Master's thesis
Master's Programme in Data Science

Decomposing Genome-Phenome Associations using Independent Component Analysis

Panagiotis Anastasakis

May 21, 2025

Supervisor(s): Professor Matti Pirinen

Examiner(s): Professor Matti Pirinen
Professor Antti Honkela

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Panagiotis Anastasakis

Työn nimi — Arbetets titel — Title

Decomposing Genome-Phenome Associations using Independent Component Analysis

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	May 21, 2025	75

Tiivistelmä — Referat — Abstract

Genome-wide association studies (GWASs) have revealed that complex traits are typically influenced by many genetic variants, while at the same time individual variants often affect multiple traits. Together, these overlapping associations can lead to convoluted genetic architectures that pose significant challenges for biological interpretation of GWAS findings. To address these challenges, statistical methods have been developed that utilize results from multiple GWASs to identify latent components mediating variant-trait associations, which can provide more meaningful biological insights and interpretations.

In this thesis, two such methods are explored: Decomposition of Genetic Associations (DeGAs), which uses Singular Value Decomposition to estimate latent components, and Genetic Unmixing by Independent Decomposition (GUIDE), which extends DeGAs with the additional use of Independent Component Analysis (ICA). Through extensive simulations across diverse scenarios, it is found that GUIDE has a greater capacity to recover true latent components from simulated genetic architectures.

Next, to improve the robustness of GUIDE, a novel extension of the method is proposed using the ICASSO framework, where results from multiple ICA runs are aggregated. This extension is shown to produce more reliable and consistent decompositions, while also providing a ranking for the components by their significance, which the original GUIDE method lacks.

In addition to the simulations, GUIDE and DeGAs are applied to a type 2 diabetes GWAS dataset. The components estimated by GUIDE are found to closely align with those identified in previous analyses of the same data, demonstrating its effectiveness in uncovering meaningful variant-trait association patterns. Furthermore, the components extracted by GUIDE are sparser and more independent than those of DeGAs, offering a more interpretable decomposition.

ACM Computing Classification System (CCS):
Applied computing → Life and medical sciences → Genetics
Applied computing → Life and medical sciences → Bioinformatics

Avainsanat — Nyckelord — Keywords

GUIDE, ICA, ICASSO, DeGAs, SVD, genetics

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki, Finland

Muita tietoja — Övriga uppgifter — Additional information

Acknowledgments

First, I want to express my sincere gratitude and thanks my supervisor, Professor Matti Pirinen. I am grateful for the guidance provided throughout my work, the insightful discussions and the detailed feedback given at every step of the way. I am also very thankful to have been a part of the Institute for Molecular Medicine Finland (FIMM), where I had the opportunity to work alongside many great people, whose support I truly appreciate. I also want to thank the authors of [29] for providing the data used in this work. Additionally, I gratefully acknowledge the financial support provided by the John S. Latsis Public Benefit Foundation and Bodossaki Foundation scholarships for my studies during the academic years 2023-2024 and 2024-2025. I also acknowledge the use of Large Language Models (ChatGPT) for text refining in this thesis.

I want to thank my friends, for all the fun memories throughout my studies. I also want to thank my parents, Kostas and Liana. Your support has been truly invaluable in my studies and beyond. Lastly, I want to thank my partner, Zoi. Thank you for standing by me, inspiring me, and bringing joy and strength into every step of this journey.

Contents

1 Introduction **1**

2 Background **3**

2.1 Singular Value Decomposition 3

2.2 Independent Component Analysis 4

2.2.1 Motivation 4

2.2.2 Definition of ICA 7

2.2.3 Maximization of Non-Gaussianity 8

2.2.4 Preprocessing for ICA 11

2.2.5 ICA Estimation 12

2.3 Validation of Independent Components 13

2.3.1 Unreliability of Individual ICA Runs 13

2.3.2 ICASSO 13

2.4 Genetics 16

2.5 Genome-Wide Association Studies 17

2.5.1 Overview of GWAS 17

2.5.2 Applications and Biological Insights from GWASs 18

2.6 Decomposition of Variant-Trait Associations 19

3 Methods & Data **21**

3.1 Decomposition of Genetic Associations 21

3.2 Genetic Unmixing by Independent Decomposition 22

3.2.1 GUIDE Overview 22

3.2.2 Extending GUIDE with ICASSO 24

3.3 Component Interpretation 25

3.4 Simulations 27

3.4.1 Block Structures 27

3.4.2 Polygenic Additive Structures 29

3.4.3 Matching the Components 32

3.5 Type 2 Diabetes Data 32

4 Simulation Results **35**
4.1 GUIDE's Estimate of the Number of Components 36
4.2 Simulations with Known Genetic Architectures 39
4.2.1 Block Simulation 39
4.2.2 Polygenic Additive Structure Simulation Results 39
4.3 Simulations Under Model Misspecification 44
4.4 Ranking GUIDE Components with CQI 48

5 Application to Type 2 Diabetes Data **51**
5.1 Motivation for ICASSO 51
5.2 GUIDE Clusters 52
5.3 Comparison of GUIDE & DeGAs Components 57

6 Discussion **63**

Bibliography **67**

Appendix A The FastICA Algorithm **71**

Appendix B Whitening for GUIDE **73**

Appendix C DeGAs T2D Clusters **75**

1. Introduction

Genetic variation among humans, together with environmental factors, accounts for much of the diversity in human traits and susceptibility to disease [22]. A widely used approach to investigate the genetic basis underlying complex traits is the Genome-Wide Association Study (GWAS). The aim of a GWAS is to identify genetic variants that occur more frequently in individuals with a specific trait. So far, over 5,700 GWASs have been conducted, resulting in the discovery of thousands of statistical associations between genetic variants and a wide range of traits [34]. Although individual variant-trait associations often have small effect sizes and explain only a limited proportion of trait variance, when considered in aggregate they can provide valuable insights into the biological mechanisms underlying complex diseases [31].

An important factor that complicates the interpretation of GWAS results is the highly polygenic nature of complex traits, where many variants affect a given trait, each contributing a small effect. Additionally, pleiotropy, where a single variant affects multiple traits, further complicates the identification of specific genetic pathways, requiring more elaborate statistical approaches for meaningful GWAS interpretations. One promising strategy for addressing polygenicity and pleiotropy is to utilize association results from multiple traits to identify latent components mediating these associations [23, 32, 33, 37]. These components could implicate subsets of variants and traits, potentially pointing to unique disease pathways or pathophysiological mechanisms.

This thesis explores two methodologies for estimating such latent components from GWAS summary statistics. The first is Decomposition of Genetic Associations (DeGAs) [32], which applies Singular Value Decomposition (SVD) on GWAS association statistics to obtain the components. The second is Genetic Unmixing by Independent Decomposition (GUIDE) [23], a recently developed extension of DeGAs that employs Independent Component Analysis (ICA). This work presents the details of both methods, with a particular emphasis on GUIDE, and evaluates their performance using both simulated datasets and real GWAS data.

The aims of the thesis are the following:

1. To implement the GUIDE and DeGAs methods.
2. To improve the robustness of GUIDE by addressing uncertainty in ICA through the integration of the ICASSO framework [10], which aggregates results from multiple ICA runs.
3. To conduct simulations under a wide range of conditions to evaluate the performance of the two methods.
4. To apply GUIDE and DeGAs to a type 2 diabetes (T2D) GWAS dataset from [29] and assess the results in terms of interpretability of the extracted components and their potential to reveal biologically meaningful patterns.

This thesis is organized as follows. Chapter 2 covers the theoretical foundations, specifically SVD, ICA and ICASSO, also discussing main concepts in genetics and GWASs. Chapter 3 presents DeGAs and GUIDE, where an extension of GUIDE is proposed to account for the uncertainty in ICA estimates. Additionally, strategies for simulating genetic architectures are discussed and an overview of the T2D data analyzed in this thesis is given. Chapter 4 contains the results of the simulations, providing a comprehensive evaluation of GUIDE and its comparison with DeGAs, while Chapter 5 shows the results from the applications on the T2D data. Finally, Chapter 6 concludes the thesis.

2. Background

In this Chapter, the main concepts and methods that have been used throughout the thesis are discussed. In Section 2.1, the Singular Value Decomposition is presented. Section 2.2 provides a detailed background on the principles of Independent Component Analysis (ICA), while Section 2.3 discusses ICASSO, a method to obtain more reliable results in ICA. Section 2.4 introduces basic concepts of genetics and Section 2.5 provides an overview of Genome-Wide Association Studies. Finally, Section 2.6 briefly discusses the decomposition of variant-trait associations, motivating the work conducted in this thesis.

2.1 Singular Value Decomposition

The Singular Value Decomposition (SVD) is one of the most important matrix factorization methods and a fundamental tool in numerical linear algebra [9]. It decomposes a rectangular matrix into a product of three simpler matrices: two orthogonal ones (rotations) and one diagonal matrix. For a real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, its SVD is given as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2.1}$$

where

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix whose columns $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are called left-singular vectors of $\mathbf{A}$,
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ is also an orthogonal matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are called right-singular vectors of $\mathbf{A}$ and
- $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a rectangular diagonal matrix with non-negative entries σ_i, $i = 1, ..., \min\{m, n\}$ that are called singular values of $\mathbf{A}$.

Although the SVD of a matrix is not unique, we can always choose a decomposition so that the singular values are ordered in decreasing order in the diagonal of $\mathbf{\Sigma}$. Through the SVD, the rank of $\mathbf{A}$ is given by the number of non-zero singular

values. Denoting $\text{rank}(\mathbf{A}) = r \leq \min\{m, n\}$, and assuming a decreasing ordering of the singular values in $\mathbf{\Sigma}$, we can rewrite the SVD as

$$\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \tag{2.2}$$

where $\mathbf{U}_r \in \mathbb{R}^{m \times r}$ and $\mathbf{V}_r \in \mathbb{R}^{n \times r}$ contain the first r columns of $\mathbf{U}$ and $\mathbf{V}$ respectively and $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$ is a diagonal matrix with positive entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. This formulation shows that $\mathbf{A}$ can be expressed as a sum of r rank-1 matrices $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$, also referred to as components of $\mathbf{A}$. By including the first $L < r$ components, we obtain a lower-rank approximation of $\mathbf{A}$, called the truncated SVD (tSVD):

$$\tilde{\mathbf{A}}_L = \mathbf{U}_L \mathbf{\Sigma}_L \mathbf{V}_L^T, \tag{2.3}$$

where now $\text{rank}(\tilde{\mathbf{A}}_L) = L$. The tSVD of $\mathbf{A}$ is guaranteed to be the best L-rank approximation in terms of the Frobenius norm (denoted as $||\cdot||_F$), which is defined as the square root of the sum of the squares of all the entries of $\mathbf{A}$. In other words, it holds that

$$||\mathbf{A} - \tilde{\mathbf{A}}_L||_F = \min_{\mathbf{B}}\{||\mathbf{A} - \mathbf{B}||_F \mid \text{rank}(\mathbf{B}) = L\}. \tag{2.4}$$

This property makes SVD one of the most common methods to approximate a matrix using a low-rank representation, retaining the maximum amount of information of the original data, as measured by the Frobenius norm.

2.2 Independent Component Analysis

Independent Component Analysis (ICA) is a computational technique used to estimate statistically independent sources from observed mixed signals without prior knowledge of the mixing process or the original sources. This problem, also known as Blind Source Separation (BSS), was discussed during the 1980s and various solutions had been proposed under different names, serving as predecessors of ICA. During the following years, the concept of ICA was more rigorously defined by Jutten and Hérault [19] in 1991 and later by Comon [7] in 1994. Since then, several methods have been developed to optimize the ICA model, leading to applications in fields such as signal processing, neuroscience (e.g., EEG and MEG analysis) and finance [15].

2.2.1 Motivation

Perhaps the most popular example of a source separation problem is the "cocktail party problem". Suppose that three people are talking simultaneously at a party and their

conversations are recorded by three microphones located in different places across the room. Given a time index t, we can denote the individual signals produced by the people as $s_1(t), s_2(t)$ and $s_3(t)$. The signals mix as they travel through the air, and the microphones perceive a combination of these signals, denoted as $x_1(t), x_2(t)$ and $x_3(t)$. Assuming that the mixing procedure is linear and constant across time, we can express the mixed signals as a linear combination of the independent sources:

$$\begin{aligned} x_1(t) &= \alpha_{11}s_1(t) + \alpha_{12}s_2(t) + \alpha_{13}s_3(t), \\ x_2(t) &= \alpha_{21}s_1(t) + \alpha_{22}s_2(t) + \alpha_{23}s_3(t), \\ x_3(t) &= \alpha_{31}s_1(t) + \alpha_{32}s_2(t) + \alpha_{33}s_3(t). \end{aligned} \tag{2.5}$$

Using a vectorized notation, we can write $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$, $\mathbf{s}(t) = [s_1(t), s_2(t), s_3(t)]^T$ and $\mathbf{A} = (\alpha_{ij})$, for $i, j = 1, 2, 3$. Then, the linear system 2.5 can be expressed as $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, with only $\mathbf{x}(t)$ known. The main idea behind ICA is that solving this system of linear equations is possible when $\mathbf{A}$ is unknown, under the assumption that the sources $s_1(t), s_2(t), s_3(t)$ are statistically independent for every instant of time [16].

To demonstrate how ICA can identify the original sources from their mixtures, let's consider a simple example. Suppose we have three independent components producing signals as

$$\begin{aligned} s_1(t) &= \sin(3t) \\ s_2(t) &= 5\cos(2t)\sin(0.5t) \\ s_3(t) &= \mathcal{L}(0, 1), \end{aligned} \tag{2.6}$$

for $t = [0, 10]$. Signal s_3 is random noise independent of time, sampled from the Laplace distribution. The Laplace probability density function is

$$f(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right), \tag{2.7}$$

where in this example we have set the mean $\mu = 0$ and scale $b = 1$. We mix the sources with the mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.5 & -2 & 3 \\ -1 & 1 & -2 \\ 3 & -1.5 & -2 \end{bmatrix}, \tag{2.8}$$

obtaining the observed data $\mathbf{x}$. Figure 2.1 shows how ICA can recover the true signals almost perfectly, only using as input their mixtures $\mathbf{x}(t)$. We notice that the ordering

of the components is not fixed, as well as that for some their signs are flipped and their variances differ from those of the original signals. As we discuss later, these ambiguities are due to ICA being able to identify the independent components up to a sign, a scaling and a permutation.

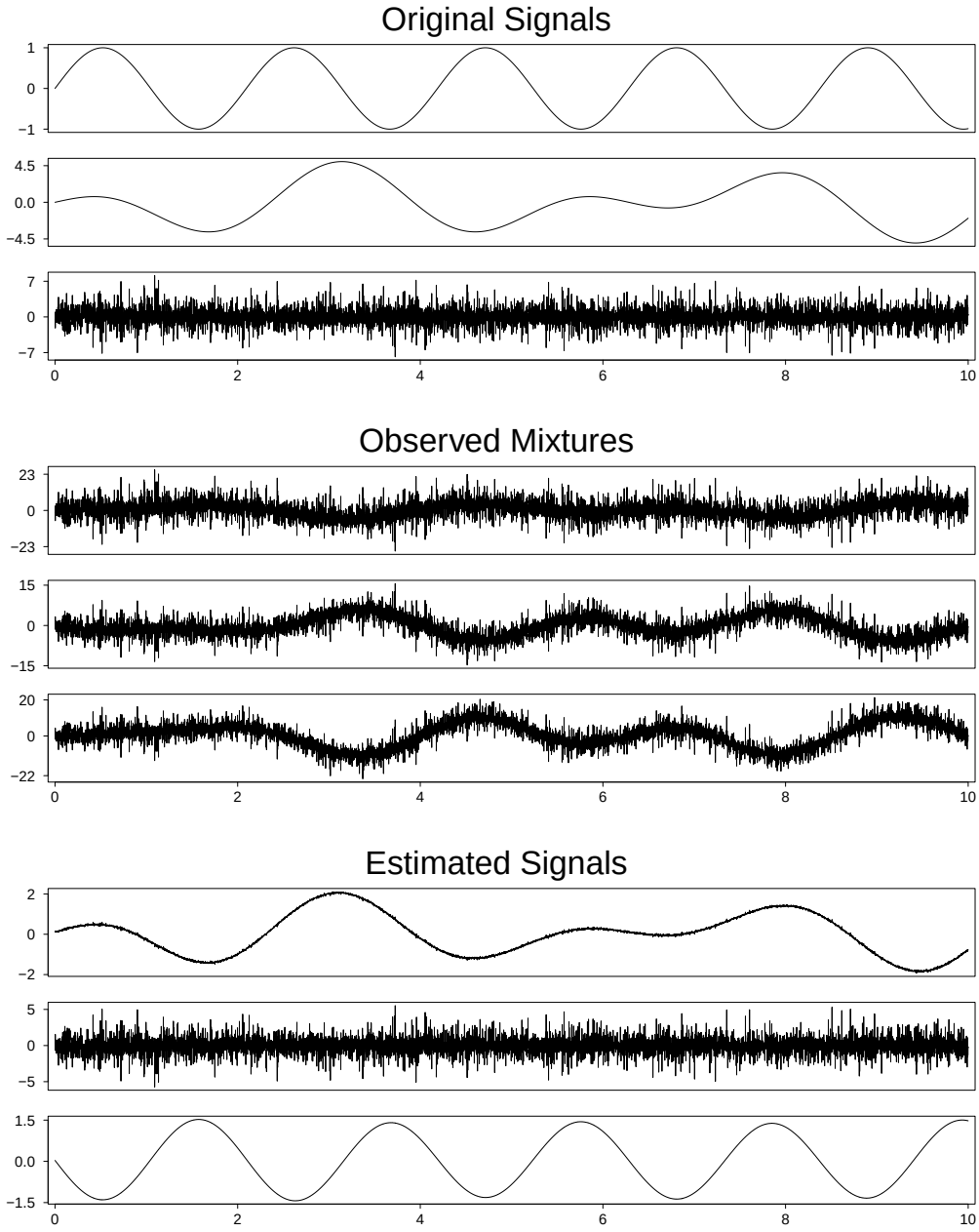

Figure 2.1: The top panel contains the original signals $\mathbf{s}(t) = [s_1(t), s_2(t), s_3(t)]^T$ and the middle one shows their mixtures $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$, occurring from the mixing matrix defined in 2.8. The bottom panel shows the estimated signals by ICA.

2.2.2 Definition of ICA

The rest of this section discusses ICA, following the general structure of [16]. To formally define ICA, we assume n linear mixtures $x_1, \ldots, x_n$, resulting from n independent components $s_1, \ldots, s_n$:

$$x_i = \alpha_{i1}s_1 + \alpha_{i2}s_2 + \ldots + \alpha_{in}s_n, \tag{2.9}$$

for $i = 1, \ldots, n$. Each mixture x_i and each independent component s_i are assumed to be random variables and not necessarily signals in time, hence the time index t is dropped. In practice, the observed values for each x_i would be a sample of this random variable, which is the input in ICA.

Using the same vectorized notation as previously with $\mathbf{x} = [x_1, \ldots, x_n]^T$, $\mathbf{s} = [s_1, \ldots, s_n]^T$ and $\mathbf{A} = (\alpha_{ij})$, for $i, j = 1 \ldots, n$, the above mixing model becomes

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{2.10}$$

From this formulation, it can be easily seen that there cannot be a unique ordering of the independent components, as they can be re-arranged in Equations 2.9 and the ICA model 2.10 will still be valid. Furthermore, the fact that the number of sources s_i equals the number of mixtures x_i means $\mathbf{A}$ is a square matrix. This simplifies the estimation problem as we can then compute its inverse $\mathbf{A}^{-1} = \mathbf{D}$ (the unmixing matrix), giving us the independent components as $\mathbf{s} = \mathbf{D}\mathbf{x}$. While useful, this condition is not necessary, as in some cases ICA can be extended to handle over-determined (more mixtures than sources) or under-determined (more sources than mixtures) linear systems [15].

In order for ICA to be able to estimate the original sources with no other information than the observed mixtures, the assumption that the components s_i are statistically independent is necessary. Independence means that the value of one component provides no information about the value of the others. In mathematical terms, the joint probability density function $f(s_1, \ldots, s_n)$ can be factorized into a product of individual densities:

$$f(s_1, \ldots, s_n) = \prod_{i=1}^{n} f_i(s_i). \tag{2.11}$$

Another important assumption made in ICA is that the components have non-Gaussian distributions. The reason is that when the components are Gaussian, it can be shown that the ICA model can only be estimated up to a rotation, hence it is not identifiable [15]. In fact, one approach for recovering independent components is to maximize their non-Gaussianity.

An intuitive justification as to why non-Gaussianity can lead to independent

components in ICA can be found in the Central Limit Theorem, which states that a mixture of independent random variables tends to be more Gaussian than the individual variables themselves. In other words, when independent sources are mixed, their sum (the observed mixtures) generally follows a more Gaussian distribution, even if the original sources are non-Gaussian. Thus, to reverse the mixing process and recover the independent components, ICA seeks to find a transformation that produces outputs as non-Gaussian as possible. Of course, if the independent components were Gaussian, this approach would fail as any linear combination of Gaussian variables is also Gaussian, making it impossible to distinguish the sources based on non-Gaussianity alone.

For the rest of this section, it is assumed that the independent components satisfy $E[s_i] = 0$ and $\text{Var}[s_i] = 1$ for all i. The zero-mean assumption holds when the observed mixtures are mean-centered, i.e. when $E[x_i] = 0$, which follows directly from 2.10. If this condition is not met, we can subtract the sample mean from each x_i to make the model zero-mean. The unit variance is due to the fact that the true variances of the independent components cannot be determined, since multiplying any of the sources s_i by a scalar in 2.10 can be canceled by diving the corresponding column $\mathbf{a}_i$ of $\mathbf{A}$ by the same scalar. To remove this ambiguity, the variance of each independent component is fixed to 1.

2.2.3 Maximization of Non-Gaussianity

Previously, we mentioned that one way to estimate the ICA model is by maximizing the non-Gaussianity of the recovered components to separate the sources. Various statistical measures of non-Gaussianity can be used for this purpose, each with its own advantages and computational properties.

Kurtosis

One of the simplest and most popular measures of non-Gaussianity is the kurtosis. Let s be a random variable such that $E[s] = 0$ and $\text{Var}[s] = 1$. The kurtosis of s is defined as

$$\kappa(s) = E[s^4] - 3. \tag{2.12}$$

Kurtosis measures the tailedness of a distribution by comparing its fourth moment to that of a standard Gaussian distribution, whose fourth moment is 3, hence has a kurtosis of zero. Random variables with positive kurtosis are called super-Gaussian or leptokurtic, while variables with negative kurtosis are called sub-Gaussian or platykur-

tic. Intuitively, super-Gaussian distributions tend to be more heavy-tailed with a sharper peak, while sub-Gaussian distributions usually have more flat densities (see Figure 2.2). It is worth noting that there are some non-Gaussian distributions with zero kurtosis, however they are considered to be uncommon [15].

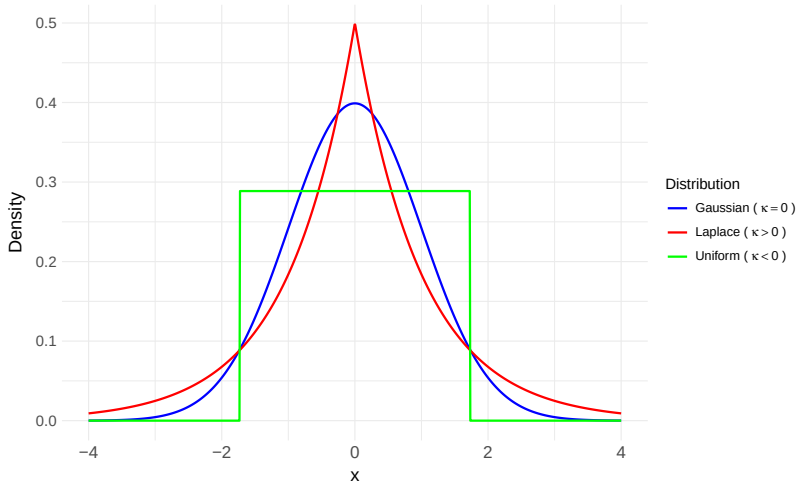

Figure 2.2: Comparison of probability density functions for distributions with different kurtosis values. The Laplace distribution has heavier tails and a sharper peak, while the Uniform distribution is more flat. The Gaussian distribution serves as a reference.

