega)$$

where $S = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i^T$, is the empirical covariance matrix.

A.1.2 Conditional Independence

Proof of Proposition 1:

Proof. From the conditional probability formula, we know that $P(S \mid T) \propto P(S \cap T)$. Additionally, we can decompose $x^T\Omega x$ into

$$x^T \Omega x = \omega_{i,j} x_i x_j + x_i \Omega_{i,-j} x_{-j} + x_j \Omega_{j,-i} x_{-i} + x_{-ij}^T \Omega_{-ij,-ij} x_{-ij}$$

Therefore, the conditional distribution of (X_i, X_j), given all other variables $B = \{X_k \mid k \neq i, j\}$, would be proportional to

$$f_{|B}(x_i, x_j) \propto \exp\{\frac{1}{2}(\omega_{i,j} x_i x_j + x_i K_{i,-j} x_{-j} + x_j K_{j,-i} x_{-i} + x_{-ij}^T K_{-ij,-ij} x_{-ij})\}$$

Which can be factored as

$$f_{|B}(x_i, x_j) \propto \exp\{\frac{1}{2}(k_{i,j} x_i x_j\} \exp\{\frac{1}{2} x_i \Omega_{i,-j} x_{-j}\} \exp\{\frac{1}{2} x_j \Omega_{j,-i} x_{-i}\} \exp\{x_{-ij}^T \Omega_{-ij,-ij} x_{-ij}\}$$

Considering that all variables other than x_i and x_j are fixed, terms not involving them x_i or x_j can be considered constants. So we can simplify the equation above to

$$f_{|B}(x_i, x_j) \propto \exp\{\frac{1}{2}(\omega_{i,j} x_i x_j\} \exp\{\frac{1}{2} \omega_{i,i} x_i^2\} \exp\{\frac{1}{2} \omega_{j,j} x_j^2\}$$

I.e., the product of two marginal probabilities and a dependent component involving $\omega_i j$, and if $\omega_{ij} = 0$, the component would equal 1, and the pdf could be seen as the product of the marginals of x_i and x_j. □

Proof of Proposition 2:

Proof. Considering A as a vertex subset and B as its complement, from the Schur Complement properties, we know that

$$|\Omega| = |\Omega_{B,B}||\Omega_{A,A} - \Omega_{A,B}(\Omega_{B,B})^{-1}\Omega_{B,A}|$$
$$\log|\Omega| = \log|\Omega_{B,B}| + \log|\Omega_{A,A} - \Omega_{A,B}(\Omega_{B,B})^{-1}\Omega_{B,A}|$$

Therefore, the optimization problem becomes

$$\begin{aligned} \max_{\Omega} \quad & \log|\Omega^0_{B,B}| + \log|\Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}| - \mathrm{tr}(S\Omega) \\ \text{s.t} \quad & \omega_{ij} = \omega^0_{ij} && \forall ij \notin M \\ & \omega_{ij} = 0 && \forall ij \notin E \end{aligned}$$

Now we can remove constant terms due to $\Omega_{B,B}$, $\Omega_{A,B}$, and $\Omega_{B,A}$ being fixed. And thus, obtain

$$
\begin{aligned}
\max_{\Omega} \quad & \log|\Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}| - \mathrm{tr}(S_{A,A}\Omega_{A,A}) \\
\text{s.t} \quad & \omega_{ij} = \omega^0_{ij} && \forall ij \notin M \\
& \omega_{ij} = 0 && \forall ij \notin E
\end{aligned}
$$

Let us conveniently add the term $\mathrm{tr}(S_{A,A}\Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A})$ to the objective function, without changing the optima, since it is a constant. Now we can use the linearity of the trace operator to group the terms $\mathrm{tr}(S_{A,A}\Omega_{A,A})$ – $\mathrm{tr}(S_{A,A}\Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A})$ into $\mathrm{tr}(S_{A,A}(\Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}))$.

Now, finally, by creating the auxiliary matrix Ω' we can rewrite the problem as

$$
\begin{aligned}
\max_{\Omega' \succeq 0} \quad & \log|\Omega'| - tr(S_{A,A}\Omega') \\
\text{s.t} \quad & \omega_{ij} = 0 && \forall ij \notin E \\
& \Omega' = \Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A},
\end{aligned}
$$

where the positive definiteness of Ω' guarantees that the resulting Ω^1 solution will also be positive definite, given that Ω^0 is also assumed to be positive definite.

