Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Computer Science

Diploma Thesis

Inference of interaction networks from multi-omics data using Bayesian networks

Alikhan Anuarbekov

Supervisor: doc. Ing. Jiří Kléma, Ph.D.
Study Programme: Open Informatics
Field of Study: Bioinformatics

May 26, 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Anuarbekov** Jméno: **Alikhan** Osobní číslo: **483420**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Bioinformatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Inference interakčních sítí z multi-omických dat pomocí bayesovských sítí

Název diplomové práce anglicky:

Inference of interaction networks from multi-omics data using Bayesian networks

Pokyny pro vypracování:

1. Familiarize yourself with the issue of multi-omic data integration.
2. Familiarize yourself with learning techniques suitable for learning and representing interaction networks, focusing on Bayesian networks and the recently proposed IntOMICS algorithm.
3. Apply the IntOMICS algorithm to the MDS data supplied by the supervisor. The data contain a circRNA, miRNA, and mRNA interaction network.
4. Evaluate the scalability and efficiency of the IntOMICS algorithm. Modify the algorithm to be applicable to the interaction network mentioned above.
5. Verify the plausibility of the interactions found (by comparison with the literature, existing a priori knowledge, parallel interaction search tools).
6. Evaluate the follow-up to the circGPA tool (what results will the application of circGPA in the MDS task lead to before and after the application of modified IntOMICS).

Seznam doporučené literatury:

Pačínková, A., & Popovici, V. (2022). Using empirical biological knowledge to infer regulatory networks from multi-omics data. BMC bioinformatics, 23(1), 1-23.

Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. Statistical applications in genetics and molecular biology, 6(1).

Ryšavý, P., Kléma, J., & Merkerová, M. D. (2022). circGPA: circRNA functional annotation based on probability-generating functions. BMC bioinformatics, 23(1), 1-23.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Jiří Kléma, Ph.D. Intelligent Data Analysis FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **06.02.2023** Termín odevzdání diplomové práce: **26.05.2023**

Platnost zadání diplomové práce: **22.09.2024**

doc. Ing. Jiří Kléma, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 26. května 2023 ....

Acknowledgements

I would like to thank my thesis supervisor doc. Ing. Jiří Kléma, Ph.D. for his support and guidance. He has provided me with the necessary feedback and resources I needed to carry out the work. Additionally, I must express my gratitude to my parents and to my friends for providing me with support throughout the process of writing this thesis. Additionally, I would like to thank CESNET Metacentrum for the offer of the computing power for my practical part of the work.

Abstract

A recent study has introduced the circGPA method for efficient GO term annotation from a given interaction network. Circular RNAs without known functions are annotated with ontological terms based on the assumption that interacting RNA molecules have similar functions and, therefore, similar annotations. However, the interaction network used in the method was constructed solely from known database interactions, which may be inactive under certain experimental setups. In this thesis, a potential solution to this problem is presented by combining empirical data specific to the experimental setup with known database interactions using the Bayesian network inference algorithm.

Initially, the IntOMICS algorithm was chosen as the model. However, it was found unsuitable for the experimental data and the circRNA interaction network of the given size. Consequently, modifications of the algorithm were necessary. These modifications resulted in partial improvements in the speed of upper-bound term computation and memory requirements. Despite the improvements, the algorithm still has limitations and does not guarantee convergence. Overall, IntOMICS was found unsuitable for the task. Therefore, an alternative algorithm stemming from Prior-incorporated Skeleton-based Stochastic Search was implemented to address this problem and the empirical interaction network was successfully generated.

Finally, the empirical interaction network was used for a selected subsample of circRNAs. Three different versions of the circGPA algorithm were tested, confirming the significant influence of gene expression data and experimental conditions on GO term annotation.

Keywords: bayesian network, circular RNA, interaction network, GO term annotation

Abstrakt

V nedávném článku byla představena metoda circGPA pro efektivní anotaci cirkulárních RNA z jejich interakční sítě. Cirkulární RNA s dosud neznámou funkcí jsou anotovány ontologickými termy na základě předpokladu, že interagující RNA se známou anotací mají příbuznou funkci a tím i anotaci. Jedním z omezení navržené metody bylo to, že interakční síť byla konstruována pouze ze předem známých databazových interakcí. Ty nemusejí být v konkretních experimentálních podminkách aktivní. V této diplomové práci je představeno potenciální řešení tohoto problému – kombinace empirických expresních, t.j. experimentálních dat, se znamými interakcemi z databazí pomocí algoritmu učení bayesovských sítí.

Jako první model byl použit algoritmus IntOMICS. Bylo však zjištěno, že IntOMICS se pro naše experimentální data a circRNA interakční sítě dané velikosti nehodí a jsou potřeba jeho modifikace. Pomocí těchto modifikací bylo dosaženo dílčích zlepšení výpočtu horní meze pravděpodobnostního ohodnocení interakčních sítí a byla dosažena vyšší výpočetní rychlost algoritmu. I přesto algoritmus vykazoval několik omezení, neměl garanci konvergence a celkově byl vyhodnocen jako nevhodný. Proto byl naimplementován alternativní algoritmus Prior-incorporated Skeleton-based Stochastic Search, který zásadní problémy vyřešil a empirická interakční síť byla vygenerována.

Konečně, empirická interakční síť byla použita k anotaci vybraných cirkulárních RNA pomocí algoritmu circGPA. Byly porovnány tři různé verze použití circGPA algoritmu a bylo dokázáno, že data genové exprese a experimentální podmínky mají významný vliv na anotaci cirkulárních RNA.

Klíčová slova: bayesovské sítě, cirkulární RNA, interakční sítě, anotace GO termy.

Contents

Introduction

Recently, a new type of RNA molecule has been discovered – the circular RNA (circRNA) [92][58]. However, most of the currently known circRNA molecules have yet unknown functionality. While it is possible to unveil their functionality through biological experiments, the cost of such experiments is expensive. As a consequence, any in-silico method available that could hint a potential functionality beforehand is needed. This gene-functionality mapping can be considered as part of the broader gene ontology (GO) annotation task, which involves annotating diseases, processes, etc.
A recent study [105] has presented a circRNA generating-polynomial annotator (circGPA), an algorithm for efficient gene ontology (GO) term annotation of circRNA molecule. The circGPA algorithm solves the problem by assuming similar functions of interacting RNA molecules [89]. Since the circRNA is known to regulate micro RNA (miRNA), therefore, it effects the functionality of other RNAs, particularly miRNA and mRNA. Thus, the functionality of circRNA molecules could be derived by analyzing already studied RNA molecules that interact with them [89]. However, in the original circGPA algorithm the interactions of molecules were constructed entirely from data collected across multiple databases. This approach combines multiple databases, which differ in the way of interaction acquisition and biological assumptions. Moreover, general experimental conditions are not considered, which may result in misleading interactions that are inactive in the specific studied setup. To overcome these limitations of the original algorithm, an empirical interaction network, which is a graph of interactions utilizing both known database interactions and experimental data, is constructed.

As evident from the preceding paragraph, the main focus of this thesis is the analysis and construction of empirical interaction networks and their subsequent application in gene ontology term annotation task [105]. The structure of a thesis is as follows.

First (chapter 1), a general theoretical background for such multi-discriplinary task is given. This chapter fulfills multiple goals at once – it introduces all the basic knowledge necessary for more advanced algorithms and concepts presented afterwards, and, at the same time, introduces the notation used in formulas and algorithms. It covers the molecular biology, graph theory and the probability theory – all the parts required for an introduction of interaction networks and algorithms associated with them.

Second (chapter 2), an experimental data and more details about the gene ontology term annotation task are presented. After a small introduction into the multi-omics data problematic, the experimental setup of the supplied data is presented. Particularly, a model of gene interaction network used in circGPA [105] is described and the list of database sources of the known RNA interactions is shown. Furthermore, a gene expression data collected from patients with Myelodysplastic syndrome (MDS) is analyzed. More precisely – 77 samples, e.g., patients diagnosed with MDS, and a set of 21952 RNA genes are measured on them. The set consists of – 17287 mRNAs, 1656 miRNAs and 3009 circRNAs. An initially given count table is normalized according to the Transcript Per Million (TPM) [99] method.

Third (chapter 3), a definition of Bayesian networks and interaction networks, e.g., a particular use case of Bayesian networks, is presented. A Bayesian network could be briefly described as the combination of Directed Acyclic Graph (DAG) and a joint probability function encoded in it. Finally, at the end of the chapter it is concluded that the only problematic part of the Bayesian network learning is the structure inference. More precisely – the learning of a Bayesian network consists of two parts, e.g., learning of its structure and subsequent parameter learning. In case of interaction networks, learned parameters, e.g., weights of each edge in Bayesian network graph, are not utilized at all since the upcoming circGPA annotation uses the structure only. Although, even without parameter learning, the structure inference remains the key problem and the reason for this is the NP-hard nature [9] of the Bayesian structure learning in general. In other words, there exists no polynomial algorithm under the $P \neq NP$ assumption [32].

Fourth (chapter 4), a comprehensive analysis of the existing methods of Bayesian network structure inference is given. The classification of existing approaches is divided into three categories: Score-and-Search, Constraint-based and Hybrid.

Score-and-Search approach is the state-of-the-art method to return an optimal network under mild assumptions [101][104], but it requires exponential memory and time complexity and, thus, restricts us from applying this method straightforwardly on the gene expression data table of size 77 x 21952. Existing algorithms try to overcome such problem by introducing the maximum in-degree limitation, e.g., to limit the maximum number of parents that each node could have [101][30]. While such limitation may be useful in general, it critically reduces the interaction network reliability. Despite all the limitation, the initial choice of an algorithm belongs to this category – the IntOMICS [104] algorithm. The reason for this choice is the fact that the IntOMICS algorithm is one of the first algorithms to introduce the empirical interaction network – a combination of gene expression data and known (prior) database interactions. Consequently, the only known formula that incorporates both of these methods is found here.

Constraint-based approach algorithms, on the other hand, have polynomial memory and time complexity, but are computed under very strict assumption of faithfullness [28] that does not hold in practice. Nevertheless, the core advantage of this type of methods is in their scalability that goes beyond thousands of variables thanks to the polynomial complexity. Moreover, if compared, results of such methods are comparable to those of early Score-and-Search methods [96][28]. Although, such methods use statistical testing instead of a previous probability score and, thus, there are no known statistical tests to incorporate both known (prior) interactions along with the gene expression data in it. Consequently – no empirical interaction network could be generated from these methods.

Hybrid approach attempts to combine advantages of two previous approaches in order to achieve better performance for a larger Bayesian network. To generalize this approach – the algorithm starts by constructing the interaction network in polynomial time and then improves it by a limited search-and-score algorithm iteratively.

Fifth (chapter 5), a more detailed description of two key algorithms is given, e.g., IntOMICS and circGPA. As mentioned above, initially, the goal is to modify the IntOMICS algorithm in order to generate an empirical interaction network of a required size. The modification of IntOMICS targets the most demanding part of the algorithm – a computation formula of super-exponential terms. Alternatively, if such modification will not succeed, a different algorithm should be chosen from a list above and implemented.

After the generation of an empirical network for supplied data, a further GO term analysis would be performed with a help of circGPA algorithm. A hypothesis tested in the analysis is whether the gene expression data crucially influences the outcome of GO term annotation. To perform such testing, three version of circGPA algorithm are presented – the original one and two modified versions that incorporate the gene expression data.

Sixth (chapter 6, an in-depth description of the chosen empirical network inference algorithm is given. Starting from an attempt to modify and improve the IntOMICS algorithm, and then proceeding to the implementation of the alternative algorithm, e.g., Prior-incorporated MMPC Skeleton-Based Stochastic Search (PiM-SK-SS). A special accent is placed on the reasoning for the choice of this particular algorithm, but technical details are also mentioned.

Lastly (chapter 7), a practical experimentation according to the above-mentioned scheme with three circGPA versions is conducted and results are summarized in the Conclusion (chapter 8).

Chapter 1

Theoretical background

This chapter discusses the biological, graph theory and statistical aspects of the thesis and the essential background necessary to understand the topic at hand. The main purpose of this chapter is to define all the basic terminology and concepts in order to reduce the large paragraphs in the complex algorithms definition.

1.1 Biological background

In this section we will focus on molecular biology, especially on its fundamental statements.

1.1.1 Deoxyribonucleic acid

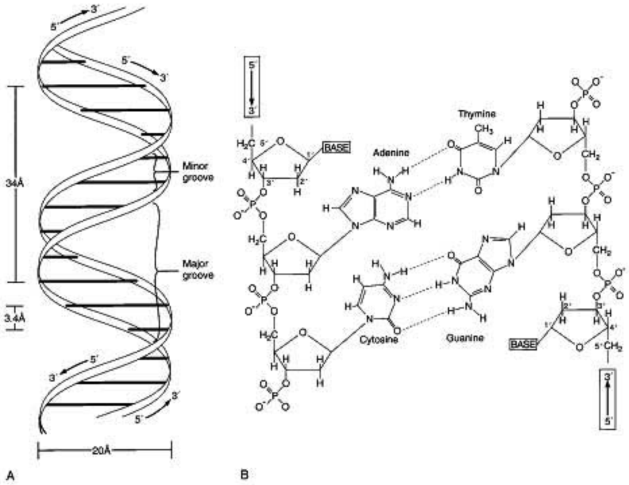


Figure 1.1: DNA structure, taken from [1]

Deoxyribonucleic acid (DNA) is a molecule consisting of two strands. Each strand is a chain of nucleotide molecules connected by a phosphate group [2]. For more detailed chemical structure, see fig 1.1.
Two strands are held together by hydrogen bonds of corresponding pairs of nucleotides. These pairs are called *complementary pairs*. All possible complementary pairs can be seen in fig 1.1.

It was proved in [1] that the DNA molecule carries the genetic information of an organism. We typically denote the DNA by a sequence of its nucleotides. The four types of nucleotides define the coding of a sequence: Adenine (**A**), Thymine (**T**), Guanine (**G**), and Cytosine (**C**) [2].
It is worth noting that complex organic molecules built from more primitive molecules are called *biopolymers*.

[1] https://www.ufrgs.br/imunovet/molecular_immunology/DNAstructureanalysis.html

1.1.2 Ribonucleic acid

Ribonucleic acid (RNA) is a biopolymer with a similar structure to the DNA molecule. There are two key differences:

- RNA is a single coil molecule
- RNA uses Uracil (**U**) nucleotide instead of Thymine (**T**)

Moreover, the sequence of RNA defines the functionality of a molecule. Change in functionality is achieved by interaction among the nucleotide pairs. For more details, see [40].

1.1.3 Protein

Protein is a chain of amino acid molecules that is called a *polypeptide*. Similarly, a protein is reprensented based on its sequence of amino acids, and the interactions of amino acid molecules define its structure and functionality. Twenty different types of amino acids determine the coding of a protein. Proteins are essential components in primary functions of an organism [57].

1.1.4 Central dogma of molecular biology

The central dogma was published in 1958 and became the keystone of modern molecular biology. The core idea of the central dogma is a transfer of information from one type of biopolymer to another. Consequently, it implies that a sequence of one type allows a cell to synthesize a different kind of biopolymer. Furthermore, the existence of coding alphabet mentioned in previous subsections was proposed in central dogma, and every coding alphabet was derived experimentally afterward [3].

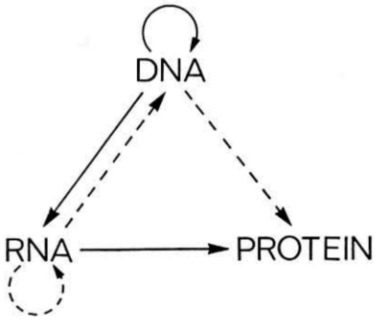


Figure 1.2: Central dogma diagram, taken from [3]

Fig. 1.2 shows us the information flow in an organism. The genetic information initially stored in DNA is transcribed into RNA. Afterward, it is translated into protein polypeptide.
Dashed arrows represent exceptional cases. For example, viruses use RNA to create pathogenic DNA sequences [67].

1.1.5 Regulatory mechanisms

The central dogma, as admitted by authors in [3], is a simplified overview of a complex biological system of a cell. One of aspects not mentioned in the dogma are effects of biopolymer parts on each other.

Gene

Gene is a subsequence of a DNA sequence that encodes a functional RNA. Likewise, the information transfer from DNA to RNA, e.g., transcription, is performed in segments. Each segment can contain one or more genes [69], [114]. We distinguish two types of genes:

- *coding genes* – subsequences that encode an mRNA, which participates in a protein synthesis
- *non-coding genes* – subsequences that encode other types of RNA with a different functionality or a subsequence that only regulates other RNA's transcription

Regulatory sequence

The regulatory sequence (site) of a DNA is a segment that does not encode RNA. Its purpose is to regulate the amount of RNA produced from neighbor genes, and it is achieved by the molecular binding of specific proteins to this site. Some regulatory sequences play an essential role in the transcription process, allowing other proteins and RNAs to begin synthesizing or blocking it [98].

Transcription factor

Transcription factors (TFs) are specific proteins that bind to regulatory sequences, thus, regulating or initiating the transcription process [98], [112].

Micro RNA

Another method of regulating a protein synthesis is the transcription of the **micro RNA (miRNA)** molecule. The primary mechanism of this regulation is the complementarity of micro RNA and mRNA sequences. This chemical bond leads to a double strand similar to the DNA and effectively prevents the mRNA from being translated [74].

Circular RNA

While most known regulatory sequences affect protein synthesis, some regulatory mechanisms also target other regulatory sequences. An example of such secondary regulation can be **circular RNA**.
The name – **circular RNA (circRNA)** suggests the unusual property of this RNA. After the transcription step, its components interact and create a loop, while previously mentioned molecules remain in their linear form.
The secondary regulation happens in a process called *miRNA sponge* when a miRNA molecule creates a bond with a circular RNA. The bond prevents the microRNA from forming a bond with the mRNA, thus regulating it [92].

DNA Methylation

DNA methylation is an epigenetic modification, e.g., a change in a DNA that does not affect the sequence but, for instance, alters its chemical properties. DNA methylation adds a methyl group molecule to the nucleotide base and modifies its chemical structure, leaving the nucleotide type unchanged for measurement. The main effect of this change is the interaction with previously mentioned regulatory proteins, which makes them less likely or completely unable to bind to the methylated segment [36].
For an illustration of a change, see fig 1.3.

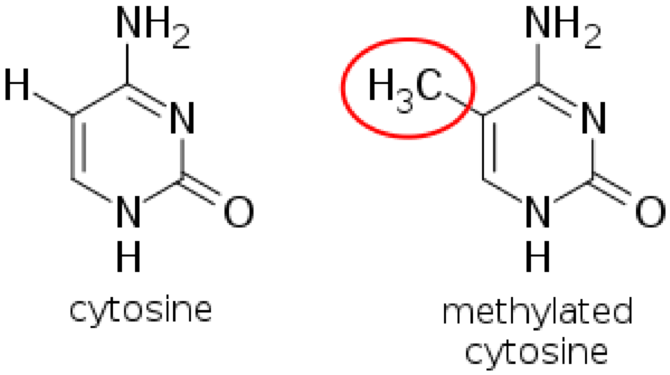


Figure 1.3: DNA methylation example, taken from [1]

[1] https://en.wikipedia.org/wiki/DNA_methylation

1.1.6 Gene expression

To summarize previously presented definitions, we will introduce a method to analyze gene interactions.
Gene expression (GE) is a process of synthesizing a functional gene product, e.g., proteins and functional RNAs. In contrast with stable DNA, the amount of each functional gene product changes based on an environment change or cell development. These quantities can be measured and used to perform a further analysis.
This research method is becoming more available due to the increasing number of data samples and databases, thus, it will be used as a primary method throughout the thesis [75].

1.2 Graph theory

In this section, we will focus on graph theory, especially on its statements needed for future use.

1.2.1 Graph

A **graph** is a set ($\mathbb{V}$, $\mathbb{E}$, ϵ), where $\mathbb{V}$ is a non-empty set of vertices, $\mathbb{E}$ is a set of edges, and ϵ is the incidence function.
If a graph is said to be a **directed graph**, then ϵ is defined as:

$$\epsilon : \mathbb{E} \to \mathbb{V} \times \mathbb{V}$$

or identically:

$$\epsilon : \mathbb{E} \to (v_1, v_2), \qquad \text{where } v_1, v_2 \in \mathbb{V}$$

If a graph is said to be an **undirected graph**, then ϵ is defined as:

$$\epsilon : \mathbb{E} \to \{v_1, v_2\}, \qquad \text{where } v_1, v_2 \in \mathbb{V}$$

1.2.2 Path and Cycle

A **path** is defined as an oriented sequence in the form of:

$$v_1, e_1, v_2, e_2, ..., e_{k-1}, v_k,$$

where $v_i \in \mathbb{V}, e_j \in \mathbb{E}$ for i=1,2,...,k and also such that:

$$\forall i = 1, 2, ..., k-1 : \quad (e_i, \{v_i, v_{i+1}\}) \in \epsilon$$

and

$$\forall i = 1, 2, ..., k;\ j = 1, 2, ..., k;\ i \neq j :$$
$$v_i \neq v_j$$

with a possible exception of $v_1 = v_k$.

A **cycle** is defined as a path where $v_1 = v_k$, e.g., starts and ends in the same vertex [5].

1.2.3 Directed path and cycle

Paths and cycles are **directed** if the underlying graph is *directed* and:

$$\forall i = 1, 2, ..., k-1 : \quad (e_i, (v_i, v_{i+1})) \in \epsilon$$

1.2.4 Connectivity

A graph is **connected** if:

$$\forall i = 1, 2, ..., k;\ j = 1, 2, ..., k;\ i \neq j :$$
$$\exists \text{ path with } v_1 = v_i \text{ and } v_k = v_j$$

1.2.5 Tree

A connected graph without *undirected* cycles is called a **Tree**.

1.2.6 Directed Acyclic Graph

A directed graph without *directed* cycles is called a **Directed Acyclic Graph (DAG)** [5].

Topological ordering

If the **DAG** is given, then there always exists a linear ordering of its vertices:

$$v_1, v_2, ..., v_n$$

such that:

$$\forall (e_i, (v_j, v_k)) \in \epsilon : \qquad j < k$$

This ordering is called a **Topological ordering** of a given **DAG** [5].

1.2.7 Skeleton of a graph

If given a *directed* graph $G = (\mathbb{V}, \mathbb{E}, \epsilon)$, we define its *Skeleton graph* as the *undirected* graph $G_{SK} = (\mathbb{V}, \mathbb{E}^{'}, \epsilon^{'})$ with the same, but undirected edges:

$$\forall (e_k, (v_i, v_j)) \in \epsilon \rightarrow \exists (e^{'}, \{v_i, v_j\}) \in \epsilon^{'}$$

1.2.8 Partially directed graphs

A *partially directed graph* is a graph that contains both directed and undirected edges. It is possible to derive the underlying *partially directed graph* of the directed graph by doing the same procedure as in the skeleton, but only for a subset of edges.

1.3 Probability theory background

In this section, we will focus on probability theory, specifically its definitions needed for further applications.

1.3.1 σ-algebra

Given a set Ω, we define its family of sets $\mathcal{A}$ as σ-algebra if it satisfy these assumptions [97][48]:

- $\emptyset \in \mathcal{A}$
- $(\forall n \in \mathbb{N})(M_n \in \mathcal{A}) \Rightarrow (\bigcup_{i=1}^{\infty} M_i) \in \mathcal{A}$
- $(\forall M \in \mathcal{A}) \Rightarrow (\Omega \setminus M) \in \mathcal{A}$

1.3.2 Probability definition

Let us have a triplet $(\Omega, \mathbb{A}, \mathbb{P})$ where:

- Ω is a set of all possible outcomes of a single sample
- $\mathbb{A}$ is a σ-algebra of Ω that represents events, e.g., all possible sets of samples
- $\mathbb{P}$ is a **probability function**:
  $\mathcal{P} : \mathcal{A} \rightarrow \langle 0, 1 \rangle$ such that:
  - $\mathcal{P}(\emptyset) = 0$
  - $\mathcal{P}(\Omega) = 1$
  - $\mathcal{P}(\bigcup_{i=1}^{n} M_i) = \sum_{i=1}^{n} \mathcal{P}(M_i) \qquad$ for $M_1, ..., M_n \in \mathcal{A}$ and $\forall i \neq j : \ M_i \cap M_j = \emptyset$

1.3.3 Random variable

The random event can also be replaced with the **random variable**:

$$\mathbb{X} : \Omega \rightarrow \mathbb{R}$$

because a measurable quantity always represents abstract events. The most trivial case will be a 0/1 discrete variable of whether the event happened.

1.3.4 Conditional Probability

We define a **conditional probability** of an event A given the knowledge that an event B has already occurred as [111]:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}$$

1.3.5 Marginalization

From the previous definitions, we can derive the following formula [110][111]:

$$\mathcal{P}(B) = \sum_{A_i \in \mathcal{A}} \mathcal{P}(B|A_i)\mathcal{P}(A_i) \tag{1.1}$$

or, alternatively, if the variable is continuous:

$$\mathcal{P}(B) = \int \mathcal{P}(B|A)\mathcal{P}(A)dA \tag{1.2}$$

or in other form:

$$\mathcal{P}(B) = \mathcal{P}(B|A)\mathcal{P}(A) + \mathcal{P}(B|\overline{A})\mathcal{P}(\overline{A}) \tag{1.3}$$

where $\overline{A} = \mathcal{A} \backslash A$ and $A_i \cap A_j = \emptyset$
We call this formula a **marginalization** of a variable A.

1.3.6 Bayes theorem

The Bayes theorem formula can be derived from the previous definitions [110]. It has the following form [107]:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(A)}{\mathcal{P}(B)}\mathcal{P}(B|A) \tag{1.4}$$

This equation can then be broken down into the following probabilities [107]:

- **Likelihood probability**: $\mathcal{P}(B|A)$
- **Prior probability**: $\mathcal{P}(A)$
- **Marginal probability**: $\mathcal{P}(B) = \mathcal{P}(B|A)\mathcal{P}(A) + \mathcal{P}(B|\overline{A})\mathcal{P}(\overline{A})$
- **Posterior probability**: $\mathcal{P}(A|B)$

1.3.7 Independence and Joint probability

We define two events A and B to be **independent** if the following holds [97][48]:

$$\mathcal{P}(A \cap B) = \mathcal{P}(A)\mathcal{P}(B)$$

Generally, the probability of two variables can be written as:

$$\mathcal{P}(A \cup B) = \mathcal{P}(A) + \mathcal{P}(B) - \mathcal{P}(A \cap B)$$

Furthermore, we denote the union of two or more events as the **Joint probability**:

$$\mathcal{P}(A_1 \cup A_2 \cup ... \cup A_n) = \mathcal{P}(A_1, A_2, ..., A_n)$$

1.4 Statistical background

The previous probability theory is needed to define a model of real-world events we observe. However, to verify a proposed model's correctness, we must introduce several terms in a statistical field.

1.4.1 Markov Chain

A Markov Chain is sequence of random variables: $X_0, X_1, ..., X_m$ sampled in a following model [42]:

- Set of states: $S = \{S_1, ..., S_n\}$, e.g., a set of possible values of random variables
- Initial state: $X_0 = S_{init}$
- Transition probability function P of changing a current state: $P(X_{next}|X_{current}, X_{previous}, ...)$
- Markov Property: $P(X_{k+1} = i|X_k = j, X_{k-1} = l, ...) = P(X_{k+1} = i|X_k = j)$, e.g., the probability of transition from the current state to another does not depend on anything except for the current state.

1.4.2 Statistical population

Statistical population is the real-world source of the events described in 1.3.2. It is a set of objects that generate events represented in a probability model and can be characterized as a Ω from a probability definition.

1.4.3 Random sampling

In contrast with the abstract model, there exists a limitation of interaction with the statistical population. The most significant limitation is that we usually cannot research the entire population due to its size and the limited resources available. Therefore, only a subset of Ω is analyzed, and a certain amount of error and inaccuracy of the measured data should also be considered. This subset is obtained by a method called **random sampling**. The sampled data is used to analyze the entire population and the results are then extrapolated [85].