In practice, non-Gaussianity is measured by some non-negative transformation such as the absolute value or the square of kurtosis. This is a computationally efficient and simple way to measure how much a random variable departs from Gaussianity and is easy to use in ICA to find the independent components. However, there are some downsides that can make it unsuitable for some applications.

By serving as a measure of tailedness of a distribution, i.e. how likely it is for the distribution to produce outliers, kurtosis can be sensitive to extreme values when estimated numerically from a sample [12]. In fact, it has been shown that for many common distributions, the kurtosis is influenced mostly by the tails, while the region within one standard deviation of the mean contributes very little [36].

Another downside of using kurtosis for quantifying non-Gaussianity is its asymmetry when measuring the extremities of super-Gaussian and sub-Gaussian signals. The kurtosis of a super-Gaussian signal is not upper-bounded, as the fourth moment of a standardized random variable can be infinitely large. On the other hand, using the Jensen inequality we can show that kurtosis has a lower bound of -2 for standardized sub-Gaussian signals:

$$\mathrm{E}[s^4] \geq (\mathrm{E}[s^2])^2 = (\mathrm{Var}(s)+\mathrm{E}[s]^2)^2 = (1+0)^2 = 1 \implies \kappa(s) = \mathrm{E}[s^4]-3 \geq -2. \quad (2.13)$$

Due to this asymmetry, using kurtosis for applications where both super-Gaussian and sub-Gaussian independent components may exist is not appropriate [15]. Nonetheless, it can be still used for evaluating how sparse the independent components are, after we have estimated them.

Negentropy

Another important measure of non-Gaussianity of a random variable is negentropy, which is an information-theoretic way to quantify the distance of a random variable from Gaussianity. To formally define negentropy, we first define the differential entropy of a continuous random variable s with density $p(s)$:

$$H(s) = -\int p(s)\log(p(s))ds. \tag{2.14}$$

The differential entropy can be viewed as a generalization of standard entropy, originally defined for discrete variables. It measures the amount of uncertainty or unpredictability of s by quantifying how dispersed the distribution is over its possible values. Importantly, a Gaussian variable has the largest differential entropy of all random variables with the same variance [25], meaning that the Gaussian distribution is the most spread out and unpredictable. This result allows the use of differential entropy as a measure of non-Gaussianity, which is achieved through the negentropy:

$$J(s) = H(g_s) - H(s), \tag{2.15}$$

where g_s is a Gaussian random variable with the same mean and variance as s. Negentropy is always non-negative, becoming zero only when s is Gaussian, while larger values can be interpreted as implying greater non-Gaussianity. Hence, maximizing negentropy is equivalent to maximizing the non-Gaussianity of s, with respect to the negentropy measure.

Defining ICA through negentropy provides additional justification as to why in the context of source separation, non-Gaussianity implies independence of the components. Specifically, another way to estimate independent components in ICA is by minimizing their Mutual Information, which is a measure of how dependent random variables are. It can be shown that minimizing the Mutual Information is in fact equivalent to maximizing the negentropy of the estimated independent components (hence their non-Gaussianity), providing that they are constrained to be uncorrelated [15].

Negentropy provides a natural and robust way of measuring the distance of a random variable from Gaussianity. Computing negentropy, however, requires approximating the density $p(s)$ in 2.14, which is generally complicated and computationally

expensive. To address this issue and make it possible to use negentropy in ICA, Hyvärinen [13] derived an efficient approximation of negentropy that retains the desirable statistical properties for maximizing non-Gaussianity. Assuming s, u are random variables with zero mean and unit variance, where u is Gaussian, and G is a non-quadratic real function, we have that:

$$J(s) \propto \left(E[G(s)] - E[G(u)]\right)^2. \tag{2.16}$$

In theory, one can select any function G as long as it is non-quadratic, but a more informed choice can lead to better approximations. In the context of ICA, two popular options proposed in [16] that are widely used today are

$$G_1(x) = \frac{1}{\alpha}\log(\cosh(\alpha x)), \;\; 1 \leq \alpha \leq 2 \tag{2.17}$$

$$G_2(x) = -\exp\left(-\frac{x^2}{2}\right). \tag{2.18}$$

2.2.4 Preprocessing for ICA

Before applying ICA, it is advisable to preprocess the data to simplify the estimation process. One preprocessing step is to mean center the data to ensure that the independent components have zero mean. Another important step is to whiten the observed mixtures $\mathbf{x}$. Whitening [20] is a linear transformation applied on a random vector $\mathbf{x}$ to obtain a new vector

$$\mathbf{z} = \mathbf{M}\mathbf{x} \tag{2.19}$$

with unit diagonal covariance matrix:

$$\mathrm{Cov}(\mathbf{z}) = E[\mathbf{z}\mathbf{z}^T] = \mathbf{I}. \tag{2.20}$$

With this transformation, $\mathbf{z}$ is said to be white, meaning that its components are uncorrelated with unit variance. From 2.10 we have

$$\mathbf{z} = \mathbf{M}\mathbf{x} = \mathbf{M}\mathbf{A}\mathbf{s} = \hat{\mathbf{A}}\mathbf{s}, \tag{2.21}$$

hence the new mixing matrix is $\hat{\mathbf{A}} = \mathbf{M}\mathbf{A}$. Whitening the data is important for ICA as it constrains the mixing matrix to be orthogonal, significantly simplifying the estimation problem:

$$E[\mathbf{z}\mathbf{z}^T] = E[\hat{\mathbf{A}}\mathbf{s}(\hat{\mathbf{A}}\mathbf{s})^T] = \hat{\mathbf{A}}E[\mathbf{s}\mathbf{s}^T]\hat{\mathbf{A}}^T = \hat{\mathbf{A}}\hat{\mathbf{A}}^T, \tag{2.22}$$

which from 2.20 implies that $\hat{\mathbf{A}}\hat{\mathbf{A}}^T = \mathbf{I}$. Since the new mixing matrix $\hat{\mathbf{A}}$ is orthogonal, the unmixing matrix applied on $\mathbf{x}$ to give the independent components is simply the transpose: $\hat{\mathbf{A}}^{-1} = \hat{\mathbf{A}}^T$.

The definition of whitening above is rather theoretical, given in terms of random variables for the mixtures. When it comes to practical use of ICA, each observed mixture x_i is represented through a set of samples. Suppose that for all x_i we observe K samples, denoted as $\mathbf{x}_i = [x_i(1), \ldots, x_i(K)]$, resulting in the data matrix where each row represents one mixture:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \tag{2.23}$$

with $\mathbf{X} \in \mathbb{R}^{n \times K}$. The ICA model then becomes

$$\mathbf{X} = \mathbf{AS}, \tag{2.24}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the mixing matrix and $\mathbf{S} \in \mathbb{R}^{n \times K}$ is a matrix where each row contains the values of one independent component across samples. Mean-centering the mixtures ensures that each row $\mathbf{x}_i$ has mean 0. To compute a whitening matrix $\mathbf{M}$, we use the empirical covariance matrix $\text{Cov}(\mathbf{X}) = (h_{ij})$, defined as:

$$h_{ij} = \frac{1}{K-1} \sum_{k=1}^{K} (x_i(k) - \bar{\mathbf{x}}_i)(x_j(k) - \bar{\mathbf{x}}_j), \tag{2.25}$$

where $\bar{\mathbf{x}}_i$ is the mean of row $\mathbf{x}_i$. Assuming the data has been mean-centered we have $\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_j = 0$ and the formula simplifies to:

$$h_{ij} = \frac{1}{K-1} \sum_{k=1}^{K} x_i(k) \cdot x_j(k). \tag{2.26}$$

2.2.5 ICA Estimation

There are several algorithms to estimate the ICA model, relying on different ideas and principles. One such method is the estimation of independent components by maximum likelihood [27], where the final unmixing matrix maximizes the probability of the observed mixtures. Another efficient ICA algorithm proposed by Bell and Sejnowski in 1995 [2] relies on the infomax principle to find the unmixing matrix that maximizes the entropy of the independent components.

In 1999, Hyvärinen developed the FastICA algorithm [14], which estimates the independent components by maximizing their non-Gaussianity using the approximation of negentropy given in 2.16. FastICA is widely regarded as one of the fastest and

most robust ICA algorithms, making it particularly popular for solving source separation problems. The algorithm allows components to be extracted either one-by-one (deflation approach) or all at once (parallel approach), providing flexibility depending on the application. For all ICA applications in this thesis, the FastICA algorithm has been used and its details are presented in Appendix A.

2.3 Validation of Independent Components

2.3.1 Unreliability of Individual ICA Runs

ICA is widely used for extracting meaningful independent components from multidimensional data, aiming to uncover potential hidden structures and patterns [15]. A major challenge in these types of analyses is understanding which of the resulting components are reliable and worth exploring further, as not all components necessarily have the same interpretational value [10]. There are two main reasons behind this uncertainty. Firstly, as with any statistical method, the quality and quantity of data influence the accuracy of the final estimates, potentially introducing statistical errors [11]. Secondly, since the estimation process of ICA usually involves the optimization of an objective function (e.g. maximization of negentropy in FastICA), different initializations are likely to converge to different local optimum points. In many optimization problems, one approach is to run the method many times, evaluating the objective function after each run and choose the run with the best local optimum. In the case of ICA, however, the optimization landscape is such that in each of these local optimum points, a different subset of the components may have been accurately estimated [11]. Consequently, different local optima can yield equally good solutions and without further investigation, it is not possible to know which run could be trusted more than others.

2.3.2 ICASSO

In response to the above considerations, Himberg and Hyvärinen developed ICASSO [10], a method for assessing the reliability of independent components by combining the outputs of multiple ICA runs. The main idea behind ICASSO is that reliable components should appear consistently across runs, perhaps only with small numerical differences. By collecting all estimated components from repeated ICA runs and clustering them based on similarity, stable components can be identified as those forming tight, well-separated clusters. In contrast, less reliable components will likely vary significantly across runs and thus will not form clear clusters.

Suppose we have our preprocessed data $\mathbf{X} \in \mathbb{R}^{n \times K}$, consisting of n mixtures and K samples. We select an algorithm for ICA estimation (e.g. FastICA) and run the optimization process M times on the same data with random initializations, combining the resulting unmixing matrices $\mathbf{D}_j$, for $j = 1, \ldots, M$ into a single matrix $\mathbf{D} = [\mathbf{D}_1^T, \ldots, \mathbf{D}_M^T]^T \in \mathbb{R}^{nM \times n}$. An alternative way for obtaining a set of unmixing matrices is by using bootstrap samples of $\mathbf{X}$, potentially again with random initializations.

Each row of the combined matrix $\mathbf{D}$ represents an unmixing vector corresponding to one component. The goal is to cluster these components based on a suitable distance metric. To quantify the dissimilarity between two components i and j of $\mathbf{D}$, ICASSO computes

$$d_{ij} = 1 - |r_{ij}|, \tag{2.27}$$

where r_{ij} is the Pearson correlation coefficient between components i and j.

To choose a clustering strategy, Himberg and Hyvärinen recommend the use of agglomerative hierarchical clustering [10]. With this method, each data point starts as its own cluster and at each step the two closest clusters are merged until all points belong to a single cluster. This process builds a nested hierarchy of clusters, which can be visualized using a dendrogram (see Figure 2.3). The structure of the dendrogram reflects the merging process of the data, from individual elements at the bottom (leafs) to larger clusters formed at higher levels (branches). Unlike other clustering techniques (such as K-means), hierarchical clustering does not require committing to a pre-specified number of clusters. Instead, one can assess the dendrogram and cut the tree at a specific height to extract the desired number of clusters.

The distance between two clusters can be defined in various ways, one of the most common being average linkage, where the pairwise distances between the members of the two clusters are averaged. Two other strategies involve taking the minimum or the maximum of the distances, but these are not recommended for ICASSO as they were found to be unreliable in clustering the independent components [10].

Following the clustering of all nM components, the next step is to determine an appropriate number clusters to retain. This decision is often challenging, as it typically depends on application-specific goals and domain knowledge [11]. Generally speaking, a simple conservative choice would be to keep the same number of components ICA was run with. Although this choice remains consistent with the initial ICA runs, it risks potentially including unreliable clusters that are not well-separated. For that reason, one could compute some quality index for the clusters and keep the ones that are more compact and isolated. One such measure introduced along with ICASSO is the Cluster

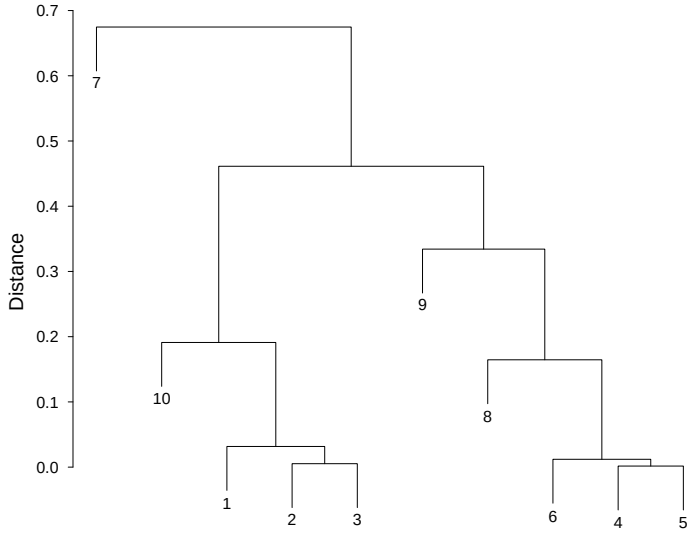

Figure 2.3: Illustration of a dendrogram from agglomerative hierarchical clustering on 10 observations. The observations begin to form clusters as we move higher up the tree, eventually merging into a single cluster.

Quality Index (CQI), defined as

$$I_q(C_m) = \frac{1}{|C_m|^2} \sum_{i,j \in C_m} |r_{ij}| - \frac{1}{|C_m||C_{-m}|} \sum_{i \in C_m} \sum_{j \in C_{-m}} |r_{ij}|, \tag{2.28}$$

where C_m is the set of indices of the components in the rows of $\mathbf{D}$ that belong to the mth cluster and C_{-m} are the indices outside of the mth cluster. CQI scores range between 0 and 1, with higher values indicating more reliable clusters.

Another recommended approach is to try different partitions and evaluate them using some quantitative measure to choose the most promising one. Furthermore, a visual inspection of the results, for example through the dendrogram or a two-dimensional projection of the clusters can help making a more informed decision.

For a given number of clusters L, ICASSO also provides a strategy to obtain L independent components, one from each cluster, that are statistically and algorithmically more reliable than those obtained from individual ICA runs. For cluster m, the unmixing vector is chosen as the medoid, defined as the vector whose sum of absolute correlations with all other vectors in the cluster is maximized. The index of the medoid of cluster m in $\mathbf{D}$ is given by

$$l_m = \arg\max_{i \in C_m} \sum_{j \in C_m} |r_{ij}|. \tag{2.29}$$

An alternative option would be to average the vectors of each cluster so that all components contribute to the final estimate. However, this method can be sensitive to noise,

as some clusters may contain outlier or noisy components. Additionally, averaging will result in new vectors that do not correspond to the outputs of any actual ICA run, making their interpretation less intuitive. In contrast, selecting the medoid ensures that each component has been produced by at least one ICA run.

2.4 Genetics

The human genome consists of approximately 3.2 billion base pairs of DNA, organized into 23 pairs of chromosomes, each pair consisting of one chromosome inherited from the mother and one from the father. Each chromosome carries hundreds of genes, which are segments of DNA that contain instructions for the production of proteins responsible for most biological functions. Our DNA is composed of four nucleotide bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T), the ordering of which determines the code for creating proteins.

Although the vast majority of DNA is identical across individuals, a small proportion varies, contributing to the diversity of observable human traits. These variations in specific DNA regions are called variants, and they are created by mutations. A common type of mutations are point mutations, which usually occur during DNA replication and refer to changes in a single nucleotide base [22]. Point mutations can be insertions, where one or more nucleotides are added to the sequence, deletions, where nucleotides are lost or substitutions, where one nucleotide is replaced by another. When a substitution occurs at a single base position, it is called Single Nucleotide Variant (SNV).

In each pair of chromosomes, an SNV can have different alleles, or alternative forms of a nucleotide base, and the specific pair of alleles an individual carries is known as their genotype. For example, the genotype at an SNV may consist of a base C in the paternal chromosome and a base T in the maternal chromosome. Typically, an SNV can have two possible alleles, for example either C or T, resulting in CC, CT or TT as the possible genotypes. If neither allele is extremely rare in the population and both appear with frequency greater than 0.01%, then the SNV is called Single Nucleotide Polymorphism (SNP). The frequency of a SNP is reported through the Minor Allele Frequency (MAF), which is the frequency of the less common allele [4].

The traits or characteristics influenced by genetic differences between humans, such as variations in SNPs, are referred to as phenotypes. These can be binary (e.g., presence or absence of a disease) or quantitative (e.g., cholesterol level, blood pressure), and are typically shaped by both genetic and environmental factors. The proportion of variance of a phenotype explained by genetic variation is known as heritability, denoted as $h^2 \in [0, 1]$. Traits with higher heritability are more strongly influenced by genetic variation, compared to non-genetic factors.

2.5 Genome-Wide Association Studies

2.5.1 Overview of GWAS

A Genome-Wide Association Study (GWAS) aims to identify statistical associations between known genetic variations across the human genome and a phenotype of interest [34]. GWASs usually focus on SNPs, analyzing whether specific variants are more common in individuals with a particular phenotype. In the case of categorical (mainly binary) phenotypes, if a SNP is found to occur significantly more often in individuals with the phenotype than those without it, it may suggest a potential genetic contribution to the phenotype. Similar conclusions can be made for quantitative traits, where the presence of a specific SNP can be associated with variation in the degree or value of a trait.

Testing for associations in a GWAS is accomplished by modeling the phenotype of an individual as a function of the SNP genotype, adjusting for demographic or environmental factors that may also affect the phenotype to avoid confounding [34]. For quantitative phenotypes, a linear regression model is typically employed, defined as

$$y_i = \mathbf{w}_i^T \boldsymbol{a} + x_i\beta + \epsilon, \tag{2.30}$$

where y_i is the value for the phenotype y of the ith individual, $\mathbf{w}$ is a vector of covariates (plus the intercept) with corresponding effects $\boldsymbol{a}$, x_i is the genotype at the modeled SNP, β is the effect of the genotype and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is normally distributed residual error. For dichotomous traits, the logistic regression model is used to model the probability p_i that the trait is present on the ith individual:

$$\log\left(\frac{p_i}{1-p_i}\right) = \mathbf{w}_i^T \boldsymbol{a} + x_i\beta. \tag{2.31}$$

In both models, the genotype $x_i \in \{0, 1, 2\}$ represents the number of copies of the reference allele of a given SNP. For instance, if the individual has the genotype CT and the reference allele is C at that SNP, then $x_i = 1$, while for the TT genotype we would have $x_i = 0$. It has been observed that the genotypes of most SNPs in a population approximately follow a binomial distribution $\text{Bin}(2, f)$, where f is the reference allele frequency. This principle is a central concept in genetics and is known as the Hardy-Weinberg Equilibrium (HWE).

A standard GWAS involves examining hundreds of thousands or even millions of SNPs independently for association with the phenotype [4]. This multiple testing framework requires a very strict significance threshold to avoid reporting false positives, which may appear by chance due to the large amount of tests conducted. For

this purpose, given that there are around 10^6 independent common variants in the human genome [17], a generally accepted p-value threshold is given from the Bonferroni correction at p-value $< \frac{0.05}{10^6} = 5 \cdot 10^{-8}$.

By requiring a very high level of significance, GWASs lose statistical power and many SNPs truly associated with the phenotype may not reach the p-value threshold. To ensure adequate statistical power for detecting true SNP effects, GWASs typically include hundreds of thousands of samples. This creates another issue as the collection of such large datasets is not always possible, although efforts on many fronts (cheaper sequencing techniques, publicly available genetic data etc.) lead to increasingly more data to be utilized for GWASs [34].

Another difficulty frequently encountered in a GWAS is identifying the causal variant behind a significant association. This is due to the tendency of nearby SNPs to be inherited together, which introduces a level of dependency between them in a population, a phenomenon known as Linkage Disequilibrium (LD). Due to potential presence of LD, a biologically significant signal in the data may not originate from the SNP tested, but from another variant in close proximity that is highly statistically correlated with it. This often results in clusters of associated SNPs in a given region, making it difficult to identify which specific variant is truly influencing the phenotype. Consequently, pinpointing specific SNPs as biologically causal requires additional steps, such as fine-mapping and functional analyses [31].

Additional challenges in interpreting specific signals in a GWAS occur due to most associations being found in non-coding regions of the genome, meaning regions where SNPs do not affect protein structure. In these regions, regulatory functions of SNPs are mostly unknown, but it is hypothesized that the risk-associated SNPs alter the expression of a critical gene [30]. A common practice for significant variants in non-coding regions is to assign them to their nearest gene, although the results of such approach can sometimes be misleading [6].

2.5.2 Applications and Biological Insights from GWASs

GWASs have become a fundamental part of modern genetics, offering a systematic way to uncover genetic variants associated with a wide range of human traits and diseases. The findings of GWASs have important clinical applications, for example by flagging regions of the DNA responsible for a disease, which can become targets during drug development [34]. Another application involves identifying high-risk individuals for certain diseases to improve patient outcomes by early prevention or treatment [31]. This can be achieved by computing a Polygenic Risk Score (PRS), which quantifies the genetic predisposition for a specific trait. There are many ways to compute PRSs, the

most simple of which is as a sum of alleles across a number of risk SNPs, weighted by the GWAS effect size for each SNP [34].

The results of GWASs also have the potential to reveal novel biological mechanisms underlying human traits and diseases. It is known that most phenotypes studied in GWASs are polygenic, often requiring thousands of genes to explain a significant proportion of their variance [34], which complicates the interpretation of GWAS associations. While laboratory experiments have yielded promising results in understanding the biology behind significant variant-trait associations [35], such experiments typically focus on individual genes and are difficult to scale across the many loci implicated in complex traits. To address this, computational tools like MAGMA [8] and DEPICT [26] have been developed to systematically interpret GWAS findings for highly polygenic traits. For a given trait, these tools aggregate GWAS signals across the genome and test for their enrichment in relevant biological pathways and functional categories.

Apart from the aforementioned polygenicity in GWASs, another factor convoluting our understanding of variant-trait associations is the pleiotropic nature of many SNPs [35]. Pleiotropy refers to the situation where genetic variants are found to be causal for multiple phenotypes, suggesting that their biological interpretation cannot be limited to only one of these phenotypes. In fact, many disease-causing variants are likely to be loaded onto important pathways that affect multiple diseases, forming complex genetic networks [3]. Hence, uncovering meaningful genetic pathways, particularly for complex traits can be aided by considering not only the directly associated genes, but also the broader network of traits influenced by those genes.

2.6 Decomposition of Variant-Trait Associations

A promising strategy for addressing the challenges of polygenicity and pleiotropy in the relationship between the human genome and phenome is to utilize results from a large number of GWASs to identify distinct genetic pathways, each involving a subset of variants and phenotypes. These pathways may correspond to separate biological mechanisms, involving smaller variant-phenotype sets, which could simplify their interpretation.

To identify such pathways, summary statistics derived from GWAS regression analyses can be utilized:

- β (Effect Size Estimate): The estimated regression coefficient representing the strength and direction of association between a genetic variant and a phenotype.
- SE (Standard Error): The standard error of the effect size estimate β, quantifying the uncertainty or variability in the estimate.