□

Proof of Corollary 1:

Proof. If M is a clique, then the restriction $\omega_{ij} = 0$ can be dropped, given that all non-edges are outside of M, and therefore, ω_{ij} is already naturally set to ω^0_{ij}, and thus equal 0, since we assume Ω^0 to be a viable solution. If it is dropped, then the problem becomes

$$
\begin{aligned}
\max_{\Omega' \succeq 0} \quad & \log|\Omega'| - tr(S_{A,A}\Omega') \\
\text{s.t} \quad & \Omega' = \Omega_{A,A} - \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A},
\end{aligned}
$$

a simple problem of the Gaussian MLE for sample covariance matrix $\Sigma_{A,A}$ with no structure restriction. Therefore, the optimum is given by

$$
\begin{aligned}
\Omega' &= (S_{A,A})^{-1} \\
\Omega^1_{A,A} &= (S_{A,A})^{-1} + \Omega^0_{A,B}(\Omega^0_{B,B})^{-1}\Omega^0_{B,A}.
\end{aligned}
$$

□

A.1.3 Equivalence of using Correlation

With better numerical stability in mind, our MLE algorithms were all executed on the data set's empirical correlation matrix, instead of the empirical covariance matrix. This does not alter the results, provided that obtained precision matrix is transformed back the original scale. To see why this is true: let's use $P = D\Omega D$, where D is the diagonal matrix

$$D = \begin{bmatrix} (s_{11})^{-1/2} & & \\ & \ddots & \\ & & (s_{kk})^{-1/2} \end{bmatrix},$$

where s_{ii} are the diagonal elements of the empirical covariance matrix S. If we replace Ω in terms of P and D, and optimize on P instead, we obtain the equivalent optimization problem

$$\begin{aligned} &\max_P \log|D^{-1}PD^{-1}| - \operatorname{tr}(SD^{-1}PD^{-1}) \\ &\text{s.t } p_{ij} = 0 \ \ \forall ij \notin E. \end{aligned}$$

From this equivalent formulation, we can calculate the solution Ω^* of the original Structured MLE problem by $\Omega^* = D^{-1}P^*D^{-1}$, where P^* is the solution to the reformulated problem.

Let us recall two matrix properties:

$$|AB| = |A||B| \tag{A.1}$$

$$\operatorname{tr}(ABC) = \operatorname{tr}(BCA) = \operatorname{tr}(CAB) \tag{A.2}$$

Applying both to the objective function of the reformulated problem:

$$\begin{aligned} \log|D^{-1}PD^{-1}| - \operatorname{tr}(SD^{-1}PD^{-1}) &= 2\log|D^{-1}| + \log|P| - \operatorname{tr}(D^{-1}SD^{-1}P) \\ &= 2\log|D^{-1}| + \log|P| - \operatorname{tr}(CP), \end{aligned}$$

where C is the empirical correlation matrix. Therefore, the optimization problem simplifies to

$$\begin{aligned} &\max_P \log|P| - \operatorname{tr}(CP) \\ &\text{s.t } p_{ij} = 0 \ \ \forall ij \notin E, \end{aligned}$$

Which is solving the MLE for the empirical correlation matrix, but maintaining the same structure restriction.

A.2 Chapter 5

A.2.1 Derivation of the Info Score

From our definition of the information score, we can apply the differential entropy formula for the multivariate gaussian, thus obtaining

$$\begin{aligned}\text{info\_score}(i) &= 2\,\mathbb{E}_y[H(\boldsymbol{X}) - H(\boldsymbol{X} \mid X_i = y)] \\ &= 2H(\boldsymbol{X}) - 2\mathbb{E}_y[H(\boldsymbol{X} \mid X_i = y)] \\ &= k + k\log(2\pi) + \log|\Sigma| - 2\,\mathbb{E}_y[H(\boldsymbol{X} \mid X_i = y)].\end{aligned}$$

For the second term, $\mathbb{E}[H(\boldsymbol{X} \mid i)]$, we know that the covariance matrix of the conditional distribution given X_i is the Schur complement

$$\Sigma' = \Sigma_{-i} - \Sigma_{i,-i}\sigma_{ii}\Sigma_{-i,i},$$

and as we can see, it does not depend on specific values of y, so the expected value operator can be ignored. From properties of the Schur complement, we also know that $|\Sigma'| = |\Sigma|/|\sigma_{ii}|$. Therefore

$$2\,\mathbb{E}_y[H(\boldsymbol{X} \mid X_i = y)] = k - 1 + (k-1)\log(2\pi) + \log|\Sigma_{-i}| - |\sigma_{ii}|$$

Putting it all together, we obtain

$$\text{info\_score}(i) = \log|\Sigma| - \log|\Sigma_{-i}| + |\sigma_{ii}| + C.$$

References

[1] El ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology*, 31(6):545–552, May 2013.

[2] Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: the worldviews of AIC and BIC. *Ecology*, 95(3):631–636, March 2014.

[3] Joshua Batson, Loïc Royer, and James Webber. Molecular cross-validation for single-cell RNA-seq. September 2019.

[4] Evan A. Boyle, Yang I. Li, and Jonathan K. Pritchard. An expanded view of complex traits: from polygenic to omnigenic. *Cell*, 169:1177–1186, June 2017.

[5] Scott L. Carter, Christian M. Brechbühler, Michael Griffin, and Andrew T. Bond. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20:2242–2250, September 2004.

[6] Gerda Claeskens and Nils Lid Hjort. *Model Selection and Model Averaging*. Cambridge University Press, January 2001.

[7] John C. Doyle, David L. Alderson, and Lun Li. The "robust yet fragile" nature of the internet. *PNAS*, 102:14497–14502, October 2005.