1.4.4 Hypothesis testing

Hypothesis

When analyzing the data, a researcher assumes a certain property of the data and tries to either prove or disprove it. The assumed property is called the *hypothesis* and it formalizes as follows:

- $\mathcal{H}_0$ = *Null hypothesis*, the assumed property does not hold
- $\mathcal{H}_A$ = *Alternative hypothesis*, the assumed property actually holds

Statistical test

In order to find the true hypothesis of these two, first – the error tolerance threshold should be stated beforehand. As mentioned earlier, any sampling, e.g. measuring of some data from the population, is accompanied with a certain error or a difference from the whole population. Thus, even if some property holds, the hypothesis may show a small error that should be accepted [85].
A common way to define an error threshold is either the **p-value** or the **α-significance level** [85].

Test statistics

Finally, in order to convert the abstract hypotheses to a mathematical form, a certain value on the data is computed. The particular choice of the value and its formula is called the *Test statistics*.
As an example, if the independence of two samples is tested, a possible choice of the test statistics is the $\chi^2 -$*test (chi-squared test)*[2]:

- Given two sample sets: $(O_{1,1}, ..., O_{1,c})$, $(O_{2,1}, ..., O_{2,c})$

[2] https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/chi-square/

- Calculate its expected value:

$$E_{ij} = \frac{\sum\limits_{k=1}^{c} O_{ik} \sum\limits_{k=1}^{2} O_{kj}}{N}$$

- Calculate the Statistical test's value:

$$\chi^2 = \sum_{i=0}^{2} \sum_{j=0}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

- Compare the computed χ^2 value to the known value from the χ^2 table (which depends only on a number of elements in samples sets) and derive the error. If it is larger than the threshold, $\mathcal{H}_0$ holds. Otherwise, the $\mathcal{H}_A$ is the true hypothesis.

1.4.5 Linear regression

Another method of further statistical analysis is a modeling of the random variables dependency. In other words, while the previous probability theory established the general probability model definition, statistical analysis introduces methods to estimate the concrete function form and, more importantly, the relationship between individual variables in a complex model [71][11].

One of simplest methods is the modeling of the variables dependency based on a linear function. More formally, given the set of so-called independent random variables $X_1, ..., X_N$ and so-called dependent random variable Y, a linear dependency is defined as the following set of equations:

$$\begin{aligned} C_0 + C_1 X_{1,1} + ... + C_N X_{1,N} &= Y_1 \\ C_0 + C_1 X_{2,1} + ... + C_N X_{2,N} &= Y_2 \\ &... \\ C_0 + C_1 X_{K,1} + ... + C_N X_{K,N} &= Y_K \end{aligned}$$

where indices denote individual samples. Or, equivalently, it is possible to write it in a matrix form:

$$\mathbb{1} C_0 + \mathbf{X}\vec{c} = \vec{y}$$

The only unknown term in equations above are coefficients that are determined from the statistical analysis. However, as mentioned above, the only possibility is to sample random variables from a population. Moreover, an inevitable error or noise should be taken into account. By combining these assumptions, a minor change in the definition of linear regression should be made.

$$\mathbb{1} C_0 + \mathbf{X}\vec{c} + \mathbb{N}(0, \sigma^2) = \vec{y}$$

Where $\mathbb{N}$ is a famous Gaussian normal distribution function [7] that is used to approximate any function.

Last term represents some unknown and immeasurable error that is added to the data. Essentially, it could be thought of as a noise in variables' value.

Finally, a specific metric function must be introduced to determine coefficients from the equation above. Typically, an *Ordinary Least Squares* minimization metric is used:

$$\underset{C}{\operatorname{argmin}}\, LSE = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{K} (Y_i - C_0 - \sum_{j=1}^{N} X_{i,j} C_j)^2$$

Now it is sufficient to solve the minimization problem to get optimal coefficients. This procedure is called a *fitting* of a model.

It should be mentioned that different metrics are being used in practice, but they will not be listed here.

Chapter 2

Data

Lastly, with all the background knowledge presented, the following chapter will be devoted to the subject of a thesis, the problem of multi-omics data. After the brief introduction into the multi-omics data problematic, a detailed description of the studied data is given. Particularly, the MDS gene expression data and prior database interactions are examined in context of an upcoming circGPA annotation task. Also, a tripartite interaction graph is assumed as a biological model.

2.1 Multi-Omics data

As shown in section 1.1, the cell of an organism undergoes a variety of complex biological processes with the presence of different biomolecules. Even though fundamental molecular interactions were discovered more than a half-century ago, up until recently, the amount of empirical data remained insufficient for more complex analysis. The situation changed with the advent of modern high-throughput methods that allowed the sequencing of an enormous amount of genetic data efficiently. Collecting these data allows for modeling a more complex and complete view of a molecular system of a cell [60][93][80].

The heterogeneous biological data raises a question of systematizing and grouping. Practically, the omics-based classification of biological data sets is used, e.g., categorizing the data based on disciplines of molecular biology that it originates from and, consequently, based on the part in the central dogma. Examples of omics groups are:

- **Genome** – DNA, e.g., original biopolymers and products of their replication
- **Transcriptome** – RNA, e.g., products of transcription of a DNA
- **Proteome** – proteins, e.g., products of translation of an RNA

and et cetera.

Such categorization originates from the history of molecular biology and biology as a whole [91]. Previously, simple abstract concepts and studies became too complex to be a part of more extensive genetics and gradually diverged into their separate fields.

For example, the initial study of Watson and Crick on DNA structure and later introduced Central Dogma (see section 1.1) was still considered part of genetics, e.g., a study of core mechanisms behind the heredity process. However, with the revolutionary discovery of DNA structure and processes behind these interactions, the need for a separate field which focuses on individual genes arose [91].

The previously mentioned techniques of massively parallel sequencing allowed the collection of an enormous amount of data [78]. This data, consequently, allowed researchers to analyze particular biopolymers and their interactions.

On top of that, a systematic view of a complex biological system is studied by a systems biology. While being the core of interaction studies, previously mentioned omics groups must be combined into the dynamic interacting network. As it follows from the text above, such comprehensive analysis results from sufficient empirical data collection in recent years. This dynamic interacting network is called an **interactome** [35][39], and it is one of the main subjects studied in this thesis.

2.2 Experimental data

The upcoming practical work is focused on a specific part of the multi-omics data domain. The studied subject is the construction of interaction networks (see next chapter 3.3) from the raw statistical data containing *RNA gene expression data samples* (see section 2.2.4). This empirical data is collected in order to derive interactions that are differentially expressed, e.g., statistically significant, in these samples.

2.2.1 Tripartite biological network

While most frameworks consider the general interaction network without any limitations on gene interaction, this thesis targets the tripartite biological model. It is based on the regulatory mechanisms observed in 3 types of RNA molecules in section 1.1.

The model follows the straightforward approach – it allows only the previously mentioned interactions. More precisely, the assumptions are that the only allowed interactions are circRNA - miRNA and miRNA - mRNA. Moreover, interactions between the same type of RNA are also forbidden. This setup leads to a tripartite graph in a form:

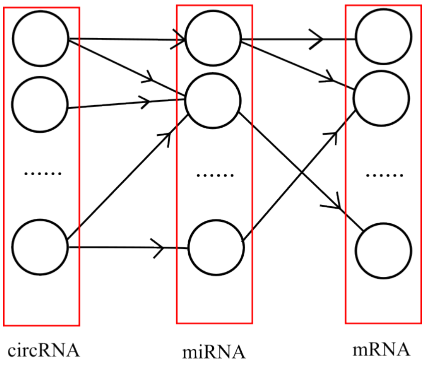


Figure 2.1: Graphical representation of tripartite biological model

However, it should be noted that circRNA has a broader spectrum of interactions with other biopolymers [92], not only with the miRNA, as listed here. Moreover, mRNAs directly impact each other's expression [91] by regulating through their products, and thus, practically, they should be allowed to interact with each other. The sole purpose of such a structure is the compatibility with the input of the circGPA (see 5.1) framework used for further analysis.

2.2.2 Usage and motivation

The core motivation of a tripartite and partly misleading model is its application to the obtained experimental data. As mentioned at the start of this chapter, the multi-omics comprehensive analysis is a recently arisen method, and the algorithms used in it still need to be improved. This thesis mainly analyzes a newly discovered circular RNA and its interactions with known biopolymers. Due to the relatively little data about the circRNA, any potential interaction discovered may suggest the path in a laboratory research.

The further motivation is to apply inferred interactions to frameworks such as circGPA to perform additional analysis. More details about the circGPA algorithm are covered in the later section 5.1. The tripartite model mentioned previously was also introduced in a circGPA paper [105].

Gene ontology annotation

To further describe the motivation, a gene ontology task will be presented. Not only that it is the output of the circGPA algorithm, but it also is the primary goal of the majority of presented frameworks and algorithms of this thesis.

A *gene ontology task* is a comprehensive vocabulary with nearly all of known genes, proteins and their interactions. On top of that, such vocabulary, compared to interaction networks, contains an information about the relationship of molecular mechanisms to the cellular components and the underlying biological processes. It is meant to be dynamic, applicable to any eukaryotic species, and, globally, tries to describe the current biological knowledge entirely [76][16].

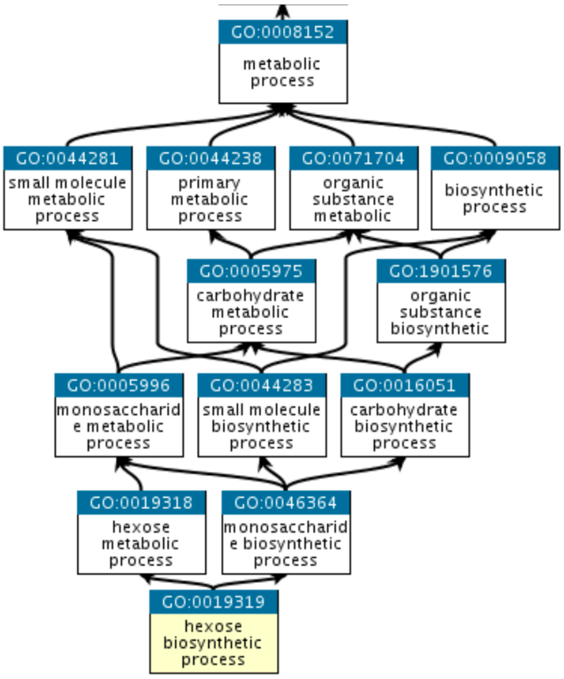

GO:0008152
metabolic process
GO:0044281
small molecule metabolic process
GO:0044238
primary metabolic process
GO:0071704
organic substance metabolic
GO:0009058
biosynthetic process
GO:0005975
carbohydrate metabolic process
GO:1901576
organic substance biosynthetic
GO:0005996
monosaccharide metabolic process
GO:0044283
small molecule biosynthetic process
GO:0016051
carbohydrate biosynthetic process
GO:0019318
hexose metabolic process
GO:0046364
monosaccharide biosynthetic process
GO:0019319
hexose biosynthetic process

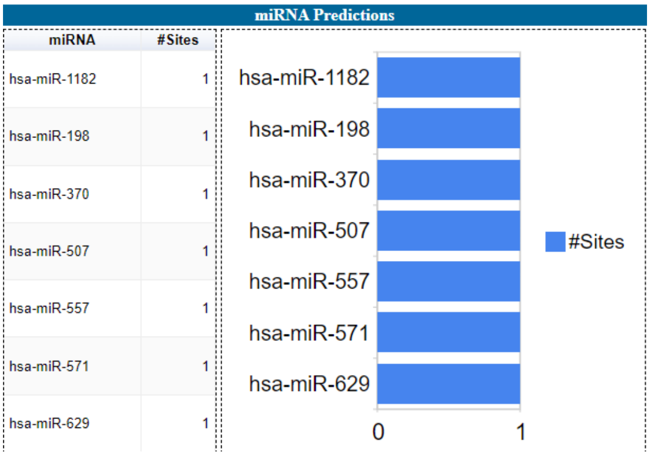
Figure 2.2: The example of the GO annotation, taken from [1]

2.2.3 Prior knowledge source

Since we are dealing with the statistical model of complex system, some prior knowledge should be used. Moreover, most frameworks used in this thesis require prior knowledge as an input to obtain adequate results. For such purposes, the following section will list most of databases with known gene interactions used afterwards.

circInteractome database

This database[2] is based on the algorithm Targetscan [58], which computes the circRNA - miRNA interaction based on features obtained empirically from different regression experiments. Since circRNA has only been recently discovered, the amount and the quality of such interactions are yet to be fully verified.
CircInteractome is the only source of prior (known) interactions used to construct a circRNA - miRNA part of the circGPA input.


miRNA Predictions

miRNA	#Sites
hsa-miR-1182	1
hsa-miR-198	1
hsa-miR-370	1
hsa-miR-507	1
hsa-miR-557	1
hsa-miR-571	1
hsa-miR-629	1

Figure 2.3: An illustration of interactions of a hsa-circ-0000010 from circInteractome database

[1] http://geneontology.org/docs/ontology-documentation/
[2] https://circinteractome.nia.nih.gov/

As it can be seen in an image above, an ID encoding of individual circ RNAs is as follows: *hsa* (species, e.g., Homo Sapiens) – *circ* (type of RNA, e.g., circRNA) – *0000010* (ID, represents an ordering based on discovery date).

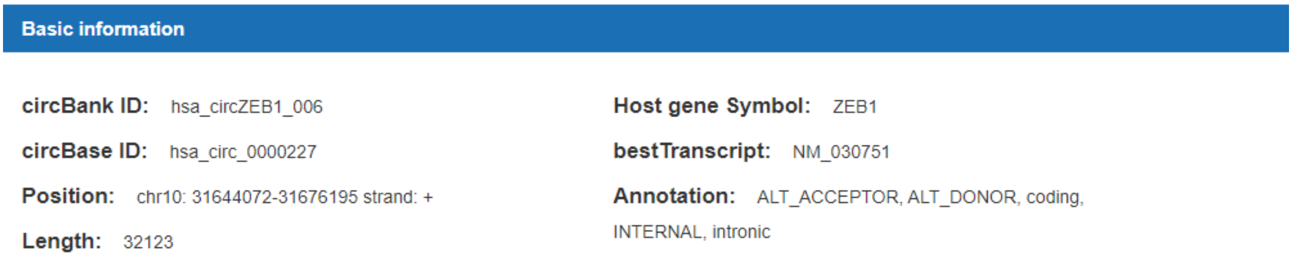
An important note to mention here is the list of circRNA molecules that would be analyzed in a further analysis:

- *hsa-circ-0000227* – has 139 interactions with different miRNAs
- *hsa-circ-0000228* – has 11 interactions with different miRNAs
- *hsa-circ-0003793* – has 10 interactions with different miRNAs

The reason for such limited set is the time complexity of each GO term annotation [105], primarily due to the tens of thousands of GO terms. Besides, circRNAs listed above have had no known functions at the time of circGPA algorithm publishing and are still mostly undiscovered. It is important to note that all three circRNAs are derived from the *ZEB1* gene.

circBank

Another source of information about the circRNA is available from [3]. However, this database is not used in collection process of known interactions in upcoming algorithms. The reason for this is the fact that database utilizes the same approach as in previous one, e.g., the TargetScan [58] algorithm, and, essentially, it is redundant to mix multiple database formats.
On the other hand, this database has an advantage over the circInteractome – it lists known annotations derived from existing statistical tools [83]. As an example, let us examine previously mentioned molecules:

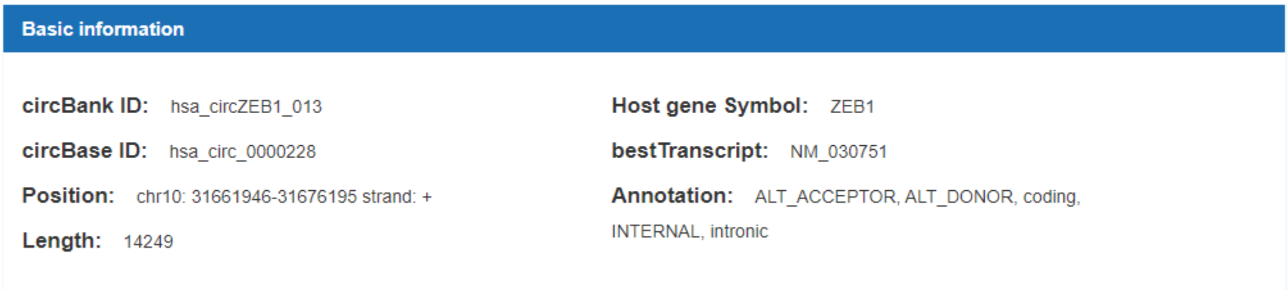


Basic information

circBank ID: hsa_circZEB1_006
circBase ID: hsa_circ_0000227
Position: chr10: 31644072-31676195 strand: +
Length: 32123

Host gene Symbol: ZEB1
bestTranscript: NM_030751
Annotation: ALT_ACCEPTOR, ALT_DONOR, coding, INTERNAL, intronic

Figure 2.4: More details about the *hsa-circ-0000227* in circBank database



Basic information

circBank ID: hsa_circZEB1_013
circBase ID: hsa_circ_0000228
Position: chr10: 31661946-31676195 strand: +
Length: 14249

Host gene Symbol: ZEB1
bestTranscript: NM_030751
Annotation: ALT_ACCEPTOR, ALT_DONOR, coding, INTERNAL, intronic

Figure 2.5: More details about the *hsa-circ-0000228* in circBank database

As seen in two figures above, the circBank database provides an annotation list for a given circRNA. While it may seem like a reason for a redundancy of a further analysis, it is quite the opposite.
First, the annotation given in the table above is referring to the paper [102], which analyses the entire group of circRNAs associated with the *ZEB1* gene. The regulation effect from those circRNAs is proven to be related with tumor genesis. However, since *ZEB1*-related circRNAs incorporate tens of circRNA molecules, the analysis

[3] http://www.circbank.cn/

is too broad and must be specified for particular circRNA molecules in it. For instance, [109] conducted another *ZEB1*-related circRNAs analysis. It showed the regulation effect on thermogenesis process, but in case of lamb as a testing subject. Such distinct annotation is yet another reason for a need of specification per circRNA molecule.

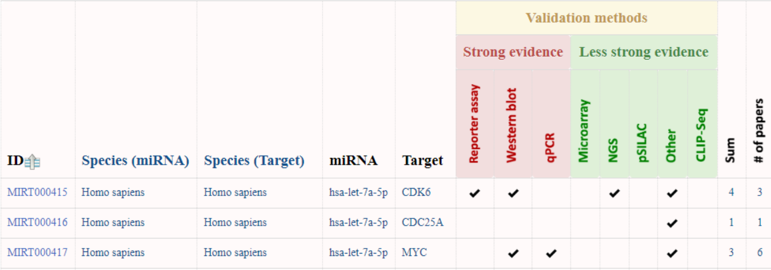
Second, an approach used in [109] is based on the same Targetscan [58] algorithm, but has been applied on animal gene expression data, in contrast with MDS human samples given in our case. Thus, it is an example of a different experimental setup. Consequently, it can be seen as a prime example that will be used to further prove the point of a critical influence of experimental setup.

MSigDB database

This is a database with annotated gene sets of humans and mice. It contains gene ontology annotations, different signatures, and computational microarray data[4].

miRTarBase database

Finally, the miRNA-mRNA interactions could be obtained from the MultiMIR database [81]. Individual interactions here are marked with the verification reliability, as shown in figure 2.6.



					Validation methods									
					Strong evidence			Less strong evidence						
ID	Species (miRNA)	Species (Target)	miRNA	Target	Reporter assay	Western blot	qPCR	Microarray	NGS	pSILAC	Other	CLIP-Seq	Sum	# of papers
MIRT000415	Homo sapiens	Homo sapiens	hsa-let-7a-5p	CDK6	✓	✓			✓		✓		4	3
MIRT000416	Homo sapiens	Homo sapiens	hsa-let-7a-5p	CDC25A							✓		1	1
MIRT000417	Homo sapiens	Homo sapiens	hsa-let-7a-5p	MYC		✓	✓				✓		3	6

Figure 2.6: Available sources of validated interactions from [5]

Moreover, the multiMIR database was used in the circGPA paper through the R programming language library. It must be noted that the MultiMIR is not the only database used to collect prior knowledge in circGPA.

2.2.4 MDS gene expression data

The data obtained for the experimentation purposes is a table of gene expression counts from 77 samples. Each sample represents a certain patient diagnosed with *Myelodysplastic syndrome (myelodysplasia, MDS)* [100][13]. There is a total of 21952 genes of interest measured on each patient, that is:

- mRNA: 17287, miRNA: 1656, circRNA 3009

All of other RNAs were omitted due to the specifics of the task. The data could be thought of as a table of positive integers with the size of 77 x 21952. Each positive integer then represents the number of occurrences of a particular RNA.

Count normalization

Counts of individual RNA genes highly differ in their absolute values based on their RNA type, conditions and properties such as *gene length*, *copy-number variations (CNV)*, etc. Because of this, a certain normalization procedure should be defined to convert them to the same scale. In the upcoming experimentation a *Transcript Per Million (TPM)* [99] method is used. The exact procedure is described as follows:

1. Separate the table of all genes by the RNA type. The resulting tables are: *mRNA*, *miRNA*, *circRNA*
2. On each of the table run the following procedure:

[4] https://www.gsea-msigdb.org/gsea/msigdb/
[5] https://mirtarbase.cuhk.edu.cn/

(a) For each gene, let it be the i-th row in the table, the input is given as the 77 count numbers, e.g., counts in each sample. The number of samples corresponds to the number of columns in the table: $(n_{i,1}, n_{i,2}, ..., n_{i,77})$

(b) Normalize every count number for each sample. More precisely, normalize each column's counts to get an expected number of occurrences per 1 transcript in the sample:

$$p_{ij} = \frac{n_{ij}}{\sum\limits_{k=0}^{77} n_{ik}} \tag{2.1}$$

(c) Normalize all of gene counts by the length of a corresponding gene. The longer the gene is, the less number of genes will be transcribed in general if no *copy-number variations*, alternative splicing, etc., are considered [55]:

$$q_{ij} = \frac{p_{ij}}{l_i}, \qquad l_i = \text{ length of gene in terms of number of nucleotides} \tag{2.2}$$

(d) Convert the normalized count number to expected counts for a million transcriptions:

$$TPM_i = q_{ij} \cdot 10^6 \tag{2.3}$$

This is the final value used in upcoming algorithms.

3. Concatenate the tables again into the final table

The per-type separation is crucial in the circRNA analysis due to the low number of occurrences of circRNA molecules in comparison to more common RNA molecules such as mRNA,miRNA, etc. If not applied, circRNA counts are effectively zeroed and the circRNA expression data is not utilized at all.

Chapter 3

Interaction network

The upcoming chapter will be devoted to interaction networks, one of analysis methods of the interactome. In order to introduce the interaction network, a definition of graphical networks and Bayesian networks is needed. After that, a problem of interaction network learning from measured data is being discussed. Particularly, a division into two sub-problems is analyzed – the parameter inference and the structure inference.

3.1 Graphical networks

The motivation for the graphical representation of probability arises from the increasing complexity of real-life systems. Monolithic models are hard to interpret and analyze; thus, the decomposition methods are used to show a probability model's internal structure and properties [26][12]. There are several types of graphical probability models, but they have following standard assumptions:

- Given a set of random variables $A_1, ..., A_N$
- We define a graph $\mathbb{G}$ with $\mathbb{V} = \{\ v_1 = A_1, ..., v_N = A_N\}$
- The graph subsequently defines a decomposition of joint probability as multiplication:

  $$\mathcal{P}(A_1, A_2, ..., A_N) = \prod_k \Psi(C_k) = \prod_k \Psi(\{A_i \in C_k\})$$

  where C_k is some subset of variables and $\Psi(C_k)$ is some general function on it
- Decomposition and its terms Ψ depend on the structure of a graph, e.g., on edges $\mathbb{E}$. So, to define a graphical probability model, it is necessary to define the terms $\Psi(C_k)$ for every possible subset C_k.

As mentioned above, there exist multiple different kinds of graphical probability models. Below are listed two of the most used ones [26]:

- **Markov Random Fields** – an undirected graph with a freedom to define different general functions Ψ. They are typically used in, for example, physical simulation models.
- **Bayesian Networks** – a directed acyclic graph with Ψ defined as the conditional probability function. They are typically used in the statistical analysis field. The reason behind this is the better interpretation of a structure thanks to directed causality of an edge.

This thesis focuses only on the second type due to the subject of interest. Thus, a more detailed definition of Bayesian networks will be given in the following subsection.

3.2 Bayesian Networks

The core idea behind the decomposition of the joint probability can be described using conditional probability decomposition. When given a joint probability function with numerous random variables, it can be decomposed as follows:

$$\mathcal{P}(A_1, A_2, ..., A_n) = \mathcal{P}(A_n | A_1, ..., A_{n-1})\mathcal{P}(A_1, ..., A_{n-1}) =$$

Subsequently, apply the same decomposition on the last term recursively:

$$\mathcal{P}(A_n|A_1,...,A_{n-1})\mathcal{P}(A_1,...,A_{n-1}) = \mathcal{P}(A_n|A_1,...,A_{n-1})\mathcal{P}(A_{n-1}|A_1,...,A_{n-2})\mathcal{P}(A_1,...,A_{n-2}) =$$

$$= [\prod_1^n \mathcal{P}(A_{i+1}|A_1,...,A_i)]\mathcal{P}(A_1)$$

Graphically, each term can be represented as a vertex in complete DAG. See figure 3.1:

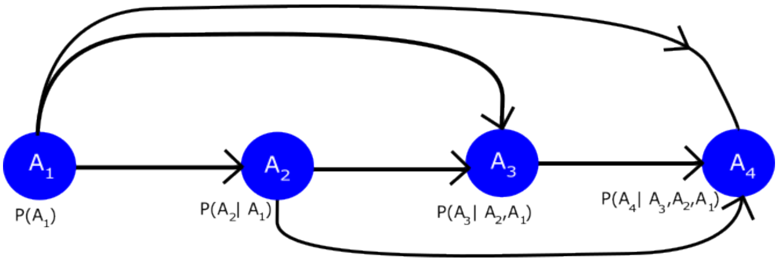


Figure 3.1: Graphical representation of decomposition for n=4

However, the number of edges in a complete DAG increases polynomially with the number of random variables, and usually, not all of the connections are needed [26][12] .

Conditional independence

The removal of an edge indicates the absence of a variable in a conditional probability term. The formal definition of such property can be written as [26][12]:

$$\mathcal{P}(A_k|A_1,.,A_{m-1},\underline{\underline{A_m}},A_{m+1},..,A_{k-1}) = \mathcal{P}(A_k|A_1,.,A_{m-1},\underline{\underline{\;}},A_{m+1},..,A_{k-1}) = \mathcal{P}(A_k|Pa(A_k))$$

where $Pa(A_k)$ denotes the parent variables of node A_k.
This property is called **conditional independence** and can be denoted as:

$$A_k \perp\!\!\!\perp A_m|\ Pa(A_k)$$

As a visualization, see figure 3.2.

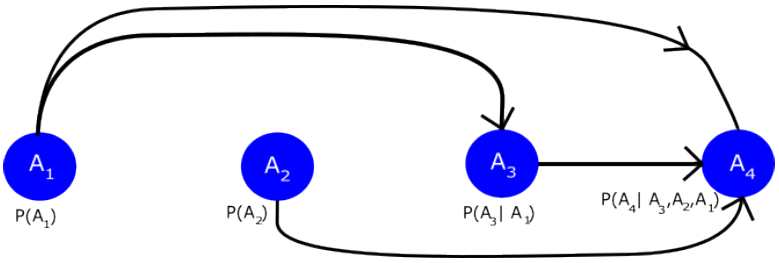


Figure 3.2: The same graph as 3.1, but with $A_1 \perp\!\!\!\perp A_2|\ \emptyset$ and $A_2 \perp\!\!\!\perp A_3|\ \emptyset$

It is worth mentioning that conditional independence goes beyond the introduced simple definition above. As shown in [26][12], conditional independence could be determined for any two variables in a graph, and it can also utilize our partial observations of some variables. This property leads to some algorithms, which this thesis will not cover since they are not employed in frameworks analyzed in the following chapters.