- z-score: Computed as $z = \frac{\beta}{\text{SE}}$, it is a standardized measure of association used for quantifying statistical significance.

- p-value: The probability of observing an association at least as extreme as the one measured, under the null hypothesis of no association.

For the methods considered in this thesis, the focus is on quantifying the strength of individual variant-trait associations, for which either effect sizes or z-scores are used. These statistics can then be aggregated into a matrix, allowing the application of various statistical techniques to extract latent components that mediate the observed associations. One such method is SVD [5, 32], which provides a natural way to obtain a latent representation of a matrix. Bayesian Non-Negative Matrix Factorization (bNMF) has also been used to decompose a GWAS summary statistics matrix into two non-negative ones that define separate clusters of variants and traits [21, 29, 33]. The main application of bNMF so far has been for Type 2 Diabetes (T2D) data, producing biologically meaningful and interpretable genetic pathways. Furthermore, factor analysis has been successfully employed to extract latent components from GWAS summary data [37].

More recently, ICA has been proposed as a method for obtaining statistically independent components of variants and traits [23]. It has been reported to show improved performance over SVD, identifying biologically meaningful pathways in complex traits, specifically Alzheimer's disease and forced expiratory volume in one second (FEV1). This thesis examines ICA as a method for decomposing variant-trait associations, analyzing the statistical properties of such an approach and discussing their implications on the inferred biology of the extracted latent components.

3. Methods & Data

This chapter outlines the methods used and developed in this thesis, as well as the data they were applied to. All analyses and methodologies described herein were independently implemented for the purposes of this work using the R programming language. Section 3.1 discusses the Decomposition of Genetic Associations (DeGAs), a method for decomposing variant-trait associations using SVD. Section 3.2 presents Genetic Unmixing by Independent Decomposition (GUIDE), the primary method studied in this thesis, utilizing ICA for variant-trait decomposition. It covers the standard methodology before introducing a novel extension based on ICASSO to obtain more reliable and consistent decompositions.

Section 3.3 discusses statistical tools for comparing and interpreting different decompositions. Section 3.4 presents simulation strategies to compare GUIDE and DeGAs, proposing new approaches for simulating GWAS summary statistics. Finally, Section 3.5 presents the Type 2 Diabetes (T2D) dataset, analyzed in Chapter 5.

3.1 Decomposition of Genetic Associations

The use of SVD to decompose a matrix of GWAS summary statistics data was introduced in 2019 by Tanigawa et al. [32] under the method Decomposition of Genetic Associations (DeGAs). Given a set X of n genetic variants and a set T of p traits, the input in DeGAs is a matrix $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{n \times p}$ where each entry w_{ij} is a GWAS summary statistic (either an effect size or a z-score) from a univariate regression of the ith variant in X onto the jth trait in T. The matrix is organized so that rows correspond to variants and columns correspond to phenotypes. Prior to decomposition, $\mathbf{W}$ is mean-centered and scaled phenome-wise so that each column vector has zero mean and unit variance. Then, the SVD of $\mathbf{W}$ is computed and the first L components are retained, resulting in the following low-rank approximation:

$$\mathbf{W} \approx \mathbf{U}_L \mathbf{\Sigma}_L \mathbf{V}_L^T, \tag{3.1}$$

where $\mathbf{U}_L = [\mathbf{u}_1, \cdots, \mathbf{u}_L] \in \mathbb{R}^{n \times L}$ and $\mathbf{V}_L = [\mathbf{v}_1, \cdots, \mathbf{v}_L] \in \mathbb{R}^{p \times L}$ contain the left and right singular vectors corresponding to variants and traits respectively, and $\mathbf{\Sigma}_L =$

$\text{diag}(\sigma_1, \ldots, \sigma_L)$ is a diagonal matrix whose entries are the top L singular values. In this formulation, the singular vectors $\mathbf{u}_l$ and $\mathbf{v}_l$ characterize the contribution of each variant and trait to the lth component, while the corresponding singular value σ_l reflects the relative importance of that component.

When it comes to selecting the number of components L, several strategies related to SVD have been proposed in the literature (see e.g. [18]), any of which can be chosen depending on the specific context. Indicatively, with DeGAs the above-average eigenvalue criterion was used, retaining components whose eigenvalues (given by the squared singular values) exceed the average [32]. In some applications involving SVD on GWAS summary statistics data, L was chosen based on a predefined threshold of cumulative variance explained [5], or by experimenting with different values [28].

3.2 Genetic Unmixing by Independent Decomposition

A recent alternative for decomposing a GWAS summary statistics matrix $\mathbf{W}$ representing associations between a set of variants X and a set of traits T was proposed by Lazarev et al. [23], introducing the method Genetic Unmixing by Independent Decomposition (GUIDE). In GUIDE, it is assumed that the associations between genetic variants and traits are mediated by a set of latent independent components, whose effects are combined additively to form $\mathbf{W}$. According to the Central Limit Theorem, these sums tend to become increasingly Gaussian and thus have higher entropy as more components are added. Reversing the summation of the original components to recover them can therefore be attempted by identifying a structure consisting of components whose entropy is minimized, a problem solved by ICA.

Motivated by the above, GUIDE extends the DeGAs framework by assuming that the components obtained from the tSVD of $\mathbf{W}$ are mixtures of some underlying latent components. These components are then estimated by applying ICA simultaneously to the variant and trait matrices, $\mathbf{U}_L$ and $\mathbf{V}_L$, introducing an additional step to get the final decomposition. In Section 3.2.1, the original GUIDE methodology is outlined and in Section 3.2.2 I propose a novel extension of the method that incorporates ICASSO to improve component reliability.

3.2.1 GUIDE Overview

GUIDE begins by mean-centering both the rows and columns of $\mathbf{W}$. Then, given a user-specified number of components L, the tSVD of $\mathbf{W}$ is computed as in 3.1. The resulting weights $\mathbf{U}_L$ and $\mathbf{V}_L$ are concatenated row-wise to get $\mathbf{G} = (\mathbf{U}_L^T | \mathbf{V}_L^T)$, a matrix

of size $L \times (n+p)$, which is the input for ICA. Now, as mentioned in 2.2.4, it is useful to whiten the data $\mathbf{G}$ before ICA. This is ensured by the following operation (see Appendix B for details):

$$\mathbf{G} \leftarrow \sqrt{\frac{n+p-1}{2}} \cdot \mathbf{G}. \tag{3.2}$$

ICA is then applied with the FastICA algorithm on the whitened matrix $\mathbf{G}$, solving the problem

$$\mathbf{G}_u = \mathbf{D}\mathbf{G}, \tag{3.3}$$

where $\mathbf{G}_u \in \mathbb{R}^{L\times(n+p)}$ is the (unknown) unmixed matrix and $\mathbf{D} \in \mathbb{R}^{L\times L}$ is the (also unknown) orthogonal unmixing matrix. FastICA is run using the contrast function $G(x) = \log(\cosh(x))$ to approximate the negentropy in 2.16 and unless otherwise specified, the components are extracted in parallel. Finally, the GUIDE weights are given by

$$\begin{aligned} \tilde{\mathbf{W}}_{XL} &= \mathbf{U}_L\mathbf{D}^T \\ \tilde{\mathbf{W}}_{TL} &= \mathbf{V}_L\mathbf{D}^T, \end{aligned} \tag{3.4}$$

where $\tilde{\mathbf{W}}_{XL} \in \mathbb{R}^{n\times L}$, $\tilde{\mathbf{W}}_{TL} \in \mathbb{R}^{p\times L}$ contain the variant and trait weights for each component. These weights can be directly extracted from the unmixing output as $\mathbf{G}_u = (\tilde{\mathbf{W}}_{XL}^T|\tilde{\mathbf{W}}_{TL}^T)$.

From this formulation, we can see that the GUIDE and DeGAs weights differ only by a rotation operation, estimated with ICA. This means that they can be interpreted in the same way, so that each column of the weight matrices characterizes the contribution of the variants and traits to the respective component. Note however that while in DeGAs the components are naturally ordered according to their significance, such ordering is not available in GUIDE due to the arbitrary permutation introduced by ICA.

Another difference between the two methods can be found in terms of independence of the resulting components. With tSVD, the columns of the variant weight matrix $\mathbf{U}_L$ and trait weight matrix $\mathbf{V}_L$ each form orthogonal sets. Since each column represents one component, this means that the components are also orthogonal to one another (with respect to the variants and traits respectively), hence uncorrelated. Uncorrelatedness implies linear independence, a condition that ICA strengthens by maximizing the non-Gaussianity of the components through the negentropy, which constitutes a more general notion of independence (see Section 2.2.3). This can be seen as making the GUIDE components statistically more independent than the DeGAs-derived components.

Regarding the choice of L, GUIDE proposes a method based on the non-identifibility of Gaussian components by ICA. The main idea is that when a dataset

consists of a mixture of signals occurring from both non-Gaussian and Gaussian sources, ICA can reliably recover the non-Gaussian components, but will arbitrarily rotate the Gaussian ones [15]. Consequently, if the true number of non-Gaussian components is L, running ICA with $K > L$ will yield an unmixing matrix in which L rows correspond to independent, non-Gaussian components, while the remaining $K - L$ rows are random unmixing vectors. Hence, running ICA again with a different initialization will retain the non-random components, but the random ones will be different. Thus, by computing the correlations between the rows of the two unmixing matrices, one will be able to match the L true ones, but not the rest, hence L will be identifiable as the number of high correlations found. By the same reasoning, ICA can be run J times so that for each pair of unmixing matrices out of the $\binom{J}{2}$ possible combinations, the number of highly correlated rows (e.g. $|r| > 0.95$) is computed, the median of which can be the final estimate of L.

This approach relies on the fundamental properties of ICA model estimation, which should indicate which components are non-Gaussian and hence correspond to true latent factors. Despite the theoretical guarantees of this method, I demonstrate in Section 4.1 that it is actually not so effective in practice, either due to insufficient data or violations of the assumption that the latent components are truly independent.

3.2.2 Extending GUIDE with ICASSO

As we discussed in Section 2.3.1, the ICA model is estimated through an iterative optimization procedure that begins with a random initialization. This stochasticity often leads to convergence to different local optima, corresponding to different sets of independent components, which may provide equally good decompositions. This is particularly relevant in this application of ICA in GWAS data as the summary statistics matrix can be very large, making the optimization landscape of ICA highly complicated. Indicatively, some of the datasets used as input for DeGAs and GUIDE consisted of over 200,000 SNPs and hundreds of traits. In such high-dimensional settings, consistent convergence with random initializations is unlikely, making reliance on a single ICA run potentially unstable and the resulting components less trustworthy [10].

To address this uncertainty, I propose incorporating ICASSO into the GUIDE framework to obtain more reliable components. Following the standard ICASSO procedure (see Section 2.3.2), FastICA is run I times with random initializations to solve Equation 3.3, obtaining multiple unmixing matrices $\{\mathbf{D}_i\}_{i=1}^{I}$, which are aggregated into a single matrix $\mathbf{T} = [\mathbf{D}_1^T \cdots \mathbf{D}_I^T]^T \in \mathbb{R}^{LI \times L}$. Then, pairwise correlations between the LI components in $\mathbf{T}$ are computed and agglomerative hierarchical clustering is applied with average linkage. To remain consistent with the GUIDE framework, a partition

into L clusters is selected, the same number ICA is run with. The final unmixing matrix $\hat{\mathbf{D}}$ is constructed by selecting the medoid from each cluster to be the unmixing vector of one component, corresponding to one row $\mathbf{d}_l^T$ of $\hat{\mathbf{D}}$, for $l = 1, \dots L$. Finally, the variant and trait weights are computed by multiplying the tSVD matrices $\mathbf{U}_L, \mathbf{V}_L$ with $\hat{\mathbf{D}}^T$ as shown in Equations 3.4.

By utilizing information from multiple ICA runs, this new approach produces components that are both statistically and algorithmically more stable. Importantly, it remains consistent with the GUIDE methodology, as each final component is guaranteed to have appeared in at least one ICA solution. Additionally, ICASSO offers a natural way to assess and order the reliability of components through the Cluster Quality Index (CQI), defined in 2.28. Since higher CQI scores indicate greater consistency across ICA runs, they can potentially serve as a measure of component significance in GUIDE, similar to the singular values for DeGAs.

3.3 Component Interpretation

Different methodologies for decomposing variant-trait associations will result in different sets of components, each with its own biological properties and implications. Understanding which method provides a more meaningful decomposition is not a simple task, as there is no ground truth to compare the components with. Instead, performance must be assessed relative to existing biological knowledge. In this section, I discuss quantitative measures that can facilitate the interpretation of GUIDE and DeGAs components, helping to understand their biological relevance.

First, to quantify how much each variable and phenotype contributes to a given component, the contribution scores can be used [23]. For DeGAs, the variant and phenotype contribution scores are defined as $\mathbf{U}_L^{\circ 2} = (u_{il}^2)$ and $\mathbf{V}_L^{\circ 2} = (v_{jl}^2)$ respectively, where $i = 1, \dots, n$, $j = 1, \dots, p$ and $l = 1, \dots L$. Similarly, the GUIDE contribution scores are $\tilde{\mathbf{W}}_{XL}^{\circ 2} = (w_{il}^2)$ for the variants and $\tilde{\mathbf{W}}_{TL}^{\circ 2} = (w_{jl}^2)$ for the phenotypes. The symbol $\circ$ denotes the element-wise exponent of a matrix.

Because the columns of $\mathbf{U}_L$ and $\mathbf{V}_L$ are orthonormal, hence have unit length, the DeGAs contribution scores for a given component always sum to 1. It can easily be shown that the same holds for GUIDE. From Equation 3.4, the lth column of $\tilde{\mathbf{W}}_{XL}$ is given as $\mathbf{U}_L\mathbf{d}_l$, where $\mathbf{d}_l$ is the lth row (transposed) of the unmixing matrix $\mathbf{D}$. Since $\mathbf{D}$ is orthogonal, $\mathbf{d}_l$ has unit length, therefore

$$||\mathbf{U}_L\mathbf{d}_l||^2 = (\mathbf{U}_L\mathbf{d}_l)^T(\mathbf{U}_L\mathbf{d}_l) = \mathbf{d}_l^T\mathbf{U}_L^T\mathbf{U}_L\mathbf{d}_l = \mathbf{d}_l^T\mathbf{I}_L\mathbf{d}_l = \mathbf{d}_l^T\mathbf{d}_l = 1. \tag{3.5}$$

By the same reasoning, each column of $\tilde{\mathbf{W}}_{TL}$ also has unit length, therefore the GUIDE variant and phenotype contribution scores for a given component sum to 1. It is worth

noting that this property is retained if we implement GUIDE with ICASSO, as even though the unmixing matrix occurs from the combination of many unmixing vectors $\mathbf{d}_l^T$, hence may not be orthogonal, the individual vectors still have unit length.

Apart from quantifying the contribution of each variant and phenotype to a given component, contribution scores also provide a principled basis for identifying which variants and phenotypes are loaded to each component [23]. Since there are n variant contribution scores and p phenotype contribution scores for each component, each summing to 1, if these scores were uniformly distributed, then each variant and phenotype would contribute $\frac{1}{n}$ and $\frac{1}{p}$ respectively. Hence, the ith variant can be considered to be loaded to the lth component if its contribution exceeds this uniform baseline, that is, $u_{il}^2 > \frac{1}{n}$ in DeGAs or $w_{il}^2 > \frac{1}{n}$ in GUIDE. Similarly, the jth phenotype can be considered to belong to component l if $v_{jl}^2 > \frac{1}{p}$ for DeGAs or $w_{jl}^2 > \frac{1}{p}$ for GUIDE.

When it comes to biological interpretation of a derived decomposition, having sparse components that involve only a small number of variants and phenotypes may be preferable to more dense ones. Sparse components can be easier to interpret, while they may point more directly to specific biological pathways or mechanisms, without involving non-relevant associations. To assess the sparsity of a decomposition, one can count the number of variants and phenotypes that are loaded to each component and compare that to the total number involved in the data. An alternative measure of sparsity is kurtosis, discussed in Section 2.2.3 in the context of random variables. For a sample $\mathbf{x} = [x_1, \ldots, x_k]$, the kurtosis is

$$\kappa(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{x_i - \bar{\mathbf{x}}}{s_k} \right)^4 - 3, \tag{3.6}$$

where $\bar{\mathbf{x}}$ is the sample mean and s_k is the sample standard deviation of $\mathbf{x}$. To quantify the overall sparsity of a component, the kurtosis can be computed over its combined variant and trait weights. For DeGAs, the input in Equation 3.6 for the kurtosis of the lth component is the concatenation of the lth columns of the weight matrices $\mathbf{U}_L$ and $\mathbf{V}_L$, while for GUIDE, the corresponding weights are taken from the lth columns of $\tilde{\mathbf{W}}_{XL}$ and $\tilde{\mathbf{W}}_{TL}$.

Another desirable property that can lead to more interpretable components is independence, which suggests minimal overlap of information. To quantify the independence between two components, the weighted Jaccard similarity index can be used, referred to here simply as the Jaccard index. Given two vectors $\mathbf{x} = [x_1, \ldots, x_n]$ and $\mathbf{y} = [y_1, \ldots, y_n]$ with $x_i, y_i > 0$, the Jaccard index is defined as

$$J(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}. \tag{3.7}$$

This measure ranges from 0 to 1, with higher values indicating greater similarity between the two vectors. In this thesis, the Jaccard index is used to measure overlap of information between two components, separately for variants and traits. The input vectors are the absolute values of either variant or trait weights associated with each component, where weights corresponding to lower than uniform contribution scores are set to be exactly 0 to ensure that the index is determined only by significant weights.

3.4 Simulations

In this section, I discuss simulation strategies for comparing the performance of GUIDE and DeGAs across diverse settings. I first present a simulation strategy for simple block-structured GWAS matrices, before introducing new, more intricate simulations of polygenic additive structures.

3.4.1 Block Structures

A block matrix can be interpreted as a matrix that has been subdivided into distinct blocks of smaller matrices. One example of such a structure is a diagonal block matrix, where non-zero blocks $\mathbf{B}_l$ are found across the "diagonal" and all other values are 0:

$$\mathbf{W} = \begin{bmatrix} \mathbf{B}_1 & & \mathbf{0} \\ & \mathbf{B}_2 & \\ & & \ddots \\ \mathbf{0} & & \mathbf{B}_L \end{bmatrix}. \tag{3.8}$$

If $\mathbf{W} \in \mathbb{R}^{n \times p}$ contains summary statistics from a set X of n variants to a set T of p traits, then the distinct blocks can be understood as separate components representing associations between different subsets of variants and traits. By having null effect sizes outside the diagonal defined by the blocks, we essentially assume that the components do not overlap, hence each variant and trait is loaded onto one component only. In reality, however, a genetic architecture with perfectly separated blocks as in Equation 3.8, hence independent, is unlikely to exist. This means that despite being able to identify components of concentrated effects, we expect that at least some of the variants and traits will be found in multiple components.

To simulate block-like genetic architectures, we assume $\mathbf{W}$ can be factorized as

$$\mathbf{W} = \mathbf{W}_{XL}\mathbf{W}_{TL}^T, \tag{3.9}$$

where $\mathbf{W}_{XL} \in \mathbb{R}^{n \times L}$ and $\mathbf{W}_{TL} \in \mathbb{R}^{p \times L}$ are weight matrices, representing the associations from genetic variants and traits respectively to the components.

Figure 3.1 visualizes such a structure for 9 variants, 6 traits and 3 components, highlighting the presence of dependencies between the components. The edges from the variants correspond to the weights of $\mathbf{W}_{XL}$ and the edges from the traits are captured in $\mathbf{W}_{TL}$. In $\mathbf{W}_{XL}$, the lth column represents the weights from the variants to the lth latent component and similarly the lth column of $\mathbf{W}_{TL}$ contains the weights from the same component to the traits. For both $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$, absence of an edge clearly implies a zero entry in the weight matrix.

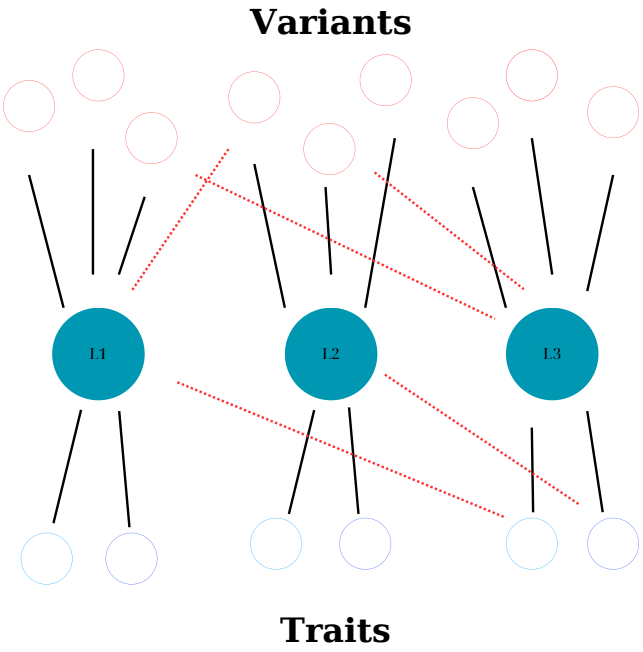

Figure 3.1: Graphical illustration of block-like genetic architecture where dependencies between components are introduced. The black edges represent non-zero weights defining the main components, while the red dashed edges correspond to effects outside of the block diagonal of $\mathbf{W}$, making the components dependent.

For simplicity, we assume that the blocks are of equal size and not overlapping, meaning that the number of non-zero weights to and from each component is the same, or that L divides both n and p. To simulate $\mathbf{W}_{XL}$, for each of the L columns, random weights sampled from $\text{Unif}(-\alpha, \alpha)$ are assigned to the non-null $\frac{n}{L}$ variants of this column, where $\alpha > 0$. For the lth column, the non-null variants are $f_n(l) = \{(l-1)\frac{n}{L} + 1, \ldots, l\frac{n}{L}\}$, for $l = 1, ..., L$, so that

$$\mathbf{W}_{XL}^{(f_n(l),l)} \sim \text{Unif}(-\alpha, \alpha). \tag{3.10}$$

For each of the remaining $n - \frac{n}{L}$ variants of the lth column (i.e. the variants that are not in the set $f_n(l)$), an association with the lth component is introduced with probability

P_{dep} by assigning a non-zero weight from the same distribution Unif$(-\alpha, \alpha)$. For $\mathbf{W}_{TL}$, the same process is repeated, again iterating through the columns for the traits:

$$\mathbf{W}_{TL}^{(f_p(l),l)} \sim \text{Unif}(-\alpha, \alpha). \tag{3.11}$$

Same as with $\mathbf{W}_{XL}$, for the remaining $p - \frac{p}{L}$ traits of the lth column, a non-zero weight, sampled from Unif$(-\alpha, \alpha)$ is assigned with probability P_{dep}.

After simulating $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$, Gaussian noise is added to each from $\mathcal{N}(0, \sigma_{\text{A}}^2)$. Finally, they are multiplied to get $\mathbf{W} = \mathbf{W}_{XL}\mathbf{W}_{TL}^T$ and additional noise is added from $\mathcal{N}(0, \sigma_{\text{B}}^2)$ to get the final summary statistics matrix. Here, σ_{A} and σ_{B} are hyperparameters for the standard errors that control the level of induced noise. By adding the second layer of noise, it is ensured that $\mathbf{W}$ will be a full rank matrix. Without this step, the rank would be at most L:

$$\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{W}_{XL}\mathbf{W}_{TL}^T) \leq \min(\text{rank}(\mathbf{W}_{XL}), \text{rank}(\mathbf{W}_{TL}^T)) \leq L \tag{3.12}$$

With this simulation strategy, we can interpret P_{dep} as the level of dependency between components. For $P_{\text{dep}} = 0$, the components are fully independent, while as P_{dep} increases, they become more dependent. When $P_{\text{dep}} = 1$, the structure is completely random.

