SAPIENZA
UNIVERSITÀ DI ROMA

A Comparative Analysis of Algorithms for Identifying Cancer Driver Pathways

Faculty of Information Engineering, Computer Science and Statistics
Bachelor's Degree in Computer Science

Alessio Bandiera
ID number 1985878

Advisor
Prof. Ivano Salvo

Academic Year 2023/2024

A Comparative Analysis of Algorithms for Identifying Cancer Driver Pathways
Bachelor's Thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: alessio.bandiera02@gmail.com

TODO.

Abstract

TODO.

Contents

Chapter 1

Introduction

placeholder. introduzione alla tesi

1.1 Cancer

1.1.1 Overview

Cancer is a group of diseases characterized by uncontrolled cell proliferation, which allows cells to infiltrate into organs and tissues, thereby altering their functions and structure. There are more than 100 types of cancer, which can be grouped into broader categories such as carcinomas, sarcomas, leukemias and others [37].

Estimates predicted more than 2 million new cases of cancer of any site in the United States in 2024, with over 600,000 deaths [28]. According to SEER [11] 22, cancer is most frequently diagnosed among individuals aged 65-74, who also represent the age group with the highest cancer-related mortality, accounting for approximately 30% of cases in both diagnoses and deaths [4].

Cancer is often preceded by a range of symptoms, some of which may be subtle or easily overlooked. Possible signs and symptoms include persistent coughing, changes in bowel habits, unexplained bleeding, lumps, unexplained weight loss, persistent pain, yellowing or itchy skin, and feeling tired or unwell without a clear reason [36].

The exponential growth of cancer is driven by mutations in cellular DNA, which encodes the instructions for cell development and multiplication, therefore errors in these instructions can lead to cancerous transformation. These genetic mutations can arise from several factors, including random chance or exposure to carcinogens [31]. These and other causes will be discussed in the following sections.

1.1.2 Causes

The development of cancer is a complex, *multistep process* influenced by various factors, making it too simplistic to attribute cancer to a single cause. While genetic mutations can occur randomly through errors in DNA during cell division, or be

inherited from a parent [31], many agents — such as radiation, chemicals, and viruses — have been found to induce cancer.

Radiation and many chemical carcinogens work by damaging DNA and *causing mutations*. These are known as **initiating agents** because they trigger genetic changes that lead to cancer. For example, solar ultraviolet radiation, chemicals in tobacco smoke, and *aflatoxin* are well-documented carcinogens. Tobacco smoke, in particular, is a major cause of lung cancer and is also linked to cancers of the oral cavity, throat, larynx, esophagus, and other areas. It is estimated that smoking contributes to a significant portion of all cancer deaths.

In contrast, some carcinogens — known as **tumor promoters** — facilitate cancer development by *stimulating cell proliferation* rather than by inducing mutations. Tumor formation in animal models typically require both an initiating agent and a promoter to facilitate the growth of mutated cells. For instance, hormones (especially estrogens) play a role as tumor promoters in certain cancers.

Additionally, some viruses are known to cause cancer in both animals and humans, such as those linked to liver cancer and cervical carcinoma. These viral-induced cancers highlight the broader impact of carcinogens and underscore their role in both viral and non-viral cancer development [9].

In summary, the various ways in which different factors contribute to cancer emphasize the complexity of the disease and underscore the importance of developing effective treatment approaches, which will be explored in later sections.

1.1.3 Mutations in cancer development

The fundamental feature of cancer development is **tumor clonality**, meaning tumors often develop from single cells that start to proliferate abnormally. However, the clonal origin of tumors does not mean that the initial progenitor cell had all the features of a cancer cell from the start. Instead, cancer evolves through a multistep process in which cells *gradually acquire malignant characteristics* through a series of **alterations**. This multistep nature is indicated by the fact that most cancers develop later in life. For example, the incidence of colon cancer increases markedly with age, showing a dramatic rise as individuals grow older. This steep age-related increase suggests that cancer typically results from **multiple abnormalities** accumulated over many years.