Benefit

The essential advantage and practical benefit is the simplification of the probability function. Initially, the probability function was a complicated joint probability function of N variables:

$$\mathcal{P}(A_1, A_2, ..., A_n) \tag{3.1}$$

Nevertheless, after the decomposition, it was broken into several conditional probabilities of lower parameter dimensions. For a more detailed illustration, see [26].

Definition

We can now fully define a Bayesian network by combining and formalizing statements mentioned above.

Bayesian Network

- Given an ordered set of random variables: A_1, ..., A_N
- Given that only possible edges are from the topological ordering of variables in DAG
- Given a set of conditional independencies: $A_i \perp\!\!\!\perp A_j | \; Pa(A_i)$

We define their joint probability as a decomposition into the multiplication of N different conditional probabilities:

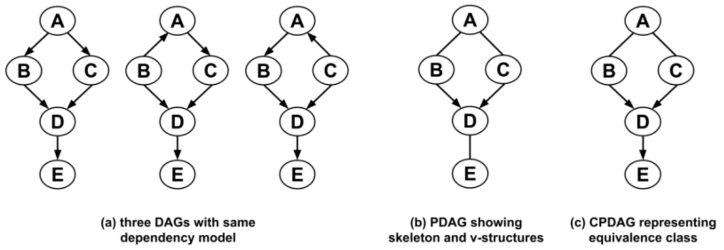
$$\mathcal{P}(A_1, ..., A_n) = [\prod_{i=1}^{n} \mathcal{P}(A_{i+1}|Pa(A_{i+1})]\mathcal{P}(A_1)$$

3.2.1 Equivalence classes and causality effect of the Bayesian networks

The important property of Bayesian networks is the fact that, even if structures of two networks differ, their corresponding probability distributions could still be the same. Alternatively, one may consider the set of networks with the same data fitting score, e.g. *score-equivalent Bayesian networks*. However, even if both networks are equivalently good estimates, the important note to remember is the logical adequacy when choosing between two networks.
As implied in [15], under natural assumptions, an edge that is present in the Bayesian network could be interpreted as the causality of one variable to another. However, in some cases, the direction learned could be misleading and illogical. This is when the prior knowledge about the variables should taken into consideration in order to evaluate an inferred causal effect.

V-structures

The equivalence class of the *Bayesian Networks* could be defined more comprehensively using the *v-structure* and the corresponding *Partial DAG* [96]. A v-structure in a *DAG G* is an ordered triple of nodes *(i, j, k)* such that G contains the directed edges $i \rightarrow j$ and $k \rightarrow j$, and i and k are not adjacent in G [25].



(a) three DAGs with same dependency model

(b) PDAG showing skeleton and v-structures

(c) CPDAG representing equivalence class

Figure 3.3: An illustration of the v-structure that defines the equivalence class. Taken from [96]

As it is stated in [96], any two Bayesian networks have the same underlying joint probability if and only if they have the equivalent v-structure for any appropriate triplet of variables. An equivalence could then be seen as the choice of an orientation of individual undirected edges in the *PDAG*.
For more details and examples of equivalence classes see [20] or [1].

[1] https://www.ime.usp.br/~ddm/courses/mac6916/equivalences/

3.3 Interaction networks

As it was introduced earlier in section 2.1, the interactome is an interaction network of a biological system. It can be seen as a graph network with biopolymers, e.g., macromolecules, as its nodes and their interactions as edges. From here, it is evident that Bayesian network methods could be applied to interactome [18].
This section presents different methods of constructing a Bayesian network from gene expression data.
From the definition of Bayesian networks, we could deduce that:

- Each random variable corresponds to the biopolymer, e.g., each gene will be mapped as one-to-one to some network node.

- The probability of each random variable represents the rate of regulation on a normalized scale. Even though gene expression data is given in absolute counts, it is then converted to a zero-one scale based on the model of choice. On top of normalization performed in section 2.2.4, a selected data fitting probability function (see section 4.1.1) defines a further normalization to zero-one scale.

- The presence of an edge is a sign of a non-zero effect of another biopolymer. Also, if mild assumptions are made [15], the direction implies causality, e.g., the causing gene and the affected gene. The existence of an edge is a typical task that is needed to be accomplished from given expression data. Then, such interactions are used as a set of factors further analyzed from the experimental point of view.

However, it is worth mentioning that each organism has a slightly different regulation rate due to its current environment and organism condition. Additionally, it is possible to have equivalent networks with the same joint distribution but different edge structures ([20] or see section 3.2.1). In this case, biological interpretation is needed to choose the correct network.
To summarize the previously mentioned statements, the construction of the Bayesian interaction network could be divided into two parts:

- Constructing an edge structure, e.g., specifying which nodes are connected by an edge and specify the direction of an edge.

- Given the structure, compute all needed conditional probabilities of each node given its parents, as it is specified by the Bayesian network definition (see section 3.2).

Upcoming learning methods are categorized according to these two criteria.

3.3.1 Learning with known structure

The first learning method set is based on the assumption that an edge structure is given. Thus, only unknown variables are conditional probabilities of each node given its parents.

Exact methods

In such cases, the solution typically consists of solving a single equation by the Maximum likelihood estimation [37][12]:

$$\max L = \frac{1}{K}\sum_{l=1}^{K}\sum_{i=1}^{N}\log P(A_i|Pa(A_i), D_l) \tag{3.2}$$

where $D = D_1, ..., D_K$ are the individual gene expression samples.

Alternatively, if it is assumed that non-observable variables exist or some information on counts is missing, then an EM algorithm [12][6] could be used to learn such a network. However, EM-based approach is not naturally used in a gene expression task. On the other hand, such techniques are utilized in Dynamical Bayesian networks [29], which will not be covered in detail.

Approximate methods

Previously mentioned explicit approaches could be computationally costly and sometimes cannot be used in practice. In such cases, more convenient methods to use are so-called sampling approximation methods.
As the name suggests, approximation methods are based on the idea that precise computation of conditional probabilities is impractical and infeasible. Thus, an alternative algorithmic approach is used to iteratively converge to a local minimum that approximates the previously defined *maximum likelihood estimate*.

- **Gibbs Sampling**
  One of the most used methods is the Monte Carlo sampling technique. It has many applications in different statistical and machine-learning fields [113]. However, only the Gibbs sampling version of Monte-Carlo applied for Bayesian networks is covered due to the subject of interest.
  The Gibbs sampling Monte-Carlo has the following algorithm [17][115]:

  1. Start with an initial state of each variable.
  2. Iterate until convergence:
     (a) Uniformly pick a random variable
     (b) Choose randomly from samples that satisfy all of other fixed variables value.
     (c) Fix a new value according to the picked sample
     (d) Save a new visited state in memory
  3. Construct a conditional probability function based on samples visited above

An assumption of known structure

The previously mentioned situation of computationally costly approach is typically related to the fact that an arbitrary subset of joint distribution variables needs to be sampled from the network. This is not the case in our research and, thus, the probabilities are only needed for parametrization of edges of a network, which is polynomial in the number of nodes and is a solvable task. For an example of such algorithm see [23].

The problem, however, arises with the assumption of known structure. As mentioned earlier in section 2.2.1, the primary source of such interactions is an expert knowledge, e.g., in our case – databases with experimentally or algorithmically proved interactions. Although, as it was mentioned in the same chapter, recently discovered biopolymers, such as circRNA, do not have sufficient known interactions in databases yet. Moreover, while being proven algorithmically, several interactions could be misleading in our particular case, as such interactions could be negligible or inactive in a particular experimental condition.

Finally, learning without a known structure is indeed a NP-hard task [9]. In other words, the inference of a structure of a general Bayesian network cannot be polynomially solved under the assumption of $P \neq NP$ [9]. Thus, certain techniques and assumptions about the Bayesian network are needed for given enormous input data. The research in this field is the primary interest for this thesis and, because of the vastness of approaches used, the entire next chapter is devoted to the overview of such methods.

Chapter 4

A global review of Bayesian network structure learning

This chapter comprehensively reviews relevant Bayesian network structure learning approaches. The structure of the chapter will be as follows. For each category of algorithms we will follow the same schematic; first – general components of any underlying inference algorithm are presented, such as scoring metrics and optimization goals. Next – several state-of-the-art algorithms are listed with a brief description of an algorithm. Finally, after all three categories of existing Bayesian network structure learning approaches are presented – a table with their properties comparison is shown, and, based on this table, the most suitable one for our MDS/circGPA task will be chosen in the subsequent chapters.

4.1 Search-and-Score methods

The first category of Bayesian network structure learning algorithms is based on the heuristic search of the space containing some representation of structures [96]. First, the definition of basic concepts is introduced – scoring metric, optimization goal and search space representation.

4.1.1 Scoring metrics

As for any optimization task, structure learning is based on score maximization with a particular set of parameters. Furthermore, the inference is conducted from samples; thus, the score must represent the underlying probability distribution of the data. Consequently, the maximization task is the same as computing the maximum likelihood estimate of the posterior:

$$score(\mathcal{G}, \mathcal{D}) \propto P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{D})}$$

where $\mathcal{G}$ is a structure and $\mathcal{D}$ are the data given.
The typical choice for the score is the simplification of the Bayes theorem formulation as:

$$score(\mathcal{G}, \mathcal{D}) \propto P(\mathcal{D}|\mathcal{G})P(\mathcal{G})$$

where it is needed to define both parts of the score [70]:

- $P(\mathcal{D}|\mathcal{G})$ = data fitting score
- $P(\mathcal{G})$ = structure learning score

The typical assumption used in all of state-of-the-art algorithms is the independence of individual variable scores:

$$P(\mathcal{G}|\mathcal{D}) = \prod_{i=1}^{n} P(A_i|Pa(A_i), \mathcal{D}) \tag{4.1}$$

In the remain of this subsection, the most common formulations of these score components are presented.

Data fitting score

When dealing with the data's fitting score, the individual scoring functions differ depending on the type of values stored in each bayesian network variable. To be more precise, the scoring functions could be formulated either for the Bayesian network with all discrete variables or for the network with at least one continuous variable.

1. **BDe, Bayesian Dirichlet metric**
   In the case of a fully discrete Bayesian network, the Bayesian Dirichlet metric is commonly used [70]. The core idea behind this scoring function is that since every variable has a discrete, e.g., finite, set of states, we can approximate its likelihood as Multinomial distribution [4]:

$$P(\mathbf{X_i}|\mathbf{Pa}(\mathbf{X_i})) \sim \mathit{Multinomial}(\mathbf{X_i}|\mathbf{Pa}(\mathbf{X_i})) \tag{4.2}$$

   Furthermore, since all possible values are known beforehand and are finite, the underlying probability could be reduced to a table in form:

$$P(\mathbf{X_i}|\mathbf{Pa}(\mathbf{X_i})) = P(\mathbf{X_i} = r_i|\mathbf{Pa}(\mathbf{X_i}) = q_j) \tag{4.3}$$

   where r_i is a possible value of the variable $\mathbf{X_i}$ and the q_j is a tuple of the parent variables' values. Such multinomial distribution has a known formula based on the unknown parameter θ. However, instead of trying to approximate the unknown parameter θ, the following approach is used instead:
   Assume that:

   - $P_{\mathcal{D}}(\theta|\mathbf{Pa}(\mathbf{X_i})$ to be a conjugate prior [10] to the multinomial distribution, e.g., *Dirichlet distribution*
   - Parameter independence, e.g., the parameter θ_i of the variable X_i is independent of any other θ_j
   - No missing data is allowed

   consequently, this unknown parameter could be removed if the continuous-variable Bayes theorem marginalization is applied:

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G}, \theta)P(\theta|\mathcal{G})d\theta = \tag{4.4}$$

$$= \prod P_{\mathcal{D}}(\mathbf{X_i}|\mathbf{Pa}(\mathbf{X_i})) = \prod \int P_{\mathcal{D}}(\mathbf{X_i}|\mathbf{Pa}(\mathbf{X_i}, \theta)P_{\mathcal{D}}(\theta|\mathbf{Pa}(\mathbf{X_i})d\theta = \tag{4.5}$$

   then, the entire equation 4.5 can be solved in a closed form [70]:

$$= \prod_{i=1}^{N} \prod_{j=1}^{q} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=1}^{r} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})} \tag{4.6}$$

   where Γ is the Gamma function [61], n_{ijk} are counts that could be derived from the data only, r is the number of all possible values of r_i and the q is the number of all possible tuple values of q_j. And the α_{ijk} is the parameter that defines the particular type of the BD score [70]:

   - The most commonly used BD metric is the **Bayesian Dirichlet equivalent uniform (BDeu)**: $\alpha_{ijk} = \frac{\alpha}{(r_i q_i)}$, where α is some constant.

   For more details see[1] or [70].

2. **BGe, Bayesian Gaussian equivalent metric**
   Key assumptions and the core idea is similar to the previous case. However, this scenario is if any continuous, e.g., non-discrete, variable is present. Consequently, the approximation is made using the Gaussian normal distribution. Discrete variables are then used as coefficients in a mixture of the corresponding parent Gaussians [70].
   Furthermore, the posterior is computed as a whole, not per variable, as in the **BDe** case:

$$P(\mathcal{D}|\mathcal{G}) \sim \mathcal{N}(\mathcal{D}|\mu, \mathbf{W})$$

[1] https://www.cs.helsinki.fi/u/bmmalone/probabilistic-models-spring-2014/ScoringFunctions.pdf

Given that Gaussian normal distribution has two parameters, e.g., mean(μ) and covariance matrix($\mathbf{W}$), let us define two distributions of these parameters:

$$P_{\mathcal{D}}(\mu|\mathcal{G},\mathbf{W}) \sim \mathcal{N}(\nu,\alpha_\mu\mathbf{W})$$

$$P_{\mathcal{D}}(\mathbf{W}|\mathcal{G}) \sim \mathcal{W}(\alpha_w,\mathbf{T})$$

where $\mathcal{W}$ is the Wishart distribution. These distributions are then used to construct a conjugate prior function. In order to derive the conjugate prior, let us apply the Bayes theorem:

$$P_{\mathcal{D}}(\mu,\mathbf{W}|\mathcal{G}) = P_{\mathcal{D}}(\mu|\mathcal{G},\mathbf{W})P_{\mathcal{D}}(\mathbf{W}|\mathcal{G}) \sim \mathcal{N}(\nu,\alpha_\mu\mathbf{W})\mathcal{W}(\alpha_w,\mathbf{T}) = \mathcal{NW}(\nu,\alpha_\mu,\mathbf{T},\alpha_w)$$

where $\mathcal{NW}(-)$ is the Normal-Wishart distribution[2]. This leads to an equation [66]:

$$P(\mathcal{D}|\mathcal{G}) = \int P(\mathcal{D}|\mathcal{G},\mu,\mathbf{W})P(\mu,\mathbf{W}|\mathcal{G})d\mu\ d\mathbf{W} = \tag{4.7}$$

which can be solved in a closed form by using the information from samples only:

$$P(\mathcal{D}|\mathcal{G}) = (\frac{\alpha_\mu}{N+\alpha_\mu})^{l/2}\frac{\Gamma_l((N+\alpha_w-n+l)/2)}{\pi^{lN/2}\Gamma_l((\alpha_w-n+l)/2)}\frac{|T_{\mathbf{YY}}|^{\alpha_w-n+l}}{|S_{\mathbf{YY}}|^{N+\alpha_w-n+l}} \tag{4.8}$$

where N is the number of samples, n is the number of nodes.
This thesis will not cover every variable, precise definition of the formula above nor the exact calculation due to its complexity. For a more detailed approach, see original papers [95][50][56].

Structure learning score

The next component of the scoring metric is the structure learning score. As [22] points out, the structure score term is not dependent on the data given, and at the early stages of Bayesian network research, it did not receive enough attention. However, as [22] states, the later research has shown that with the data sample size being significantly smaller than the number of variables, the structure score becomes dominant.

1. **Uniform or complexity penalty**
As mentioned above, the early-day Bayesian network research tended to eliminate the term by setting it to the uniform constant. The reasons behind such assumptions were rather pragmatic – the lack of computing power did not allow for a more complex approach [22]:

$$P(\mathcal{G}) \propto 1$$

Alternatively, if the prior is used as a complexity regularization, the term could be considered a penalty for a more dense network [22]:

$$P(\mathcal{G}) \propto \prod_{i=1}^{n}\binom{n-1}{|Pa_{\mathcal{G}}(X_i)|}^{-1}$$

2. **Prior knowledge function**
Otherwise, there exists another approach for the structure score. Since a structure term is defined as a prior distribution of network structures, prior knowledge about particular edges can be incorporated into the score. Example of such score function form is [104][51]:

$$P(\mathcal{G}) = \frac{e^{-\beta}E(\mathcal{G})}{Z(\beta)} \tag{4.9}$$

where $E(\mathcal{G})$ is so-called *Energy function*:

$$E(G) = \sum_{j=1}^{N}\varepsilon(X_j,X_{pa_j(G)}) \tag{4.10}$$

$$\varepsilon(X_j,X_{pa_j(G)}) = \sum_{i\in X_{pa_j}}(1-B_{ij}) + \sum_{i\notin X_{pa_j}}B_{ij} \tag{4.11}$$

[2] https://www.hellenicaworld.com/Science/Mathematics/en/NormalWishartdistribution.html

where B_{ij} is the incorporation of the priod knowledge in form of:

$$B_{ij} = \begin{cases} \langle 0, 0.5) \,, & \text{A prior knowledge about the absence of an edge } i \to j \\ 0.5, & \text{No prior knowledge is given for an edge } i \to j \\ (0.5, 1\rangle \,, & \text{A prior knowledge about the presence of an edge } i \to j \end{cases} \tag{4.12}$$

and the exact value in a given interval represents the power of the prior knowledge.

Information-theoretic scores

This is the class of score functions from the information theory that aims to prevent overfitting of the data by penalizing the dense models. In comparison to previously defined scoring functions, e.g., Bayesian-decomposable score functions, Information-theoretic functions are not defined in a multiplicative manner. Their general form could be expressed as [96]:

$$S(G, D) = \text{log} p(D|G) - \Delta(D, G) \tag{4.13}$$

And the particular choice of each of these two functions lead to the particular score [96]. For example:

- **Log-likelihood score**
  Represents the data-only inference score, no network complexity penalty

  $$\log p_{LL}(D|G) = \sum\sum\sum N_{ijk} \text{log} \frac{N_{ijk}}{N_{ij}} \tag{4.14}$$

  $$\Delta(D, G) = 0 \tag{4.15}$$

  where N_{ijk} is the count of variable x_i being in state j and having its k^{th} parent set configuration.

- **AIC score**
  Penalizes the complexity by computing a number of degrees of freedom of the discrete data. The data fitting score remains the same as in the Log-likelihood score.

  $$\log p(D|G) = \log p_{LL}(D|G) \tag{4.16}$$

  $$\Delta(D, G) = \sum_{i=1}^{n} (r_i - 1) q_i \tag{4.17}$$

  where r_i is the number of states of the node variable x_i, q_i is the number of all possible parents sets of the variable x_i.

The main benefit of the Information-theoretic score is the absence of parameters that are being estimated from the data. Thus, a score could be used without any assumption placed on the data, but if certain assumption or prior knowledge about the studied data could be placed, the previous scores are preferred [96].

4.1.2 Optimization goal

The algorithm output is the next component needed to establish the Score-and-Search learning. In other words, given the scoring metric, the maximization can be performed in the following ways:

- **Maximum A Posteriori**
  The most straightforward method is an attempt to compute a global score maximum, which can be formalized as [70]:

  $$\arg\max_{\mathcal{G}} score(\mathcal{G}, \mathcal{D}) = \arg\max_{\mathcal{G}} P(\mathcal{G}|\mathcal{D}) \tag{4.18}$$

- **Bayesian model averaging**
  The other possibility is the consideration of the bayesian approach for the estimation. The given data samples could represent any possible Bayesian network, each with the probability given by the score. Then, to calculate the optimal estimation, it is necessary to compute the expectation value of the network

distribution. A possible way to average over all structures is to analyze each so-called *structural feature*, e.g., each possible edge in a DAG [22]:

$$P(f|\mathcal{G}) = \sum_{\mathcal{G}} f(\mathcal{G}) P(\mathcal{G}|\mathcal{D}) \tag{4.19}$$

where $f(\mathcal{G})$ is 1 if the *structural feature* f is present in the network $\mathcal{G}$ and 0 otherwise. Moreover, it is necessary to point out that since a Bayesian network is a directed graph, each direction of an edge is considered to be a distinct *structural feature f.*

The last note on the optimization goal is that above-mentioned equations are both super-exponential and impractical to be computed explicitly. Because of this property, nearly all of algorithms analyzed in this section either use some property of the Bayesian network, for example, equivalence classes of multiple structures, or simply approximate by the local optimum.

4.1.3 Search space

The space of structures must contain either all or a particular subset of the possible Bayesian network structures. The main difference is how the algorithm interacts with structure space and how it determines the neighborhood of the particular structure.

- **Structures space**
  The most straightforward way to encode structures in space is to store them as a graph. Then, the neighborhood could be considered an edge modification, where all adjacent structures have a difference of single edge change, for example, edge addition, removal, or reversal [104].
  The advantage of such representation is the simplicity of interacting with it, which allows using existing graph traversal algorithms. There exist, however, several significant disadvantages.

  - The representation does not enforce the DAG property of the Bayesian network and, thus, the structure space contains structures not allowed by the task. Even though the search algorithm does not fall into such unacceptable structures, the spatial neighborhood creates an asymmetry between the structures, leading to a sharp and edgy surface with peaking local maxima. Consequently, the search algorithm depends on a starting point and converges to the nearest local maximum [22].
  - Due to the super-exponential growth of the number of structures, the traversal through such space becomes much slower as the number of variables grows [22][64].

  The possible solution for the problems above is a global connection via some Bayesian network property. Examples of a global neighborhood are the Markov Blanket Resampling [41] and the reversible jump [33] that change several graph edges simultaneously.

- **Order space**
  In order to partially solve the previously mentioned problems, a different space of structures could be defined. The key concept here is to utilize the DAG property of the Bayesian network, especially its topological ordering property [22].
  A particular topological ordering of variables is denoted as follows:

$$(U_1, U_2, ..., U_n) = (U_1 \prec U_2 \prec ... \prec U_n) = \prec (A_1, A_2, ..., A_n), \qquad U_i \in \mathcal{V} \tag{4.20}$$

  where $\prec$ is the notation for a particular topological ordering.
  Therefore, given a particular ordering, we define a limited subspace of the whole structure space from previous. This limitation effectively reduces the number of all possible structures to exponential complexity:

$$Pa(U_1) = 2^0 \text{ possible parent combinations, } Pa(U_2) = 2^1 \text{ par.combinations, } Pa(U_3) = 2^2, \text{ ...} \tag{4.21}$$

  Which effectively combines into the total number of structures for a given ordering [101]:

$$\sum_{k=0}^{n} 2^k = 2^n - 1 \tag{4.22}$$

which is orders of magnitude less than the total number of possible structures: $2^{\binom{n}{2}}$

To summarize the above-mentioned representation, the entire space combines two components – the ordering and the corresponding subspace. The neighborhood of a structure is now enriched with the connection to structures far away in a structured space but connected via order similarity or equality. The order neighborhood is a single move in an order space, such as a variable position swap [22][101]:

$$(U_1, ..., U_i, ..., U_j, ..., U_n) \rightarrow (U_1, ..., U_j, ..., U_i, ..., U_n) \tag{4.23}$$

Such connectivity makes the space smoother, and the traversal is more likely to find the same optima, even when starting from far starting points.

However, since the scoring metric mentioned previously is applied to a single structure only, the scoring of a particular ordering needs to be introduced. In order to compute an ordering score, it is necessary to apply the per-variable decomposition of the previous score function [101]:

$$P(\prec|\mathcal{D}) = \sum_{G \in \mathcal{G}_\prec} P(G|\mathcal{D}) = \sum_{G \in \mathcal{G}_\prec} \prod_{A_i \in \mathcal{V}} P(A_i|Pa(A_i), \mathcal{D}) = \prod_{i=1}^{n} \sum_{Pa(A_i) \in \mathcal{G}_\prec} P(A_i|Pa(A_i), \mathcal{D}) \tag{4.24}$$

where $\mathcal{G}_\prec$ is the subspace of structures that follow the ordering $\prec$.

Lastly, it is necessary to mention that the ordering could be enhanced even further with more subset divisions, such as partial orderings [45].

4.1.4 Space traversal strategies

Given the search space, it is necessary to define an algorithm to find the best possible structure among them. However, as mentioned earlier, the straightforward exhaustive search is intractable due to the super-exponential number of structures. Moreover, it was shown that the optimal Bayesian structure inference is an NP-hard problem, even when the limitation for the maximum number of parents is set [44]. Consequently, most algorithms focus on approximating the optimal structure.

- **Greedy hill climbing**
  This particular version of greedy hill climbing algorithm was taken from [28], but rewritten in the algorithmic form. The idea is straightforward – proceed to do small local moves until the local maximum is found:

Algorithm 1 Greedy hill climbing method

```
procedure GHC($\mathcal{G}$, G, S)
    Input: Search space $\mathcal{G}$, initial point G, scoring function S
    Output: Best found point from $\mathcal{G}$
    $G_r \leftarrow$ G
    $S_r \leftarrow$ S(G)
    while $G_r$ changed after iteration do
        for every possible move in Search do
            $G_{cand} \leftarrow$ apply(G, move)
            if $S(G_{cand}) > S_r$ then
                $S_r \leftarrow S(G_{cand})$
                $G_r \leftarrow G_{cand}$
            end if
        end for
    end while
    return highest-scoring DAG $G_r$
end procedure
```

- **MCMC**
  Alternatively, it is possible to randomly sample individual elements from the search space to simulate the

underlying posterior distribution. In other words, individual elements become states of the Markov chain, and the probability of each move is calculated according to their scores [96]. The output of the algorithm is the stationary, e.g., converged state at the end of a simulation.
However, the MCMC itself is not a particular algorithm, but a general approach for the search space traversal.

  - **Metropolis-Hastings**
    One of the most used MCMC-based sampling variant [104][51] is the Metropolis-Hastings algorithm. The algorithm can be described as applying the previously introduced Gibbs sampling algorithm to a high-dimensional case. The high-dimensional elements that are being sampled are the search space elements, e.g., edge structures, order spaces, etc. Then, the algorithm is applied as follows for Bayesian network learning [64]:

    1. Start with some initial search space element
    2. Define a distribution Q of all possible subsequent states. It could be a single-edge move, e.g., edge addition, edge deletion, edge reversal [104], or a more significant transition, such as Markov blanket resampling (MBR) [41][63] or ordering swap.
    3. Iterate until convergence:
       (a) Sample a new edge structure from distribution: $\sim$ Q
       (b) Calculate an acceptance ratio:

       $$\alpha = \frac{f(\text{new structure})}{f(\text{old structure})} \quad \propto \quad \frac{P(\text{new structure})}{P(\text{old structure})} \tag{4.25}$$

       where $f = P(\text{data}|\text{network}) \cdot P(\text{network}) \cdot Q(\text{network}|\text{other network})$
       (c) Accept or reject based on the acceptance ratio as the probability of acceptance.
    4. Return the converged edge structure

- **Exact traversal methods**
  Finally, there exist certain methods that conduct the exhaustive search on the entire search space and return the highest scoring Bayesian network globally.