3.4.2 Polygenic Additive Structures

The block-structured genetic simulation framework described in the previous section can offer valuable insights into the fundamental properties of DeGAs and GUIDE. However, its distinct block-like patterns are unlikely to represent real genetic architectures. Consequently, it is insufficient for drawing conclusions about the expected performance of these methods on actual genetic data. To evaluate GUIDE and DeGAs in more realistic scenarios, I simulated GWAS summary statistics data based on a polygenic additive model, using a similar (but not the same) approach as in [37].

Same as before, the goal is to simulate a genetic architecture comprising of n variants in a set X and p traits in a set T, mediated by L components, which can be fully characterized by $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$. We start by considering n independent genetic variants $x_1, ..., x_n$ with minor allele frequencies (MAFs) $f_i \sim \text{Unif}(0.01, 0.5)$ for $i = 1, ..., n$ and p quantitative phenotypes $y_1, ..., y_p$. True effects β_{ij} for the ith variant on the jth phenotype are generated as linear combinations of L components, which can be expressed as

$$\beta_{ij} = \mathbf{W}_{XL}^{(i,\cdot)} \left(\mathbf{W}_{TL}^{(j,\cdot)}\right)^T, \tag{3.13}$$

where the indexing $(m, \cdot)$ denotes the mth row of a matrix. In a vectorized form,

Equation 3.13 is equivalent to

$$\mathbf{B} = \mathbf{W}_{XL}\mathbf{W}_{TL}^{T}, \tag{3.14}$$

where $\mathbf{B} \in \mathbb{R}^{n \times p}$ contains the true effect sizes of each variant onto each trait. To generate weights for $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$, two scenarios are considered:

1. The weights of the underlying architecture follow a Gaussian distribution.
2. The weights are super-Gaussian, following a Laplace distribution.

For simplicity, the means of Gaussian and Laplace distributions are set to 0 for both $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$. Assuming that the independent variants have additive heritability h^2 over each phenotype, the variance of the weights is selected to be

$$\begin{aligned} \mathrm{Var}\Big(\mathbf{W}_{XL}^{(i,\cdot)}\Big) &= \frac{h^2}{n \cdot L \cdot \mathrm{Var}(x_i)} \\ \mathrm{Var}(\mathbf{W}_{TL}^{(j,\cdot)}) &= 1, \end{aligned} \tag{3.15}$$

where under Hardy-Weinberg equilibrium we have that $\mathrm{Var}(x_i) = 2f_i(1 - f_i)$, which is the variance of $\mathrm{Bin}(2, f_i)$. The variance of $\mathbf{W}_{XL}$ suggests that each variant explains the same amount of heritability for each trait. The weights are then sampled from

$$\begin{aligned} \mathbf{W}_{XL}^{(i,\cdot)} &\sim \mathcal{N}\Big(0, \frac{h^2}{n \cdot L \cdot \mathrm{Var}(x_i)}\Big) \\ \mathbf{W}_{TL}^{(j,\cdot)} &\sim \mathcal{N}(0, 1) \end{aligned} \tag{3.16}$$

for the Gaussian setting and

$$\begin{aligned} \mathbf{W}_{XL}^{(i,\cdot)} &\sim \mathcal{L}\left(0, \sqrt{\frac{h^2}{2 \cdot n \cdot L \cdot \mathrm{Var}(x_i)}}\right) \\ \mathbf{W}_{TL}^{(j,\cdot)} &\sim \mathcal{L}\left(0, \sqrt{\frac{1}{2}}\right) \end{aligned} \tag{3.17}$$

for the Laplace setting, where the Laplace distribution is parameterized according to the probability density function in Equation 2.7, using the mean (set to 0 for both $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$) and scale b. The scale was chosen to retain the variances in 3.15, as the Laplacian variance is $V = 2b^2$, which solving for scale gives us $b = \sqrt{\frac{V}{2}}$. Since heritability here only controls the variance of the sampling distributions, it is set at a constant value $h^2 = 0.1$.

The current simulation strategy would result in most effect sizes to be non-zero, meaning that the underlying genetic architecture would consist of dense components, each associated with a large number of variants and phenotypes. In practice, however,

it is unlikely that all variant-phenotype associations will be significant and we expect that there some null effects in the data. To account for this, sparsity is introduced in the simulated data by randomly setting some of the weights to be exactly 0, following an approach similar to that described in [24]. This is done component-wise, setting each weight of the lth column of $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$ to be 0 with probability P_l, requiring at the same time that every column has at least one non-zero value:

$$\begin{aligned} \mathbf{W}_{XL}^{(\cdot,l)} &\leftarrow (1 - \boldsymbol{\theta}_{XL}^{l}) \cdot \mathbf{W}_{XL}^{(\cdot,l)} \\ \mathbf{W}_{TL}^{(\cdot,l)} &\leftarrow (1 - \boldsymbol{\theta}_{TL}^{l}) \cdot \mathbf{W}_{TL}^{(\cdot,l)}, \end{aligned} \tag{3.18}$$

where $\boldsymbol{\theta}_{XL}^{l}$ and $\boldsymbol{\theta}_{TL}^{l}$ are column vectors of length n and p respectively and each of their entry follows a Bernoulli distribution with success probability P_l.

To fully explore the capacity of GUIDE and DeGAs, I consider three scenarios; one where there is no sparsity ($P_l = 0$) and two for low and high sparsity:

$$P_l \sim \begin{cases} \text{Unif}(0, 0.2), & \text{if sparsity is low} \\ \text{Unif}(0.2, 0.7), & \text{if sparsity is high.} \end{cases} \tag{3.19}$$

After generating the effect size matrix $\mathbf{B} = \mathbf{W}_{XL}\mathbf{W}_{TL}^{T}$, noise $\mathbf{E}$ is added, where $E_{ij} \sim \mathcal{N}(0, \text{SE}_{ij}^2)$, with SE_{ij} being the standard error of the effect size β_{ij} of the ith variant onto the jth phenotype. Based on the linear regression model for quantitative traits (modeled as in Eq. 2.30 without any covariates), the standard error is computed as

$$\text{SE}_{ij} = \sqrt{\frac{\text{Var}(y_j) - \beta_{ij}^2 \cdot \text{Var}(x_i)}{N_{\text{samples}} \cdot \text{Var}(x_i)}}. \tag{3.20}$$

Now, because the effect sizes are small, we can simplify the formula above assuming that $\text{Var}(y_j) - \beta_{ij}^2 \cdot \text{Var}(x_i) \approx \text{Var}(y_j)$, which we can then set to 1 by further assuming that the phenotypes are processed by quantile normalization. The quantity N_{samples} corresponds to the sample size of the simulated GWAS, with more samples leading to increased statistical power and consequently, smaller standard error. Here, since the number of samples is not relevant to the way variants and phenotypes are analyzed, all effects can be considered to occur from well powered studies with the same sample size $N_{\text{samples}} = 10^5$, therefore getting:

$$\text{SE}_{ij} = \sqrt{\frac{1}{10^5 \cdot \text{Var}(x_i)}}. \tag{3.21}$$

Finally, to take into consideration the uncertainty behind each effect estimate β_{ij}, we compute the z-score matrix $\mathbf{Z}$ by dividing with the standard errors:

$$z_{ij} = \frac{\beta_{ij}}{\text{SE}_{ij}}. \tag{3.22}$$

3.4.3 Matching the Components

When reconstructing the generated weights $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$, neither of the methods studied in this thesis can preserve the true ordering. With DeGAs, the components are ordered by the singular values, which are not relevant to the simulation, while in GUIDE, the use of ICA leads to randomly permuted weights. Hence, to compare how efficiently DeGAs and GUIDE recover the simulated genetic architecture, the estimated components must first be matched with the true ones.

One way to correct for the permutation and sign differences between the simulated and estimated components is by matching them through their correlation. This is achieved as follows: first, the $L \times L$ correlation matrix is computed, which contains the cross-correlations between the true and estimated vectors of the weights of each component. Each component vector contains the weights to all variants and traits, therefore its length is $n + p$. Then, the highest (in absolute value) correlation is identified, corresponding to a pair of simulated and estimated weights. The next highest correlation is found in a similar manner, excluding the values for the first components, as a component should not be matched twice. This process is continued until all components have been matched and then possible sign changes are corrected.

3.5 Type 2 Diabetes Data

In addition to the simulations, I apply GUIDE and DeGAs to a T2D GWAS summary statistics dataset constructed by Smith et al. [29] and analyzed with bayesian Non-Negative Matrix Factorization (bNMF). The data consists of z-scores for 650 T2D-associated variants and 110 quantitative phenotypes related to T2D from GWASs across multiple ancestral populations.

The first step in preparing the dataset involved extracting significant (p-value $< 5 \cdot 10^{-8}$) variants from multiple T2D GWASs across different populations. Then, variants with MAF < 0.001 in any of the populations were excluded and LD pruning ($r^2 < 0.05$) was performed separately for each population's reference panel so that the remaining variants were independent in all populations. Variants with a large number of missing associations ($> 20\%$) across trait GWASs, ambiguous alleles (A/T or C/G), or that were multi-allelic were replaced by high-LD ($r^2 > 0.8$) proxies. Finally, variants were checked in the largest multi-ancestry T2D GWAS and removed if they had p-value > 0.05 or allele inconsistencies, resulting in 650 variants.

Trait selection began with 165 T2D-related quantitative phenotypes, prioritizing sex-specific and multi-ancestry GWASs. Traits were excluded if their median sample size was smaller than 5,000 or if their minimum p-value across the 650 variants was

not Bonferroni-significant (p-value $< \frac{0.05}{650}$). Highly correlated traits (r > 0.8) were also removed, retaining those with the highest variant-trait associations. The resulting 650×110 summary statistics matrix consisted of z-scores (regression coefficients divided by their standard errors), scaled by the square root of the study size, as estimated by mean sample size across all variants. The purpose of this scaling is to account for variability in sample sizes across studies and enable more uniform comparison of the phenotypes [33]. Remaining missing entries were imputed using proxy ($r^2 > 0.5$) variant z-scores or, if unavailable, the trait's median value. Further details on the preprocessing pipeline can be found in [29].

4. Simulation Results

To evaluate GUIDE and DeGAs, I conducted a series of simulations under many scenarios, which are presented in this chapter. In Section 4.1, the unreliability of GUIDE's approach for choosing an optimal number of components is demonstrated. Section 4.2 presents results from simulations where the true number of components is known, while Section 4.3 explores how each method performs when the number of components is misspecified. Section 4.4 concludes this chapter by evaluating whether CQI scores in GUIDE, provided through the novel extension with ICASSO, can provide a meaningful ranking of the components based on their relative importance. Flow chart 4.1 summarizes the topics and outcomes of each simulation conducted in this chapter.

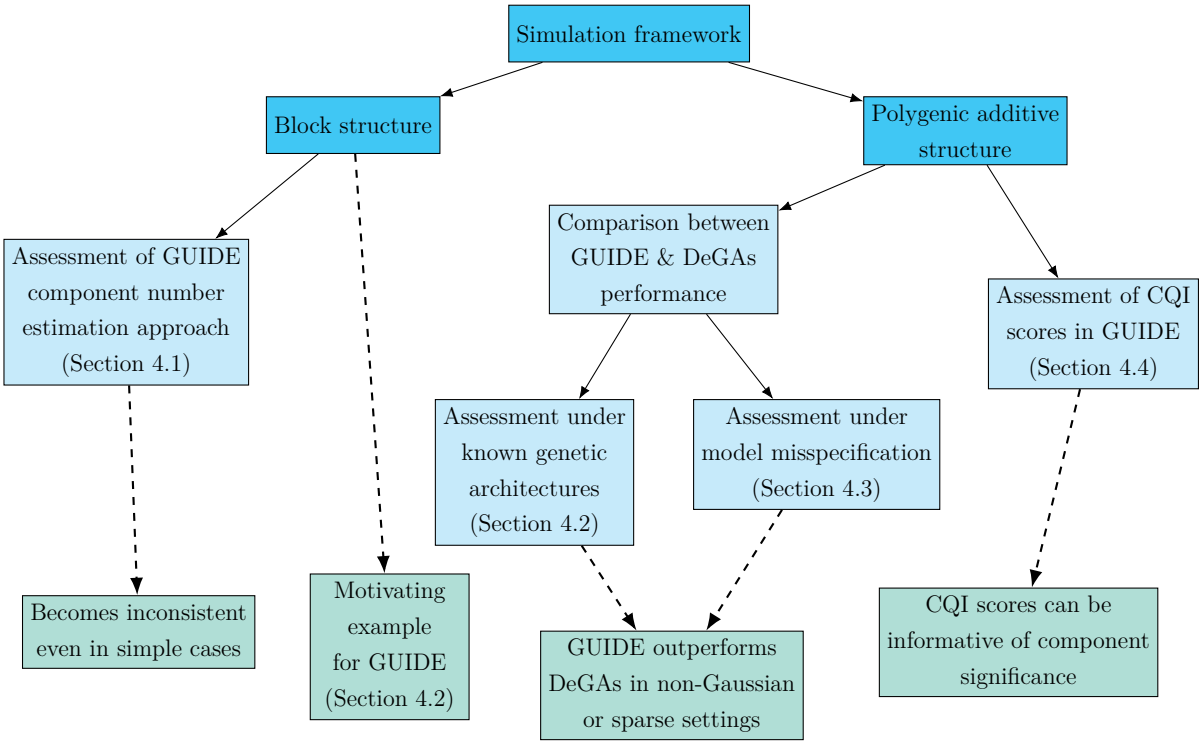

Figure 4.1: Overview of the simulation studies presented in this chapter. Two simulation frameworks are considered: block matrices and the polygenic additive models, shown in blue. These are used to assess various aspects of GUIDE and DeGAs, represented in light blue. The main results from these evaluations are highlighted in green.

4.1 GUIDE's Estimate of the Number of Components

To investigate GUIDE's utility in identifying the correct number of components of genetic architectures, all implementations in this section involve the original GUIDE method where ICA is run once, not the ICASSO-based extension. Also, in FastICA the components are extracted one-by-one and not in parallel, as parallel extraction was found to be more inconsistent for the experiments of this section.

To begin, two simple structures for block matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ are considered, listed in Table 4.1, which represent genetic architectures of different sizes. The matrices are generated as described in Section 3.4.1, with weights sampled from Unif$(-2, 2)$, assuming independent components ($P_{\text{dep}} = 0$) and setting noise standard deviations to $\sigma_{\text{A}} = 0.1$ and $\sigma_{\text{B}} = 0.5$. For each structure, 100 summary statistics matrices are generated and GUIDE's approach is used to determine the number of components. GUIDE is run with $J = 30$ unmixing matrices, initializing the method with $K_1 = 10$ and $K_2 = 30$ starting components for $\mathbf{W}_1$ and $\mathbf{W}_2$ respectively and accept matching columns between unmixing matrices when their correlation is $|r| > 0.95$.

Table 4.1: Different sizes for block matrix simulations.

	Variants	Phenotypes	Components
Structure 1 ($\mathbf{W}_1$)	200	30	5
Structure 2 ($\mathbf{W}_2$)	1000	100	10

Figure 4.2 shows histograms for the final estimates of L for the 100 simulations of the two block matrix structures. We observe that for the smaller structure 1, GUIDE rarely finds the correct number of components, consistently overestimating it, whereas it is considerably more accurate with the bigger structure 2. This result suggests that the method can become unstable with data consisting of a smaller number of variants and traits, even if their components are clear.

Now, although GUIDE's estimate of the number of components is generally correct with structure 2, some of the runs still resulted in overestimating the number of components by 1. One would expect the method to be able to fully recover the true number of components since the underlying genetic architecture is relatively simple, with no dependence between the components and small amounts of noise. Indicatively, the scree plot of the singular values for one of the simulated matrices $\mathbf{W}_2$ in Figure 4.3 clearly shows the presence of 10 components, with the rest being noise.

Based on this, a further investigation is prompted into whether GUIDE can re-

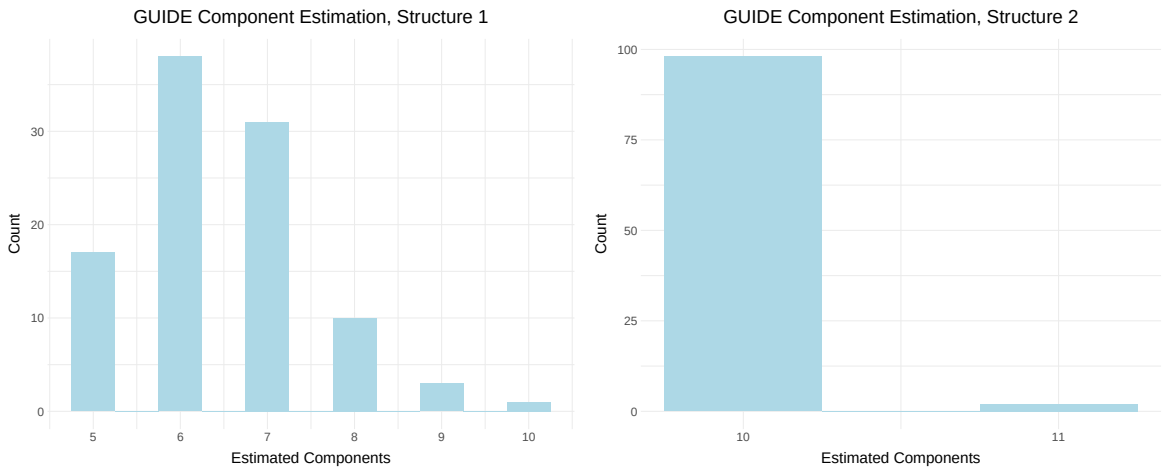

Figure 4.2: Histograms showing the selected number of components for Structure 1 (left) and Structure 2 (right) across 100 simulated summary statistics matrices, where the true number of components is 5 and 10, respectively.

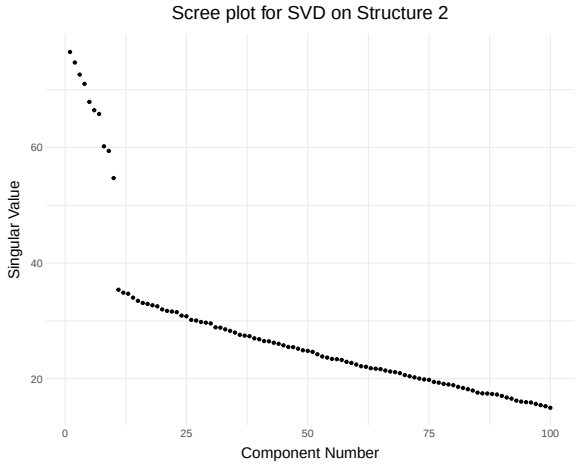

Figure 4.3: Scree plot for the singular values of a summary statistics matrix $\mathbf{W}_2$ generated based on the described structure 2.

tain its efficiency in estimating the true number of components with more complex structures. More specifically, because ICA assumes independent components, it is of interest to see what happens when this assumption is violated, i.e. when the components become dependent. With block structures, this can be simulated by selecting $P_{\text{dep}} > 0$. Using the same settings as in structure 2 previously, I generate summary statistic matrices $\mathbf{W}$ using $n = 1000$, $p = 100$ and $L = 10$, sampling weights from $\text{Unif}(-2, 2)$, with noise standard deviations $\sigma_{\text{A}} = 0.1$, $\sigma_{\text{B}} = 0.5$ and $J = 30$ ICA runs. The algorithm is initialized with $K = 30$ components and matching is considered when $|r| > 0.95$. Four different values for P_{dep} are considered, specifically 0.05, 0.1, 0.15 and 0.2. For each value, 100 summary statistic matrices are generated and the estimates of L for each are collected and displayed in Figure 4.4. We can see that as $P_{\text{dep}} > 0$

increases, the distribution of estimates moves further from the true value towards 0, meaning that GUIDE cannot identify the correct number of components.

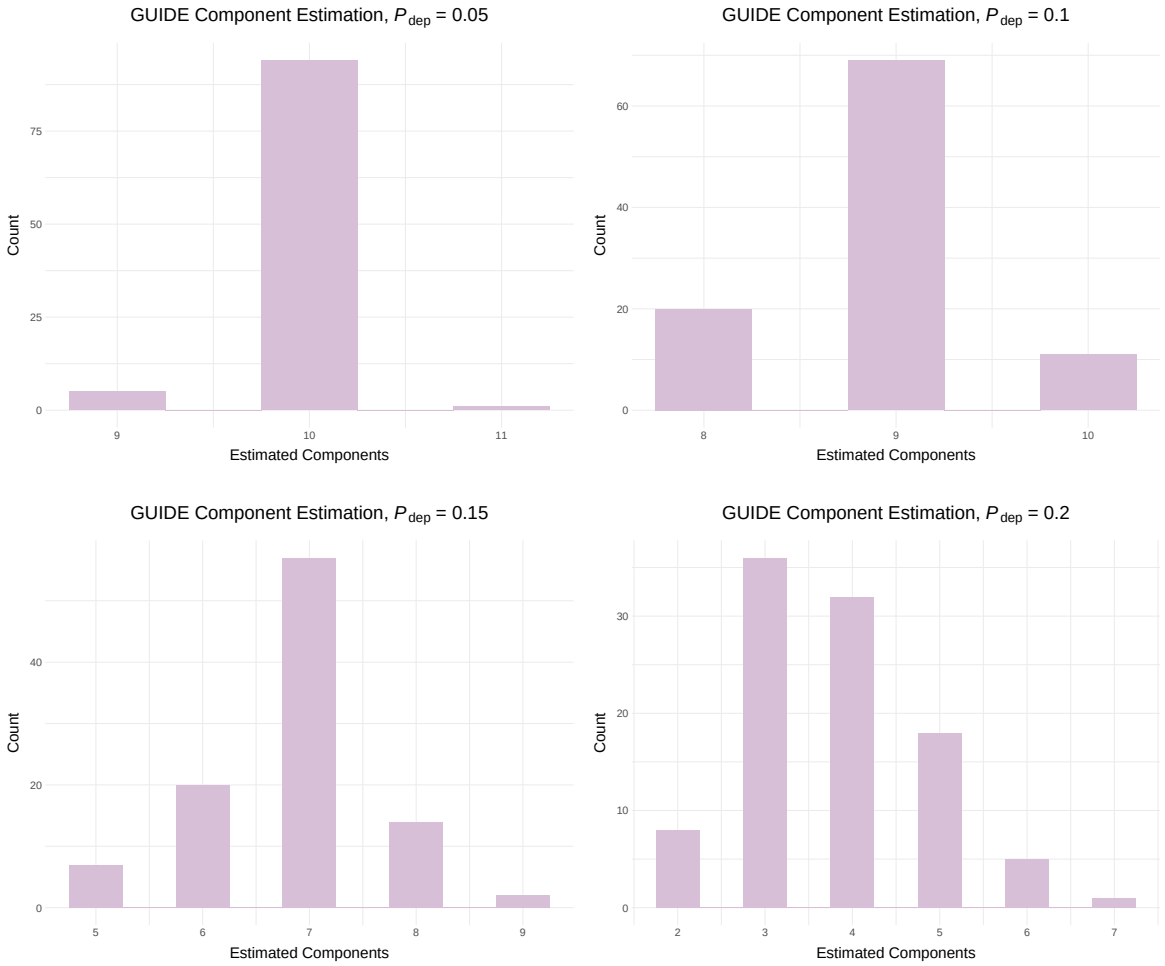

Figure 4.4: Histograms displaying the chosen components for structure 2 at different levels of dependence as determined by P_{dep}.

The examples above illustrate that GUIDE becomes very unstable in determining the number of components, even in relatively simple structures, when data are limited or the independence assumption is violated. It is also likely that this uncertainty would persist with more complex datasets, where different ICA runs may converge to distinct local optima, each corresponding to a different subset of components.