At the cellular level, cancer development is viewed as a process of mutation and selection for cells with progressively greater abilities to proliferate, survive, invade, and metastasize. The first stage, known as **tumor initiation**, involves a genetic alteration that triggers abnormal growth in a single cell, leading to the expansion of a population of clonally derived tumor cells. **Tumor progression**, continues as *additional mutations* arise within this cell population, with some mutations providing a selective advantage. As a result, cells bearing these advantageous mutations become *dominant* within the tumor, a process known as **clonal selection**. This selection continues throughout the tumor's evolution, causing it to grow more rapidly and become increasingly malignant [9].

Undoubtedly, mutations are fundamental to the development of cancer and to its progression. Therefore, to effectively combat this disease, it is essential to gain a comprehensive understanding of how these genetic alterations occur and contribute to tumor development.

1.2 Targeted therapy

1.2.1 Current cancer treatment

Research aimed at finding cancer treatment is continuously evolving due to the disease's lethality and complexity. Currently, the primary techniques used to remove, control, manage, and delay the effects of cancer include [5]:

- *surgery*, which involves the removal of the cancerous region and is generally reserved for solid tumors;
- *radiotherapy*, which uses x-rays to destroy tumor cells, aiming to target the cancerous region as precisely as possible to preserve healthy tissue; however, radiotherapy can increase the risk of developing secondary tumors, such as leukemia or sarcomas, and may lead to delayed effects like dementia, amnesia, or progressive cognitive difficulties;
- *chemotherapy*, which employs *cytotoxic* drugs to block cellular division in both cancerous and healthy cells, but they can also induce side effects in rapidly renewing tissues;
- *hormone therapy*, which alters the balance of specific hormones, potentially leading to side effects such as joint pain or osteoporosis.

Recent advancements in traditional cancer treatments like chemotherapy, radiotherapy, and surgery have contributed to a decline in cancer mortality rates over the years. However, these methods still face significant limitations, often resulting in tumor recurrence and mortality, due to their various side effects. This has prompted a shift toward **mutation-targeted therapies**, as a result of their potential to precisely target cancer cells and minimize damage to healthy cells and tissue [30, 35].

1.2.2 Overview and origin

Targeted therapy is a form of cancer treatment that targets proteins responsible for the growth, division, and spread of cancer cells, and it forms the basis of precision medicine. The targets include growth factor receptors, signaling molecules, cell-cycle proteins, and other molecules crucial for normal tissue development and homeostasis, which often become overexpressed or altered in cancer cells, leading to their aberrant function [38].

Unlike standard chemotherapy, which indiscriminately destroys both rapidly dividing cancerous and normal cells, targeted therapies specifically attack abnormal proteins produced by mutated genes. Because normal cells lack these tumor-specific mutations, targeted therapies often show a higher degree of selectivity, causing fewer off-target effects and achieving more rapid and substantial tumor reduction [35].

The concept of targeted therapy originates from the german Nobel Prize Paul Ehrlich's idea of a "*magic bullet*" [12], when he envisioned a chemical capable of specifically targeting microorganisms. Over a century later, advances in molecular biology enhanced our understanding of the mechanisms behind cancer initiation, promotion, and progression. This progress led to the development of treatments that can interfere with specific molecular targets, typically proteins, linked to tumor growth and progression [38].

1.2.3 Therapy types

Most types of targeted therapy consist of **small-molecule drugs**, which are used for targets located inside cells because their small size allows them to enter cells easily, and **monoclonal antibodies**, which are laboratory-produced proteins engineered to bind to specific targets on cancer cells. Some monoclonal antibodies help the immune system identify and destroy cancer cells by marking them, while others directly inhibit the growth of cancer cells or induce their self-destruction, and still others deliver toxins directly to cancer cells [30].

Most targeted therapies treat cancer by interfering with specific proteins that promote tumor growth and spread. This approach differs from chemotherapy, which often kills all rapidly dividing cells. The following are the different approaches that targeted therapy employs [30].