  - **Dynamic Programming**
    Previously mentioned order/partial ordering space could be easily converted into a hierarchical problem, where the solution of the global optimization problem is decomposed into the large set of much smaller search spaces. Such approach is called the dynamical programming and there exist several papers [22][44] that cover the possible algorithms. However, as the authors state, the approach is typically further enhanced by some heuristic that tries to bound the solution and cut-off unnecessary subspaces [96].
    Generally, such approaches have an exponential time complexity [44] and typically have a super-exponential memory requirement [22]. However, if certain assumptions are met, as for the certain scoring function and existing ordering, there exist a much more effective methods such as [8].

  - **Integer Linear Programming (ILP)**
    Linear programming is defined as an optimization problem in form [72]:

    $$\min\ c^T x \tag{4.26}$$

    $$\text{s.t.}\ \ Ax \geq b$$

    And the ILP could be defined as the LP problem with a restriction to the integer variables domain. The flexibility of such formulation leads to a large class of problems that could be represented using such notation. However, the key advantage is the existence of effective solvers that could solve almost any ILP formulated problem. An example of such automatic solver is .
    As a consequence, the problem solving is shifted from the search of an actual algorithm to a reduction of the task to the ILP formulation. Due to the extensive size of the existing reductions, such formulations won't be covered here. For more details refer to the original papers [38][65].

[2] https://www.gurobi.com/

- **Other methods**
  Lastly, the global review did not cover the entirety of the existing hundreds of the search-and-bound approaches and only gave a overall view for the directions that the research in this field is being conducted. The reason for such short overview is the reality of the bioinformatics field where the majority of the existing methods are limited.

4.1.5 List of representative algorithms and their properties

- **IntOMICS** [104]
  - Score: BGe + Prior
  - Optimization goal: MAP model
  - Search method: structure space, Metropolis-Hastings(MCMC)
  - Limitations: maximum in-degree parent limitation, maximum of 100s nodes
- **Partial-Ordering MCMC** [34][45]
  - Score: K2
  - Optimization goal: Bayesian model averaging
  - Search method: (partial) order space, Metropolis-Hastings(MCMC)
  - Limitations: maximum in-degree parent limitation, maximum of 100s nodes
- **Greedy Equivalence Search** [49]
  - Score: BDeu/AIC
  - Optimization goal: MAP model
  - Search method: Greedy Hill climbing
  - Limitations: maximum in-degree parent limitation, maximum of 100s nodes
- **Parent set identification** [62]
  - Score: BIC only
  - Optimization goal: MAP model
  - Search method: Dynamic Programming
  - Limitations: maximum of 1000 nodes

4.2 Constraint-based methods

An alternative approach to the Bayesian network inference is to determine the conditional independencies locally by a statistical test. As the name suggests, the constraints represent the subset of parents with a direct connection to the analyzed variable. The essential advantage of such methods is the scalability of the algorithms. While the previous search and score methods are limited to hundreds of variables, often with maximum-in-degree limitation, the local constraint methods are proven useful even with tens of thousands of variables [28][25].

4.2.1 Faithfulness assumption

However, the scalability comes with the cost of optimality. Theoretically, such algorithms return an optimal solution, but with a strict and, in most cases, unsatisfied assumption.

Conceptually: All conditional independencies, e.g., entire distribution, can be inferred from the d-separation or a graph. Formally:
For any subset $s \subset V$

$$\mathbf{X}_i \perp\!\!\!\perp \mathbf{X}_j \mid \mathbf{X}^s \iff \mathbf{X}_i \text{ is d-separated from } \mathbf{X}_j \text{ given } \mathbf{X}^s$$

where $\mathbf{X}^s = \{x \in s\}$ [47][25]

Super-set and its problems

As a consequence of faithfulness assumption, the output of any local search algorithm returns a super-set of the truth graph, since no edge, e.g., conditional dependence of two variables, can be present and not show a d-separation property in a statistical test.
The output of the algorithm often could not decide the orientation of the inferred causality and returns a *Partial DAG* or a *Skeleton*. Such output is the representation of some equivalence class of Bayesian networks (see 3.2.1). However, such assumption is considered to be restrictive and only the limited set of distributions follow it [88][54].

Generalized model formulation

The abovementioned approach could be generalized as follows:

- **Generalized metric**
  Let us define a general conditional dependence metric:

$$Assoc(X;T|Z) = \text{ strength of a conditional dependence} \tag{4.27}$$

$$X \perp\!\!\!\perp T \mid Z \Longleftrightarrow (Assoc(X;T|Z) = 0) \tag{4.28}$$

- **Generalized learning**
  Let us define a general learning procedure:
  1. Learn a skeleton/PDAG of the Bayesian network (skeleton phase)
  2. Convert the previously learned PDAG into the Bayesian network DAG (Orientation phase)

The choice of the particular statistical test and the particular learning procedure defines the algorithm. Moreover, not every constraint-based algorithm performs step 2 and could potentially return the skeleton/PDAG as the equivalence class to choose from.

4.2.2 Conditional independence statistical tests

- $\mathcal{G}^2$ **test** [28]
  - If given two discrete variables **A**, **B**, and a set of discrete variables: **C**
  - The alternative hypothesis: *The A variable is independent of B given that C effect both of them*
  - Then, the test statistics is computed as follows:

$$\mathcal{G}^2 = 2 \sum_{a,b,c} S_{abc}^{ABC} \log \frac{S_{abc}^{ABC} S_c^C}{S_{ac}^{AC}} \tag{4.29}$$

    where a,b,c are all of the possible discrete values of variables A, B, C. It should be noted that C variable is indeed a vector of variables, thus, a value is also vectorized. Finally, S_{abc}^{ABC} is the number of samples where $A = a, B = b, C = c$.
  - Finally, the $\mathcal{G}^2$ is compared to the $\mathcal{X}^2$ table value, but with the $df = (|a| - 1)(|b| - 1) \prod_{c \in C} (|c|)$. Here, the $|a|$ is the number of distinct values that the variable A has.
- **Partial correlation, continuous data** [34][90]
  This statistical test is designed to compute the dependence between two variables, while, at the same time, assuming the influence of all the other variables [90].

  - If given two continuous variables **A**, **B**, and a set of continuous variables: **C**
  - The alternative hypothesis: *The A variable is independent of B given that C effect both of them*
  - Then, the test statistics is computed as follows:

∗ Compute the correlation matrix:

$$R(x, y) = \frac{\sum_i (x_i - \text{mean}(x_i))(y_i - \text{mean}(y_i))}{(x_i - \text{mean}(x_i))^2 (y_i - \text{mean}(y_i))^2} \tag{4.30}$$

∗ Compute each sub-matrix r_{MN} by taking M variables on the rows and N variables on columns

∗ Compute the partial correlation as:

$$r_{ABC} = \frac{r_{AB} - r_{AC} r_{BC}}{\sqrt{1 - r_{AC}^2}\sqrt{1 - r_{BC}^2}} \tag{4.31}$$

∗ Compare its value to the *t-student distribution* [31] table value with parameter $df = n - 3$. The number n is then the total number of variables in samples.

4.2.3 Algorithms

- **PC algorithm**
  This particular version of PC algorithm is taken from [87]. The core limitation is the maximum in-degree parent limitation that is denoted as K in the subsequent code:

Algorithm 2 Peter and Clark (PC) algorithm

```
procedure PC-ALGORITHM SKELETON PHASE(V, $\mathcal{D}$)
    Form the complete undirected graph G on nodes V
    k = 0
    sepset = ($\emptyset$ for every ordered pair $V_i \to V_j$)
    repeat
        repeat
            Select an ordered pair of variables $V_i$ and $V_j$ that are adjacent in G, such that $|adj(G; V_i)/\{V_j\}| \geq k$
            for every subset S with $|adj(G; V_i)/\{V_j\}| = k$ do
                if $V_i \perp\!\!\!\perp V_j | S$ then
                    Delete directed edge $V_i \to V_j$
                    Update sepset($V_i, V_j$) = sepset($V_j, V_i$) = S
                end if
            end for
        until all ordered pairs of adjacent variables $V_i$ and $V_j$ with $|adj(G; V_i)/\{V_j\}| \geq k$ and all subsets S with $|adj(G; V_i)/\{V_j\}| = k$ havebeen tested for CI.
        k = k + 1
    until for each ordered pair of adjacent variables $V_i$ and $V_j$, $|adj(G; V_i)/\{V_j\}| \leq k$
    return G, sepset
end procedure
```

There exist several other versions of PC-algorithm such as [25], [43], [46], etc. The reason for this many modifications is the fact that PC-algorithm is considered to be one of state-of-the-art algorithms of constraint-based type.

Algorithm 3 Peter and Clark (PC) algorithm

```
procedure PC-ALGORITHM ORIENTATION PHASE(V, G, sepset)
    for every v-structure $(v_i, v_j, v_k)$ in G of variable such that $v_i \leftrightarrow v_j$ and $v_j \leftrightarrow v_k$, but $v_i, v_k$ is not do
        if $V_j \notin$ sepset$(V_i, V_k)$ then
            Orient $v_i \rightarrow v_j$, $v_k \rightarrow v_j$
        end if
    end for
    for Every unordered edge $v_j \leftrightarrow v_k$ do
        if There exists an directed edge $v_i \rightarrow v_j$ and there is no any edge between $v_i$ and $v_k$ then
            Orient $v_j \rightarrow v_k$
        end if
    end for
    for Every unordered edge $v_i \leftrightarrow v_k$ do
        if There exists an directed path $v_i \rightarrow v_j \rightarrow v_k$ then
            Orient $v_i \rightarrow v_k$
        end if
    end for
    for Every unordered edge $v_i \leftrightarrow v_j$ do
        if There exists two directed path $v_i \rightarrow v_k \rightarrow v_j$ and $v_i \rightarrow v_l \rightarrow v_j$ and there is no any edge between $v_l$ and $v_k$ then
            Orient $v_i \rightarrow v_j$
        end if
    end for
end procedure
```

- **MMPC algorithm**
  The next subalgorithm is presented in its short, undetailed form. For a complete overview see the original paper [28]. This algorithm has no maximum in-degree parent limitation and, thus, is considered to be an alternative state-of-the-art approach:

Algorithm 4 Max-Min Parents and Children (MMPC)

```
procedure PER-NODE-MMPC(node, V, $\mathcal{D}$)
    CPC ← $\emptyset$
    repeat
        for other variable $v \in V$, $v \notin CPC$ do
            $cand_v$ ← Assoc(node, v — CPC)
        end for
        $maxCand$ ← max $_{v \in V, v \notin CPC} cand_v$
        $v_{max}$ ← argmax $_{v \in V, v \notin CPC} cand_v$
        if maxCand ¿ threshold then                ▷ Threshold is defined as global constant, typically 0.05
            $CPC \leftarrow CPC \cup v_{max}$
        end if
    until no changes in CPC
    return CPC
end procedure
```

Algorithm 5 Max-Min Parents and Children (MMPC)

```
procedure MMPC(V, $\mathcal{D}$)
    for every variable $X \in V$ do
        $\mathbf{CPC}_{\text{node}} \leftarrow$ per-node-MMPC(node, V, $\mathcal{D}$)
    end for
    for every variable $X \in V$ do
        if node $\notin$ per-node-MMPC(X, $\mathcal{D}$) then
            $\mathbf{CPC}_{\text{node}} \leftarrow \mathbf{CPC}_{\text{node}}/\{X\}$        ▷ If any of two neightbours didn't find an edge
        end if                                                  ▷ This means that they are cond. independent
    end for
    return CPC
end procedure
```

- **TLPDAG algorithm**
  Finally, there exist an approach that utilizes a similar technique as it was shown in the search-and-score algorithms. The entire task is converted to the *Convex programming* [27] problem, which is a subclass of the Linear programming mentioned above. Therefore, it again utilizes already existing solver algorithms to solve the problem and the task is to formulate the problem effectively. Again, due to the extensive size of the formulation, the exact formulation won't be covered. For more detail refer to the original paper [88].

  However, there is one significant difference in comparison to previously analyzed constrained-based algorithms. That is – the assumptions are less strict now [88]:

  - **Equal variance assumption**
    As it was shown in subsection 4.1.1, when modelling a Bayesian network's probability density, a particular distribution function is assumed. If continuous data case is assumed, the distribution is taken as the mix of Gaussian distributions, each having two parameters of mean μ_i and variance σ_i. The *equal variance* assumption is then defined as the equality of all variance parameters of such Gaussian distributions, e.g. $\sigma_i = \sigma$ for some constant σ.
  - **Degree of reconstructability**
    Let us assume that the true Bayesian network has a adjacency matrix $\mathcal{A}^0$. Then, let us compute its *precision matrix* $\Omega^0 = (I - \mathcal{A}^0)^T(I - \mathcal{A}^0)$ and E, the number of non-zero elements in adjacency matrix $\mathcal{A}^0$. On top of that, let us denote the smallest eigenvalue [79] of the precision matrix as $C_{min}(\Omega^0)$. Then, the *degree of reconstructability* assumes the existance of a lower bound in form of:

    $$C_{min}(\Omega^0) \geq \frac{4}{C \cdot n} \max\,(\log p, E) \tag{4.32}$$

    where n is the number of samples, p is the number of nodes and C is some positive constant.

  Consequently, if the assumptions are held, the algorithm returns the global optimum in polynomial time. If the assumptions are violated, which is often the case when log p $\gg n$, the algorithm still has a better consistency of results than the faithfulness assumption [88].

4.2.4 Limitations

The essential problem of all these scalable algorithms, if max in-degree of PC algorithm is not considered, is the strict assumption of faithfulness. Again, such assumption typically does not hold and the output of these algorithm, while being a comparable local optimum, are still far from the Score-and-Search method output results.
The maximum number of nodes that are still computationally feasible are listed in the table 4.1 at the end of this chapter. The numbers are presented according to their original paper confirmed experiments and, thus, some of these maxima could be misleading, but not in orders-of-magnitude.

4.3 Hybrid methods

The last approach is a combination of the previous two. As shown in [52], previously introduced local constraint methods can compute and return the computationally effective MAP model comparable to the entire search-and-score run for a tractable number of variables. However, when comparing the results with the order-MCMC using the Bayesian model averaging, the order-MCMC outperforms it. Consequently, the fine-tuning of the local search result is needed to approach precision levels of the state-of-the-art MCMC approaches.

4.3.1 General approach

To generalize a previously mentioned approach, let us define a following pseudoalgorithm:

Algorithm 6 Hybrid Bayesian network learning

```
procedure HybridLearning($\mathcal{D}$)
    for number of global iterations do
        Apply the constraint-based algorithm
        Perform a space traversal based on previous result
    end for
end procedure
```

However, it must be noted that some algorithms may not perform more than one global iteration or slightly differ in their algorithm structure. Otherwise, each step has been analyzed in detail in two previous approaches and, therefore, only the concrete hybrid algorithms are listed below.

Algorithms

- **Sparse Candidate algorithm**
  The algorithm is taken from [14]. The pairwise dependency is computed according to the metric [14][62]:
  - **Mutual Information:**

$$I(x;y) = \sum_{y}\sum_{x} P(x,y)\,\log\left(\frac{P(x,y)}{P(x)P(y)}\right) \tag{4.33}$$

  This metric is a part of the *Information-theoretic scores* class mentioned earlier 4.1.1.

Algorithm 7 Sparse Candidate algorithm

INPUT: $\mathcal{D}$ - data samples
V - variables/nodes
k - parameter
score - a score function that is decomposable per variable: $Score(G|\mathcal{D}) = \prod_{X_i \in V} Score(X_i|Pa(X_i),\mathcal{D})$

```
procedure Sparse Candidate(nodes, $\mathcal{D}$, k, score)
    Start with initial network $G_0$
    for n = 1,2,... until convergence do
        Restrict:
        for each variable $X_i \in V$ do
            Find all sets of parent variables, up to size K, and compute their pairwise dependency given $G_{n-1}$ edges
            Define a set with maximum dependency score as $C_i^n$
        end for
        Maximize:
                    Define a search space restricted by parent set $C_i^n$
                    Perform a traversal and find maximum $G_n$
    end for
end procedure
```

- **MMHC algorithm**
  The pseudo-code of an entire algorithm could be summarized as(taken from [28]):

Algorithm 8 Max-Min Hill Climbing (MMHC)

```
procedure MMHC(𝒟, V)
    Input: data 𝒟, nodes V
    Output: MAP Bayesian network
    $CPC \leftarrow MMPC(V, \mathcal{D})$
    <Perform Greedy Hill-Climbing in space restricted by skeleton CPC to subset and orient edges>
    <Allowed local moves are:  add edge, delete edge, reverse edge>
    return highest-scoring DAG from Greedy Hill-Climbing
end procedure
```

- **Skeleton-based (SK)**
  The paper [52] presented two versions of skeleton-based hybrid algorithms. As shown in the same paper, the performance of both versions are comparable and both of them could be applied interchangeably.
  - **SK-MCMC**
    First, the straightforward combination of previous score-and-search and constrained-based approaches could be performed as follows:

Algorithm 9 SK-MCMC

```
procedure SK-MCMC-ITERATION(S, G, 𝒟, V)
    Input: data 𝒟
           nodes V
           current network G
           search space S

    repeat
        Sample a number from [ 0, 1] interval
        if number is in interval with probability $\beta$ then
            $G^{'} \leftarrow$ G with global move applied in space S, for example Markov Blanket Resampling ([41])
            Define Metropolis-Hastings acceptance: $\alpha = \frac{P(G^{'})P(D|G^{'})q_{global}(G|G^{'})}{P(G)P(D|G)q_{global}(G^{'}|G)}$
            Accept $G \leftarrow G^{'}$ with probability $\alpha$ after sampling

        else if number is in interval with probability $1 - \beta$ then
            $G^{'} \leftarrow$ G with local move applied in space S, for example add edge, delete edge, reverse edge
            Define Metropolis-Hastings acceptance: $\alpha = \frac{P(G^{'})P(D|G^{'})q_{local}(G|G^{'})}{P(G)P(D|G)q_{local}(G^{'}|G)}$
            Accept $G \leftarrow G^{'}$ with probability $\alpha$ after sampling
        end if

    until Convergence
    return G
end procedure
```

Such algorithm ultimately converges to the posterior distribution and returns a global MAP network given enough iteration time. Moreover, the initial traversal in skeleton-restricted space increases the convergence rate [52]. Unfortunately, the MCMC part of the algorithm comes with all the limitations of the score-and-search – maximum in-degree and only 100s of variable maximum.

Algorithm 10 SK-MCMC

```
procedure SK-MCMC($\mathcal{D}$, V)
    Input: data $\mathcal{D}$, nodes V
    Output: MAP Bayesian network


    $CPC \leftarrow MMPC(V, \mathcal{D})$
    Initialize some Bayesian network $G$
    $G \leftarrow$ SK-MCMC-iteration(CPC space, G)
    $G \leftarrow$ SK-MCMC-iteration(structure space, G)
    return G
end procedure
```

– **SK-Stochastic Search**
Alternatively, the stochastic search [52] could be performed instead. The space traversal could be understood as performing only the local move part of the above MCMC version. Since the search has no convergence guarantee, the algorithm is essentially a extensive search space traversal, but from a good initial point estimate – the skeleton space traversal.

Algorithm 11 SK-SS inner functions

```
procedure PARTIAL-SK-SS-ITERATION($\mathcal{D}$, V, score, CPC)
    Initialize empty Bayesian network $G$
    $V_G \leftarrow \emptyset$
    for every variable $X \in V$ do
        for every variable $Y \in$ neighbour$(X, CPC)$ do
            $G_{cand} \leftarrow$ G with $X \rightarrow Y$ edge addition
            compute: score($G_{cand}$), score($G$)
            $G \leftarrow$ sample from { $G_{cand}, G$} according to the scores
            $V_G \leftarrow V_G \cup G$
        end for
    end for

    for every variable $X \in V$ do
        for every variable $Y \in$ neighbour$(X, CPC)$ do
            if edge $(X \rightarrow Y)$ in G then
                $G_{cand} \leftarrow$ G with $X \rightarrow Y$ edge deletion
            else
                $G_{cand} \leftarrow$ G with $X \rightarrow Y$ edge addition
            end if
            compute: score($G_{cand}$), score($G$)
            $G \leftarrow$ sample from { $G_{cand}, G$} according to the scores
            $V_G \leftarrow V_G \cup G$
        end for
    end for
    return $V_G$
end procedure
```

Algorithm 12 SK-SS inner functions

```
procedure SK-SS-ITERATION($\mathcal{D}$, V, score, $V_G$)
    Initialize Bayesian network $G$ by sampling from $V_G$
    $G_{optim} \leftarrow G$
    for every variable $X \in V$ do
        for every variable $Y \in V$ do
            if edge $(X \rightarrow Y)$ or edge $(Y \rightarrow X)$ in G then
                skip
            end if
            $G_{cand} \leftarrow$ G with $X \rightarrow Y$ edge addition
            compute: score($G_{cand}$), score($G$)
            $G \leftarrow$ sample from $\{ G_{cand}, G\}$ according to the scores
            $V_G \leftarrow V_G \cup G$
            if score($G$) > score($G_{optim}$) then
                $G_{optim} \leftarrow G$
            end if
        end for
    end for

end procedure
```

Algorithm 13 SK-SS

```
procedure SK-STOCHASTICSEARCH($\mathcal{D}$, V, score)
    Input: data $\mathcal{D}$, nodes V
    Output: MAP Bayesian network

    $CPC \leftarrow MMPC(V, \mathcal{D})$
    $V_G \leftarrow \emptyset$
    for i = 1...I global iterations do
        $V_{cand} \leftarrow$ partial-SK-SS-iteration($\mathcal{D}$, V, score)
        $V_G \leftarrow V_G \cup V_{cand}$
    end for
    $G_{optim} \leftarrow$ empty network
    for i = 1...I global iterations do
        $G_{cand} \leftarrow$ SK-SS-iteration($\mathcal{D}$, V, score, $V_G$)
        if score($G_{cand}$) > score($G_{optim}$) then
            $G_{optim} \leftarrow G_{cand}$
        end if
    end for
    return
end procedure
```

Such algorithm has no convergence guarantee, but it has of a property of:

- **Anytime algorithm** – The more iterations the algorithm runs, the better result it returns. The reasons for such behaviour come from the nature of a stochastic search – it uses the constrain-based algorithm as a better initial point and slowly explores the search space around. In other words, it could be thought of as a local move only version of the previous MCMC algorithm.

4.4 Method comparison and summary

Name	Type	Limitation/Assumption	Run time	Max. variables
Greedy Equivalence Search	Search-and-Score	Max in-degree	Polynomial	100s
Dynamic Programming	Search-and-Score	No	Exponential	100s
Integer Linear Programming	Search-and-Score	Max in-degree	Polynomial	100s
IntOMICS/Structure MCMC	Search-and-Score	Max in-degree	Polynomial (1 iter)	100s
Order/Partial Order MCMC	Search-and-Score	Max in-degree	Polynomial (1 iter)	100s
PC-algorithm (all versions)	Constraint-based	Max in-degree	Polynomial	1000s
MMPC	Constraint-based	Faithfulness	Polynomial	10000s
TLPDAG	Constraint-based	Degree-of-reconstructability	Polynomial	1000s
MMHC	Hybrid	Faithfulness	Polynomial	10000s
SK-SS	Hybrid	Anytime algorithm	Polynomial (1 iter)	10000s
SK-MCMC	Hybrid	Max in-degree	Polynomial	100s

Table 4.1: Comparison of all the algorithms mentioned

To summarize the table above – Search-and-Score methods, while returning a global optimum, require a global move, such as Markov Blanket Resampling (MBR) [41], or some complex space definition, such as order space [101]. Consequently, their memory complexity rises to exponential and, thus, is unsuitable for more than 100s of nodes. On top of that, to achieve hundreds of nodes, the maximum in-degree limitation is being placed on a network.

Constraint-based methods are scalable to 10000s of variables, but return local minima only. The reason for locality are strict assumptions placed for the data.

The only solution left for large networks is to use a suitable hybrid approach for available computation time. Consequently, a comparably good result is obtained for a given time limit.

Chapter 5

Frameworks and Experimentation goals

In this chapter a more detailed description of two key algorithms, e.g. IntOMICS and circGPA, is presented, as well as the intention of their usage in future experimentation. More precisely, the circGPA algorithm, as it was mentioned in previous chapters, is a framework capable of efficient GO term annotation. In order to further improve the algorithm, an effect of experimental setup is being analyzed by comparing three different circGPA versions, two of which are being enchanced by the MDS data supplied. Finally, the IntOMICS algorithm is a part of a pipeline in one of three versions and, thus, more details about its inner structure are given and a problem of enormous input data, unsuitable for an original algorithm, is being discussed.

5.1 circGPA

5.1.1 Algorithm description

As presented earlier, the *circGPA* [105] framework is not the Bayesian network learning framework but a method for further analysis. Even though the thesis follows the paper's tripartite biological model (see 2.2.1) of three RNA types, the approach could also be generalized for other biopolymers.
The entire algorithm's pipeline consists of these 4 steps:

1. Construction of the interaction network.
2. Introducing a statistic based on the interaction network.
3. Converting the probability function of the introduced statistic into the generating function.
4. Solving the annotation task:
   - Each miRNA and mRNA already has a set of GO terms annotated with it
   - Compute number of paths from each circRNA to individual miRNA and mRNA, which leads to the number of path to particular GO terms from a given circRNA.
   - Summarize and order GO terms by the statistic's p-value that is computed based on the number of paths
   - Output an ordered sequence of GO terms associated with a given circRNA

In step 1 the interaction network is constructed based on combined known interactions from different databases (see 2.2.1).
However, as addressed in chapters 3 and 2, the known interactions, e.g., prior knowledge about the interactions, are very limited in the general case when no assumption about the studied organism is given. This issue is further amplified by the fact that little information is known about circRNA interactions. Also, several such interactions are obtained from algorithmic frameworks similar to IntOMICS and are yet to be experimentally confirmed.

5.1.2 circGPA correlation (circGPA-corr)

In order to address such issues a novel modified circGPA approach is yet to be published. Essentially, the following approach is used to modify the circGPA annotation pipeline:

1. Obtain gene expression data to get information about the particular organism with specific conditions, e.g., certain disease-inflicted groups.
2. Extract previously known prior interactions from the different databases, such as miRTarBase, and circInteractome (see 1.1), and use them as the second source of information.
3. For each prior interaction compute a weight that is based on the correlation of two RNA genes in the expression data
4. Compute the weighted sum of paths from each circRNA to particular GO terms
5. Output a weighted ordered sequence of GO terms associated with a given circRNA

This version of circGPA algorithm is called **circGPA corr**.

5.1.3 Bayesian network with prior knowledge (circGPA-BN)

Alternatively, a circGPA could be applied to a more sophisticated network instead of a straightforward prior knowledge usage. As it follows from the previous sections, the subject of interest of the thesis is the modification of step 1.