More generally, although several methods for decomposing variant-trait associations offer suggestions for choosing a number of components, no definitive solution exists. Given this inherent uncertainty, it is particularly relevant to explore how GUIDE and DeGAs perform when the number of components in a simulated genetic architecture is either underestimated or overestimated.

4.2 Simulations with Known Genetic Architectures

In this section, it is assumed that we have access to the true genetic architecture, meaning that the number of components is known. I begin by illustrating GUIDE's accuracy over DeGAs when it comes to decomposing genetic architectures originating from clear and well-separated underlying signals, specifically those of a block matrix. This analysis is then extended to more elaborate simulations based on polygenic additive structures, aiming to provide valuable insights onto the capacity of each method for decomposing real genetic data.

4.2.1 Block Simulation

To illustrate how GUIDE, unlike DeGAs, has the potential to capture true underlying components perfectly, even when these are not fully independent, I consider a block matrix consisting of $n = 3000$ genetic variants and $p = 200$ traits, with $L = 20$ latent components. Weight matrices $\mathbf{W}_{XL}, \mathbf{W}_{TL}$ are generated by sampling from Unif$(-2, 2)$, with noise standard deviations $\sigma_{\mathrm{A}} = 0.3, \sigma_{\mathrm{B}} = 0.5$ and dependence probability $P_{\mathrm{dep}} = 0.1$. Since the block simulation scheme under low dependence probability should result in well-separated signals which ICA should be able to easily identify, GUIDE is applied with only one ICA run, not using the ICASSO extension. After correcting for permutation and possible sign changes for GUIDE (as described in Section 3.4.3), repeating the same procedure for DeGAs's output, the weights from the variants and the traits to the components are plotted in Figure 4.5. From there, it is clear that GUIDE manages to recover the true weights, while DeGAs is considerably less accurate.

4.2.2 Polygenic Additive Structure Simulation Results

For the next simulation, z-score matrices are generated under a polygenic additive model, sampling $\mathbf{W}_{XL}$ and $\mathbf{W}_{TL}$ from both Gaussian and Laplace distributions. For each distribution, two data sizes are considered, listed in Table 4.2, and three sparsity levels: none, low, and high.

Table 4.2: Different sizes for polygenic additive structure simulations.

	Variants	Phenotypes	Components
Configuration 1 (**C1**)	500	50	5
Configuration 2 (**C2**)	1,000	100	10

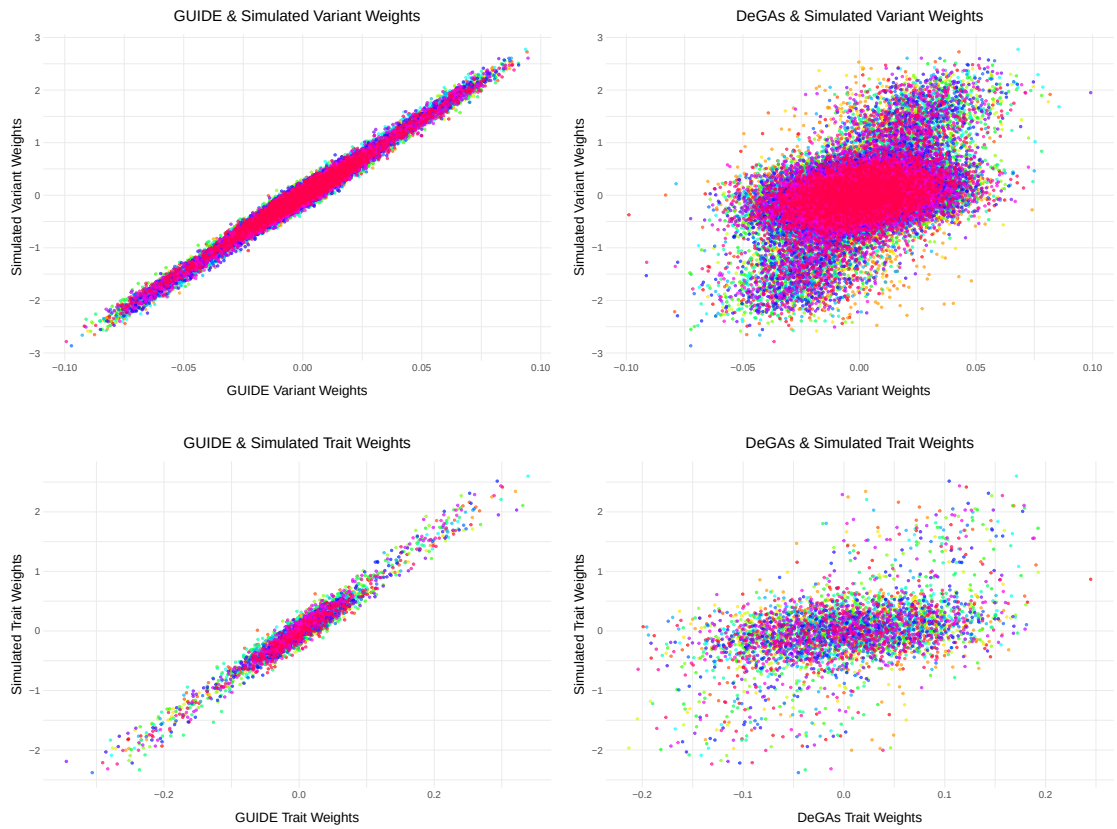

Figure 4.5: Scatter plots comparing the simulated variant and trait weights with the estimated ones using GUIDE and DeGAs. Each color represents a different component.

For each setting and data size, 300 z-score matrices are generated. GUIDE and DeGAs are then run on each with the true number of components to obtain estimates $\tilde{\mathbf{W}}_{XL}, \tilde{\mathbf{W}}_{TL}$. For this experiment, GUIDE is implemented with the proposed ICASSO extension (see Section 3.2.2), where ICA is run $I = 20$ times. After correcting the weights of both methods for permutation and sign changes, the accuracy of GUIDE and DeGAs is assessed by computing the correlation between estimates and ground truth, separately for the variant and phenotype weights. Specifically, $\text{Cor}(\mathbf{W}_{XL}, \tilde{\mathbf{W}}_{XL})$ and $\text{Cor}(\mathbf{W}_{TL}, \tilde{\mathbf{W}}_{TL})$ are calculated, defined as

$$\begin{aligned} \text{Cor}(\mathbf{W}_{XL}, \tilde{\mathbf{W}}_{XL}) &= \frac{1}{L}\sum_{l=1}^{L} \text{Cor}(\mathbf{W}_{XL}^{(\cdot,l)}, \tilde{\mathbf{W}}_{XL}^{(\cdot,l)}) \\ \text{Cor}(\mathbf{W}_{TL}, \tilde{\mathbf{W}}_{TL}) &= \frac{1}{L}\sum_{l=1}^{L} \text{Cor}(\mathbf{W}_{TL}^{(\cdot,l)}, \tilde{\mathbf{W}}_{TL}^{(\cdot,l)}), \end{aligned} \tag{4.1}$$

where L is the number of components. These expressions represent the average correlation between the true and estimated variant and phenotype weights for each component.

In addition to capturing the ground truth as accurately as possible, we are interested in obtaining components that are maximally sparse, a condition that improves

their interpretability as then information is conveyed in a more dense and concentrated manner. To measure the sparsity of the resulting estimates, the kurtosis for the components produced by GUIDE and DeGAs is also reported. The kurtosis is computed as in Equation 3.6 for the concatenated variant and trait weights of each component and the resulting values are averaged across all L components:

$$\kappa(\tilde{\mathbf{W}}_{XL}|\tilde{\mathbf{W}}_{TL}) = \frac{1}{L}\sum_{l=1}^{L}\kappa(\tilde{\mathbf{W}}_{XL}^{(\cdot,l)}|\tilde{\mathbf{W}}_{TL}^{(\cdot,l)}). \tag{4.2}$$

Figures 4.6 and 4.7 show the results of the simulation across both configurations, separately for Gaussian and Laplace settings. When the underlying weight distribution is Gaussian without induced sparsity, GUIDE performs similarly to DeGAs in recovering the true components. This outcome is expected, as ICA relies on the non-Gaussianity of components for signal separation, hence under Gaussian components it fails to improve the tSVD estimation. It is worth noting that despite the comparable performance, GUIDE's recovered components are more sparse than those of DeGAs, as seen from their increased kurtosis values.

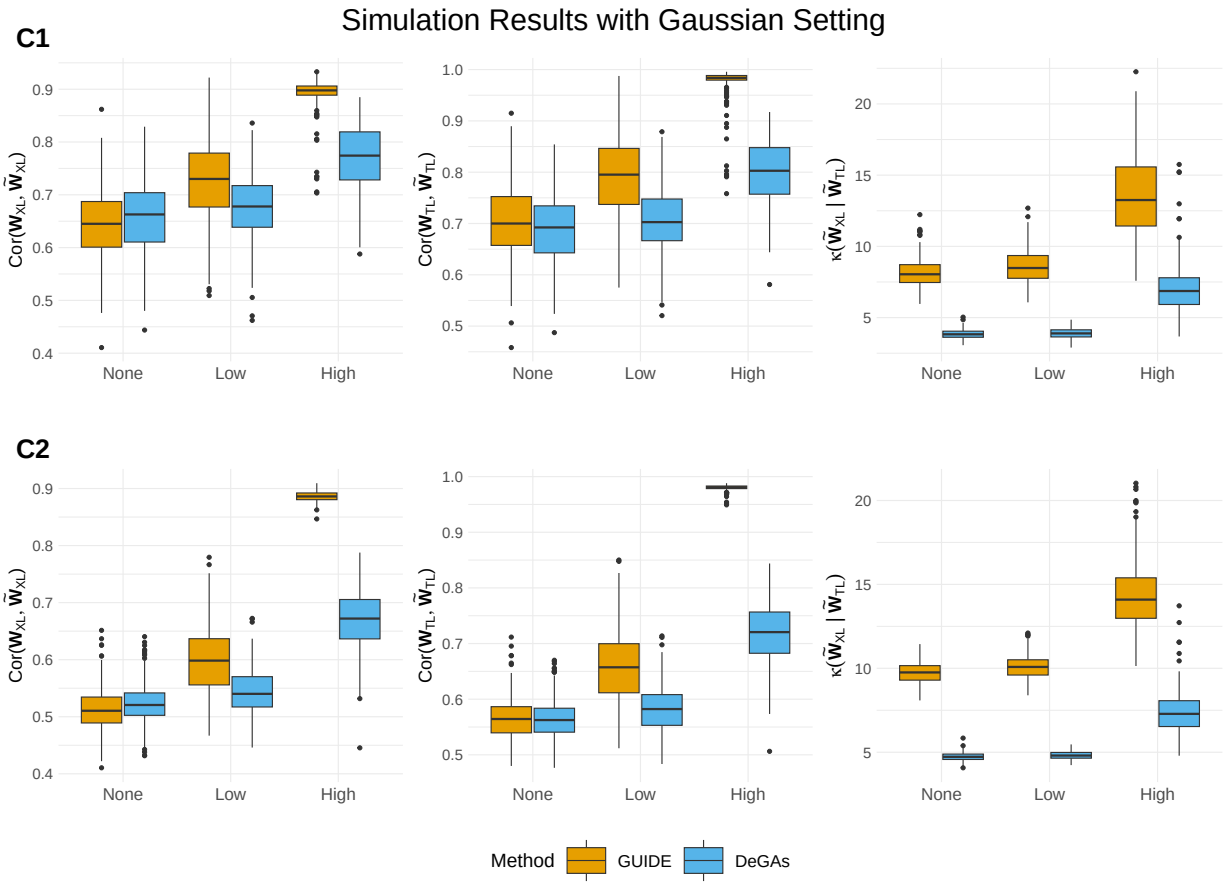

Figure 4.6: Boxplots comparing the performance of GUIDE and DeGAs across simulations where the weights are sampled from the Gaussian distribution. Three levels of sparsity are considered: "None", "Low" and "High" (described in Section 3.4.2), shown in the x-axis of each plot. Each row corresponds to a different configuration (**C1**, **C2**) from Table 4.2.

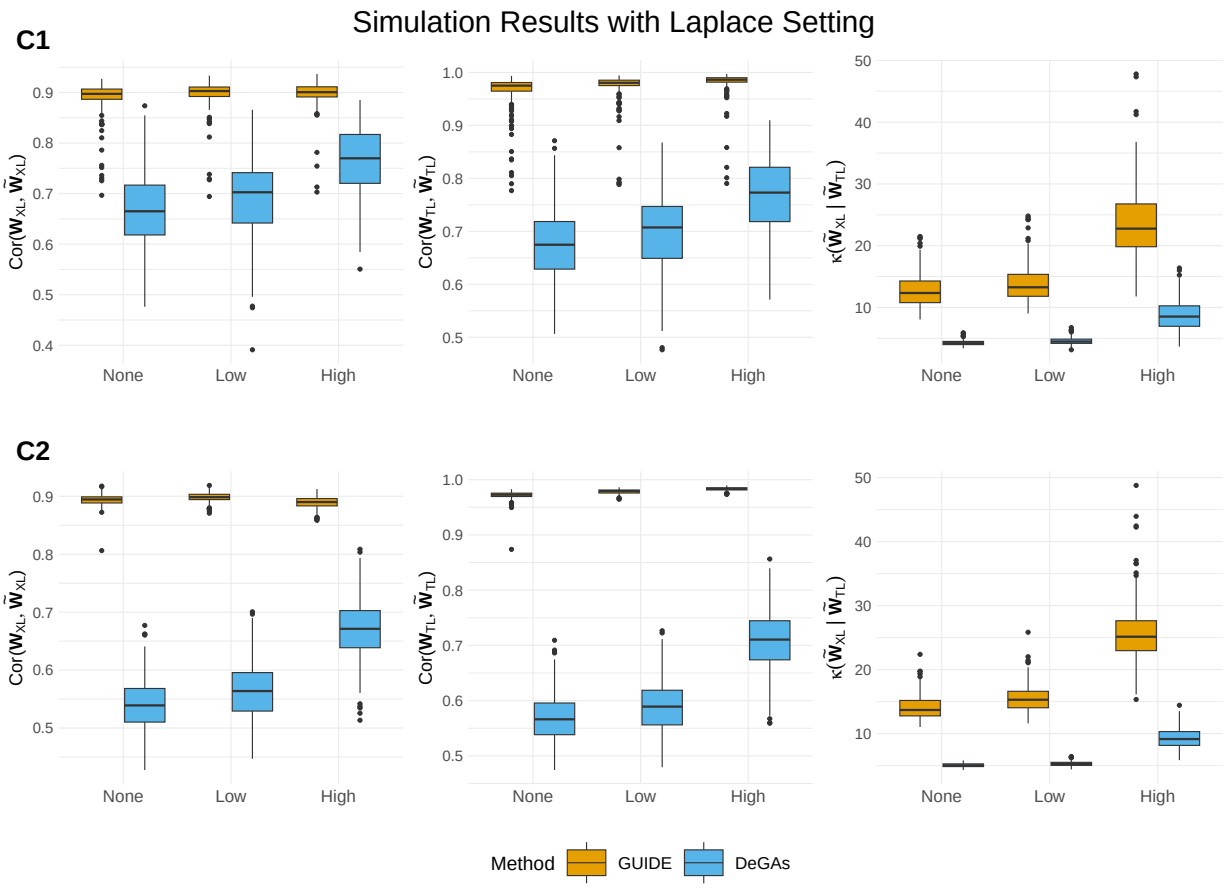

Figure 4.7: Boxplots comparing the performance of GUIDE and DeGAs across simulations where the weights are sampled from the Laplace distribution. Three levels of sparsity are considered: "None", "Low" and "High" (described in Section 3.4.2), shown in the x-axis of each plot. Each row corresponds to a different configuration (**C1**, **C2**) from Table 4.2.

When sparsity is introduced into the genetic architecture, GUIDE becomes more efficient at recovering the components while also yielding more kurtotic solutions. Under low sparsity, GUIDE begins to perform better than DeGAs, though the margin of improvement is not substantial. When sparsity is high the components clearly deviate from Gaussianity, despite being sampled from a Gaussian distribution, as many weights are set to zero, allowing GUIDE to capture them accurately and significantly outperform DeGAs.

Looking at Figure 4.7, we observe that when the weights are sampled from a Laplace distribution, GUIDE estimates the true components very accurately with correlations consistently above 0.8 across both data sizes and sparsity levels. In contrast, DeGAs is not as successful, performing similarly to when the weights are Gaussian. Furthermore, GUIDE solutions again have higher kurtosis than those of DeGAs, as expected. It is also worth noting that because the true weights are now super-Gaussian,

hence naturally more kurtotic, their accurate estimation leads to sparser recovered components compared to the Gaussian case. This is reflected in the higher kurtosis values observed with GUIDE as the weights follow a Laplace distribution.

Focusing on GUIDE, we notice that for both data sizes, the variant weight correlations are generally lower than the trait correlations. This difference arises because, unlike $\mathbf{W}_{TL}$, the variance of $\mathbf{W}_{XL}$ is not constant across variants as it depends on their individual variances, which in turn are determined by their MAF. Variants with lower MAF have lower variance, which in turn leads to greater weight variances, as seen from 3.15. Consequently, variants with lower MAF are more likely to have weights further from zero, behaving as outliers in the overall weight distribution, hence driving the corresponding effect sizes to also be more extreme. At the same time, variants with low MAF have greater standard errors, meaning that when their effect sizes are divided by the standard error, the resulting z-scores will have, on average, values within a normal range that do not reflect the potential outlier weights they originated from. Therefore, the weights of these estimates may not be captured as well, leading to lower overall correlations compared to $\mathbf{W}_{TL}$.

To illustrate this effect, a large genetic architecture is considered where weights are sampled from a Laplace distribution with low sparsity. Z-scores are simulated for 3,000 variants, 150 traits, and 20 components, then GUIDE is applied again with the ICASSO extension for 20 ICA runs, correcting for permutation and sign changes. Figure 4.8 shows plots for the estimated weights against the true ones, separately for the variants and the traits. We can clearly see that while the trait weights are captured almost perfectly, the variant estimates start diverging for more extreme values.

Figure 4.8: Scatter plots comparing simulated variant and trait weights with the estimated ones using GUIDE, each color representing a different component.

In Figure 4.9, one of the components has been isolated to plot its variant weights, highlighting those corresponding to the lowest 5% of MAFs among the 3,000 simulated

variants. The figure confirms that many of the more extreme values, which GUIDE does not capture as accurately as the rest, correspond to variants with the lowest MAFs. We can also see that these points are consistently underestimated in magnitude, leading to a lower overall correlation between the simulated and estimated variant weights. These observations suggest that in practice, the contribution of rare variants in individual components may be systematically underestimated with GUIDE due to the increased uncertainty when estimating their effects across phenotypes.

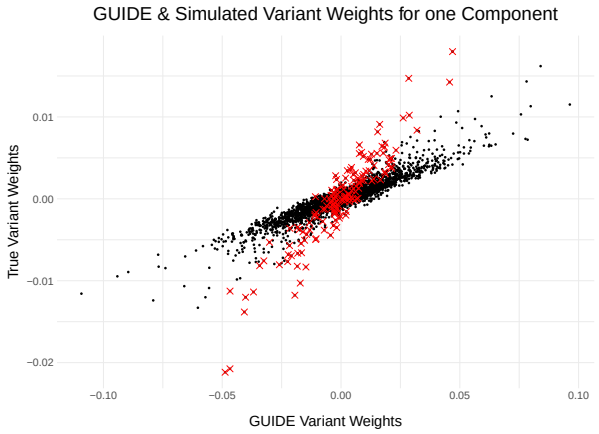

Figure 4.9: Scatter plot comparing the simulated and GUIDE-estimated variant weights for one component. The red cross points correspond to the variants with the lowest 5% of MAFs.

Regarding the overall findings from the simulations in this section, it is worth noting that although ICASSO was used with GUIDE to obtain more stable components, additional tests conducted (but not included in this thesis) showed that the results were nearly identical to those from standard GUIDE, where a single ICA run is considered. This is likely because the data were generated under a true underlying genetic architecture, which ICA is able to capture by converging consistently to the same components across runs. As a result, combining multiple ICA runs with ICASSO has little to no impact, since the components are virtually identical in each run.

4.3 Simulations Under Model Misspecification

Having established GUIDE's efficiency over DeGAs in recovering genetic components when their number is correctly specified, I now compare their performance under model misspecification, i.e. when the number of components is either overestimated or underestimated. As before, z-scores are generated under a polygenic additive model, sampling variant and trait weights from Gaussian and Laplace distributions. For each distribution, a relatively large genetic architecture is considered, consisting of 1,000 variants, 100 phenotypes and 10 components, where weights are sampled under low

and high sparsity. Smaller datasets were also tested, giving similar results but with increased variability.

For every distribution and sparsity level, 200 z-score matrices are generated. For each matrix, both GUIDE with ICASSO for $I = 20$ ICA runs and DeGAs are applied, specifying the number of components to be $\tilde{L} = 2, 3, 4, ..., 30$, covering scenarios of both underspecification ($\tilde{L} < 10$) and overspecification ($\tilde{L} > 10$). The components are then matched in a similar way as to when the correct number of components is known. In the case of underspecification, all $\tilde{L}$ estimated components are matched with $\tilde{L}$ of the true ones so that the pairwise correlations are maximized, thus having $\tilde{L}$ pairs in total. Similarly, when the components are overspecified, all 10 true components are matched with 10 of the estimated ones. To assess how each method performs, the correlations between estimates and ground truth as defined in 4.1 are computed, averaging the correlation of the $\min(\tilde{L}, 10)$ matched components pairs. Figures 4.10 and 4.11 show the results of this simulation.

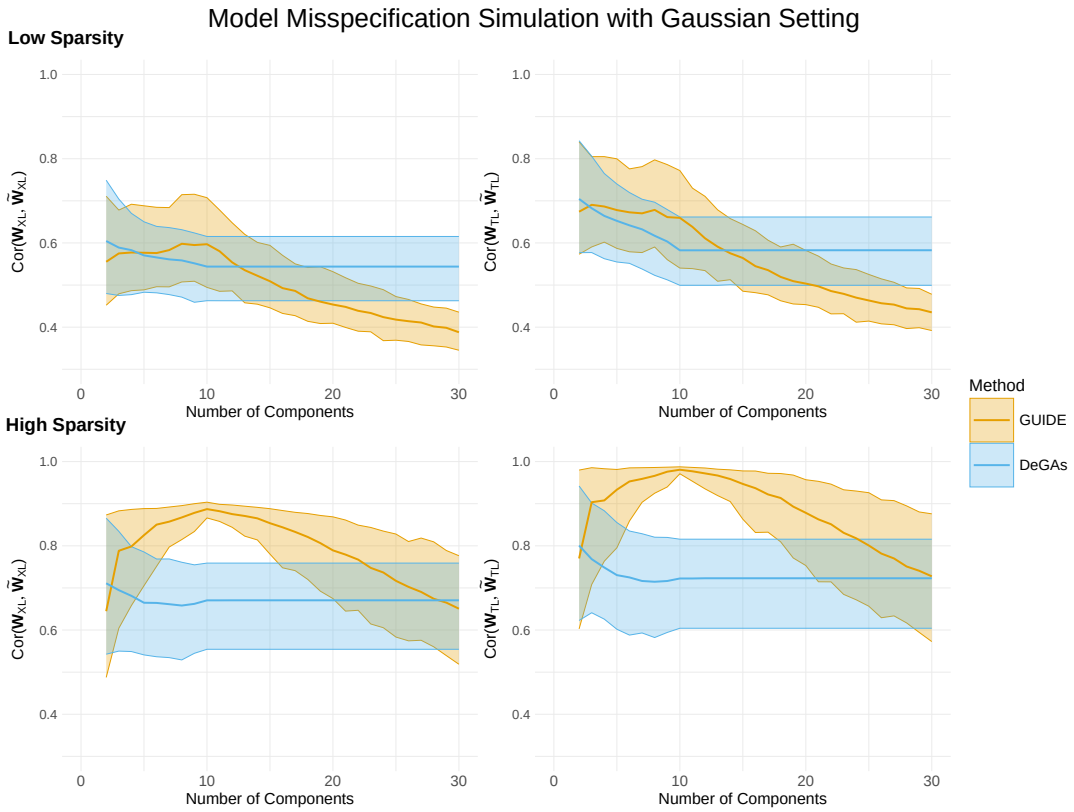

Figure 4.10: Line plots comparing the performance of GUIDE and DeGAs under model misspecification when the weights follow a Gaussian distribution. The top row corresponds to low sparsity in the genetic architecture, while the bottom row represents high sparsity. The x-axis denotes the number of components $\tilde{L}$ used in GUIDE and DeGAs and the y-axis measures the correlation between the estimated and true weights for variants (left column) and traits (right column). The solid lines indicate the median correlation across the 200 generated structures, with the shaded regions representing the 95% quantile range (from 2.5% to 97.5%).