- *Immunotherapy.* Cancer cells can often evade detection by the immune system. Certain targeted therapies mark cancer cells, making them easier for the immune system to identify and destroy, while others enhance the immune system's ability to fight cancer more effectively.
- *Signal interruption.* Targeted therapies can interrupt signals that cause cancer cells to grow and divide uncontrollably. Cells normally divide in response to specific signals binding to proteins on their surface. However, some cancer cells present changes in the proteins that tell them to divide without the signals. Targeted therapies can block these proteins, slowing the uncontrolled growth of cancer.
- *Angiogenesis inhibition.* The process through which new blood vessels form is called angiogenesis; beyond a certain size tumors need new blood vessels, thus the tumor sends signals to start angiogenesis. Some targeted therapies can disrupt the signals that trigger this process, preventing the formation of a blood supply, and restricting the tumor's size.
- *Cell-killing agents delivery.* Some monoclonal antibodies are combined with substances like toxins, chemotherapy drugs, or radiation. These antibodies

bind to targets on the surface of cancer cells, delivering the cell-killing agents directly into the cells, causing them to die. Most importantly, cells without these targets remain unharmed.

- *Apoptosis activation.* Cancer cells often evade the natural process of cell death, known as apoptosis, which initiates when cells become damaged or are no longer needed. Some targeted therapies can trigger apoptosis in cancer cells, leading to their death.

- *Hormone therapy.* Some types of breast and prostate cancer require specific hormones to grow. Hormone therapies block the body's production of growth hormones or prevent them from acting on cells, including cancer cells.

The diverse strategies employed by targeted therapies highlight the innovative approaches being developed to treat cancer more precisely. As research advances, these methods will continue to evolve, potentially improving outcomes and reducing side effects compared to traditional treatments.

1.2.4 Drawbacks and side effects

Like all cancer treatment, targeted therapy also has limitations, and often works best when combined with other types of targeted therapies or additional cancer treatments like chemotherapy and radiation [30].

In particular, developing drugs for certain targets can be challenging due to factors including the target's structural complexity, its function within the cell, or a combination of both. Moreover, cancer cells can develop resistance to targeted therapy, which may occur if the target itself mutates, rendering the therapy unable to interact with it effectively. Alternatively, resistance can arise if cancer cells adapt and find new growth mechanisms that do not rely on the target [30].

As for side effects, in general, targeted molecular therapies have good toxicity profiles. However, side effects differ from person to person, even among those undergoing the same cancer treatment [27], and some patients may be highly sensitive to these drugs and may develop specific and severe toxicities [38].

The most common side effects of targeted therapy are diarrhea and liver issues, but they may also include problems with blood clotting and wound healing, high blood pressure, fatigue, mouth sores, nail changes, loss of hair color, and skin problems. In rare cases, a perforation may occur in the wall of the esophagus, stomach, small intestine, colon, rectum, or gallbladder. Medications are available to manage many of these side effects, either by preventing them or treating them once they arise. Additionally, most side effects of targeted therapy subside after the treatment is completed [30].

In conclusion, although targeted therapy shows promise with generally manageable side effects, it has limitations such as potential drug resistance and varying individual responses. Effective management of these side effects and ongoing research are essential to improving treatment outcomes and patient care.

1.2.5 Drugs targeting mutations

As mentioned earlier, mutations play a crucial role in the growth and development of cancer. Targeted therapy allows for precise targeting of the mutations that enable cancer to continue its progression. In particular, oncogenic gene mutations may be druggable in several ways [35]:

- some oncogenic gene mutations encode proteins that are structurally or functionally different from the wild-type (WT), normal version of the protein; these differences create an opportunity for developing targeted therapies, because a drug can be designed specifically to bind to these unique features, and inhibit the protein's activity, without affecting the WT protein in healthy cells;
- gene mutations often result in the abnormal activation of some protein, through mechanisms like a *gain-of-function mutation* or *gene amplification*; although these proteins are considered druggable, the mutation does not necessarily change the protein in a way that allows for mutant-specific targeting, i.e. drugs may also target the WT version of the protein present in healthy cells, potentially leading to more side effects;
- some oncogenic mutations create novel molecular dependencies or vulnerabilities in cancer cells, which can be exploited by targeted therapies; these are called *actionable mutations* because they provide new targets for drug development that are specific to cancer cells and do not exist in normal cells.

While truly druggable mutations in the first category are relatively rare, many overactive or amplified targets still offer effective therapeutic opportunities due to their elevated expression levels or the significant dependence of cancer cells on these specific proteins. Additionally, mutations that currently lack targeted therapy options can still function as biomarkers to guide other therapeutic decisions [35].