1. Obtain gene expression data to get information about the particular organism with specific conditions, e.g., certain disease-inflicted groups.
2. Extract previously known prior interactions from the different databases, such as miRTarBase, and circInteractome (see 1.1), and use them as the second source of information.
3. Apply the IntOMICS algorithm or other Bayesian network inference algorithm to construct the empirical interaction network, which results in the interaction network tuned for the studied case.
4. Solve the annotation task with the prior knowledge matrix replaced by empirical interaction network
5. Output an ordered sequence of GO terms associated with a given circRNA

5.1.4 Experimentation goal

In order to compare the three pipelines of circGPA annotation presented above, some test statistic should be introduced to compare the ordered sequences of individual circRNAs:

Spearman's rank order correlation

The formula is taken from [86]:

- Given two difference ordered sequences of the same set elements: $(x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n)$
- Compute their ranks, e.g., ordering indices: $(i_1, ..., i_n), i_k \in \{1, ..., n\}$ $\quad (j_1, ..., j_n), j_k \in \{1, ..., n\}$
- Compute the correlation coefficient $r \in [\,-1, 1]$:

$$r = 1 - \frac{6 \sum_{k=0}^{n} (i_k - j_k)^2}{n(n^2 - 1)} \tag{5.1}$$

- Compare the correlation coefficient's test statistic:

$$t = r\sqrt{\frac{n-2}{1-r^2}} \tag{5.2}$$

  to a *t-distribution* [24] with $df = n - 2$

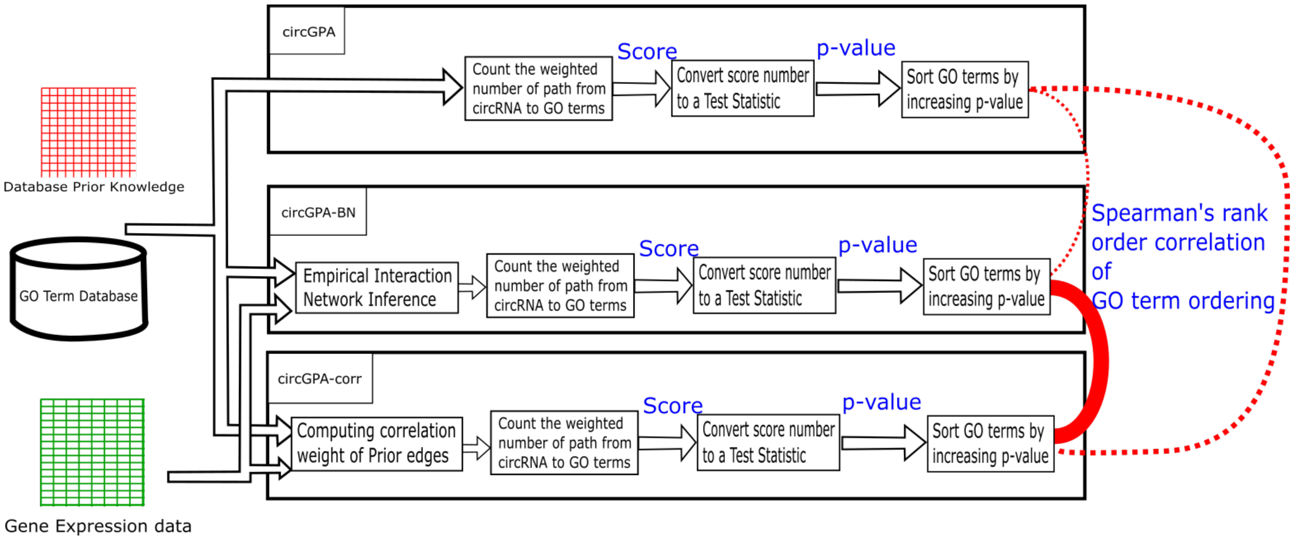

circGPA
Count the weighted number of path from circRNA to GO terms
Score
Convert score number to a Test Statistic
p-value
Sort GO terms by increasing p-value
Database Prior Knowledge
circGPA-BN
Empirical Interaction Network Inference
Count the weighted number of path from circRNA to GO terms
Score
Convert score number to a Test Statistic
p-value
Sort GO terms by increasing p-value
Spearman's rank order correlation of GO term ordering
GO Term Database
circGPA-corr
Computing correlation weight of Prior edges
Count the weighted number of path from circRNA to GO terms
Score
Convert score number to a Test Statistic
p-value
Sort GO terms by increasing p-value
Gene Expression data

Figure 5.1: The schematic of the circGPA version's comparison

Pipeline comparison

Then, if our assumptions about the significance of gene expression data and experimental conditions are true, the Spearman's rank order correlation should output the following situation:

- *SpearmanRankOrder* (circGPA-BN, circGPA-corr) should output a high correlation p-value that is orders of magnitude higher than:
- *SpearmanRankOrder* (circGPA, circGPA-corr), *SpearmanRankOrder* (circGPA, circGPA-BN)

For an illustration see figure 5.1.

However, as mentioned earlier, due to the enormous number of GO terms involved, the computation time of single circRNA is demanding and only a subset of three circRNA is being studied. Theoretically, the circGPA algorithm is modelled per-circRNA, e.g., each circRNA could be analyzed independently of others and no problem. Nonetheless, an orders-of-magnitude difference between the circRNA and GO term numbers may cause the Spearman's rank order correlation test to be ineffective and highly erroneous [73].
To overcome such problems, several additional technique are performed on top of it:

- Each value of Spearman's rank order correlation test is supplied with its standard deviation [19]. The result is then presented in form of a confidence interval, e.g., $value \pm stddev$
- Furthermore, in order to test the ordering correlation even more, a Venn's diagram of first 100 GO term overlapping is presented. The reason for this is high number of non-correlated GO terms that have no path from a given circRNA. Thus, a high overlapping of first, e.g., most significant, GO terms in rank ordering proves the correlation more precisely.
- All the more, previous two tests have an important drawback. While the empirical data may change the weight of individual interactions in case of circGPA-corr, the only allowed interactions are still the ones given by the prior database knowledge. Thus, a space of allowed paths is still constrained, which may result in stronger correlation between circGPA/circGPA-corr.
  To overcome such issue, one may consider to compare the ordering of particular interactions that are related with the experimental setup only, e.g., in our case with the Myelodysplastic syndrome (MDS). That way, if experimental conditions actually influence the observed interactions, then the set of experimentally-related interactions in both circGPA-corr/circGPA-BN would shift to the front. More precisely, the mean of their position should be lower in circGPA-corr/circGPA-BN than in circGPA.

5.2 IntOMICS

IntOMICS algorithm corresponds to the Empirical Interaction network inference box in the schematic shown in figure 5.1.

5.2.1 Algorithm description

The primary idea of the algorithm is based on a combination of both empirical, e.g., studied case-specific data, with the known general prior interactions.
As the initial point, the already mentioned probability functions and transition distribution from the Metropolis-Hastings algorithm will be specified:

- Q(network), e.g., the edge structure transition function is defined as the random action chosen from the single edge move, reverse edge move, and the Markov Blanket resampling [41].
- P(network) will be defined as the probability of prior knowledge satisfaction. E.g., the more the network satisfies the prior interactions, the more probable this part of the posterior probability will be. Particularly [104]:
$$P(\text{edge structure}) = P(\mathbb{G}) = \frac{e^{-\beta E(\mathcal{G})}}{Z(\beta)} \tag{5.3}$$
where $E(\mathcal{G})$ is the energy function that penalizes for not satisfying the prior and β is another parameter to learn. The β is defined as the "power" of the prior interaction versus the empirical interaction. The range of β values is $[0.5, 1]$.
- $P(\text{data}|\text{edge structure})$, on the other hand, forces the network to satisfy the gene expression data interactions derived from the individual gene counts. It is defined based on the BGe score [50], which is the metric that, at its core, computes the correlation but assumes the continuous variable domain with Gaussian multivariate distribution [7]. Even though our counts could be thought of as discrete variables, the IntOMICS algorithm generally allows the continuous features mixed with them, for example, Copy Number Variations (CNV) and Methylation data [104].

As it is clear from the second term, the conventional Metropolis-Hastings algorithm should be modified to include the additional β parameter.
The modification is straightforward. The additional parameter is sampled after the edge structure step with the same Metropolis-Hastings algorithm. However, in order to sample it, we should define a transition distribution and other probability terms:

- $Q(\beta)$ is defined as the uniform probability with variance given by the change of acceptance rate, more precisely – $U(\text{mean} = \beta_{\text{current}}, \text{var} = \sigma)$.
- Acceptance is then defined as:
$$A_\beta = \frac{P(\mathcal{G}|\beta_{\text{new}})}{P(\mathcal{G}|\beta_{\text{old}})} \tag{5.4}$$

Then, a simple version of an algorithm will look like:

1. Run Metropolis-Hastings step on edge structure $\mathcal{G}$
2. Run Metropolis-Hastings step on β
3. Update β acceptance rate based on previous result. Update the σ as well.

However, due to the significant violations of the Markov chain assumptions, the convergence of this so-called Adaptive MCMC simulation is not guaranteed under Markov Chain conditions. It involves several more complex conditions in sampling that need to be done to converge [64].
Although, even with convergence guaranteed, the Adaptive Monte Carlo Markov Chain approximation methods tend to run an impractically long time. This drawback leads to the problem of space complexity when the algorithm runtime limits our input data size. Then, to solve such limitations, different techniques are applied on top of the algorithm to increase the convergence speed. As one such technique, [64] introduces the adaptive MCMC algorithm that first runs in an adaptive form and then runs a conventional MCMC algorithm with the Markov Chain assumptions fulfilled. In details:

1. **1st adaption phase** – runs until the rate of acceptance in Metropolis-Hastings passes a certain threshold.
2. **Transient phase** – fine-tune the beta variance value σ.
3. **2nd adaption phase** – runs until the rate of acceptance in Metropolis-Hastings passes a certain threshold and a fixed beta value is found.
4. **Sampling phase** – conventional MCMC algorithm, e.g., previously introduced Metropolis-Hastings without any changes but starting from the previous converged state.

5.2.2 Limitations of algorithm

The reason for the application of this particular algorithm is the empirical knowledge approach used in it. Unfortunately, as mentioned earlier, the original paper [104] already stated the existence of problems with large network inference.

To understand a source of such limitation, the MCMC approach from a previous chapter should be reviewed:

- MCMC approach, while being one of the state-of-the-art method of Bayesian network learning, is being limited by the following facts:
  - Initially, the computation of Metropolis-Hastings score changes are trivial and does not generate any problem with scalability.
  - However, the Metropolis-Hastings approach violates core Markov assumptions and, thus, the convergence to a global optimum is not guaranteed. In practice, a local-only version of MCMC is highly dependent on a choice of starting point in space traversal, which results in poor local minima.
  - In order to solve this issue, a global move, such as Markov Blanket Resampling (MBR) [41] or more complex space definition, such as Order Space [101], are introduced. The mechanism behind this change is straightforward – either allowing an algorithm to jump and escape from the local minima, or relaxing the search space to remove such problem at all.

    Thus, as a drawback, the computation process of those methods involves an exponential complexity in either space or time domain. Generally, while analyzing interaction networks, maximum in-degree limitation is considered, effectively reducing the exponent to a polynomial with a pre-defined low constant.

    In case of tripartite graph presented earlier, this assumption is not possible, since it breaks the path counter of the circGPA algorithm.
- On top of above-mentioned issues, it must be recalled that the sole reason of IntOMICS and Score-and-Search methods choice is the empirical interaction network. Certainly, if an empirical interaction network is not taken into account, more suitable approaches would be chosen in the early stages. However, the IntOMICS and its counterparts in Score-and-Search domain are the only known algorithms to combine both database (prior) knowledge along with the gene expression data.

5.2.3 Experimentation goal

Consequently – the core task to solve is either to modify and apply the IntOMICS algorithm to a high-dimensional network or, if not possible, to implement an alternative algorithm that computes the MAP estimate of empirical interaction network.

Chapter 6

Empirical interaction network inference algorithm

This chapter is devoted to the problematic of the empirical interaction network inference. As hinted in previous chapters, the originally proposed IntOMICS [104] algorithm has several limitations that have not been resolved yet. Therefore, the structure of this chapter follows the progression of an algorithm choice during the work on this thesis. More precisely, the chapter starts from an attempt to improve the existing IntOMICS algorithm and then proceeds to implement an alternative algorithm to solve the core problem of the MCMC approach.

6.1 Computation improvement of an IntOMICS algorithm

As admitted by the authors of the original paper [104], the algorithm suffers from two fundamental problems:

- Input feature size limitation – up to *tens* of features only
- The time complexity of an algorithm – *superexponential* if not limited by the maximum parent number

The possible solution for both of the problems is proposed in the following section.

An improved formula

The entire derivation of the term is excessively long and it was decided to move the derivation to the separate appendix B. In the end the formula is modified as:

$$Z(\beta) = \prod_{j} \prod_{\text{TYPE}} \sum_{K} \binom{N^j_{\text{type}}}{K} exp\left(-\beta \left[N^j_{\text{type}} C_{\text{type}} + (1 - 2C_{\text{type}})K\right]\right)$$

Histogram addition

The further modification does not change the asymptotic complexity but improves the absolute speed of an improved algorithm. The strategy is easy. If two nodes share an identical set of numbers N_j^{type}, they have the same value in the prior summation. In order to capture this, the histogram count variable is introduced:

$$H(N^j) = H : (N^j_{\text{absent}}, N^j_{\text{noPK}}, N^j_{\text{GE}}, N^j_{\text{TF}}, N^j_{\text{nonGE}}) \to \mathbb{N}_0^+$$

This histogram counts every set of prior edge types of a node. For more details, see the example in the following subsections.

Final form

Lastly, another hidden modification is presented in IntOMICS[1] with a motivation of numerical stability. The entire term Z(b) is converted into its logarithm and further used with the logarithms of other terms.

[1] https://gitlab.ics.muni.cz/bias/intomics

$$\log\, Z(\beta) = \sum_{H} H(N^j) \log \left[\prod_{\text{type}} \sum_{K=0}^{N^j_{\text{type}}} \binom{N^j_{\text{type}}}{K} e^{-\beta\left[N^j_{\text{type}} C_{\text{type}} + (1-2C_{\text{type}})K\right]} \right] \tag{6.1}$$

Improved algorithm

For each node J, the followings values should be computed:

- $(N^j_{\text{absent}}, N^j_{\text{noPK}}, N^j_{\text{GE}}, N^j_{\text{TF}}, N^j_{\text{nonGE}})$

Following that, a histogram of such values should be computed:

- $H(N^j)$ – count of nodes with $(N^j_{\text{absent}}, N^j_{\text{noPK}}, N^j_{\text{GE}}, N^j_{\text{TF}}, N^j_{\text{nonGE}})$

Finally, the log partition function for a particular β can be computed as:

$$\log\, Z(\beta) = \sum_{N^j : H} H(N^j) \log \left[\prod_{\text{type}} \sum_{K=0}^{N^j_{\text{type}}} \binom{N^j_{\text{type}}}{K} e^{-\beta\left[N^j_{\text{type}} C_{\text{type}} + (1-2C_{\text{type}})K\right]} \right] \tag{6.2}$$

Additionally, for an illustration of the work of the improved algorithm, see the example in appendix C.

6.1.1 Practical implementation

All the mentioned modifications are available through GitLab [2].
The previously introduced improved version of the IntOMICS algorithm is available in the following folder on the Gitlab page:

- *~/IntOMICS_orig/*

The modified IntOMICS without the improved algorithm, but with several error fixes, is available on the:

- *~/IntOMICS_noimproved/*

The original source code contains a few implementation errors that prevent the algorithm from running on a sparse interaction network such as our tripartite biological model. The solution to this problem involves the modification of a single file:

- **pf_UB_est.R**

6.1.2 IntOMICS algorithm evaluation

Transient phase convergence

Firstly, the problem with transient phase convergence was encountered. As a possible reason, the convergence time of the IntOMICS algorithm is highly dependent on the number of absent edges, e.g., on the sparsity of the graph. As a result, the IntOMICS algorithm runs an absurdly long amount of time for the tripartite model with gene expression input.
For example, if comparing the original IntOMICS data with the tripartite model data:

- **(IntOMICS, GSE127960_WT)** – 16 input genes, 0 absent edges, 5 prior edges – **113 iterations** at the Transient phase
- **(Tripartite model)** – 18 input genes, 261 absent edges, 6 prior edges – **1746 iterations** at the Transient phase

Moreover, the multiple experiments revealed a high dependence on the start point, defined by the random generator seed. For example:

- seed=5, **(Tripartite model)** – 1746 iterations at the Transient phase

[2] https://gitlab.fel.cvut.cz/anuarali/pim-sk-ss

- seed=6, **(Tripartite model)** – not converged, more than 12000 iterations at the Transient phase
- seed=112, **(IntOMICS, GSE127960_WT)** – 146 iterations at the Transient phase
- seed=113, **(IntOMICS, GSE127960_WT)** – 456 iterations at the Transient phase

Consequently, the tripartite model converged for a specific random seed; thus, an algorithm could be used to generate the empirical interaction network. Nonetheless, the proposed tripartite biological model is unsuitable for the algorithm since convergence is not guaranteed. The convergence time is an order of magnitude higher than the same size interactions network without the restrictions.

Key points of the implementation

- The improved algorithm requires less memory to be stored in the memory to compute the partition function – only the histogram. To compare, the original IntOMICS stores up to 3 variables of an super-exponential size for the entire lifespan of an algorithm:

  1. *all_parents_config* – all possible parent combinations for every variable, needed for both of the terms below
  2. *BGe_score_config* – the BGe score computed for every parent combination, needed for a global Markov Blanket Resampling (*MBR*) move
  3. *Energy_score_config* – the partition function computed for every parent combination, needed for *Upper Bound* computation

  As shown above, the improved algorithm allows to compute the partition function in both polynomial time and memory for the energy function defined in IntOMICS. Thus, only two of three terms is used, effectively reducing the memory consumption by 33%.

- The improved algorithm requires fewer calculations to be performed for our research case. Since the IntOMICS algorithm is performed on the empirical data, most interactions are either unknown, e.g., have no prior, or are explicitly forbidden, e.g., absent prior, as is the case in the tripartite biological model used in this thesis. In both cases, the complexity drops from super-exponential to polynomial for certain parts of the algorithm.

- The only major problem of an improved algorithm is the problem with the maximum allowed node parent number. As introduced in [104][64], the limitation of all nodes' in-degree is performed with the motivation of calculating *Markov Blanket Resampling* (*MBR*).

Conclusion on the algorithm effectiveness

Despite all the optimization performed, the IntOMICS algorithm has been proven to be incapable of learning a tripartite model task for the required number of genes. The problem, however, lies in the MCMC approach as a whole, not in this particular algorithm. As mentioned in the chapter 4, while the MCMC, judging by the number of related algorithms and papers [96], has been the primary state-of-the-art approach for the theoretical learning of the Bayesian networks up until recently. However, majority of these MCMC algorithms failed in generalizing the approach for the large networks with thousands of continuous variables, which is the case for the practical application.

Consequently, by considering all the theoretical obstacles, the IntOMICS algorithm was found to be unsuitable for the further analysis and should be replaced with a more sophisticated one. As hinted previously, the solution for this scalability problem could be found in hybrid approaches.

6.2 Prior-incorporated MMPC Skeleton-Based Stochastic Search (PiM-SK-SS)

6.2.1 Theoretical overview

A recently published paper [52] presented an approach that combines scalable constraint-based methods with the further search-and-score fine-tuning of the results. There are the following key factors for the choice of this particular approach as an alternative for the IntOMICS:

1. **Universality** – instead of providing a strict specific setup or assumptions, the algorithm is presented as conceptual approach, where every part is straightforward to implement depending on the needs of a task.
2. **Scoring metric** – As a result of the previous property, the SK-SS allows the usage of any score function that is compatible with the previous MCMC approaches. While such effect may sound like unnecessary, in fact – it allows the incorporation of the same empirical-based approach as it is used in IntOMICS algorithm [104].
3. **No memory nor maximum parent limitation** – As compared to other novel MCMC-based approaches presented in the same paper [52], the SK-SS algorithm does not require any global moves such as the *Markov Blanket Resampling (MBR)*. Consequently, no super-exponential number of terms is needed to be stored in memory and no need for a maximum parent limitation is needed, as it was the case in the IntOMICS.
4. **Anytime algorithm** – The algorithm does not need to run the entire run time for a sophisticated output. Due to the time limitations, the algorithm will run for the entirety of the time left and the best achieved result will be presented.

Ultimately, with all of the arguments presented, a new approach could be summarized as follows:

1. Apply the constraint-based algorithm on the input gene expression data and construct an undirected skeleton graph
2. Apply the SK-SS algorithm on the skeleton graph returned from step 1 with the scoring function that utilizes the prior knowledge.
3. Iterate the SK-SS algorithm until either:
   - The change of a highest score network is lower than a predefined threshold
   - The number of iterations exceeded the predefined maximum
4. Return a visited directed Bayesian network with a highest score

6.2.2 Max-Min Parents-and-Children (MMPC)

For the task of inferring a network with thousands of variables the MMPC algorithm [28] is an obvious choice, since it is considered to be one of the state-of-the-art algorithms in a constrained-based field. Particularly, in paper [28] it was shown that the comparable-size network, e.g., tiled-ALARM network with approximately 10,000 variables, has successfully been learned in a trackable time, which was the largest learned Bayesian network at the time [28].
There exists a publicly available implementation from *BNlearn* [34] library in R programming language. It was used as a first step of the algorithm implemented afterwards.
However, several adjustments were required to effectively implement the tripartite graph model. The reason for this is the generalized *bnlearn* interface that stores each blocked interaction as two number in a table. Consequently, in order to block all the forbidden interactions, e.g., everything except circRNA-miRNA, miRNA-mRNA, it requires:

$$21952 * 21952(\text{entire table size}) - 17287 * 1656(\text{miRNA-mRNA}) - 1656 * 3009(\text{circRNA-miRNA}) = \tag{6.3}$$
$$= 448280128 \text{ rows of two number}$$

which is hundreds of GB RAM when the original implementation is used. In order to overcome this problem, the tripartite structure was hard-coded into the original library's code through library modification.

Practical implementation

Again, the implementation of adjusted algorithm is available through the same GitLab page [3]https://gitlab.fel.cvut.cz/anuarali/pim-sk-ss. More precisely in those files:

- *∼/source/mmpc_optimized.R* – contains modified code of the original *bnlearn* library
- *∼/workflow.Rmd* – contains code block to access the original library's code in order to overwrite it with modified one from above. See **MMPC application** section in there

Run time

- 16 CPU, 5 GB RAM, Metacentrum
- 2 hours 11 minute 32 seconds

6.2.3 SK-SS implementation

Scoring function

The first key point to define is the scoring metric. The original paper [52] used the following scoring function for the presented example:

- BDeu score for $P(D|G)$
- Uniform complexity penalty for $P(G)$

Instead of this, since the algorithm allows for any suitable scoring function to be used, the slightly modified scoring function from the IntOMICS algorithm is used:

- BGe score for $P(D|G)$, but with a difference in computation compared to the original IntOMICS/BiDAG [104]:
  - Assume a simplified probability for circRNA nodes:

    $$P(D|\text{circRNA}, \emptyset) = 1$$

    This is due to the fact that since no input edges are allowed for this type of RNA, the score does not change during the stochastic search. Moreover, since the BGe score assumes the decomposition of the score per node:

    $$P(D|G) = \prod_{v} P(D|v, Pa(v))$$

    Abovementioned simplification can be thought of as a multiplication by a constant given by the $\prod\limits_{w \in \text{circRNA}} \frac{1}{P(D|w,\emptyset)}$, but practically it allows us to reduce the final value to be more numerically stable in the 64-bit *double* type's format precision [59].

- Modified prior score:
  The key difference is the fact that, unlike the original IntOMICS algorithm, the SK-SS algorithm has no pre-defined procedure of inferring a true β value of the prior probability term. Thus, instead, a different approach used in [51] could be applied. Instead of inferring the true β value, the marginalization across all possible values is computed, which, consequently, leads to the uncertainty score rather than the point estimate. This term is considered to be more robust [51].
  However, first, let us simplify the formula to compute it for a single instance of a graph explicitly:

  $$\frac{1}{\beta_H - \beta_L} \int\limits_{\beta_L}^{\beta_H} e^{-\beta E(G)} d\beta = \text{ assume that E(G) is constant w.r.t to } \beta \text{, denote K} = \frac{1}{\Delta\beta} \int\limits_{\beta_L}^{\beta_H} e^{-\beta K} d\beta =$$

  $$= \text{ solve as a simple definite(Riemann) integral} = \frac{1}{\Delta\beta} \left[-\frac{1}{K} e^{-\beta K} \right]_{\beta_L}^{\beta_H} = \frac{1}{\Delta\beta} \frac{1}{K} \left[e^{-\beta K} \right]_{\beta_H}^{\beta_L} = \text{ substitute E(G)} =$$

  $$= \boxed{\frac{1}{\beta_H - \beta_L} \frac{1}{E(G)} \left[e^{-\beta_L E(G)} - e^{-\beta_H E(G)} \right]} \tag{6.4}$$

  Now, it is necessary to choose both bounds adequately:

  - IntOMICS sets a hardcoded limitation of $\beta \geq 0.5$ [104], because of this we set $\beta_L = 0.5$
  - Bayesian Prior [51] computes the optimal range (based on the AUC curve) to be equal to $\Delta\beta = 10$, because of this we set $\beta_H = \beta_L + \Delta\beta = 10.5$

Such modification incorporates the prior knowledge, thus, allowing us to apply the SK-SS algorithm on larger data than the IntOMICS did. On top of that, it still utilizes the prior knowledge in the same way as the IntOMICS algorithm and essentially achieves the same result.

Technical decisions

During the implementation a few technical problems have been encountered. The core problem of the algorithm is the space complexity. Consider a line from the SK-SS algorithm:

```
$V_G \leftarrow V_G \cup G$
```

While looking like a straightforward simple operation, the problem is in the size of individual G subgraphs. Even if the most efficient method of an adjacency matrix storage is used, that is 1 bit per 1 cell, the total size of the adjacency matrix is in fact:

$$17287 * 1656 \text{ (mRNA x miRNA)} + 1656 * 3009 \text{ (miRNA x circRNA)} = 33610176 \text{ bit} = 4,2 \text{ megabytes} \quad (6.5)$$

Considering that the number of saved subgraphs is, in worst case, the same as the number of undirected edges in skeleton, such straightforward approach is impractical.
To address such issue, a following data structure and an approach is introduced:

- **One-action-per-subgraph**
  If the SK-SS algorithm is to be analyzed, the stochastic search nature gives us the hint about the more effective method of the subgraph storage. Each subsequent subgraph G is being constructed from the previous one by performing a single move. Thus, instead of saving the entire subgraph adjacency matrix, the adjacency-local-action linked list could be used to store the chain of visited subgraphs:

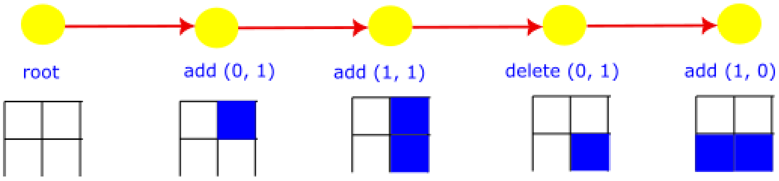


Figure 6.1: The illustration of the adjacency-local-action linked list

- **G-Trie subgraphs**
  However, the problem with the straightforward approach is the violation of the set property – that is, a repeating item should be stored in the set only once despite the number of additions. A possible solution that was implemented is the usage of G-Trie data structure from [53]. For an illustration see the figure below:

  The schematic above could be summarized as follows:

  1. Sample a new move – addition or deletion of an edge on a certain coordinate
  2. Take a coordinate and either add a new local move node or delete a previously inserted one. However, the linked list should be held as sorted by some sorting rule
  3. Use the sorted linked list as a certificate and insert it into the Trie data structure – a way to store a strings efficiently and to determine if the certificate is already present in Trie
  4. **Sampling from** V_G

     (a) Initialize an empty adjacency graph
     (b) Sample an index from a range of 1 to the number of unique subgraphs
     (c) use In-Order traversal in Trie graph and perform one local change at the time – based on the node visited. When going down the graph – perform a move, when going up – perform an inverse move.
     (d) Count the number of leaf nodes encountered and continue up until the encountered index is equal to the sampled index.
     (e) Return the adjacency graph

  However, during the implementation the following problem has been encountered:

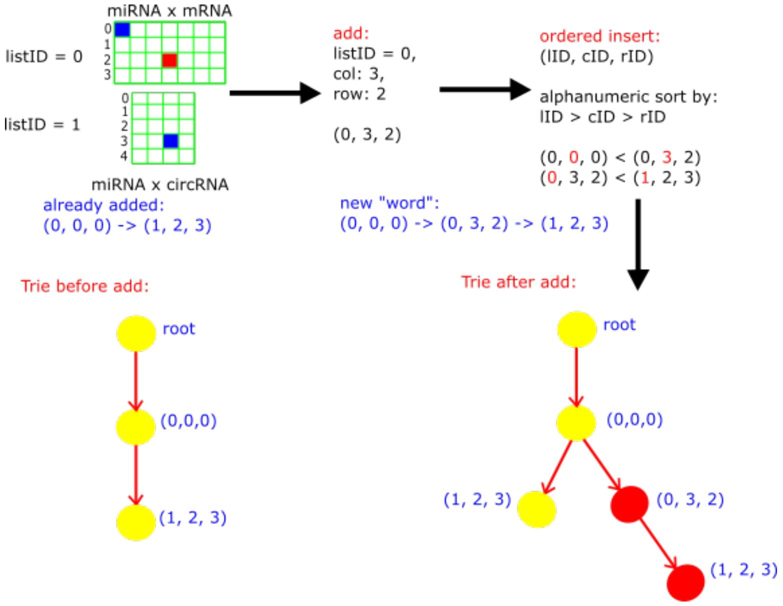


Figure 6.2: The schematic of the algorithm

- **Problem:** Too many nodes in comparison to the number of inserted subgraphs (Example: 2000 subgraphs vs 400k nodes)
  - **Possible solution:** Convert 1-change-per-node Trie to multiple-changes-per-node Trie. For more details see *Compressed trie* on https://www.geeksforgeeks.org/types-of-tries/
  - **Unsolvable part:** Even with the *Compressed trie* usage, the memory overhead is impractical – after half of the partial SK-SS loop the memory requirement is above 200 GB RAM.