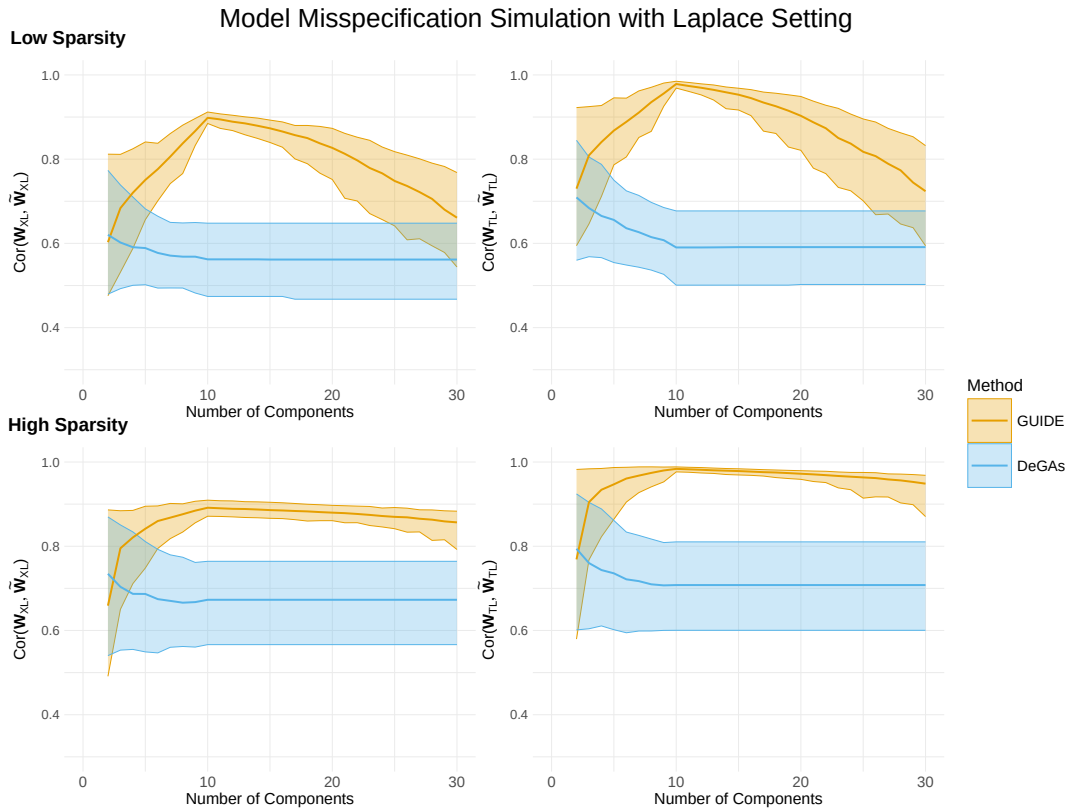

Figure 4.11: Line plots comparing the performance of GUIDE and DeGAs under model misspecification when the weights follow a Laplace distribution. Using the same layout as in Figure 4.10, the correlation of variant and trait weights with the true ones for different values of $\tilde{L}$ is displayed, for low and high sparsity using the median and the 2.5% to 97.5% quantile range.

When the weights are Gaussian with low sparsity, GUIDE and DeGAs perform similarly as the true number of components is underestimated. For $\tilde{L} > 10$, GUIDE's performance gradually declines, eventually falling below that of DeGAs as the number of estimated components significantly exceeds the true number. Under high sparsity, we observe that GUIDE's performance improves as we approach the true number of components, outperforming DeGAs, with both variant and trait correlations being maximized at $\tilde{L} = 10$, the correct estimate. Moving away from the true component number, the uncertainty of GUIDE increases as suggested by the wider quantile ranges, while the overall performance drops.

When the weights are sampled from a Laplace distribution, GUIDE achieves higher component correlations than DeGAs for both variants and traits. Under low sparsity, GUIDE's correlations are maximized when the method is run with the correct number of components, while when the number of components is misspecified, the correlation drops and the uncertainty around the estimates increases. Interestingly, for high weight sparsity, GUIDE's performance is again maximized at $\tilde{L} = 10$, but as the number of components is overestimated, the correlation remains almost stable,

only decreasing gradually. This suggests that GUIDE can still accurately recover the true weights even when their number is significantly overestimated. It is also worth mentioning that in both Gaussian and Laplace settings, GUIDE's correlation of the variant weights to the true ones is again lower than for the trait weights for both sparsity levels, reflecting the aforementioned uncertainty surrounding variants with lower MAFs.

Now, across both figures and all panels, we notice that once the true number of components is reached, DeGAs's median correlation and quantile range stabilize. This behavior is due to the leading components remaining unchanged in SVD, even as additional components are extracted. Since these leading components generally capture more structured patterns and each explains more variance than the subsequent ones, they match better with the true components. Therefore, the same dominant components are consistently selected, irrespective of how many additional components are included.

Another interesting conclusion is about ICA's ability to recover the true components as we overestimate them, which is the main idea of GUIDE's component selection strategy. If this property held in practice, then for every $\tilde{L} > 10$, the true components would remain identifiable, and correlations with the ground truth would remain stable, similar to what is observed with DeGAs. However, our simulations showed this is not generally the case, as correlations tended to decrease when more components were added (except in settings with highly sparse Laplacian weights), suggesting that the theoretical robustness of ICA to overestimation may not hold in practice. In fact, in ICA, overestimation is known to produce spurious results due to arbitrary splitting, hence reducing the accuracy with which true components are recovered [1]. While real datasets do not have a ground truth, these findings suggest that choosing too many components without considering the dimensionality of the data may lead to less meaningful and informative results.

As in the previous section, additional tests were conducted by running GUIDE with a single ICA run and performance was found to be comparable to that achieved using ICASSO. This suggests that under the polygenic additive model simulations, the optimization landscape is simple enough that repeated ICA runs offer no substantial advantage. The utility and importance of ICASSO are therefore explored more thoroughly in Chapter 5 using the T2D dataset, where there is no ground truth for ICA to converge each time to.

4.4 Ranking GUIDE Components with CQI

As discussed in Section 3.2, while DeGAs's components are naturally ranked based on their significance through the singular values, GUIDE's ordering is random as ICA components can only be identified up to a permutation. By extending GUIDE with ICASSO, however, one can utilize information from multiple ICA runs and compute the CQI, a score that quantifies how reliably a component is estimated relative to others. The aim of this section is to investigate whether CQI scores can have a similar interpretational value to that of singular values in terms of prioritizing more important components, compared to redundant noise components.

To assess the capacity of CQI in identifying important components, I apply GUIDE with ICASSO for $I = 20$ ICA runs to the same z-score matrices as before, using $\tilde{L} = 20$ components which are ranked in decreasing order based on their CQI values. At the same time, for each simulated matrix the 10 true components are matched with 10 of the GUIDE-estimated ones to maximize the pairwise correlations between them. The goal is to determine whether the GUIDE matched components, which can be interpreted to be the most significant for a given structure, have higher CQI scores than the unmatched ones. To evaluate this, curves are constructed for each distribution and sparsity level that capture the number of matched components within a subset of estimated components with the k highest CQI scores, for $k = 1, ..., \tilde{L}$. Ideally, we would find that the first 10 CQI-ordered components are the same as the 10 matched ones, indicating that CQI effectively prioritizes the most relevant components. If CQI scores were unrelated to component significance, we would expect the top k components to contain, on average, $\frac{k}{2}$ matched ones, since 10 out of the $\tilde{L} = 20$ total components are significant. This baseline provides a reference for evaluating CQI's performance.

Figure 4.12 captures the results of this analysis for GUIDE across Gaussian and Laplace samplers with low and high sparsity. Looking at the top left panel, we observe that when the genetic structure follows a Gaussian distribution with low sparsity, CQI-based sorting does not provide any information about which components are significant as it does not differ from the average performance of a random ordering. Under high sparsity, CQI scores become informative regarding component significance. Across the 200 datasets, the median outcome is that the top 6 components ranked by CQI are all matched components, Among the top 10 ranked components, the median number of matched components is 8, with 95% of cases falling between 5 and 9.

When the weights follow a Laplace distribution with low sparsity, CQI-based sorting performs similarly to the Gaussian case under high sparsity. Interestingly, when sparsity is high, the median outcome is that all 10 highest CQI scores correspond to matched components, suggesting that CQI can be very efficient in prioritizing more

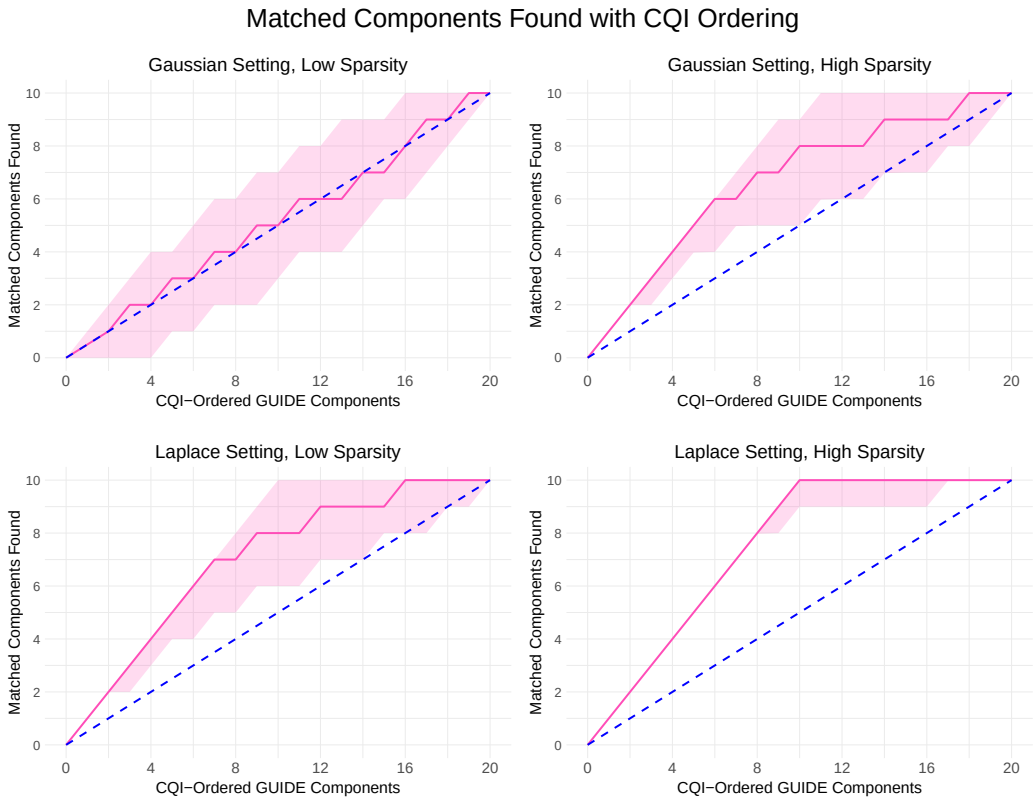

Figure 4.12: Line plots evaluating the effectiveness of CQI-based ordering in identifying the most significant components for GUIDE when the number of components ($\tilde{L} = 20$) is overspecified. The x-axis represents the ordered components according to their CQI score and the y-axis contains the number of matched components that have been observed in every subset of $k \leq \tilde{L}$ sorted components. The solid lines indicate the median number of matched components within each subset of length k across the 200 generated datasets and the shaded regions capture the 95% quantile range (from 2.5% to 97.5%). The blue dashed line represents the case where CQI is not associated with component significance, hence for every k, half of the components correspond to matched ones.

significant components.

Overall, these findings suggest that CQI scores can support a more informed and interpretable analysis when using GUIDE with ICASSO, in terms of ordering the components according to their importance. However, the simulations also demonstrated that higher CQI scores did not always identify more significant components, indicating that it should not be the sole criterion for assessing component relevance. Therefore, I conclude that while CQI offers a useful measure of component reliability and consistency, domain-specific biological understanding of the data remains essential for interpreting and validating the identified components.

5. Application to Type 2 Diabetes Data

In this chapter, I present results from analyzing the T2D dataset using both GUIDE and DeGAs. In the original study that introduced this dataset [29], the authors applied bayesian Non-Negative Matrix Factorization (bNMF) with $L = 12$ components. To enable a direct comparison with their findings, I use the same number of components to initialize GUIDE and DeGAs.

In Section 5.1, the use of ICASSO in GUIDE is motivated by demonstrating that individual ICA runs can become unreliable, often converging to different subsets of components. Section 5.2 presents the GUIDE-derived T2D components and compares them to those obtained with bNMF. In Section 5.3, GUIDE is compared with DeGAs in terms of the amount of overlapping information between the two methods, as well as in terms of sparsity and independence, two properties that can lead to more interpretable components.

5.1 Motivation for ICASSO

In this thesis, a novel extension of the standard GUIDE methodology has been proposed through the incorporation of ICASSO, where the ICA algorithm is run multiple times to obtain more reliable components than those from a single ICA run. To motivate this extension, it is shown here using the T2D dataset how individual ICA runs with different initializations can become unreliable, yielding different subset of components, an issue which can be mitigated with ICASSO.

First, GUIDE is applied on the T2D data $J = 150$ times with $L = 12$ components, producing unmixing matrices $\mathbf{D}_1, \ldots, \mathbf{D}_J$. The ICASSO-extended version of GUIDE is then run the same amount of times, yielding unmixing matrices $\hat{\mathbf{D}}_1, \ldots, \hat{\mathbf{D}}_J$, each derived from $I = 100$ ICA runs. To compare the reliability of component estimation with single ICA runs and ICASSO, an approach similar to that proposed in Section 3.2 for choosing components with GUIDE is employed. Specifically, for each of the two methods, and for each pair of unmixing matrices, the correlations between components

are computed, counting the number of matches with absolute correlation $|r| > 0.95$. Ideally, the estimated components would be identical across runs, differing only by permutation and sign, meaning that each comparison between two unmixing matrices would result in 12 matches.

Figure 5.1 summarizes the results of all $\binom{J}{2}$ comparisons per method, separately for individual ICA runs and ICASSO-based runs. For single ICA runs, we observe that although in many comparisons all 12 components are matched, others show considerably fewer matches, indicating that ICA often converges to different component subsets. In contrast, ICASSO yields significantly more consistent results, with components that are mostly the same across runs, occasionally differing by only one, and rarely by two. This shows that GUIDE with ICASSO is more consistent on the T2D data, justifying its use over the standard GUIDE methodology.

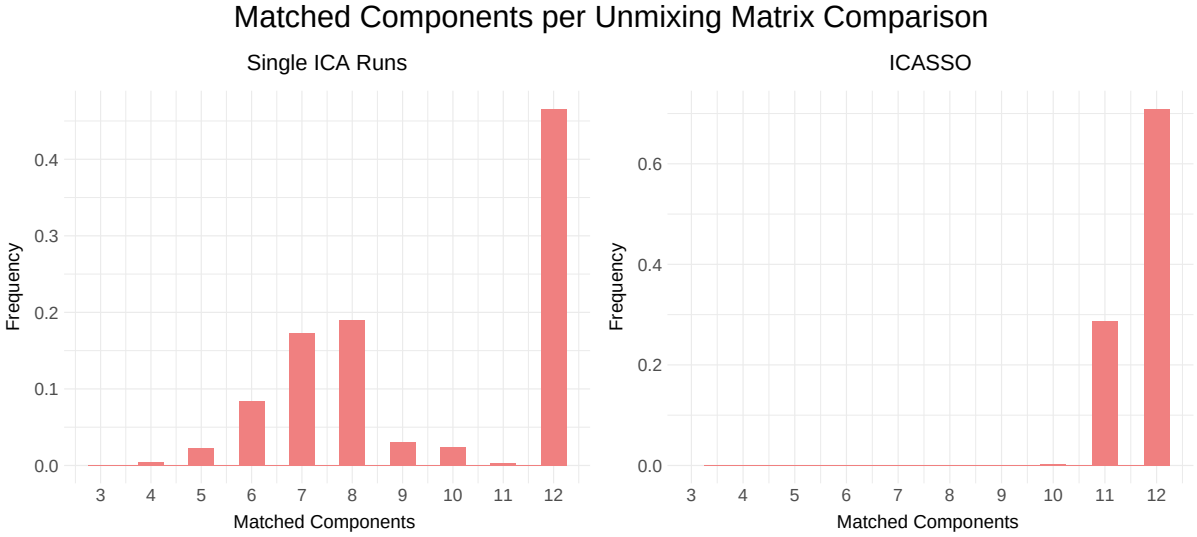

Figure 5.1: Histograms showing the consistency of component estimation in GUIDE using a single ICA run (left) and using ICASSO (right). Each histogram displays the number of components that can be matched across pairs of runs with correlation $|r| > 0.95$.

5.2 GUIDE Clusters

In the previous section, it was demonstrated how extending GUIDE with ICASSO leads to more consistent and thus more reliable component estimation. This section presents the results of applying GUIDE to the T2D dataset, again using 12 components and 100 ICA iterations for ICASSO. For each component, the CQI score is calculated, which is used to rank the components. Components with larger CQI scores appear more consistently across ICA runs and can therefore be considered more reliable in capturing variant-trait associations. The CQI scores are visualized in Figure 5.2, showing higher

values for the top components, followed by a gradual decrease and then a sharper decline for the lower ranked components.

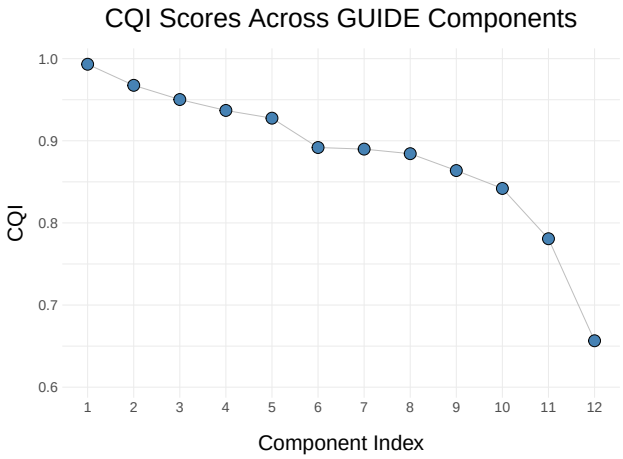

Figure 5.2: CQI scores for GUIDE components on the T2D dataset, shown in decreasing order.

As previously mentioned, the T2D dataset was initially analyzed using bNMF [29], a matrix factorization technique in which a non-negative data matrix is decomposed into two lower-rank, also non-negative matrices. As with GUIDE and DeGAs, these matrices represent variant and trait weights, where each column corresponds to one component. An important difference, however, is that the component weights in bNMF are constrained to be non-negative. This constraint implies that variants and traits can only be positively associated with components, which can limit their biological interpretation. Nonetheless, the data can be preprocessed such that either the variants or the traits (but not both) are allowed to load positively or negatively onto components. In [29], the preprocessing was performed so that trait weights could be of either direction.

In the bNMF analysis, the resulting components were interpreted as clusters of variant and trait associations, shown in Figure 5.3. The GUIDE components can be interpreted in the same way, so that the lth column of the variant and trait weight matrices corresponds to the weights of all variants and traits to the lth cluster. Large absolute weights suggest strong associations with the cluster, whereas near-zero weights suggest little to no contribution. To facilitate a direct comparison with bNMF, the GUIDE components are visualized as clusters of variants and traits, ordered by their CQI scores, presented in Figure 5.4. Following the same approach as in [29], variant IDs are replaced with the name of the gene they belong to, or, if they are located in non-coding regions, with the name of the nearest gene. DeGAs was also applied to the same dataset using the same number of components, and the resulting clusters are displayed in Appendix C.

To interpret the GUIDE clusters and compare them with those identified by

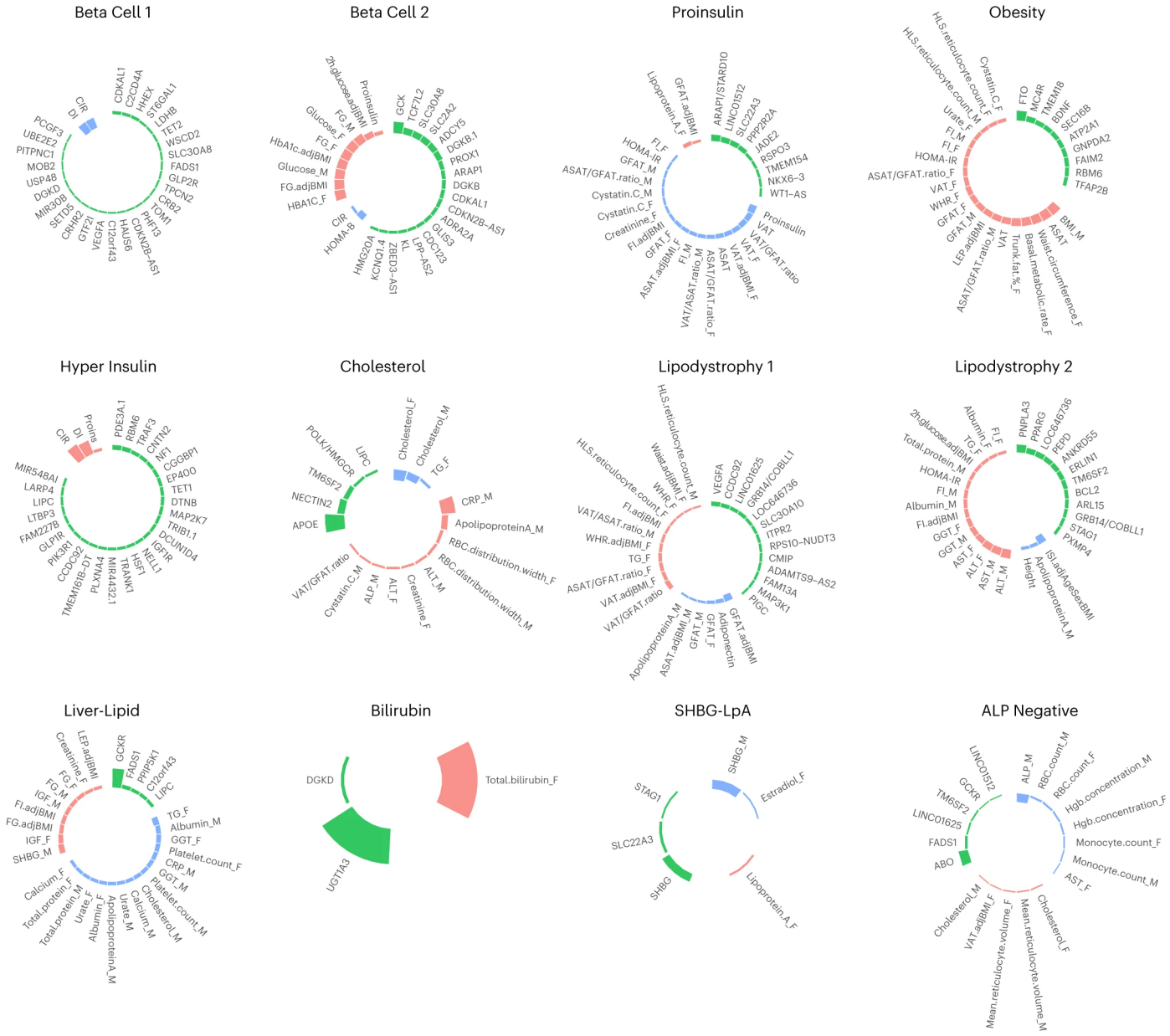

Figure 5.3: Clusters of T2D data from bNMF, presented by Smith et al. [29]. The figure is reproduced with permission from Springer Nature.

bNMF, it is important to keep in mind that ICA components are accurate up to a sign change. This means that while the absolute magnitudes of variant and trait weights indicate the strength of their association with a cluster, the signs themselves are arbitrary and may be flipped across different ICA runs. Nonetheless, the relative structure of the weights, meaning which variants and traits are loaded in the same or opposite directions, remains consistent. If some traits and variants appear with positive weights and others with negative weights in one ICA run, then in another run, all signs will either remain the same or be flipped together, but their relative pattern will be preserved. Therefore, when comparing the clusters with those of bNMF, we

Figure 5.4: GUIDE T2D genetic clusters, ordered by decreasing CQI scores, showing the variants and traits with the highest contribution scores. For each cluster, up to 30 elements (variants and traits) with the largest weights are displayed. Only elements with above-average contribution scores within each component are included. Instead of displaying variant IDs, the names of the genes in which the variants belong to are shown, while if they are located in a non-coding region, the nearest gene is used. Trait labels include the subscripts "_M" and "_F" to denote male and female specific traits respectively. Bars extending outward represent one direction of the weights: green for variants and red for traits. In contrast, purple and blue bars pointing inward represent weights in the opposite direction for variants and traits respectively. Within each direction, elements are ordered by the magnitude of their weights.

focus the relative directions of variants and traits within each cluster, instead of the actual signs of their weights.