[8] Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, December 2007.

[10] Cho-Jui Hsieh, Matyas A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[11] Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, et al. Quantitative single-cell rna-seq with unique molecular identifiers. *Nature Methods*, pages 163–166, February 2014.

[12] Christopher A Jackson, Dayanne M Castro, Giuseppe-Antonio Saldi, Richard Bonneau, and David Gresham. Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments. *eLife*, 9, January 2020.

[13] Jana Janková and Sara van de Geer. Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9(1), January 2015.

[14] Jana Janková and Sara van de Geer. Honest confidence regions and optimality in high-dimensional precision matrix estimation. *TEST*, 26(1):143–162, sep 2016.

[15] Itamar Kanter and Tomer Kalisky. Single cell transcriptomics: methods and applications. *Frontiers in Oncology*, 5:14497–14502, March 2015.

[16] Raya Khanin and Ernst Wit. How scale-free are biological networks. *Journal of Computational Biology*, 13:810–818, May 2006.

[17] Solt Kovács, Tobias Ruckstuhl, Helena Obrist, and Peter Bühlmann. Graphical elastic net and target matrices: Fast algorithms and software for sparse precision matrix estimation, 2021.

[18] Manik Kuchroo, Jessie Huang, Patrick Wong, Jean-Christophe Grenier, Dennis Shung, Alexander Tong, et al. Multiscale PHATE identifies multimodal signatures of COVID-19. *Nature Biotechnology*, 40(5):681–691, February 2022.

[19] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, February 2004.

[20] TI Lee. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298:799–804, October 2002.

[21] Weidong Liu. Gaussian graphical model estimation with false discovery rate control. *The Annals of Statistics*, 41(6), dec 2013.

[22] Teri A. Manolio, Francis S. Collins, Nancy J. Cox, David B. Goldstein, et al. Finding the missing heritability of complex diseases. *Nature*, page 747–753, October 2009.

[23] Ronald M. Nelson, Mats E. Pettersson, and Örjan Carlborg. A century after fisher: time for a new paradigm in quantitative genetics. *Trends in Genetics*, 29:669–676, October 2013.

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, nov 2011.

[25] Shaun M. Purcell, Jennifer L. Moran, Menachem Fromer, et al. A polygenic burden of rare disruptive mutations in schizophrenia. *Nature*, page 185–190, January 2009.

[26] V. Srinivasa Rao, K. Srinivas, G. N. Sujini, and G. N. Sunand Kumar. Protein-protein interaction detection: Methods and analysis. *International Journal of Proteomics*, 2014:1–12, February 2014.

[27] Marsha L. Richmond. Women in the early history of genetics: William bateson and the newnham college mendelians, 1900-1910, March 2001.

[28] Max Roser, Cameron Appel, and Hannah Ritchie. Human height. *Our World in Data*, 2013. https://ourworldindata.org/human-height.

[29] Julian D. Schwab, Silke D. Kühlwein, Nensi Ikonomi, Michael Kühl, and Hans A. Kestler. Concepts in boolean network modeling: What do they all mean? *Computational and Structural Biotechnology Journal*, 18:571–582, 2020.

[30] T. P. Speed and H. T. Kiiveri. Gaussian markov distributions over finite graphs. *The Annals of Statistics*, 14(1), March 1986.

[31] Terrence Tao. When is correlation transitive? `https://terrytao.wordpress.com/2014/06/05/when-is-correlation-transitive/`. Accessed: 2023-08-06.

[32] Duc Tran, Hung Nguyen, Bang Tran, Carlo La Vecchia, Hung N. Luu, and Tin Nguyen. Fast and precise single-cell data analysis using a hierarchical autoencoder. *Nature Communications*, 12(1), February 2021.

[33] Caroline Uhler. Gaussian graphical models. In Marloes Maathuis, Mathias Drton, Steffen Lauritzen, and Martin Wainwright, editors, *Handbook of Graphical Models*, chapter 9, pages 219–236. CRC Press, 2018.

[34] David van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J. Carr, Cassandra Burdziak, Kevin R. Moon, Christine L. Chaffer, Diwakar Pattabiraman, Brian Bierie, Linas Mazutis, Guy Wolf, Smita Krishnaswamy, and Dana Pe'er. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.e27, July 2018.

[35] Hao Wang. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 7(4), December 2012.

[36] Ivan Rodrigo Wolf, Rafael Plana Simões, and Guilherme Targino Valente. Three topological features of regulatory networks control life-essential and specialized subsystems. *Scientific Reports*, 11, December 2021.

[37] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4, August 2005.

[38] Rong Zhang, Zhao Ren, and Wei Chen. SILGGM: An extensive r package for efficient statistical inference in large-scale gene networks. *PLOS Computational Biology*, 14(8):e1006369, August 2018.

[39] Haitao Zhao and Zhong-Hui Duan. Cancer genetic network inference using gaussian graphical models. *Bioinformatics and Biology Insights*, 13:117793221983940, January 2019.