Consequently, since the problem is persistent and unsolvable, the only solution left is the non-uniform sampling from the set. That way – instead of storing the entire Trie, only the linked list mentioned above is required to be held in memory.
On top of that, to compensate the violated uniformity an algorithm – instead of running multiple iterations, the algorithm runs for 1 iteration only, but in multiple independent instances.

Practical implementation

The implementation of both algorithms is available through the same GitLab page [4]. More precisely in those files:

- *~/source/skeleton.R* – constains R language's interface as well as the callable functions along with the documentation are placed here
- *~/source/skeleton.cpp* – contains C++ implementation of the R functions from the file above

6.2.4 Run time and memory

The entire hard computation was performed entirely on Metacentrum resources [5].
Because of the shortage of the time, the algorithm run for ≈1 week of pure CPU time on 4 machines and the best result of these three is taken as the output of the thesis. Ideally, the number of iterations should be computed based on the number of new subgraphs after each upcoming iteration. Although, since it is not feasible to keep track of the unique subgraphs due to the space complexity, the other possibility is to track the change of best scoring graph after each N full instance runs of an algorithm.

[4] https://gitlab.fel.cvut.cz/anuarali/pim-sk-ss
[5] https://metavo.metacentrum.cz/

Chapter 7

Experimentation

This chapter is devoted to the practical part of the thesis. The core problem that is being solved in this chapter – is whether the experimental setup has a crucial influence in the gene ontology annotation task. Since the empirical interaction network has already been generated in the previous chapter, the subsequent statistical analysis on three circGPA versions is performed and the significance of an experimental setup is tested. Finally, the biological analysis of the inferred GO annotations concludes the entire thesis.

7.1 A comparison of three circGPA version

The R language's in-built implementation of Spearman rank correlation (Spearman RC) in combination with *SpearmanCI* [1] library is used. *SpearmanCI* has an in-built computation of the standard deviation [19] and the in-build library computes the p-value. The schematic of the code is:

- *cor.test(x=sequence1, y=sequence2, method = 'spearman')*
- *spearmanCI(x=sequence1, y=sequence2)*

Next, due to the extensive time of computation of each circRNA's ordered GO term sequence, only 3 pre-computed circRNA are tested:

- *hsa_circ_0000227*
- *hsa_circ_0000228*
- *hsa_circ_0003793*

On top of that, instead of comparing the ranking order straightforwardly, the p-value sequences are used as an input to the Spearman's rank-order test. This is an equivalent test and has no effect on the result.

7.1.1 Results

The triplet of circRNA genes from above is being tested on all three versions of an circGPA algorithm and the results are summarized below. As mentioned in chapter 5, the reported values are: p-value, confidence interval and the Venn's diagram representing an overlap of first 100 GO terms:

circRNA	Pipeline 1	Pipeline 2	p-value of Spearman RC	r of Spearman RC
hsa_circ_0000227	circGPA	circGPA-corr	$< 2.2\text{e-}16$	0.219 ± 0.02
hsa_circ_0000227	circGPA	circGPA-BN	2.515e-07	-0.051 ± 0.02
hsa_circ_0000227	circGPA-corr	circGPA-BN	0.002	0.030 ± 0.02

Table 7.1: Comparison of pipelines in *hsa_circ_0000227*

[1] https://cran.r-project.org/web/packages/spearmanCI/index.html

circRNA	Pipeline 1	Pipeline 2	p-value of Spearman RC	r of Spearman RC
hsa_circ_0000228	circGPA	circGPA-corr	2.313e-14	0.075 ± 0.02
hsa_circ_0000228	circGPA	circGPA-BN	0.001	0.033 ± 0.02
hsa_circ_0000228	circGPA-corr	circGPA-BN	0.2255	0.012 ± 0.02

Table 7.2: Comparison of pipelines in *hsa_circ_0000228*

circRNA	Pipeline 1	Pipeline 2	p-value of Spearman RC	r of Spearman RC
hsa_circ_0003793	circGPA	circGPA-corr	3.318e-06	0.046 ± 0.02
hsa_circ_0003793	circGPA	circGPA-BN	7.218e-07	0.049 ± 0.02
hsa_circ_0003793	circGPA-corr	circGPA-BN	0.1973	0.013 ± 0.02

Table 7.3: Comparison of pipelines in *hsa_circ_0003793*

To clarify the interpretation of the table – a null hypothesis is the absence of rank correlation between two sequences, e.g., the independence of two sequences. A p-value then represents the probability of null hypothesis acceptance, e.g., the more p-value is, the stronger a connection between two sequences' rank-order is.

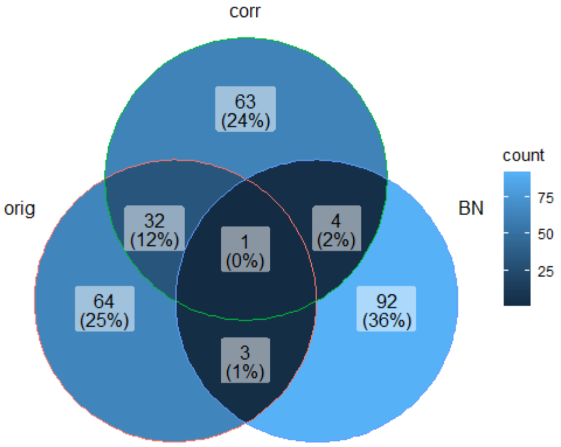


Figure 7.1: Overlapping of first 100 GO terms in *hsa_circ_0000227*

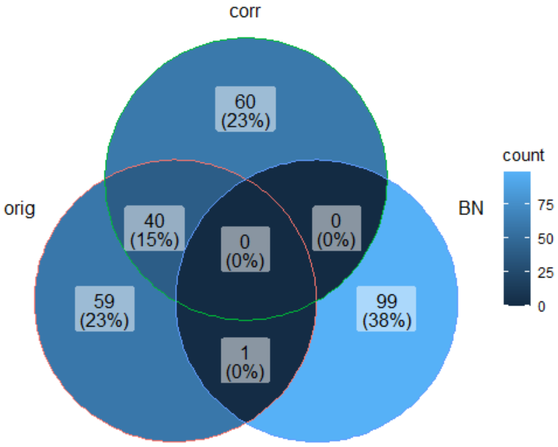


Figure 7.2: Overlapping of first 100 GO terms in *hsa_circ_0000228*

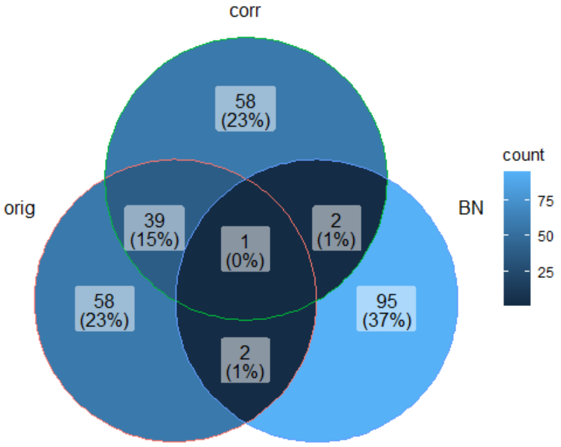


Figure 7.3: Overlapping of first 100 GO terms in *hsa_circ_0003793*

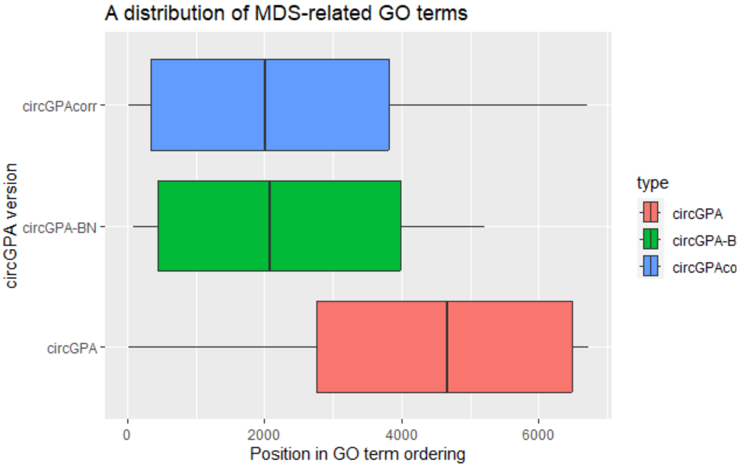


Figure 7.4: Positions of MDS-related GO terms in *hsa_circ_0000227*

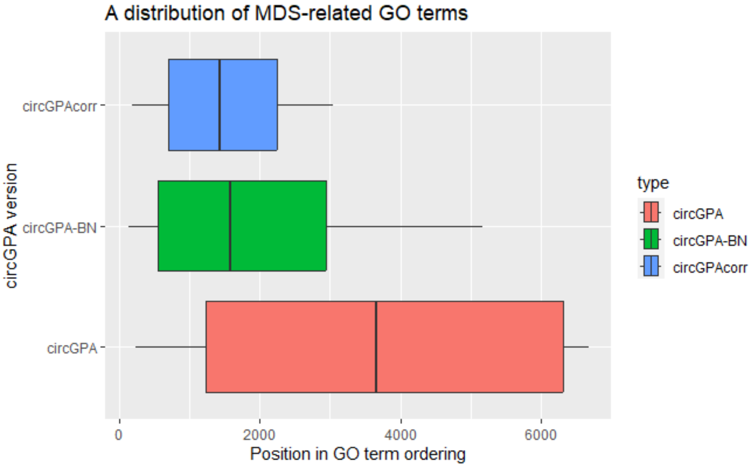


Figure 7.5: Positions of MDS-related GO terms in *hsa_circ_0000228*

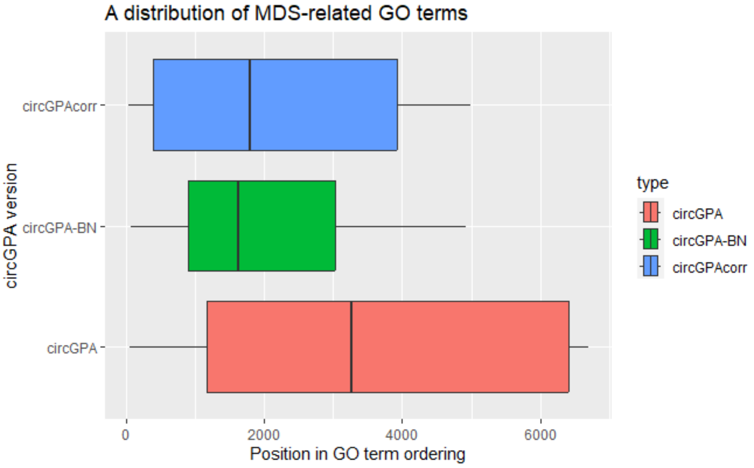


Figure 7.6: Positions of MDS-related GO terms in *hsa_circ_0003793*

7.2 Discussion

7.2.1 Summarized results

Unfortunately, if only the Spearman's rank-order test is taken into account, the results are essentially opposite to the assumption from chapter 5. The possible cause of this paradox is the limitation of Spearman's rank-order test – an assumption of *monotonic relationship* between two p-value sequences [2]. Even though such relationship is being observed with significant p-value, as mentioned earlier in chapter 5, the correlation of two sequences may be influenced by a large portion of GO terms that have a nearly zero score and, thus, make the results misleading.

Furthermore, the same result was observed on Venn's diagrams, where the overlapping of circGPA/circGPAcorr is significant and the overlapping of these two version with circGPA-BN is negligible. Although, result can be interpreted as follows – the experimental setup has an influence, which can be observed in significant differences in circGPA/circGPAcorr, e.g., nearly half of the analyzed GO terms differ just by adding experimental setup weights. Thus, the initial hypothesis works only in case of circGPA/circGPAcorr comparison.
However, if the studied interaction network is not limited to prior knowledge interactions only, then general biological processes interfere with a tripartite biological model. This statement will be proven in the upcoming section, where few leading GO terms of each version are analysed. Particularly, all circGPA versions show us

[2] https://statistics.laerd.com/spss-tutorials/spearmans-rank-order-correlation-using-spss-statistics.php

fundamental processes in a cell, where the ordering of these terms is of no use.

Fortunately, such scenario has been thought out in advance and the corresponding alternative testing method is presented in boxplots in figures 7.4, 7.5, 7.6. The situation in all three cases follows the assumption – the distribution of positions of experimentally-related GO terms is being shifted towards the start, making them more significant in two empirical cases. For more information on the MDS-related gene ontology annotations list see appendix D.

To summarize the above-mentioned result – the experimental condition influence is confirmed according to two statistical tests. Nevertheless, it was shown that two of circGPA algorithm versions are limited by the prior knowledge interactions and the straightforward interpretation of Spearman's rank-order test results could be misleading. Due to this ambiguity, the circGPA pipeline must be treated cautiously and, ideally, an additional modification or follow-up algorithm is to be added in order to ensure the consistency.

Finally, at the end of this chapter, a biological analysis of the outputs of three versions of circGPA are compared in order to suggest the possible GO annotation. Again, results could be misleading or redundant, but an effect of the empirical data is out of question. Moreover, a difference in annotated terms may hint a possible path for further in-vitro experimentations.

7.2.2 Biological analysis

Let us begin by looking at the first few leading GO terms of selected circRNAs. Then, a following information could be obtained:

- As stated in chapter 2, a recent study [109] has annotated the entire circ-*ZEB1* group. However, the paper had experiments to be tested on lamb, which is a completely different experimental setup.
  - The gene expression result demonstrated that circRNAs from circ-ZEB1 group regulate brown adipocytes differentiation and thermogenesis, e.g., heat production process, through circZEB1/miR-326–3p pathway.
- *hsa_circ_0000227*:
  - circGPA output:
    - **GOBP_REGULATION_OF_MRNA_CATABOLIC_PROCESS**
      Any process that modulates the rate, frequency, or extent of a mRNA catabolic process, e.g., for example when you digest food and the molecules break down in the body for use as energy [3].
    - **GOBP_REGULATION_OF_CELL_CYCLE_G1_S_PHASE_TRANSITION**
      Any signalling pathway that modulates the activity of a cell cycle's G1-S transition phase [4].
  - circGPAcorr output:
    - **GOBP_POSITIVE_REGULATION_OF_CATABOLIC_PROCESS**
      Any process that activates or increases the frequency, rate or extent of the chemical reactions and pathways resulting in the breakdown of substances [5].
    - **GOMF_TRANSCRIPTION_COREGULATOR_ACTIVITY**
      A transcription regulator activity that modulates the transcription of specific gene [6]
  - circGPA-BN output:
    - **GOBP_MAINTENANCE_OF_LOCATION**
      Any process in which a cell, substance or cellular entity, such as a protein complex or organelle, is maintained in a location and prevented from moving elsewhere [7].
    - **GOBP_TUBE_FORMATION**
      Creation of the central hole of a tube in an anatomical structure through which gases and/or liquids flow [8].

[3] https://www.ebi.ac.uk/QuickGO/term/GO:0061013
[4] https://www.ebi.ac.uk/QuickGO/term/GO:1902806
[5] https://www.ebi.ac.uk/QuickGO/term/GO:0009896
[6] https://www.ebi.ac.uk/QuickGO/term/GO:0003712
[7] https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/GOBP_MAINTENANCE_OF_LOCATION.html
[8] https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/GOBP_TUBE_FORMATION.html

- **Summary**:
  - *hsa-circ-0000227* is a part of circ-*ZEB1* group, which has been proven to regulate a fundamental process in mammals [109] – a heat production.
  - Thermogenesis is a catabolic process and both of catabolic GO terms that have occurred in circGPA versions further proving the relationship with thermogenesis [84].
  - Moreover, the G1-S cell cycle phase annotation is exactly where the intracellular thermogenesis takes place. Particularly, the G1 phase is a moment of a highest temperature [94].
  - Other annotations do not provide any concrete information on circRNA annotation, since transcription, location and tube formation are, basically, fundamental cell processes and could be a part of the above-mentioned thermogenesis regulation effect.

- *hsa_circ_0000228*:
  - circGPA output:
    - **GOMF_MRNA_BINDING**
      Binding to messenger RNA (mRNA), an intermediate molecule between DNA and protein. mRNA includes UTR and coding sequences, but does not contain introns [9].
    - **GOCC_RNAI_EFFECTOR_COMPLEX**
      Any protein complex that mediates the effects of small interfering RNAs on gene expression. Most known examples contain one or more members of the Argonaute family of proteins [10].
  - circGPAcorr output:
    - **GOCC_RNAI_EFFECTOR_COMPLEX**
      Already mentioned above
    - **GOBP_RESPONSE_TO_ARSENIC_CONTAINING_SUBSTANCE**
      Any process that results in a change in state or activity of a cell or an organism (in terms of movement, secretion, enzyme production, gene expression, etc.) as a result of an arsenic stimulus from compounds containing arsenic, including arsenates, arsenites, and arsenides [11].
  - circGPA-BN output:
    - **GOMF_PHOSPHATIDYLINOSITOL_3_KINASE_BINDING**
      Binding to a phosphatidylinositol 3-kinase, any enzyme that catalyzes the addition of a phosphate group to an inositol lipid at the 3' position of the inositol ring [12].
    - **GOMF_DEAMINASE_ACTIVITY**
      Catalysis of the removal of an amino group from a substrate, producing a substituted or nonsubstituted ammonia (NH3/NH2R) [13].
  - **Summary**:
    - A paper [77] shows that the application of an arsenic-containing compound is a possible treatment for MDS.
    - According to [68], the RNA interference (RNAi) is a biologic process by which RNA molecules induce sequence-specific inhibition of target gene expression or translation. Nowadays, RNAi is being used for targeted gene silencing, thus, possibly hinting experimental conditions of MDS-diagnosed patients.
    - Other annotations do not provide any concrete information on circRNA annotation, since 3-kinase binding, aminase activity or mRNA binding are, basically, fundamental cell processes.

- *hsa_circ_0003793*:
  - circGPA output, same as above:
    - **GOMF_MRNA_BINDING**
      Binding to messenger RNA (mRNA), an intermediate molecule between DNA and protein. mRNA includes UTR and coding sequences, but does not contain introns [14].

[9] https://www.ebi.ac.uk/QuickGO/term/GO:0003729
[10] https://www.gsea-msigdb.org/gsea/msigdb/geneset_page.jsp?geneSetName=GOCC_RNAI_EFFECTOR_COMPLEX
[11] https://www.ebi.ac.uk/QuickGO/term/GO:0046685
[12] https://www.ebi.ac.uk/QuickGO/term/GO:0043548
[13] https://www.ebi.ac.uk/QuickGO/term/GO:0019239
[14] https://www.ebi.ac.uk/QuickGO/term/GO:0003729

    - **GOBP_MUSCLE_CELL_PROLIFERATION**
      The multiplication or reproduction of endothelial, e.g., muscle cells, resulting in the expansion of a cell population [15].
      It is important to point out that according to [16] this annotation is one of the MDS' most associated annotation.
  - circGPAcorr output:
    - **GOBP_MUSCLE_CELL_PROLIFERATION**
      Already mentioned above.
    - **GOCC_TORC2_COMPLEX**
      A protein complex that contains at least TOR (target of rapamycin) and Rictor (rapamycin-insensitive companion of TOR), or orthologs of, in complex with other signaling components [17].
  - circGPA-BN output:
    - **GOMF_DEAMINASE_ACTIVITY**
      The enzyme catalyzes the deamination of dCTP to deoxyuridine triphosphate (dUTP) [18].
    - **GOBP_POSITIVE_REGULATION_OF_LEUKOCYTE_CELL_CELL_ADHESION**
      Any process that modulates the frequency, rate or extent of leukocyte cell-cell adhesion [19].
  - **Summary:**
    - Two of circGPA versions show us the cell proliferation – one of the tumor's associated annotation [21].
    - TORC2 complex and leukocyte cell-cell adhesion annotation hints the immunosuppressant effect of a studied circRNA
    - The tumor annotation could either be a hint of circRNA's participation in cancer-related processes, as it was shown in [102], or a consequence of the MDS experimental setup.

[15] https://www.ebi.ac.uk/QuickGO/term/GO:0001935
[16] http://ctdbase.org/detail.go?type=disease&acc=MESH%3AD009190&view=phenotype
[17] https://www.ebi.ac.uk/QuickGO/term/GO:0031932
[18] https://www.sciencedirect.com/topics/medicine-and-dentistry/deaminase
[19] https://www.ebi.ac.uk/QuickGO/term/GO:1903037

Chapter 8

Conclusion

In this thesis, the IntOMICS algorithm was applied to the gene expression data with the tripartite biological model. As experiments showed, the tripartite model is an exceptional case where the algorithm's convergence breaks for specific starting points. To be more precise, the convergence of IntOMICS has been achieved only for specific starting points, while for the others, the convergence has yet to be achieved even after hours of running. Moreover, the time complexity, e.g., the number of iterations needed to converge, is orders of magnitude larger for the tripartite model than the same size input of the original IntOMICS. Ultimately, even if convergence is to be solved, the core problem is super-exponential memory requirements.

On the other hand, the IntOMICS algorithm, as shown in chapter 6, can be improved, particularly in the upper-bound computation term, which is one of three super-exponential problematic terms. The improved version of the algorithm changes the memory and time complexity of this term into the polynomial domain. However, even with such modification applied, other parts of the algorithm were found to be impossible to change due to the fundamental limitation of the MCMC approach in Bayesian network learning.

In order to overcome such limitations, an alternative hybrid type algorithm, Prior-incorporated MMPC Skeleton Based Stochastic Search (PiM-SK-SS), has been implemented.

The implementation preserved the scoring metric from the IntOMICS algorithm, essentially allowing the combination of both prior knowledge and gene expression data. Although, while the algorithm has been implemented, it differs from the original paper in the sampling phase. More precisely, the uniformity of visited subgraphs has been violated due to the size of input network. Otherwise, the algorithm is identical to the original one and the empirical interaction network has been successfully generated.

Next, the circGPA pipelines comparison have been carried out. Unfortunately, the originally intended straightforward application of the Spearman's rank-order correlation test has failed and showed an opposite result to the assumed hypothetical scenario. Additionally, a further analysis via the Venn's diagram representation of first 100 GO terms confirmed the negative result. While there is a limitation of Spearman's rank-order test – the monotonic sequence relationship assumption, the initial hypothesis is rejected based on two tests.

Nevertheless, if another testing approach is to be introduced, e.g., the analysis of previously inferred MDS-related annotations, then the situation changes. This analysis of MDS-related annotations has shown that the application of the experimental data significantly changes the output of these annotations. Notably, their ordering shifted towards the start, making them more significant in the overall ranking.

Theoretically, this selective behavior is due to constraints on the interaction network in the circGPA/circGPA-corr case. In other words, two of the previously mentioned versions are strictly limited by the prior (database) knowledge and could not consider an interaction outside of it. Thus, the correlation of orderings in such restricted space could cause a high Spearman's rank-order correlation that violates the previous statistical test. On top of that, a significant impact of gene expression data is also observed in circGPA/circGPA-corr Venn's diagram.

Consequently, it is possible to conclude that experimental setup does have a crucial influence on the output. However, the selected analysis method is imperfect and should be enhanced by a follow-up algorithm that has yet to be discovered.

Finally, since the experimental setup has been proven to have a crucial influence on the studied interaction network, a biological analysis of the GO term annotation result was performed.

For *hsa-circ-0000227* it was confirmed that the result of a recent study [109] does indeed hold in our case for this molecule. Several GO terms suggest that above-mentioned molecule participates in a fundamental mammalian thermogenesis process, e.g., in a process of a heat generation.

For *hsa-circ-0000228*, two GO terms related with MDS treatment were found. It may hint the medication taken by the patients during samples collection.

For *hsa-circ-0003793*, it shows the strong sign of cancer (tumor) genesis, since it yields a strong cell proliferation tumor annotation in two versions. On top of that, it shows a immunosuppressant effect annotation combined with leukocyte regulation.

Bibliography

[1] A. D. HERSHEY and M. CHASE, "Independent functions of viral protein and nucleic acid in growth of bacteriophage.," *The Journal of general physiology*, vol. 36, no. 1, pp. 39–56, May 1952. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/12981234/, (visited 03.10.2022).

[2] J. Watson and F. Crick, "Molecular structure of nucleic acids," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953. [Online]. Available: https://www.mskcc.org/teaser/1953-nature-papers-watson-crick-wilkins-franklin.pdf, (visited 03.10.2022).

[3] F. Crick *et al.*, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970. [Online]. Available: https://cs.brynmawr.edu/Courses/cs380/fall2012/CrickCentralDogma1970.pdf, (visited 03.10.2022).

[4] R. C. Griffiths, "A CHARACTERIZATION OF THE MULTINOMIAL DISTRIBUTION," *Australian Journal of Statistics*, vol. 16, no. 1, pp. 53–56, Apr. 1974. DOI: 10.1111/j.1467-842x.1974.tb00914.x. [Online]. Available: https://doi.org/10.1111/j.1467-842x.1974.tb00914.x, (visited 11.03.2023).

[5] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*, 1976. [Online]. Available: https://archive.org/details/graphtheorywitha0000bond, (visited 30.10.2022).

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977, ISSN: 00359246. [Online]. Available: http://www.jstor.org/stable/2984875 (visited on 01/16/2023), (visited 16.01.2022).

[7] S. M. Stigler, "Gauss and the Invention of Least Squares," *The Annals of Statistics*, vol. 9, no. 3, pp. 465–474, 1981. DOI: 10.1214/aos/1176345451. [Online]. Available: https://doi.org/10.1214/aos/1176345451, (visited 20.11.2022).

[8] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, Oct. 1992. DOI: 10.1007/bf00994110. [Online]. Available: https://doi.org/10.1007/bf00994110, (visited 05.03.2023).

[9] D. M. Chickering, "Learning bayesian networks is np-complete," in *Learning from Data: Artificial Intelligence and Statistics V*, D. Fisher and H.-J. Lenz, Eds. New York, NY: Springer New York, 1996, pp. 121–130, ISBN: 978-1-4612-2404-4. DOI: 10.1007/978-1-4612-2404-4_12. [Online]. Available: https://doi.org/10.1007/978-1-4612-2404-4_12, (visited 02.05.2023).

[10] D. Fink, "A compendium of conjugate priors," *Technical Report.*, Jan. 1997, (visited 11.03.2023).

[11] Y. University, *Linear regression*, 1997. [Online]. Available: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm, (visited 21.11.2022).

[12] K. Murphy, *A brief introduction to graphical models and bayesian networks*, 1998. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Bayes/bayes.html, (visited 04.11.2022).