A particularly interesting result of this application of GUIDE with ICASSO is that most of the identified T2D clusters closely resemble those found using bNMF. For instance, GUIDE Cluster 1 has a direct correspondence to bNMF cluster "Cholesterol". Some of the top-weighted genes, specifically APOE, NECTIN2, TM6SF2 and POLK appear in both clusters and in the same order of weight magnitude. Among the traits with loadings of the same sign, the top two (CRP_M and ApolipoproteinA_M) also match, with further overlap observed among subsequent ones. As previously discussed, bNMF allows loadings from both directions only for traits. Looking at the opposite direction, GUIDE again recovers traits overlapping with bNMF, specifically Cholesterol_F, Cholesterol_M and TG_F. Regarding the opposite direction for the genes, which bNMF does not capture, GUIDE identifies a distinct set of significant genes, the top of which are LPL, C12orf43, and MAP3K1_1.

By comparing the rest of GUIDE and bNMF clusters in the same way, we can see that GUIDE Cluster 2 aligns closely with the bNMF cluster labeled "Beta Cell 2", sharing overlapping sets of genes and traits with consistent relative weight directions. Similarly, GUIDE Cluster 3 corresponds to "Beta Cell 1" in bNMF, while GUIDE Cluster 4, which is considerably sparser, containing only three variants and one trait, can be matched with the bNMF cluster "Bilirubin".

GUIDE Cluster 5 does not have a direct match among the bNMF clusters, however it shares some variants and traits with bNMF cluster "Proinsulin". Specifically, the genes ARAP1 and NKX6-3 appear with opposite signs from traits PROINS (proinsulin) and Lipoprotein.A_F in GUIDE, which is also the case in this bNMF cluster. GUIDE Cluster 9 also shares some similarities with the same bNMF cluster, lacking a match with any other cluster. Specifically, the genes PPP2R2A_1 and WT1-AS are loaded with the opposite sign of traits vatGfatRatio, vat, vatAsatRatio_M, vatadjbmi_F and vat_F, all related to visceral adipose tissue, a pattern which is also found in bNMF cluster "Proinsulin". For both GUIDE Clusters 5 and 9, it is likely that additional variants and traits are shared between them and bNMF's "Proinsulin", but do not appear among the top 30 elements visualized in the cluster plots.

GUIDE Cluster 6 can be matched to bNMF cluster "ALP Negative", with most top variants and traits overlapping and loaded in the same relative direction. The same correspondence can be found between GUIDE Cluster 7 and bNMF cluster "Liver-Lipid", as well as GUIDE Cluster 8 with bNMF cluster "Lipodystrophy 2". Furthermore, GUIDE Cluster 11 captures the same top variants and traits as bNMF cluster "Obesity", while GUIDE Cluster 12 also matches closely with bNMF cluster "Lipodystrophy 1".

Unlike all other GUIDE clusters, GUIDE Cluster 10 does not align closely with any specific bNMF cluster. Although it contains traits associated with abdominal subcutaneous adipose tissue (ASAT), which appear across bNMF clusters "Proinsulin", "Obesity", and "Lipodystrophy 1", the set of contributing variants differs from all these, suggesting a potentially unique biological signal not isolated by bNMF.

Overall, the visual inspection of the clusters revealed that there is a clear one-to-one correspondence between most GUIDE clusters and those identified by bNMF. The only exceptions were GUIDE Clusters 5 and 9, which show partial overlap with the bNMF cluster "Proinsulin" and GUIDE Cluster 10, which appears distinct from all bNMF counterparts. This result is significant for two reasons. First, it confirms GUIDE's effectiveness in uncovering meaningful patterns of variant-trait associations, while also validating the novel extension with ICASSO for improving component stability. Second, the fact that two different methodologies, based on separate assumptions and optimization strategies, converged to such similar results suggests that the identified clusters are more likely to reflect biologically meaningful signals related to T2D, potentially representing important underlying pathways involved in the disease.

5.3 Comparison of GUIDE & DeGAs Components

The main focus of this section is to compare the GUIDE and DeGAs T2D components in terms of sparsity and independence, properties that can lead to more interpretable components. Prior to this comparison, given that GUIDE is informed by the DeGAs weights, which are rotated through the use of ICA, it is of interest to quantify how much this rotation alters the information conveyed, compared to the initial SVD weights in DeGAs. For this purpose, the Jaccard indices are computed between the GUIDE and DeGAs components, which quantify the degree of information overlap between each method's components. They are computed separately for variants and traits, as described in Section 3.3, considering only significant weights, i.e. those with above-average contribution scores, setting the rest to be exactly 0.

The resulting matrices of Jaccard indices are presented as heatmaps in Figure 5.5, where the GUIDE and DeGAs components appear in decreasing order of CQI scores and singular values respectively. We observe that although there is some overlap of information (e.g. between GUIDE component 11 and DeGAs component 2, GUIDE component 7 and DeGAs component 4, or GUIDE component 10 and DeGAs component 10), in general the Jaccard indices are low, suggesting that the patterns captured by each methods are mostly distinct. Also, the overlap between GUIDE and DeGAs components for the traits is smaller than for the variants.

Now, to assess the independence of each of GUIDE and DeGAs produced com-

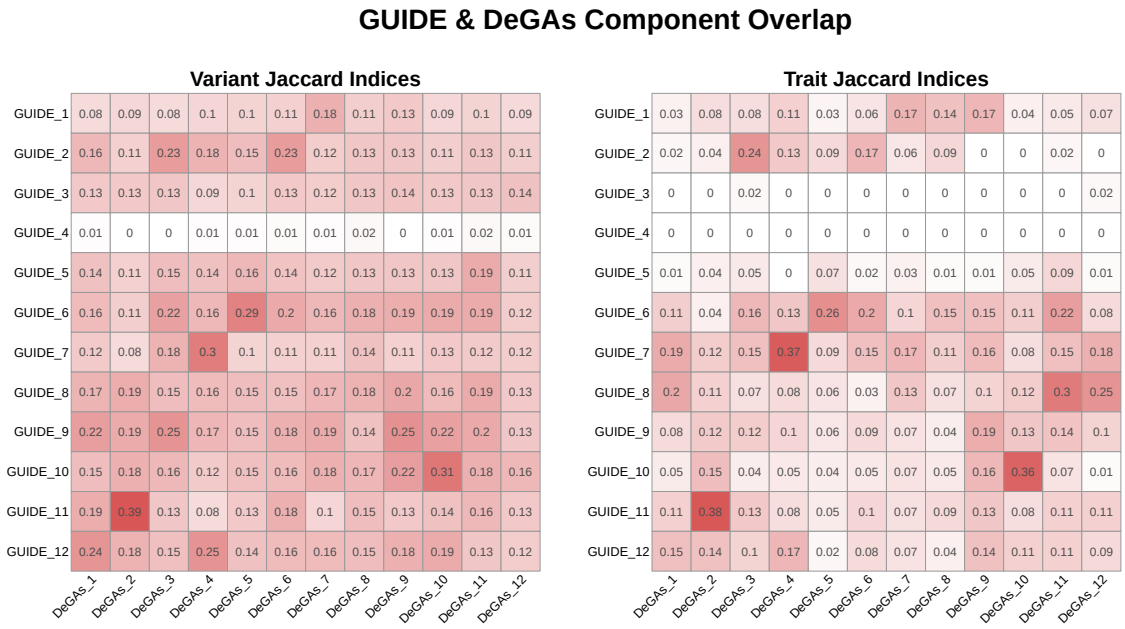

Figure 5.5: Heatmaps of Jaccard indices between GUIDE and DeGAs components, computed separately for variants (left) and traits (right). The GUIDE components are ordered by their CQI scores, while the DeGAs components are ordered by their singular values.

ponents, pairwise Jaccard indices are computed between components, separately for variants and traits, used as a measure of overlap in their significant weights. Figures 5.6 and 5.7 show the Jaccard index heatmaps for GUIDE and DeGAs respectively, where the components are again ordered according to their significance, as measured by the CQI and singular values respectively. It is clear that the GUIDE components are more independent as the pairwise Jaccard indices are lower than for their DeGAs counterparts, with most indices remaining below 0.2, while higher overlap than that is observed only in a few component pairs. In contrast, DeGAs shows consistently higher Jaccard indices across component pairs, suggesting greater information overlap between its components.

To evaluate the sparsity of the GUIDE and DeGAs components, their kurtosis values are reported as well as the number of variants and traits loaded onto each component. A variant or trait is considered loaded if its contribution score exceeds the average, which in this case corresponds to thresholds of $\frac{1}{650}$ for variants and $\frac{1}{110}$ for traits. These metrics are visualized in Figure 5.8.

We observe that GUIDE's components generally have higher kurtosis than their DeGAs counterparts, exceeding 25 for half of the components, a value not reached by DeGAs. The highest kurtosis value for GUIDE exceeds 350, corresponding to the very sparse Cluster 4 presented in Figure 5.4 which only loads 1 trait and 3 variants with significant weights.

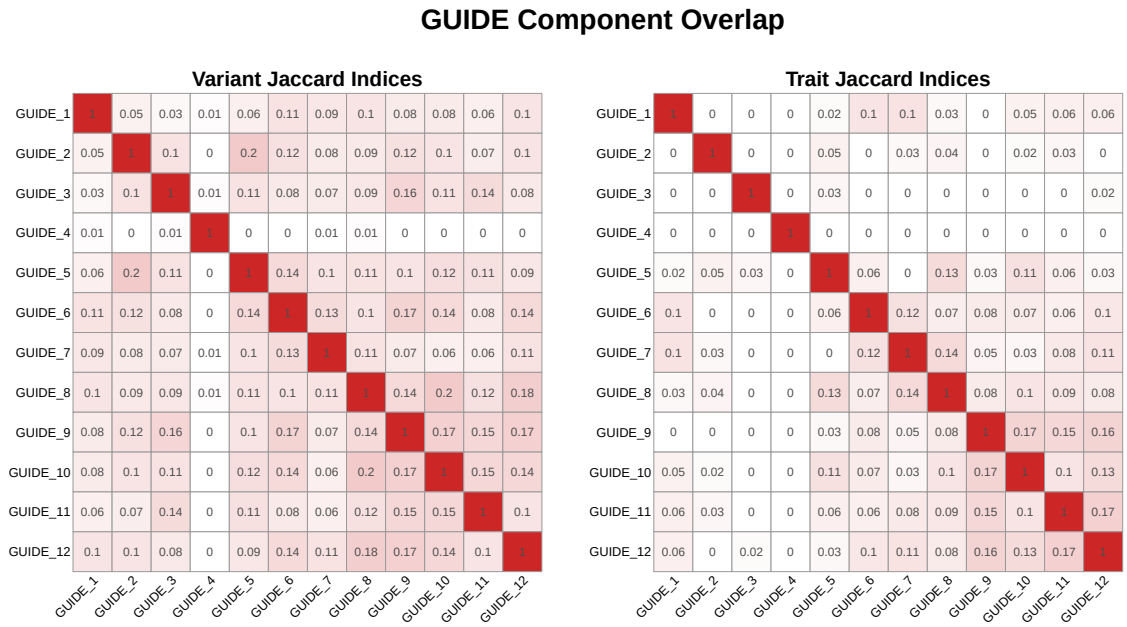

Figure 5.6: Heatmaps of Jaccard indices between GUIDE components, computed separately for variants (left) and traits (right).

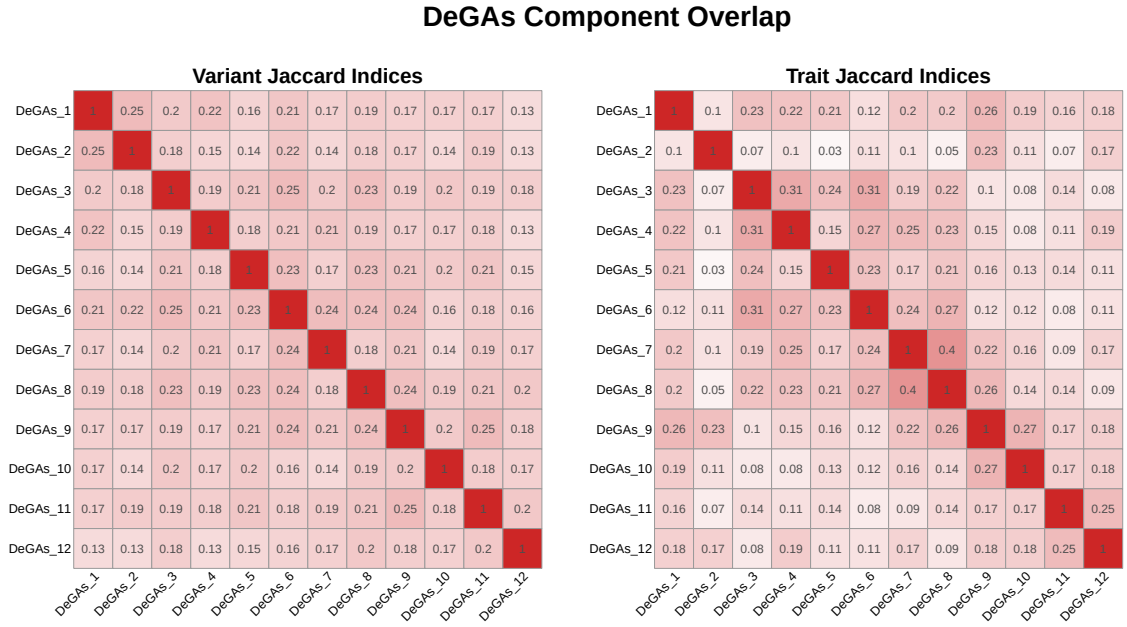

Figure 5.7: Heatmaps of Jaccard indices between DeGAs components, computed separately for variants (left) and traits (right).

In terms of variant loadings, GUIDE generally assigns fewer variants per component, but with greater variability across components. On average, each component loads 109 variants, but the number can range from fewer than 10 to over 180. In contrast, DeGAs components load an average of 146 variants, with a range of 110 to 170 per component. Interestingly, while GUIDE has fewer variants loaded on each compo-

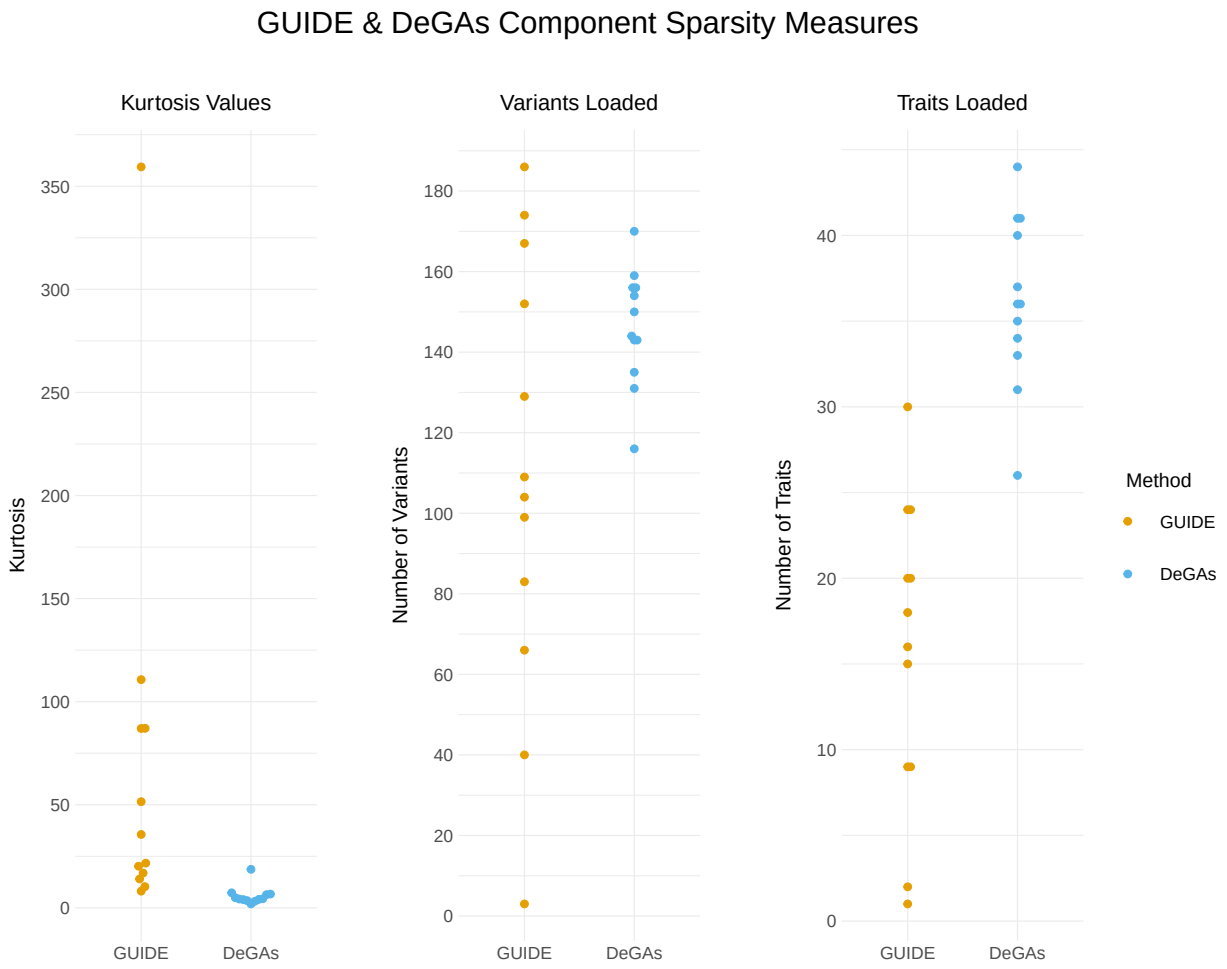

Figure 5.8: Comparison of GUIDE and DeGAs in terms of different sparsity measures, specifically kurtosis (left), number of variants loaded per component (middle) and number of traits loaded per component (right). A variant or trait is considered to be loaded to a component when its contribution score is above the uniform baseline, as defined in Section 3.3.

nent on average, it captures nearly as many total variant associations as DeGAs: out of 650 T2D variants, 537 are loaded in at least one GUIDE component, compared to 542 for DeGAs.

Similar conclusions can be drawn for the trait loadings, where GUIDE components are again sparser. No more than 30 traits are assigned to any individual GUIDE component, with an average of 16 traits per component. In contrast, nearly all DeGAs components include at least 30 traits, averaging 36 per component. The total number of T2D traits appearing in at least one component is 90 for GUIDE and 106 for DeGAs, out of 110 in total. Hence GUIDE captures slightly fewer traits than DeGAs while loading on average less than half traits on each component.

Now, apart from having sparse components, it is also desirable that a small subset of variants or traits accounts for most of the variance within each component, so that

the signal is more concentrated and thus potentially more interpretable. To quantify this, the contribution scores are used, which, as shown in Section 3.3, sum to 1 across both variants and traits for each component in GUIDE and DeGAs. Each score can then be interpreted as the proportion of variance explained by a given variant or trait. To assess how concentrated the signals are within GUIDE and DeGAs components, the cumulative contribution scores for variants and traits are computed. To compute cumulative contribution scores for each component, the individual contribution scores are sorted in descending order and their cumulative sum is calculated, which becomes 1 when all variant/trait contributions are included.

Figure 5.9 visualizes the cumulative contribution scores for the variants and the traits. While for some GUIDE components, the variant contribution scores accumulate faster, for others the progression is similar to that of DeGAs, suggesting comparable levels of signal concentration across the two methods. On the other hand, the traits in GUIDE provide more concentrated signals per component than in DeGAs, as indicated from the faster accumulation of contribution scores.

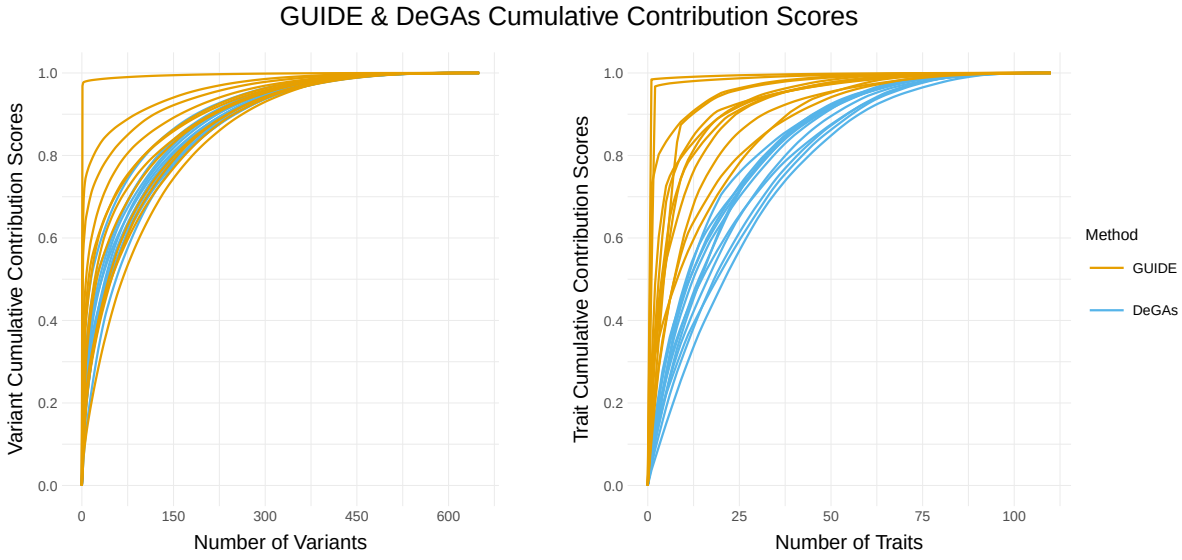

Figure 5.9: Cumulative Contribution Scores of GUIDE and DeGAs components, computed separately for variants (left) and traits (right). The x-axis contains the number of variants and traits respectively and the y-axis captures the cumulative contribution scores. Each curve represents one component.