[13] I. Avivi, H. Rosenbaum, Y. Levy, and J. Rowe, "Myelodysplastic syndrome and associated skin lesions: A review of the literature," *Leukemia Research*, vol. 23, no. 4, pp. 323–330, Apr. 1999. DOI: 10.1016/s0145-2126(98)00161-1. [Online]. Available: https://doi.org/10.1016/s0145-2126(98)00161-1, (visited 02.05.2023).

[14] N. Friedman, I. Nachman, and D. Pe'er, "Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm.," Jan. 1999, pp. 206–215. DOI: 10.13140/2.1.1125.2169, (visited 09.05.2023).

[15] J. Peña, J. Lozano, and P. Larrañaga, "Learning bayesian networks for clustering by means of constructive induction," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1219–1230, Nov. 1999. DOI: 10.1016/s0167-8655(99)00089-6. [Online]. Available: https://doi.org/10.1016/s0167-8655(99)00089-6, (visited 02.05.2023).

[16] M. Ashburner, C. A. Ball, J. A. Blake, *et al.*, "Gene ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, May 2000. DOI: 10.1038/75556. [Online]. Available: https://doi.org/10.1038/75556, (visited 17.01.2023).

[17] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000. [Online]. Available: https://www.cs.ubc.ca/~arnaud/doucet_godsill_andrieu_sequentialmontecarloforbayesfiltering.pdf, (visited 16.01.2022).

[18] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," en, *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000, (visited 15.01.2022).

[19] E. Park and Y. J. Lee, "ESTIMATES OF STANDARD DEVIATION OF SPEARMAN's RANK CORRELATION COEFFICIENTS WITH DEPENDENT OBSERVATIONS," *Communications in Statistics - Simulation and Computation*, vol. 30, no. 1, pp. 129–142, Mar. 2001. DOI: 10.1081/sac-100001863. [Online]. Available: https://doi.org/10.1081/sac-100001863, (visited 14.05.2023).

[20] D. M. Chickering, "Learning equivalence classes of bayesian-network structures," *The Journal of Machine Learning Research*, vol. 2, pp. 445–498, 2002, (visited 15.01.2022).

[21] P. A. Andreeff M Goodrich DW, *Proliferation*, 2003. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK13035/, (visited 17.05.2023).

[22] N. Friedman and D. Koller, *Machine Learning*, vol. 50, no. 1/2, pp. 95–125, 2003. DOI: 10.1023/a:1020249912095. [Online]. Available: https://doi.org/10.1023/a:1020249912095, (visited 05.03.2023).

[23] B. Das, "Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem," Dec. 2004. [Online]. Available: https://www.researchgate.net/publication/1858795_Generating_Conditional_Probabilities_for_Bayesian_Networks_Easing_the_Knowledge_Acquisition_Problem, (visited 02.05.2023).

[24] S. Dowdy, S. Weardon, and D. Chilko, "Student's t distribution," in Jan. 2005, pp. 179–210, ISBN: 9780471267355. DOI: 10.1002/0471477435.ch8, (visited 14.05.2023).

[25] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the pc-algorithm," *J. Mach. Learn. Res*, vol. 8, Nov. 2005, (visited 05.03.2023).

[26] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/Bishop-PRML-sample.pdf, (visited 04.11.2022).

[27] M. Grant, S. P. Boyd, and Y. Ye, "Disciplined convex programming," 2006, (visited 04.05.2023).

[28] I. Tsamardinos, L. Brown, and C. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine Learning*, vol. 65, pp. 31–78, Oct. 2006. DOI: 10.1007/s10994-006-6889-7, (visited 05.03.2023).

[29] Y. Zhang, Z. Deng, H. Jiang, and P. Jia, "Dynamic bayesian network (dbn) with structure expectation maximization (sem) for modeling of gene network from time series gene expression data.," Jan. 2006, pp. 41–47. [Online]. Available: https://www.researchgate.net/publication/221051781_Dynamic_Bayesian_Network_DBN_with_Structure_Expectation_Maximization_SEM_for_Modeling_of_Gene_Network_from_Time_Series_Gene_Expression_Data, (visited 16.01.2022).

[30] A. V. Werhli and D. Husmeier, *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, 2007. DOI: doi:10.2202/1544-6115.1282. [Online]. Available: https://doi.org/10.2202/1544-6115.1282, (visited 30.12.2022).

[31] Z. Yang, K.-T. Fang, and S. Kotz, "On the student's t-distribution and the t-statistic," *Journal of Multivariate Analysis*, vol. 98, no. 6, pp. 1293–1304, Jul. 2007. DOI: 10.1016/j.jmva.2006.11.003. [Online]. Available: https://doi.org/10.1016/j.jmva.2006.11.003, (visited 02.05.2023).

[32] L. Fortnow, "The status of the p versus np problem," *Commun. ACM*, vol. 52, pp. 78–86, Sep. 2009. DOI: 10.1145/1562164.1562186, (visited 13.05.2023).

[33] P. Green and D. Hastie, "Reversible jump mcmc," *Genetics*, vol. 155, Jan. 2009, (visited 16.03.2023).

[34] M. Scutari, "Learning bayesian networks with the bnlearn r package," 2009. DOI: 10.48550/ARXIV.0908.3817. [Online]. Available: https://arxiv.org/abs/0908.3817, (visited 09.03.2023).

[35] M. Vidal, "A unifying view of 21st century systems biology," *FEBS Letters*, vol. 583, no. 24, pp. 3891–3894, Nov. 2009. DOI: 10.1016/j.febslet.2009.11.024. [Online]. Available: https://doi.org/10.1016/j.febslet.2009.11.024, (visited 15.01.2022).

[36] B. Jin, Y. Li, and K. D. Robertson, "Dna methylation: Superior or subordinate in the epigenetic hierarchy?" *Genes & Cancer*, vol. 2, no. 6, pp. 607–617, 2011, PMID: 21941617. DOI: 10.1177/1947601910393957. eprint: https://doi.org/10.1177/1947601910393957. [Online]. Available: https://doi.org/10.1177/1947601910393957, (visited 04.10.2022).

[37] K. Miura, "An introduction to maximum likelihood estimation and information geometry," *Interdisciplinary Information Sciences (IIS)*, vol. 17, Nov. 2011. DOI: 10.4036/iis.2011.155, (visited 15.01.2022).

[38] M. Studeny and D. Haws, *On polyhedral approximations of polytopes for learning bayes nets*, 2011. arXiv: 1107.4708 [math.ST]. [Online]. Available: https://arxiv.org/abs/1107.4708, (visited 04.05.2023).

[39] M. Vidal, M. E. Cusick, and A.-L. Barabási, "Interactome networks and human disease," en, *Cell*, vol. 144, no. 6, pp. 986–998, Mar. 2011, (visited 15.01.2022).

[40] Y. Wan, M. Kertesz, R. C. Spitale, E. Segal, and H. Y. Chang, "Understanding the transcriptome through rna structure.," *Nature reviews. Genetics*, vol. 12, no. 9, pp. 641–55, Sep. 2011. DOI: 10.1038/nrg3049. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3858389/, (visited 03.10.2022).

[41] A. Bui and C.-H. Jun, "Learning bayesian network structure using markov blanket decomposition," *Pattern Recognition Letters*, vol. 33, Dec. 2012. DOI: 10.1016/j.patrec.2012.06.013, (visited 16.01.2023).

[42] K. Chan, C. Lenard, and T. Mills, "An introduction to markov chains," Dec. 2012. DOI: 10.13140/2.1.1833.8248, (visited 04.05.2023).

[43] D. Colombo and M. Maathuis, "Order-independent constraint-based causal structure learning," *Journal of Machine Learning Research*, vol. 15, Nov. 2012, (visited 04.05.2023).

[44] M. Koivisto, *Advances in exact bayesian structure discovery in bayesian networks*, 2012. DOI: 10.48550/ARXIV.1206.6828. [Online]. Available: https://arxiv.org/abs/1206.6828.

[45] T. Niinimaki, P. Parviainen, and M. Koivisto, "Partial order MCMC for structure discovery in bayesian networks," *CoRR*, vol. abs/1202.3753, 2012. arXiv: 1202.3753. [Online]. Available: http://arxiv.org/abs/1202.3753, (visited 16.03.2023).

[46] J. Ramsey, J. Zhang, and P. L. Spirtes, *Adjacency-faithfulness and conservative causal inference*, 2012. DOI: 10.48550/ARXIV.1206.6843. [Online]. Available: https://arxiv.org/abs/1206.6843, (visited 04.05.2023).

[47] J. Zhang and P. L. Spirtes, *Strong faithfulness and uniform consistency in causal inference*, 2012. DOI: 10.48550/ARXIV.1212.2506. [Online]. Available: https://arxiv.org/abs/1212.2506, (visited 05.03.2023).

[48] K. a. J. Š. ZVÁRA, *Pravděpodobnost a matematická statistika*. 2012.

[49] J. I. Alonso-Barba, L. delaOssa, J. A. Gámez, and J. M. Puerta, "Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes," *International Journal of Approximate Reasoning*, vol. 54, no. 4, pp. 429–451, Jun. 2013. DOI: 10.1016/j.ijar.2012.09.004. [Online]. Available: https://doi.org/10.1016/j.ijar.2012.09.004, (visited 04.05.2023).

[50] D. Geiger and D. Heckerman, *Learning gaussian networks*, 2013. DOI: 10.48550/ARXIV.1302.6808. [Online]. Available: https://arxiv.org/abs/1302.6808, (visited 16.01.2023).

[51] S. Isci, H. Dogan, C. Ozturk, and H. H. Otu, "Bayesian network prior: Network analysis of biological data using external knowledge," *Bioinformatics*, vol. 30, no. 6, pp. 860–867, Nov. 2013. DOI: 10.1093/bioinformatics/btt643. [Online]. Available: https://doi.org/10.1093/bioinformatics/btt643, (visited 13.03.2023).

[52] A. Masegosa and S. Moral, "New skeleton-based approaches for bayesian structure learning of bayesian networks," *Applied Soft Computing*, vol. 13, pp. 1110–1120, Feb. 2013. DOI: 10.1016/j.asoc.2012.09.029, (visited 15.03.2023).

[53] P. Ribeiro and F. Silva, "G-tries: A data structure for storing and finding subgraphs," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 337–377, Feb. 2013. DOI: 10.1007/s10618-013-0303-4. [Online]. Available: https://doi.org/10.1007/s10618-013-0303-4, (visited 13.04.2023).

[54] C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu, "Geometry of the faithfulness assumption in causal inference," *The Annals of Statistics*, vol. 41, no. 2, Apr. 2013. DOI: 10.1214/12-aos1080. [Online]. Available: https://doi.org/10.1214/12-aos1080, (visited 05.03.2023).

[55] V. Grishkevich and I. Yanai, "Gene length and expression level shape genomic novelties," *Genome Research*, vol. 24, no. 9, pp. 1497–1503, Jul. 2014. DOI: 10.1101/gr.169722.113. [Online]. Available: https://doi.org/10.1101/gr.169722.113, (visited 02.05.2023).

[56] J. Kuipers, G. Moffa, and D. Heckerman, "Addendum on the scoring of gaussian directed acyclic graphical models," *The Annals of Statistics*, vol. 42, no. 4, Aug. 2014. DOI: 10.1214/14-aos1217. [Online]. Available: https://doi.org/10.1214/14-aos1217, (visited 05.03.2023).

[57] G. Náray-Szabó and A. Perczel, "Protein structure and dynamics," *INTERNATIONAL JOURNAL OF TERRASPACE SCIENCE AND ENGINEERING*, vol. 6, pp. 7–16, Jan. 2014. [Online]. Available: https://www.researchgate.net/publication/280623704_Protein_structure_and_dynamics, (visited 03.10.2022).

[58] V. Agarwal, G. W. Bell, J.-W. Nam, and D. P. Bartel, "Predicting effective microRNA target sites in mammalian mRNAs," *eLife*, vol. 4, Aug. 2015. DOI: 10.7554/elife.05005. [Online]. Available: https://doi.org/10.7554/elife.05005, (visited 17.01.2023).

[59] J. Bertrane, P. Cousot, R. Cousot, *et al.*, "Static analysis and verification of aerospace software by abstract interpretation," *Found. Trends Program. Lang*, vol. 2, Jan. 2015, (visited 02.05.2023).

[60] J. A. Reuter, D. V. Spacek, and M. P. Snyder, "High-throughput sequencing technologies," en, *Mol. Cell*, vol. 58, no. 4, pp. 586–597, May 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4494749/, (visited 04.11.2022).

[61] J. Salah, "A note on gamma function," *nternational Journal of Modern Sciences and Engineering Technology (IJMSET)*, Sep. 2015, (visited 11.03.2023).

[62] M. Scanagatta, C. P. de Campos, G. Corani, and M. Zaffalon, "Learning bayesian networks with thousands of variables," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/2b38c2df6a49b97f706ec9148ce48d86-Paper.pdf, (visited 04.05.2023).

[63] C. Su and M. E. Borsuk, "Improving structure mcmc for bayesian networks through markov blanket resampling," *Journal of Machine Learning Research*, vol. 17, no. 118, pp. 1–20, 2016. [Online]. Available: http://jmlr.org/papers/v17/su16a.html, (visited 17.01.2023).

[64] J. Yang and J. S. Rosenthal, "Automatically tuned general-purpose MCMC via new adaptive diagnostics," *Computational Statistics*, vol. 32, no. 1, pp. 315–348, Sep. 2016. DOI: 10.1007/s00180-016-0682-2. [Online]. Available: https://doi.org/10.1007/s00180-016-0682-2, (visited 16.01.2023).

[65] M. Bartlett and J. Cussens, "Integer linear programming for the bayesian network structure learning problem," *Artificial Intelligence*, vol. 244, no. C, pp. 258–271, 2017. DOI: 10.1016/j.artint.2015.03.003, (visited 04.05.2023).

[66] A. Bekker, J. van Niekerk, and M. Arashi, "Wishart distributions: Advances in theory with bayesian application," *Journal of Multivariate Analysis*, vol. 155, pp. 272–283, Mar. 2017. DOI: 10.1016/j.jmva.2016.12.002. [Online]. Available: https://doi.org/10.1016/j.jmva.2016.12.002, (visited 11.03.2023).

[67] C. J. Burrell, C. R. Howard, and F. A. Murphy, "Chapter 4 - virus replication," in *Fenner and White's Medical Virology (Fifth Edition)*, C. J. Burrell, C. R. Howard, and F. A. Murphy, Eds., Fifth Edition, London: Academic Press, 2017, pp. 39–55, ISBN: 978-0-12-375156-0. DOI: https://doi.org/10.1016/B978-0-12-375156-0.00004-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123751560000047, (visited 03.10.2022).

[68] X. Chen, L. S. Mangala, C. Rodriguez-Aguayo, X. Kong, G. Lopez-Berestein, and A. K. Sood, "RNA interference-based therapy and its delivery systems," *Cancer and Metastasis Reviews*, vol. 37, no. 1, pp. 107–124, Dec. 2017. DOI: 10.1007/s10555-017-9717-6. [Online]. Available: https://doi.org/10.1007/s10555-017-9717-6, (visited 20.05.2023).

[69] P. Portin and A. Wilkins, "The Evolving Definition of the Term "Gene"," *Genetics*, vol. 205, no. 4, pp. 1353–1364, Apr. 2017, ISSN: 1943-2631. DOI: 10.1534/genetics.116.196956. eprint: https://academic.oup.com/genetics/article-pdf/205/4/1353/42193126/genetics1353.pdf. [Online]. Available: https://doi.org/10.1534/genetics.116.196956, (visited 04.10.2022).

[70] M. Scutari, "Dirichlet bayesian network scores and the maximum relative entropy principle," 2017. DOI: 10.48550/ARXIV.1708.00689. [Online]. Available: https://arxiv.org/abs/1708.00689, (visited 09.03.2023).

[71] K. Kumari and S. Yadav, "Linear regression analysis study," *Journal of the Practice of Cardiovascular Sciences*, vol. 4, p. 33, Jan. 2018. DOI: 10.4103/jpcs.jpcs_8_18, (visited 20.11.2022).

[72] G. Lancia and P. Serafini, "Integer linear programming," in Jan. 2018, pp. 43–66, ISBN: 978-3-319-63975-8. DOI: 10.1007/978-3-319-63976-5_4, (visited 04.05.2023).

[73] L. Lin, "Bias caused by sampling error in meta-analysis with small sample sizes," *PLOS ONE*, vol. 13, no. 9, Z. Chen, Ed., e0204056, Sep. 2018. DOI: 10.1371/journal.pone.0204056. [Online]. Available: https://doi.org/10.1371/journal.pone.0204056, (visited 14.05.2023).

[74] J. O'Brien, H. Hayder, Y. Zayed, and C. Peng, "Overview of MicroRNA biogenesis, mechanisms of actions, and circulation," en, *Front. Endocrinol. (Lausanne)*, Aug. 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6085463/, (visited 30.10.2022).

[75] K. P. Singh, C. Miaskowski, A. A. Dhruva, E. Flowers, and K. M. Kober, "Mechanisms and measurement of changes in gene expression," en, *Biol. Res. Nurs.*, vol. 20, no. 4, pp. 369–382, Jul. 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6346310/, (visited 30.10.2022).

[76] "The gene ontology resource: 20 years and still GOing strong," *Nucleic Acids Research*, vol. 47, no. D1, pp. D330–D338, Nov. 2018. DOI: 10.1093/nar/gky1055. [Online]. Available: https://doi.org/10.1093/nar/gky1055, (visited 02.05.2023).

[77] P. Zhao, J.-b. Liang, Z.-y. Deng, *et al.*, "Association of gene mutations with response to arsenic-containing compound qinghuang powder in patients with myelodysplastic syndromes," *Chinese Journal of Integrative Medicine*, vol. 25, no. 6, pp. 409–415, Apr. 2018. DOI: 10.1007/s11655-018-2977-3. [Online]. Available: https://doi.org/10.1007/s11655-018-2977-3, (visited 20.05.2023).

[78] A. Conesa and S. Beck, "Making multi-omics data accessible to researchers," *Scientific Data*, vol. 6, no. 1, Oct. 2019. DOI: 10.1038/s41597-019-0258-4. [Online]. Available: https://doi.org/10.1038/s41597-019-0258-4, (visited 15.01.2022).

[79] B. Ghojogh, F. Karray, and M. Crowley, *Eigenvalue and generalized eigenvalue problems: Tutorial*, 2019. DOI: 10.48550/ARXIV.1903.11240. [Online]. Available: https://arxiv.org/abs/1903.11240, (visited 08.05.2023).

[80] J. S. Hawe, F. J. Theis, and M. Heinig, "Inferring interaction networks from multi-omics data," *Frontiers in Genetics*, vol. 10, 2019, ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00535. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fgene.2019.00535, (visited 04.11.2022).

[81] H.-Y. Huang, Y.-C.-D. Lin, J. Li, *et al.*, "miRTarBase 2020: updates to the experimentally validated microRNA–target interaction database," *Nucleic Acids Research*, vol. 48, no. D1, pp. D148–D154, Oct. 2019, ISSN: 0305-1048. DOI: 10.1093/nar/gkz896. eprint: https://academic.oup.com/nar/article-pdf/48/D1/D148/31697874/gkz896.pdf. [Online]. Available: https://doi.org/10.1093/nar/gkz896, (visited 05.11.2022).

[82] Y. Le, "Screening and identification of key candidate genes and pathways in myelodysplastic syndrome by bioinformatic analysis," *PeerJ*, vol. 7, e8162, Nov. 2019. DOI: 10.7717/peerj.8162. [Online]. Available: https://doi.org/10.7717/peerj.8162, (visited 18.05.2023).

[83] M. Liu, Q. Wang, J. Shen, B. B. Yang, and X. Ding, "Circbank: A comprehensive database for circRNA with standard nomenclature," *RNA Biology*, vol. 16, no. 7, pp. 899–905, Apr. 2019. DOI: 10.1080/15476286.2019.1600395. [Online]. Available: https://doi.org/10.1080/15476286.2019.1600395, (visited 14.05.2023).

[84] M. D. Lynes, S. D. Kodani, and Y.-H. Tseng, "Lipokines and thermogenesis," *Endocrinology*, vol. 160, no. 10, pp. 2314–2325, Jul. 2019. DOI: 10.1210/en.2019-00337. [Online]. Available: https://doi.org/10.1210/en.2019-00337, (visited 17.05.2023).

[85] C. Pramesh, "An introduction to statistics: Understanding hypothesis testing and statistical errors," *Indian Journal of Critical Care Medicine*, vol. 23, no. S3, 2019. DOI: 10.5005/jp-journals-10071-23259. [Online]. Available: https://doi.org/10.5005/jp-journals-10071-23259, (visited 02.05.2023).

[86] J. Pum, “Chapter six - a practical guide to validation and verification of analytical methods in the clinical laboratory,” in ser. Advances in Clinical Chemistry, G. S. Makowski, Ed., vol. 90, Elsevier, 2019, pp. 215–281. DOI: https://doi.org/10.1016/bs.acc.2019.01.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S006524231930006X, (visited 10.05.2023).

[87] M. Tsagris, “Bayesian network learning with the pc algorithm: An improved and correct variation,” *Applied Artificial Intelligence*, vol. 33, no. 2, pp. 101–123, 2019. DOI: 10.1080/08839514.2018.1526760. eprint: https://doi.org/10.1080/08839514.2018.1526760. [Online]. Available: https://doi.org/10.1080/08839514.2018.1526760, (visited 04.05.2023).

[88] Y. Yuan, X. Shen, W. Pan, and Z. Wang, “Constrained likelihood for reconstructing a directed acyclic gaussian graph,” *Biometrika*, vol. 106, pp. 109–125, Mar. 2019. DOI: 10.1093/biomet/asy057, (visited 05.03.2023).

[89] J. Cardenas, U. Balaji, and J. Gu, “Cerina: Systematic circRNA functional annotation based on integrative analysis of ceRNA interactions,” *Scientific Reports*, vol. 10, no. 1, Dec. 2020. DOI: 10.1038/s41598-020-78469-x. [Online]. Available: https://doi.org/10.1038/s41598-020-78469-x, (visited 13.05.2023).

[90] I. Erb, “Partial correlations in compositional data analysis,” *Applied Computing and Geosciences*, vol. 6, p. 100 026, 2020, ISSN: 2590-1974. DOI: https://doi.org/10.1016/j.acags.2020.100026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590197420300082, (visited 16.03.2023).

[91] A. M. Giani, G. R. Gallo, L. Gianfranceschi, and G. Formenti, “Long walk to genomics: History and current approaches to genome sequencing and assembly,” en, *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 9–19, 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6926122/, (visited 30.12.2022).

[92] M. Lu, “Circular RNA: Functions, applications and prospects,” *ExRNA*, vol. 2, no. 1, Mar. 2020. DOI: 10.1186/s41544-019-0046-5. [Online]. Available: https://exrna.biomedcentral.com/articles/10.1186/s41544-019-0046-5#citeas, (visited 30.10.2022).

[93] I. Subramanian, S. Verma, S. Kumar, A. Jere, and K. Anamika, “Multi-omics data integration, interpretation, and its application,” en, *Bioinform. Biol. Insights*, vol. 14, p. 1 177 932 219 899 051, Jan. 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7003173/, (visited 04.11.2022).

[94] R. Yamanaka, Y. Shindo, K. Hotta, N. Hiroi, and K. Oka, “Cellular thermogenesis compensates environmental temperature fluctuations for maintaining intracellular temperature,” *Biochemical and Biophysical Research Communications*, vol. 533, no. 1, pp. 70–76, Nov. 2020. DOI: 10.1016/j.bbrc.2020.08.110. [Online]. Available: https://doi.org/10.1016/j.bbrc.2020.08.110, (visited 20.05.2023).

[95] D. Geiger and D. Heckerman, “Parameter priors for directed acyclic graphical models and the characterization of several probability distributions,” 2021. DOI: 10.48550/ARXIV.2105.03248. [Online]. Available: https://arxiv.org/abs/2105.03248, (visited 05.03.2023).

[96] N. K. Kitson, A. C. Constantinou, Z. Guo, Y. Liu, and K. Chobtham, *A survey of bayesian network structure learning*, 2021. DOI: 10.48550/ARXIV.2109.11415. [Online]. Available: https://arxiv.org/abs/2109.11415, (visited 15.03.2023).

[97] M. Navara, *Pravděpodobnost a matematická statistika*. 2021. [Online]. Available: https://cmp.felk.cvut.cz/~navara/stat/PMS_ebook.pdf, (visited 30.10.2022).

[98] Z. M. Patel and T. R. Hughes, “Global properties of regulatory sequences are predicted by transcription factor recognition mechanisms,” *Genome Biology*, vol. 22, no. 1, Oct. 2021. DOI: 10.1186/s13059-021-02503-y. [Online]. Available: https://genomebiology.biomedcentral.com/articles/10.1186/s13059-021-02503-y, (visited 04.10.2022).

[99] Y. Zhao, M.-C. Li, M. M. Konaté, *et al.*, “TPM, FPKM, or normalized counts? a comparative study of quantification measures for the analysis of RNA-seq data from the NCI patient-derived models repository,” *Journal of Translational Medicine*, vol. 19, no. 1, Jun. 2021. DOI: 10.1186/s12967-021-02936-w. [Online]. Available: https://doi.org/10.1186/s12967-021-02936-w, (visited 02.05.2023).

[100] L. Y. Dotson JL, *Myelodysplastic syndrome*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK534126/, (visited 02.05.2023).

[101] J. Kuipers, P. Suter, and G. Moffa, “Efficient sampling and structure learning of bayesian networks,” *Journal of Computational and Graphical Statistics*, vol. 31, no. 3, pp. 639–650, Jan. 2022. DOI: 10.1080/10618600.2021.2020127. [Online]. Available: https://doi.org/10.1080/10618600.2021.2020127, (visited 05.03.2023).

[102] W. Liu, L. zheng, R. Zhang, *et al.*, "Circ-ZEB1 promotes PIK3ca expression by silencing miR-199a-3p and affects the proliferation and apoptosis of hepatocellular carcinoma," *Molecular Cancer*, vol. 21, no. 1, Mar. 2022. DOI: 10.1186/s12943-022-01529-5. [Online]. Available: https://doi.org/10.1186/s12943-022-01529-5, (visited 20.05.2023).

[103] M. D. MERKEROVA, J. KLEMA, D. KUNDRAT, *et al.*, "Noncoding RNAs and their response predictive value in azacitidine-treated patients with myelodysplastic syndrome and acute myeloid leukemia with myelodysplasia-related changes," *Cancer Genomics - Proteomics*, vol. 19, no. 2, pp. 205–228, 2022. DOI: 10.21873/cgp.20315. [Online]. Available: https://doi.org/10.21873/cgp.20315, (visited 18.05.2023).

[104] A. Pačínková and V. Popovici, "Using empirical biological knowledge to infer regulatory networks from multi-omics data," *BMC Bioinformatics*, vol. 23, no. 1, Aug. 2022. DOI: 10.1186/s12859-022-04891-9. [Online]. Available: https://doi.org/10.1186/s12859-022-04891-9, (visited 05.11.2022).

[105] P. Ryšavý, J. Kléma, and M. D. Merkerová, "circGPA: circRNA functional annotation based on probability-generating functions," *BMC Bioinformatics*, vol. 23, no. 1, Sep. 2022. DOI: 10.1186/s12859-022-04957-8. [Online]. Available: https://doi.org/10.1186/s12859-022-04957-8, (visited 05.11.2022).

[106] N. Tuerxun, J. Wang, F. Zhao, *et al.*, "Bioinformatics analysis deciphering the transcriptomic signatures associated with signalling pathways and prognosis in the myelodysplastic syndromes," *Hematology*, vol. 27, no. 1, pp. 214–231, Feb. 2022. DOI: 10.1080/16078454.2022.2029256. [Online]. Available: https://doi.org/10.1080/16078454.2022.2029256, (visited 18.05.2023).

[107] Y. Youssef, "Bayes theorem and real-life applications," Jun. 2022. [Online]. Available: https://www.researchgate.net/publication/361402449_Bayes_Theorem_and_Real-life_Applications, (visited 30.10.2022).