In summary, although the components of GUIDE and DeGAs only differ by a rotation operation, estimated with ICA, it was found that the information conveyed by each method, as measured from the Jaccard index, differs significantly. At the same time, in terms of structure, GUIDE yields sparser and more independent components with higher kurtosis and less overlap. It also loads fewer variants and traits per component on average, in most cases with more concentrated signals within each

component.

Overall, these findings confirm the results reported in the study introducing GUIDE [23], where, as in this work, a comprehensive comparison with DeGAs was made in terms of sparsity and independence. In this application with the T2D data, these properties lead to the conclusion that GUIDE yields more distinct and concentrated representations of T2D-associated patterns, potentially offering more interpretable and biologically meaningful insights into the genetic architecture of the disease.

6. Discussion

In this thesis, the challenges posed by polygenicity and pleiotropy in the interpretation of GWAS results were addressed through methods for decomposing variant-trait associations. Two approaches, GUIDE [23] and DeGAs [32], were implemented and presented, both of which aim to identify latent components that mediate these associations. To improve the robustness of GUIDE, a novel extension incorporating the ICASSO framework [10] was proposed to account for the potential unreliability of individual ICA runs.

The performance of both methods was evaluated across a wide range of simulated genetic architectures. Specifically, simulations of block structured data were conducted, as well as simulations based on polygenic additive structures. For the latter, variant and trait weights to each component were sampled from either Gaussian or Laplace distributions, while different levels of sparsity were considered. In addition to the simulations, the methods were applied to a type 2 diabetes (T2D) GWAS dataset from [29], and the results were assessed in terms of their consistency with previous analyses and the interpretability of the extracted components.

Through the simulations, I found that the method proposed in [23] for estimating the number of components of a genetic architecture with GUIDE becomes inconsistent and unreliable even in simple scenarios. Moreover, in polygenic additive structure simulations with a known number of components, GUIDE outperformed DeGAs in terms of recovering the true weights when these were either sampled either from a Laplace distribution or a Gaussian with high sparsity. In contrast, when the sampling distribution was Gaussian with little to no sparsity, GUIDE and DeGAs were found to perform similarly. This outcome was expected, as ICA cannot estimate purely Gaussian components [15], hence when the weights are not sparse enough to clearly deviate from Gaussianity, the ICA part of GUIDE does not provide any additional benefit over SVD in DeGAs regarding their estimation.

When the number of components was misspecified in the models, GUIDE again outperformed DeGAs under Laplacian weights, or highly sparse Gaussian weights. As the number of components was overestimated, the performance of DeGAs remained unchanged. On the other hand, GUIDE's performance across all simulations deterio-

rated as the number of components was increasingly overestimated, falling below that of DeGAs when the simulated weights were Gaussian with low sparsity. This deterioration is consistent with known limitations of ICA, where overestimating the number of components can lead to arbitrary splitting [1], therefore reducing accuracy of recovering the true components. In addition to the above comparisons, the CQI scores of GUIDE's components, provided through the use of ICASSO, were assessed in terms of prioritizing more important components and it was found that they can indeed be informative of component significance.

In the application to the T2D dataset, it was demonstrated that GUIDE can become unstable when applied with individual ICA runs, often leading to inconsistent subsets of components across runs. Since each subset implies a different interpretation, the results from individual runs were less reliable. This issue was addressed with the proposed ICASSO extension, which improved the reliability of component estimation, yielding more stable and reproducible results. Specifically, with ICASSO it was found that in most cases, the T2D components were identical across GUIDE runs, occasionally differing by one component and very rarely by two. For that reason, further analysis of the T2D data was conducted with ICASSO.

When comparing the results of GUIDE to those of the original analysis with bNMF [29], an one-to-one match was found between most components of the two methods. An important difference in these matches, however, involved the direction of the variant loadings. While GUIDE can output both positive and negative weights, bNMF is constrained to provide a non-negative decomposition, allowing only positive associations between variants and traits. In [29], this constraint was relaxed by preprocessing the T2D data in a way that only the trait weights could imply negative associations. Because the limitation regarding the variant weights remained, GUIDE was able to uncover additional structure in the components matched with bNMF by capturing both positive and negative variant loadings, whereas bNMF captured variant associations in only one direction. Overall, the re-discovered components with GUIDE not only confirm the findings reported in [29], but also demonstrate GUIDE's capacity to extract meaningful and interpretable components, while revealing additional information regarding variant-trait associations relative to bNMF.

In addition to the comparison with bNMF, GUIDE's T2D components were also compared with those from DeGAs, with GUIDE weights showing greater sparsity and independence. This suggests that GUIDE may lead to more interpretable components, where patterns of associations between variants and traits are captured in a more distinct and concentrated way. These findings are consistent with those reported in the original GUIDE study [23], indicating that the benefits of GUIDE in terms of interpretability are preserved across different datasets and applications.

More generally, given GUIDE's effectiveness in decomposing both simulated and real data, it is important to recognize its limitations, arising from the assumptions made with the use of ICA. Specifically, GUIDE relies on the assumption that the underlying variant-trait associations are mediated by a set of statistically independent, non-Gaussian latent components. However, this assumption may not hold for all traits. For instance, if the biological mechanisms underlying a given trait are highly co-dependent or overlapping, the independence assumption is violated, making GUIDE less suitable for such analyses. Additionally, certain biological signals may be better characterized by Gaussian distributions, which ICA is unable to recover. In such cases, applying GUIDE could lead to spurious or misleading associations, which should be taken into consideration when evaluating its appropriateness. Given this limitation, a promising topic for further research would be to develop diagnostic tests capable of identifying potential violations of the ICA assumptions in GUIDE.

Regarding the novel integration of ICASSO with GUIDE, this thesis demonstrated its ability to produce more reliable components compared to standard ICA. In addition to improving stability, ICASSO has the potential for addressing the challenging task of selecting the number of components, a decision that, in this work, was made heuristically based on the bNMF analysis. For example, one could run GUIDE with greater number of components, and then choose a subset of top L based on their CQI scores. This would allow GUIDE to consider additional components from the SVD of the summary statistics matrix, which may provide useful information that would have otherwise been discarded. Alternatively, GUIDE could be run across a range of component numbers, with ICASSO clustering quality used to identify an optimal value. The utility of both of approaches has already been demonstrated in [11], where fMRI and MEG data were analyzed with ICASSO, showing promising results. Therefore, future work could explore how these strategies can be incorporated within the GUIDE framework as they could potentially offer a principled method for selecting the number of components.

Finally, the comparison with bNMF on the T2D data also highlighted promising directions for future research. While the current analysis relied on visual inspection of top variants and traits across components, extending this to a quantitative assessment of component overlap would provide a more rigorous comparison. Additionally, it would be of interest to investigate the criteria with which bNMF prioritizes a single direction for variant loadings, in contrast to GUIDE's ability to capture bidirectional associations. Such comparisons could be carried out in the context of a broader study, involving both simulated and real datasets to understand the differences and similarities between the two methods.

Bibliography

[1] C. F. Beckmann and S. M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE transactions on medical imaging*, 23(2):137–152, 2004.

[2] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[3] E. A. Boyle, Y. I. Li, and J. K. Pritchard. An expanded view of complex traits: from polygenic to omnigenic. *Cell*, 169(7):1177–1186, 2017.

[4] W. S. Bush and J. H. Moore. Chapter 11: Genome-wide association studies. *PLoS computational biology*, 8(12):e1002822, 2012.

[5] D. I. Chasman, F. Giulianini, O. V. Demler, and M. S. Udler. Pleiotropy-based decomposition of genetic risk scores: association and interaction analysis for type 2 diabetes and CAD. *The American Journal of Human Genetics*, 106(5):646–658, 2020.

[6] E. H. Z. Chua, S. Yasar, and N. Harmston. The importance of considering regulatory domains in genome-wide analyses–the nearest gene is often wrong! *Biology Open*, 11(4):bio059091, 2022.

[7] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[8] C. A. De Leeuw, J. M. Mooij, T. Heskes, and D. Posthuma. MAGMA: generalized gene-set analysis of GWAS data. *PLoS computational biology*, 11(4):e1004219, 2015.

[9] G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU press, 2013.

[10] J. Himberg and A. Hyvarinen. Icasso: software for investigating the reliability of ica estimates by clustering and visualization. In *2003 IEEE XIII workshop on neural networks for signal processing (IEEE cat. No. 03TH8718)*, pages 259–268. IEEE, 2003.

[11] J. Himberg, A. Hyvärinen, and F. Esposito. Validating the independent components of neuroimaging time series via clustering and visualization. *Neuroimage*, 22(3):1214–1222, 2004.

[12] P. J. Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

[13] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in neural information processing systems*, 10, 1997.

[14] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[15] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley Sons, Ltd, 2001.

[16] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[17] International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–1320, 2005.

[18] I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[19] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

[20] A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

[21] H. Kim, K. E. Westerman, K. Smith, J. Chiou, J. B. Cole, T. Majarian, M. von Grotthuss, S. H. Kwak, J. Kim, J. M. Mercader, et al. High-throughput genetic clustering of type 2 diabetes loci reveals heterogeneous mechanistic pathways of metabolic disease. *Diabetologia*, 66(3):495–507, 2023.

[22] W. S. Klug, M. R. Cummings, C. A. Spencer, M. A. Palladino, and D. Killian. *Concepts of genetics*. Number Ed. 11. Prentice Hall Upper Saddle River, NJ, 2016.

[23] D. Lazarev, G. Chau, A. Bloemendal, C. Churchhouse, and B. M. Neale. GUIDE deconstructs genetic architectures using association studies. *bioRxiv*, 2024.

[24] A. R. Omdahl, J. S. Weinstock, R. Keener, S. B. Chhetri, M. Arvanitis, and A. Battle. Sparse matrix factorization robust to sample sharing across GWAS reveals interpretable genetic components. *bioRxiv*, pages 2024–11, 2024.

[25] A. Papoulis and U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1221 Avenue of the Americas, New York, 4th edition, 2002.

[26] T. H. Pers, J. M. Karjalainen, Y. Chan, H.-J. Westra, A. R. Wood, J. Yang, J. C. Lui, S. Vedantam, S. Gustafsson, T. Esko, et al. Biological interpretation of genome-wide association studies using predicted gene functions. *Nature communications*, 6(1):5890, 2015.

[27] D. T. Pham, P. Garrat, and C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774, 1992.

[28] S. Sakaue, M. Kanai, Y. Tanigawa, J. Karjalainen, M. Kurki, S. Koshiba, A. Narita, T. Konuma, K. Yamamoto, M. Akiyama, et al. A cross-population atlas of genetic associations for 220 human phenotypes. *Nature genetics*, 53(10):1415–1424, 2021.

[29] K. Smith, A. J. Deutsch, C. McGrail, H. Kim, S. Hsu, A. Huerta-Chagoya, R. Mandla, P. H. Schroeder, K. E. Westerman, L. Szczerbinski, et al. Multi-ancestry polygenic mechanisms of type 2 diabetes. *Nature medicine*, 30(4):1065–1074, 2024.

[30] Y. G. Tak and P. J. Farnham. Making sense of GWAS: using epigenomics and genome engineering to understand the functional relevance of snps in non-coding regions of the human genome. *Epigenetics & chromatin*, 8:1–18, 2015.

[31] V. Tam, N. Patel, M. Turcotte, Y. Bossé, G. Paré, and D. Meyre. Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics*, 20(8):467–484, 2019.

[32] Y. Tanigawa, J. Li, J. M. Justesen, H. Horn, M. Aguirre, C. DeBoever, C. Chang, B. Narasimhan, K. Lage, T. Hastie, et al. Components of genetic associations across 2,138 phenotypes in the UK biobank highlight adipocyte biology. *Nature communications*, 10(1):4064, 2019.

[33] M. S. Udler, J. Kim, M. von Grotthuss, S. Bonàs-Guarch, J. B. Cole, J. Chiou, C. D. Anderson on behalf of METASTROKE and the ISGC, M. Boehnke, M. Laakso, G. Atzmon, et al. Type 2 diabetes genetic loci informed by multi-trait associations point to disease mechanisms and subtypes: a soft clustering analysis. *PLoS medicine*, 15(9):e1002654, 2018.

[34] E. Uffelmann, Q. Q. Huang, N. S. Munung, J. De Vries, Y. Okada, A. R. Martin, H. C. Martin, T. Lappalainen, and D. Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, 2021.

[35] P. M. Visscher, N. R. Wray, Q. Zhang, P. Sklar, M. I. McCarthy, M. A. Brown, and J. Yang. 10 years of GWAS discovery: biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

[36] P. H. Westfall. Kurtosis as peakedness, 1905–2014. RIP. *The American Statistician*, 68(3):191–195, 2014.

[37] Z. Zhang, J. Jung, A. Kim, N. Suboc, S. Gazal, and N. Mancuso. A scalable approach to characterize pleiotropy across thousands of human diseases and complex traits using GWAS summary statistics. *The American Journal of Human Genetics*, 110(11):1863–1874, 2023.

Appendix A. The FastICA Algorithm

To extract one independent component, we want to maximize the non-Gaussianity of $\mathbf{d}^T\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{n\times K}$ is a mean-centered and whitened data matrix consisting of n mixtures and K samples and $\mathbf{d}$ is a vector of length n. This is achieved by maximizing $J(\mathbf{d}^T\mathbf{X})$ through its approximation in Equation 2.16. Starting from a random vector $\mathbf{d}$, the FastICA algorithm iteratively updates $\mathbf{d}$ as follows until convergence:

1. $\mathbf{d} \leftarrow \mathbf{X}G'(\mathbf{X}^T\mathbf{d}) - \mathbf{d}G''(\mathbf{d}^T\mathbf{X})\mathbf{1}_K$
2. $\mathbf{d} \leftarrow \frac{\mathbf{d}}{||\mathbf{d}||}$

Here, $\mathbf{1}_K$ is a unit vector of length K, G is the contrast function used to approximate the negentropy, while G' and G'' are the first and second derivatives of G. The scaling of $\mathbf{d}$ in step 2 is to satisfy the assumption that the independent components have unit variance.

The above algorithm extracts one of the independent components. Estimating all n independent components requires maximizing the non-Gaussianity of $\mathbf{d}_1^T\mathbf{X}, \ldots, \mathbf{d}_n^T\mathbf{X}$ while ensuring that each unmixing vector converges to a different local optimum. One way to achieve this is to run the single-component iteration algorithm for all components separately, decorrelating the next weight vector $\mathbf{d}_{c+1}$ from all the previous estimated ones $\mathbf{d}_1, \ldots, \mathbf{d}_c$ after each iteration. For that, starting from a random initialization, the following steps are repeated until convergence:

1. $\mathbf{d}_{c+1} \leftarrow \mathbf{X}G'(\mathbf{X}^T\mathbf{d}_{c+1}) - \mathbf{d}_{c+1}G''(\mathbf{d}_{c+1}^T\mathbf{X})\mathbf{1}_K$
2. $\mathbf{d}_{c+1} \leftarrow \frac{\mathbf{d}_{c+1}}{||\mathbf{d}_{c+1}||}$
3. $\mathbf{d}_{c+1} \leftarrow \mathbf{d}_{c+1} - \sum_{j=1}^{c}(\mathbf{d}_{c+1}^T\mathbf{d}_j)\mathbf{d}_j$
4. $\mathbf{d}_{c+1} \leftarrow \frac{\mathbf{d}_{c+1}}{||\mathbf{d}_{c+1}||}$

The first two steps of the iteration are as in the single-unit case. Step 3 uses the Gram-Schmidt process to decorrelate $\mathbf{d}_{c+1}$ from $\mathbf{d}_1, \ldots, \mathbf{d}_c$, before normalizing the weight vector again.

An alternative way to obtain the independent components is to compute them in parallel and decorrelate them all at once after each iteration. For that, we initialize $\mathbf{d}_1, \ldots, \mathbf{d}_n$ and for each we run one iteration of the single-unit FastICA algorithm. Then, for $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]^T$ we perform the following update:

$$\mathbf{D} \leftarrow (\mathbf{D}\mathbf{D}^T)^{-\frac{1}{2}}\mathbf{D}, \tag{A.1}$$

repeating until all weight vectors have converged. The matrix $(\mathbf{D}\mathbf{D}^T)^{-\frac{1}{2}}$ can be obtained using the eigendecomposition $\mathbf{D}\mathbf{D}^T = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ as $(\mathbf{D}\mathbf{D}^T)^{-\frac{1}{2}} = \mathbf{Q}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T$.

Appendix B. Whitening for GUIDE

As mentioned in 2.2.4, to learn a whitening transformation for the ICA data we can use the empirical covariance formula $\text{Cov}(\mathbf{G}) = (h_{ij})$ with

$$h_{ij} = \frac{1}{n+p-1} \sum_{k=1}^{n+p} (g_{ik} - \bar{\mathbf{g}}_i)(g_{jk} - \bar{\mathbf{g}}_j), \tag{B.1}$$

where $\bar{\mathbf{g}}_m$ is the mean of the mth row of $\mathbf{G}$. Since $\mathbf{G} = (\mathbf{U}_L^T | \mathbf{V}_L^T)$, we have that $\mathbf{g}_m = (\mathbf{u}_m | \mathbf{v}_m)$, where $\mathbf{u}_m$ and $\mathbf{v}_m$ are the mth columns of $\mathbf{U}_L$ and $\mathbf{V}_L$ respectively. Because both the rows and columns of $\mathbf{W}$ are mean-centered, the columns of $\mathbf{U}_L$ and $\mathbf{V}_L$ are also mean-centered, meaning that $\bar{\mathbf{u}}_m = \bar{\mathbf{v}}_m = 0$. Therefore, $\bar{\mathbf{g}}_m = \bar{\mathbf{u}}_m + \bar{\mathbf{v}}_m = 0$. Formula B.1 now becomes

$$h_{ij} = \frac{1}{n+p-1} \sum_{k=1}^{n+p} g_{ik} \cdot g_{jk}. \tag{B.2}$$

Here, the term $\sum_{k=1}^{n+p} g_{ik} \cdot g_{jk}$ is the inner product between the ith and jth row of $\mathbf{G}$, denoted as $\langle \mathbf{g}_i, \mathbf{g}_j \rangle$, which we can rewrite as:

$$\langle \mathbf{g}_i, \mathbf{g}_j \rangle = \langle (\mathbf{u}_i | \mathbf{v}_i), (\mathbf{u}_j | \mathbf{v}_j) \rangle = \langle \mathbf{u}_i, \mathbf{u}_j \rangle + \langle \mathbf{v}_i, \mathbf{v}_j \rangle. \tag{B.3}$$

Since the columns of $\mathbf{U}_L$ and $\mathbf{V}_L$ are orthonormal, for $i = j$ we have $\langle \mathbf{g}_i, \mathbf{g}_j \rangle = 1+1 = 2$, otherwise $\langle \mathbf{g}_i, \mathbf{g}_j \rangle = 0 + 0 = 0$. Hence,

$$h_{ij} = \begin{cases} \frac{2}{n+p-1} & , i = j \\ 0 & , i \neq j. \end{cases} \tag{B.4}$$

We can see from this formulation that the covariance matrix is already diagonal with constant diagonal elements, which however are not 1. To ensure that $h_{ij} = 1$ for $i = j$, we can simply multiply $\mathbf{G}$ by the constant $\sqrt{\frac{n+p-1}{2}}$ since then for $i = j$ we get

$$\langle \mathbf{g}_i, \mathbf{g}_j \rangle = \langle (\mathbf{u}_i | \mathbf{v}_i), (\mathbf{u}_j | \mathbf{v}_j) \rangle = \left(\sqrt{\frac{n+p-1}{2}} \right)^2 + \left(\sqrt{\frac{n+p-1}{2}} \right)^2 = n+p-1 \tag{B.5}$$

Then, eq. B.2 becomes

$$h_{ij} = \begin{cases} 1 & , i = j \\ 0 & , i \neq j. \end{cases} \tag{B.6}$$

meaning that $\text{Cov}(\mathbf{G}) = \mathbf{I}_L$, thus $\mathbf{G}$ is whitened.

Appendix C. DeGAs T2D Clusters

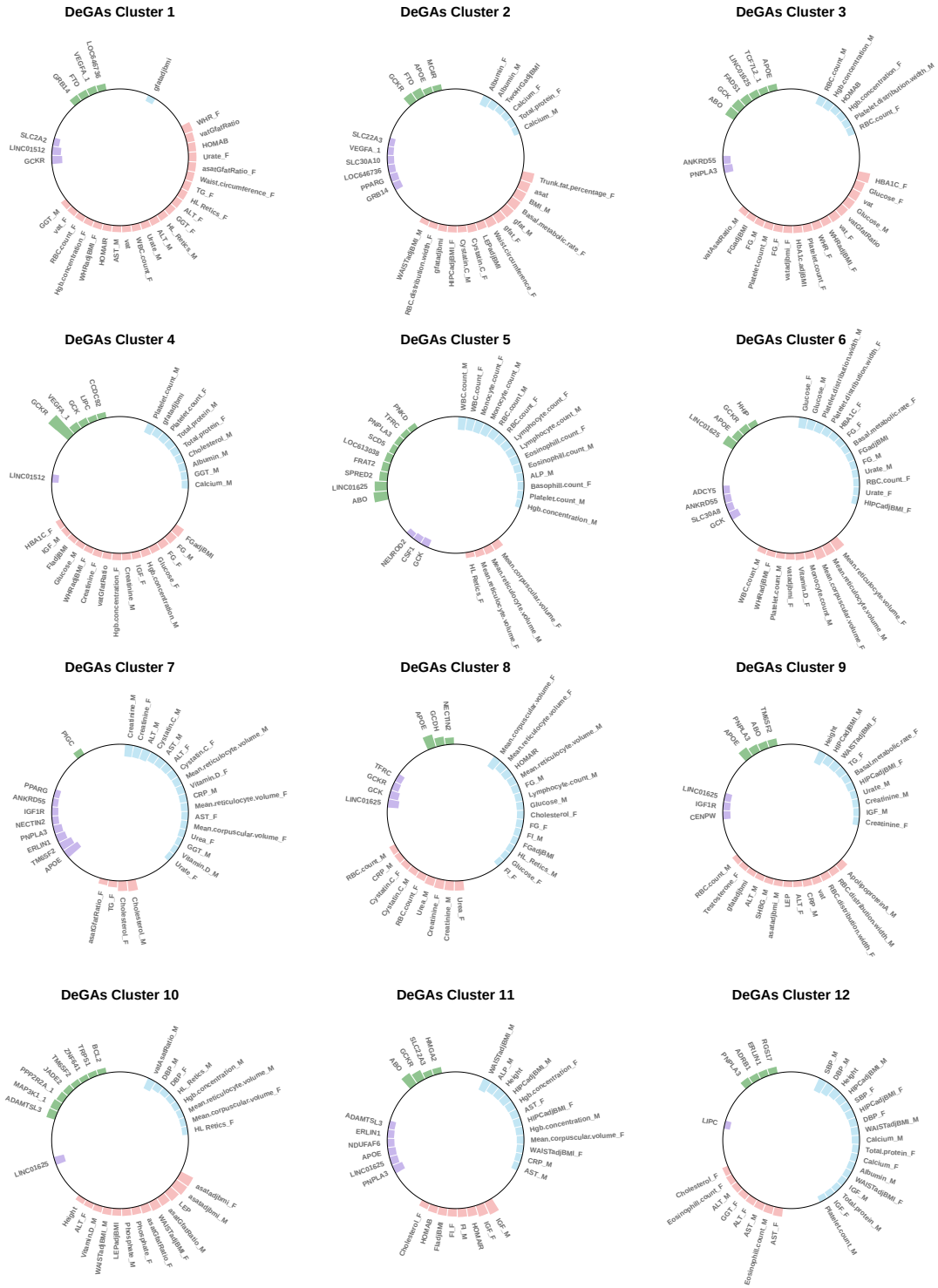

Figure C.1: DeGAs T2D genetic clusters, ordered by decreasing singular values, showing the variants and traits with the highest contribution scores. Variant and trait weights are visualized using the same scheme as in Figure 5.4, where the GUIDE clusters are shown.