[108] P. Xin, M. Li, J. Dong, H. Zhu, and J. Li, "Bioinformatics gene analysis of potential biomarkers and therapeutic targets of osteoarthritis associated myelodysplastic syndrome," *Frontiers in Genetics*, vol. 13, Mar. 2023. DOI: 10.3389/fgene.2022.1040438. [Online]. Available: https://doi.org/10.3389/fgene.2022.1040438, (visited 18.05.2023).

[109] X. Zhang, X. Liu, T. Jiang, *et al.*, "Circular RNA circZEB1 regulates goat brown adipocytes differentiation and thermogenesis through miR-326–3p," *Small Ruminant Research*, vol. 218, p. 106 884, Jan. 2023. DOI: 10.1016/j.smallrumres.2022.106884. [Online]. Available: https://doi.org/10.1016/j.smallrumres.2022.106884, (visited 14.05.2023).

[110] *Bayes theorem, cuemath*. [Online]. Available: https://www.cuemath.com/data/bayes-theorem/, (visited 30.10.2022).

[111] *Conditional probability*. [Online]. Available: http://www.stat.yale.edu/Courses/1997-98/101/condprob.htm, (visited 30.10.2022).

[112] J. A. Cooper, *Gene*. [Online]. Available: https://www.britannica.com/science/transcription-factor, (visited 04.10.2022).

[113] D. P. Kroese, *Why the monte carlo method is so important today*. [Online]. Available: https://people.smp.uq.edu.au/DirkKroese/ps/whyMCM_fin.pdf, (visited 16.01.2023).

[114] K. Rogers, *Gene*. [Online]. Available: https://www.britannica.com/science/gene, (visited 04.10.2022).

[115] B. University, *Bayesian networks: Inference and learning*. [Online]. Available: https://people.eecs.berkeley.edu/~russell/classes/cs194/f11/lectures/CS194%5C%20Fall%5C%202011%5C%20Lecture%5C%2022.pdf, (visited 16.01.2023).

Appendix A

List of abbreviations used

DNA Deoxyribonucleic acid

RNA Ribonucleic acid

TF Transcription factor

mRNA Messenger RNA

miRNA Micro RNA

circRNA Circular RNA

CPD Conditional probability distribution

DAG Directed acyclic graph

MBR Markov Blanket Resampling

PDAG Partial Directed Acyclic Graph

LP Linear programming

ILP Integer Linear programming

GO Gene ontology

circGPA circRNA generating-polynomial annotator

MDS Myelodysplastic syndrome

BN Bayesian Network

DEGs Differentially Significant Genes

BDeu score Bayesian Dirichlet equivalent uniform score

BGe score Bayesian Gaussian equivalent score

Appendix B

Derivation of an improved IntOMICS algorithm

The [104][30] introduces the following prior distribution over all possible networks:

$$P(G|\beta) = \frac{e^{-\beta E(G)}}{Z(\beta)} = \frac{e^{-\beta E(G)}}{\sum\limits_{G \in \mathcal{G}} e^{-\beta E(G)}} \tag{B.1}$$

As well as the definition of a so-called *Energy function*:

$$E(G) = \sum_{j=1}^{N} \varepsilon(X_j, X_{pa_j(G)}) \tag{B.2}$$

$$\varepsilon(X_j, X_{pa_j(G)}) = \sum_{i \in X_{pa_j}} (1 - B_{ij}) + \sum_{i \notin X_{pa_j}} B_{ij} \tag{B.3}$$

Notably, the IntOMICS algorithm [104] has a slightly modified definition of the prior matrix B than [30]:

$$B_{ij} = \begin{cases} 0 & \text{if prior about the absence of the directed edge } i \to j \text{ is given} \\ \text{noPK}_{\text{belief}} & \text{if no prior knowledge is given for the directed edge } i \to j \\ 1 & \text{if prior about the presence is given and } i \text{ is a gene node} \\ 0.5 < \text{nonGE}_{\text{belief}} \leq 1 & \text{if prior about the presence is given and } i \text{ is a CNV/METH node} \\ 0.5 < \text{TF}_{\text{belief}} < 1 & \text{if prior about the presence is given and } i \text{ is a TF node} \end{cases} \tag{B.4}$$

where $\text{GE}_{\text{belief}}$, $\text{TF}_{\text{belief}}$ and $\text{nonGE}_{\text{belief}}$ are the constants with the default value [104]:

- $\text{noPK}_{\text{belief}} = 0.5$
- $\text{nonGE}_{\text{belief}} = 0.5$
- $\text{TF}_{\text{belief}} = 0.75$

Modification of partition function

Furthermore, [104][30] also suggests the following modification of an equation:

$$Z(\beta) = \prod_j \sum_{X_{pa_j}} e^{-\beta \varepsilon(X_j, X_{pa_j(G)})} \tag{B.5}$$

Decomposition per edge prior type

Following the B prior matrix definition B.4, the following decomposition per prior edge type can be performed:

$$\varepsilon(X_j, X_{pa_j(G)}) = \sum_{i \in X_{pa_j}} (1 - B_{ij}) + \sum_{i \notin X_{pa_j}} B_{ij} =$$

$$= \left[\sum_{i_{\text{absent}} \in X_{pa_j}} (1-0) \;+\; \sum_{i_{\text{absent}} \notin X_{pa_j}} 0 \right] +$$

$$+ \left[\sum_{i_{\text{noPK}} \in X_{pa_j}} (1 - \text{noPK}_{\text{belief}}) \;+\; \sum_{i_{\text{noPK}} \notin X_{pa_j}} \text{noPK}_{\text{belief}} \right] +$$

$$+ \left[\sum_{i_{\text{GE}} \in X_{pa_j}} (1-1) \;+\; \sum_{i_{\text{GE}} \notin X_{pa_j}} 1 \right] +$$

$$+ \left[\sum_{i_{\text{nonGE}} \in X_{pa_j}} (1 - \text{nonGE}_{\text{belief}}) \;+\; \sum_{i_{\text{nonGE}} \notin X_{pa_j}} \text{nonGE}_{\text{belief}} \right] +$$

$$+ \left[\sum_{i_{\text{TF}} \in X_{pa_j}} (1 - \text{TF}_{\text{belief}}) \;+\; \sum_{i_{\text{TF}} \notin X_{pa_j}} \text{TF}_{\text{belief}} \right] =$$

where the index i_{TYPE} is the shorthand for a subset of a given type of prior edges of a node, e.g.:

$$i_{\text{TYPE}} = \{i : B_{ij} = \text{TYPE}_{\text{belief}}\} \qquad\qquad \text{TYPE} \in \{\textit{absent}, \textit{noPK}, \textit{GE}, \textit{nonGE}, \textit{TF}\}$$

Let us further define the individual subset summations as *Partial node energy*:

$$\varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)}) = \left[\sum_{i_{\text{TYPE}} \in X_{pa_j}} (1 - \text{TYPE}_{\text{belief}}) \;+\; \sum_{i_{\text{TYPE}} \notin X_{pa_j}} \text{TYPE}_{\text{belief}} \right]$$

which effectively converts the original term into:

$$\varepsilon(X_j, X_{pa_j(G)}) = \sum_{\text{TYPE}} \varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})$$

Because of that, the partition function can be computed as follows:

$$Z(\beta) = \prod_j \sum_{X_{pa_j}} e^{-\beta \varepsilon(X_j, X_{pa_j(G)})} = \prod_j \sum_{X_{pa_j}} e^{-\beta \sum\limits_{\text{TYPE}} \varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})} =$$

$$= \prod_j \sum_{X_{pa_j}} \prod_{\text{TYPE}} e^{-\beta \varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})}$$

The individual prior edge types are constant and are calculated based on an input. Thus, such decomposition is performed once at the initial input analysis only. Moreover, it is possible to apply the same modification as in B (**Modification of partition function**) due to the independence of individual terms.

$$Z(\beta) = \prod_j \sum_{X_{pa_j}} \prod_{\text{TYPE}} e^{-\beta \varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})} = \prod_j \prod_{\text{TYPE}} \sum_{X_{pa_j}} e^{-\beta \varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})} \tag{B.6}$$

General explicit form

The upcoming subsection lists necessary properties that lead to the explicit form of a partition function. Let us consider the previously derived per prior edge type term:

$$\varepsilon_{\mathrm{TYPE}}(X_j, X_{pa_j(G)}) = \left[\sum_{i_{\mathrm{TYPE}} \in X_{pa_j}} (1 - \mathrm{TYPE}_{\mathrm{belief}}) \quad + \sum_{i_{\mathrm{TYPE}} \notin X_{pa_j}} \mathrm{TYPE}_{\mathrm{belief}} \right]$$

For simplicity, let us consider the following notation and index range:

$$i_{\mathrm{TYPE}} \coloneqq t_i, \quad i \in \langle 1, N^j_{\mathrm{type}} \rangle, \qquad C_{\mathrm{type}} = \mathrm{TYPE}_{\mathrm{belief}} \qquad N^j_{\mathrm{type}} = \text{ number of edges with } (B_{ij} = \mathrm{TYPE})$$

On top of that, let us introduce the more comprehensive definition of $X_{pa_j}(G)$:

$$g_j \in \{0,1\}^{N^j_{\mathrm{type}}}, \qquad g^{t_i}_j = \begin{cases} 0, \text{if edge } t_i \to j \text{ is not included in G} \\ 1, \text{if edge } t_i \to j \text{ is included in G} \end{cases} \tag{B.7}$$

Consequently, the term is converted into the following form:

$$\varepsilon_{\mathrm{TYPE}}(X_j, X_{pa_j(G)}) = \varepsilon_{\mathrm{TYPE}}(g_j) = \sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}_j (1 - C_{\mathrm{type}}) + \sum_{t_i=1}^{N^j_{\mathrm{type}}} (1 - g^{t_i}_j) C_{\mathrm{type}} =$$

Or, equivalently:

$$= \sum_{t_i=1}^{N^j_{\mathrm{type}}} \left[(1 - C_{\mathrm{type}}) g^{t_i}_j + (1 - g^{t_i}_j) C_{\mathrm{type}} \right] = \sum_{t_i=1}^{N^j_{\mathrm{type}}} \left[g^{t_i}_j - g^{t_i}_j C_{\mathrm{type}} + C_{\mathrm{type}} - g^{t_i}_j C_{\mathrm{type}} \right] =$$

$$= \sum_{t_i=1}^{N^j_{\mathrm{type}}} C_{\mathrm{type}} + (1 - 2C_{\mathrm{type}}) \sum_{t_i=1}^{N^j_{\mathrm{type}}} \left[g^{t_i}_j \right] = N^j_{\mathrm{type}} C_{\mathrm{type}} + (1 - 2C_{\mathrm{type}}) \sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}_j$$

$$\boxed{\varepsilon_{\mathrm{TYPE}}(X_j, X_{pa_j(G)}) = \varepsilon_{\mathrm{TYPE}}(g_j) = N^j_{\mathrm{type}} C_{\mathrm{type}} + (1 - 2C_{\mathrm{type}}) \sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}_j} \tag{B.8}$$

Consequently, the following properties hold for the derived term:

1. Sum of all numbers of prior edge types is N:
$$\sum_{\mathrm{type}} N^j_{\mathrm{type}} = N$$

2. Total number of all possible g_j for a given node and prior edge type is $2^{N^j_{\mathrm{type}}}$
The proof is straightforward – since every edge is chosen independently, we can multiply all possible cases of each $g^{t_i}_j$. Moreover, the following holds: $\sum_{k=0}^{N_{\mathrm{type}}} = 2^{N_{\mathrm{type}}}$

3. For any two different parent sets $g_{j,1} = (g^{t_1}_{j,1}, ..., g^{t_{N^j_{\mathrm{type}}}}_{j,1})$ and $g_{j,2} = (g^{t_1}_{j,2}, ..., g^{t_{N^j_{\mathrm{type}}}}_{j,2})$, the following holds:
$$\sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}_{j,1} = \sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}_{j,2} \qquad \Rightarrow \qquad \varepsilon_{\mathrm{TYPE}}(g_{j,1}) = \varepsilon_{\mathrm{TYPE}}(g_{j,2})$$
The proof is again straightforward – since the only non-constant variable in the $\varepsilon_{\mathrm{TYPE}}(g)$ term is the $\sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}$, the equality holds.
Following the property, it is more convenient to characterize the term by the summation:
$$\text{Denote: } K = \sum_{t_i=1}^{N^j_{\mathrm{type}}} g^{t_i}, \qquad K \in \{0, ..., N^j_{\mathrm{type}}\}$$
$$\varepsilon_{\mathrm{TYPE}}(g_j) = \varepsilon_{\mathrm{TYPE}}(K) = N^j_{\mathrm{type}} C_{\mathrm{type}} - (1 - 2C_{\mathrm{type}}) K$$

4. The number of possible g_j with the given K is $\binom{N^j_{\text{type}}}{K}$
   The number of all possible g_j is the same as the number of all possible solutions of:

$$K = \sum_{t_i=1}^{N^j_{\text{type}}} g_j^{t_i} \qquad\qquad s.t.\ \ g_j^{t_i} \in \{0,1\}$$

   which is the number of unordered subsets of K elements that will be chosen to be ones, e.g., $\binom{N^j_{\text{type}}}{K}$. Moreover, this leads to the transformation of the partition function:

$$\sum_{X_{pa_j(G)}} e^{-\beta\varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})} = \sum_{g} e^{-\beta\varepsilon_{\text{TYPE}}(g)} = \sum_{K=0}^{K=N^j_{\text{type}}} \binom{N^j_{\text{type}}}{K} e^{-\beta\varepsilon_{\text{TYPE}}(K)}$$

Combining all the abovementioned properties allows us to formulate the final form of the partition function:

$$Z(\beta) = \prod_j \prod_{\text{TYPE}} \sum_{X_{pa_j}} e^{-\beta\varepsilon_{\text{TYPE}}(X_j, X_{pa_j(G)})} = \prod_j \prod_{\text{TYPE}} \sum_K \binom{N^j_{\text{type}}}{K} e^{-\beta\varepsilon_{\text{TYPE}}(K)} =$$

$$= \boxed{\prod_j \prod_{\text{TYPE}} \sum_K \binom{N^j_{\text{type}}}{K} exp\left(-\beta\left[N^j_{\text{type}} C_{\text{type}} + (1 - 2C_{\text{type}})K\right]\right)}$$

Rewrite for specific IntOMICS implementation and constants

The next step in algorithm improvement depends on the original implementation of IntOMICS in R code [104][1]. Mainly, the source code [1] contains modifications not mentioned in the original paper [104].

After the substitution of the default constant values, individual terms become:

- **Absence prior**
  The first hidden modification is the explicit forbidding of any edge with prior knowledge of absence. This corresponds to the term:

$$\varepsilon_{\text{absent}}(X_{pa_j}) = \left[\sum_{i_{\text{absent}} \in X_{pa_j}} (1-0) \;+ \sum_{i_{\text{absent}} \notin X_{pa_j}} 0\right] = \left[\sum_{i_{\text{absent}}} 0\right] = 0$$

  which is effectively canceling the term.

- **No prior knowledge and nonGE prior**
  Partial energy:

$$\varepsilon_{\text{noPK}}(X_{pa_j}) = \left[\sum_{i_{\text{noPK}} \in X_{pa_j}} (1 - \text{noPK}_{\text{belief}}) \;+ \sum_{i_{\text{noPK}} \notin X_{pa_j}} \text{noPK}_{\text{belief}}\right] = \left[\sum_{i_{\text{noPK}} \in X_{pa_j}} (1-0.5) \;+ \sum_{i_{\text{noPK}} \notin X_{pa_j}} 0.5\right] =$$

  Since constants for noPK and nonGE prior edge types are the same, they have the same formula.

$$\varepsilon_{\text{noPK}}(X_{pa_j}) = \varepsilon_{\text{nonGE}}(X_{pa_j}) = \left[\sum_{i_{\text{noPK}}} 0.5\right] = 0.5 \cdot N^j_{\text{noPK}}$$

  Partial partition:

$$\sum_{X_{pa_j}} e^{-\beta\varepsilon_{\text{noPK}}(X_{pa_j})} = \sum_g e^{-0.5\beta N^j_{\text{noPK}}} = \sum_{K=0}^{N^j_{\text{noPK}}} \binom{N^j_{\text{noPK}}}{K} e^{-0.5\beta N^j_{\text{noPK}}} =$$

$$= e^{-0.5\beta N^j_{\text{noPK}}} \sum_{K=0}^{N^j_{\text{noPK}}} \binom{N^j_{\text{noPK}}}{K} = 2^{N^j_{\text{noPK}}} e^{-0.5\beta N^j_{\text{noPK}}}$$

[1] https://gitlab.ics.muni.cz/bias/intomics

- **Gene prior**

  Partial energy:

$$\varepsilon_{\mathrm{GE}}(X_{pa_j}) = \left[\sum_{i_{\mathrm{GE}} \in X_{pa_j}} (1-1) \quad + \sum_{i_{\mathrm{GE}} \notin X_{pa_j}} 1 \right] \quad = \left[\sum_{i_{\mathrm{Ge}} \notin X_{pa_j}} 1 \right] = N^j_{\mathrm{GE}} - K$$

  Partial partition:

$$\sum_{X_{pa_j}} e^{-\beta \varepsilon_{\mathrm{GE}}(X_{pa_j})} = \sum_{g} e^{-\beta (N^j_{\mathrm{GE}} - K)} = \sum_{K=0}^{N^j_{\mathrm{GE}}} \binom{N^j_{\mathrm{GE}}}{K} e^{-\beta (N^j_{\mathrm{GE}} - K)}$$

- **TF prior**

$$\varepsilon_{\mathrm{TF}}(X_{pa_j}) = \left[\sum_{i_{\mathrm{TF}} \in X_{pa_j}} (1-0.75) \quad + \sum_{i_{\mathrm{TF}} \notin X_{pa_j}} 0.75 \right] \quad = \left[\sum_{i_{\mathrm{TF}} \in X_{pa_j}} 0.25 \quad + \sum_{i_{\mathrm{TF}} \notin X_{pa_j}} 0.75 \right] \quad =$$

$$= \left[\sum_{i_{\mathrm{TF}}} 0.25 \quad + \sum_{i_{\mathrm{TF}} \notin X_{pa_j}} 0.5 \right] \quad = 0.25 \cdot N^j_{\mathrm{TF}} + 0.5 \cdot (N^j_{\mathrm{TF}} - K) = 0.75 N^j_{\mathrm{TF}} - 0.5 \cdot K$$

Partial partition:

$$\sum_{X_{pa_j}} e^{-\beta \varepsilon_{\mathrm{TF}}(X_{pa_j})} = \sum_{g} e^{-\beta (N^j_{\mathrm{TF}} - K)} = \sum_{K=0}^{N^j_{\mathrm{TF}}} \binom{N^j_{\mathrm{TF}}}{K} e^{-\beta (0.75 N^j_{\mathrm{TF}} - 0.5 \cdot K)}$$

The derived noPK, nonGE, and TF prior terms in this form could be used only if the constant values match the default ones presented in the IntOMICS paper [104].

This representation will not be used in the following implementation nor the final formula, but it will be used in the following example computed by hand. The conducted simplification for given constants shows that if given some more assumptions, for example, $C_{\mathrm{noPK}} = 0.5$, the computation simplifies even more. However, since the IntOMICS algorithm allows a change, the final formula is presented in a general explicit form.

Appendix C

Example of improved algorithm usage

Manual computation of a partition function is performed on a simple example to demonstrate the superiority of an improved algorithm.

Input data

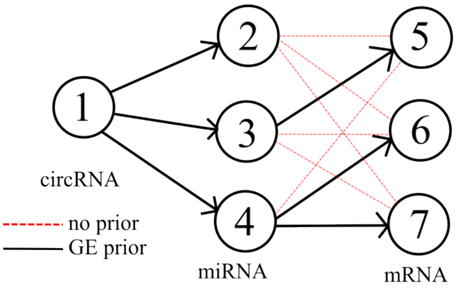


Figure C.1: Input data example

As input data, the tripartite cellular model is used. A detailed description of such a model is described in section 2.2.1.
In this particular case, the following numbers of each type of RNA are used:

- circRNA = 1
- miRNA = 3
- mRNA = 3

and the total number of feature nodes $N = 7$.

As a prior knowledge, the entire MultiMIR database (see section 2.2.3) is used. More precisely, all verified gene interactions are used as **GE priors** and are shown in figure C.1. Next, the biologically forbidden interactions from the tripartite model are marked as absent prior, which can be seen in the tripartite structure of a graph.

The individual gene names and IDs are not crucial since the example illustrates the algorithmic approach, not the particular use case in a real-world instance. Nonetheless, it must be noted that this particular setup is present in some concrete gene samples. These samples could be obtained in an upcoming R implementation of an improved algorithm (see section 6.1.1).
Furthermore, the usage of Transcription Factors (TF), Methylation (METH), and Copy Number Variations (CNV) are not shown here due to the absence of experimental data. However, the formula includes them and can be used with them.

Classical IntOMICS

By the notation of the original IntOMICS algorithm, the partition function should be computed as:

$$\log\, Z(\beta) = \sum_j \log\left[\sum_{X_{pa_j}} e^{-\beta\varepsilon(X_j, X_{pa_j(G)})}\right] \tag{C.1}$$

This equation includes the previously stated hidden modification of the logarithmic scale. On top of that, it is known that absent prior edges are canceled, so the entire prior becomes:

$$\log\, Z(\beta) = \sum_j \log\left[\sum_{X_{pa_j}} e^{-\beta\varepsilon(X_j, X_{pa_j(G)})}\right] = \tag{C.2}$$

$= \log\ (0) +$ (Since all edges going to circRNA(1) are absent priors, 1 calculation)

$+\log\ \left[e^{-\beta\cdot 0} + e^{-\beta\cdot 1}\right] +$ (miRNA(2) has only 2 possible choices, include or not the only GE prior edge)

$+\log\ \left[e^{-\beta\cdot 0} + e^{-\beta\cdot 1}\right] +$ (miRNA(3) has only 2 possible choices, 2 calculations)

$+\log\ \left[e^{-\beta\cdot 0} + e^{-\beta\cdot 1}\right] +$ (miRNA(4) has only 2 possible choices, 2 calculations)

$+\log\ \left[\sum_{g=\{0,0,0\}}^{\{1,1,1\}} e^{-\beta \sum_{i=1}^{3} g^i}\right] +$ $\big($mRNA(5) has 2^3 possible choices, each with 3 calculations, $3 * 2^3 = 24$ calculations$\big)$

$+\log\ \left[\sum_{g=\{0,0,0\}}^{\{1,1,1\}} e^{-\beta \sum_{i=1}^{3} g^i}\right] +$ (mRNA(6) has the same 24 calculations)

$+\log\ \left[e^{-\beta(0.5+0.5+1)} + ... + e^{-\beta(0.5+0.5+0)}\right] =$ (mRNA(7) has the same 24 calculations)

Let us calculate it for the $\beta = 0.5$.

$$\log Z(0.5) = \log\ (0) + 3 * \log\ (1 + e^{-0.5}) + 3 * \log\ (4 * e^{-2\cdot 0.5} + 4 * e^{-0.5}) \approx 3.39$$

Improved algorithm

Now, the same calculation will be performed using the improved algorithm. Let us start by calculating the prior type numbers for every node:

- circRNA(1) – no incoming edges, $N^1 = (0,0,0,0,0)$
- miRNA(2-4) – 1 GE prior edge, $N^2 = N^3 = N^4 = (0,0,1,0,0)$
- mRNA(5-7) – 1 GE prior edge, 2 no prior edges, $N^5 = N^6 = N^7 = (0,2,1,0,0)$

Thus, a histogram will have the following mapping:

- $H : (0,0,0,0,0) \mapsto 1$
- $H : (0,0,1,0,0) \mapsto 3$
- $H : (0,2,1,0,0) \mapsto 3$

and the final prior will look like:

$$\log Z(\beta) = 1 \cdot \log\ [0] +$$

$$3 \cdot \log\ \left[\left(\sum_{k=0}^{k=1} \binom{1}{k} \exp(-\beta\,[1-K])\right)\right] +$$

$$3 \cdot \log\ \left[\left(\sum_{k=0}^{k=1} \binom{1}{k} \exp(-\beta\,[1-K])\right) \left(2^2 \exp(-0.5\beta\cdot 2)\right)\right] =$$

for $\beta = 0.5$ it is equal to:

$$= \log\ (0) + 3 * \log\ (1 + e^{-0.5}) + 3 * \log\ ((1 + e^{-0.5}) \cdot (4 \cdot e^{-2\cdot 0.5\cdot 0.5})) \approx 3.39$$

Appendix D

GO annotations related to Myelodysplastic syndromes

The following appendix lists GO terms that have been observed with statistical significance in other experimental conditions involving Myelodysplastic syndrome (MDS). However, it must be noted that in each particular instance the inference method and the studied clinical case differ and, thus, their combination is not a perfect subset:

- A paper [82] performs a Gene Ontology (GO) term enrichment analysis. More precisely, given a gene expression data table, it looks for Differentially Significant Genes (DEGs). This paper is labeled as *biological classification of DEGs* in table below.
- A paper [106] combined multiple databases – GSE4619, GSE19429, GSE30195, and GSE58831 microarray datasets of CD34+cells for identifying the differentially expressed genes (DEGs) in the MDS. Consequently, MDS transcriptomic profiles were identified. This paper is labeled as *GO transcriptomic profile* in table below.
- A paper [103] tested various circRNA for azacitidine treatment of MDS and acute myeloid leukemia. As a step of analysis, differentially expressed genes were used to find MDS-related GO annotations. The paper is labeled as *GO pathways*.
- A paper [108] assumed that Osteoarthritis (OA) and Myelodysplastic syndrome (MDS) have a similar mechanisms involved. Thus, the intersection of their differential genes was taken using KEGG Analysis tool. The paper is labeled as *GO/KEGG Analysis tool*.

Last note: intersections are not considered, only the first appearance of GO term is listed. Moreover, several GO terms couldn't be identified by the ID given, therefore only found ones are listed.

Paper	GO ID	Description
biological classification of DEGs	GO:0006955	Immune response
biological classification of DEGs	GO:0001775	Cell activation
biological classification of DEGs	GO:0040011	Locomotion
biological classification of DEGs	GO:0005102	Receptor binding
biological classification of DEGs	GO:0045595	Cell differentiation
biological classification of DEGs	GO:0002684	Regulation of immune system
biological classification of DEGs	GO:0022407	Cell-cell adhesion

Paper	GO ID	Description
biological classification of DEGs	GO:0098911	Regulation of cell action potential
biological classification of DEGs	GO:0043565	Specific DNA binding
biological classification of DEGs	GO:0050954	Sensory perception
biological classification of DEGs	GO:0006955	Immune response
biological classification of DEGs	GO:0045746	Notch signaling pathway
biological classification of DEGs	GO:0007166	Cell surface receptor signaling pathway
biological classification of DEGs	GO:0000982	Transcription factor activity
biological classification of DEGs	GO:0009891	Biosynthetic process
biological classification of DEGs	GO:0004871	Signal transducer activity
biological classification of DEGs	GO:0008283	Cell proliferation
GO transcriptomic profile	GO:0045814	Negative regulation of gene expression epigenetic
GO transcriptomic profile	GO:0005677	Chromatin silencing complex
GO transcriptomic profile	GO:0051103	DNA ligation involved in DNA repair
GO transcriptomic profile	GO:0048024	Regulation of mrna splicing via spliceosome
GO pathways	GO:0045944	Positive regulation of transcription by RNA polymerase II
GO pathways	GO:0045944	Positive regulation of transcription by RNA polymerase II
GO pathways	GO:0045595	Regulation of cell differentiation
GO pathways	GO:0031984	Organelle subcompartment
GO/KEGG Analysis tool	GO:0048738	Cardiac muscle tissue development
GO/KEGG Analysis tool	GO:0002764	Immune response-regulating signaling pathway