Advances in targeted therapies have been significantly driven by technological progress in sequencing over the past two decades, particularly with the development of next-generation sequencing (NGS). The identification of both common and rare genetic mutations has launched research into targeted therapies against mutant proteins and aberrant molecular signaling pathways. Moreover, the discovery of the BCR-ABL fusion gene and the development of the BCR-ABL inhibitor *imatinib* marked a breakthrough in targeted cancer therapies, leading to numerous FDA-approved drugs [35]. However, the challenge of developing targeted therapies remains difficult, particularly for mutations that affect normal and cancerous proteins alike, or those for which no targeted therapies currently exist. The complexities of druggable mutations and their effects on treatment underscore the need for ongoing research and refinement in this area. Given the importance of fully understanding the role of mutations in cancer development, in order to improve targeted therapies and cancer treatment overall, research must focus on genomic mutations and their classification. The next chapter will discuss the existence of different types of mutations and the current techniques used to classify them.

Chapter 2

Pathways identification criteria

2.1 Mutations and pathways

2.1.1 Passenger and driver mutations

In the previous sections, it was discussed how mutations play a critical role in cancer development, but not every aberration that occurs in a tumor is relevant to its proliferation. In fact, cancer mutations are generally divided into two categories: **passenger mutations** and **driver mutations**. Passenger mutations do not confer direct benefits to tumor growth or development, whereas driver mutations actively contribute to cancer progression by providing an evolutionary advantage and promoting the proliferation of tumor cells. *Driver genes* are genes that harbor at least one driver mutation, though it may also contain passenger ones [34].

Although the general consensus is that mutations are divided into these two categories, some studies do not fully agree with this dichotomous model [20]. Further complicating matters, the term *driver gene* has two distinct meanings in cancer research. Originally, the *driver-versus-passenger* concept was used to differentiate mutations that provide a selective growth advantage from those that do not. According to this definition, genes without driver mutations cannot be classified as driver genes. However, many genes that have few or no driver mutations are still referred to as driver genes in the literature. This includes genes that are overexpressed, underexpressed, or epigenetically altered in tumors. Although some of these genes might contribute significantly to cancer development, classifying all of them as *driver genes* may be misleading [34].

Despite these complexities, it is generally accepted that mutations can be categorized into the two described types. This classification is essential, as identifying driver mutations can significantly advance the development of targeted therapies, which may specifically target these driver mutations directly.

2.1.2 Problems with frequency analyses

To classify mutations into these two *drivers* and *passengers*, assessing their biological function is essential, though this remains a challenging task. Numerous methods exist to predict the functional impact of mutations based on *a priori* knowledge. However, these approaches often fail to integrate information effectively across various mutation types and are limited by their reliance on known proteins, rendering them less effective for less-studied ones [22].

With the decreasing cost of DNA sequencing, it is now possible to categorize mutations by examining their frequency, as driver mutations are typically the most recurrent in patients' genomes. For instance, some methods focus on comparing the mutation frequency of an individual gene to that of others within the same or related tumors, while accounting for sequence context and gene size. For genes with a very high number of mutations, such as TP53 or KRAS, most statistical methods will strongly suggest that these genes are drivers; these highly mutated genes are often referred to as *mountains* [22, 34].

However, genes with more than one, but still relatively few mutations, termed *hills*, are more common in cancer genome landscapes. In these cases, mutation frequency and context alone are insufficient to reliably identify driver genes, as background mutation rates can vary significantly among different patients and regions of the genome. Recent research on normal cells has shown that the mutation rate can vary more than 100-fold within the genome. In tumor cells, this variation can be even greater, affecting entire genomic regions in a seemingly random manner [34].

Moreover, because driver mutations are primarily found in genes involved in cell signaling pathways, in many cases different patients harbor mutations in different pathway loci. Consequently, driver mutations can vary widely between patient samples, even within the same cancer type, resulting in minimal overlap of mutated genes across sample pairs, even from the same patient, which further reduces the statistical power of frequency-based analyses [22, 39].

Therefore, methods based solely on mutation frequency can only prioritize genes for expanded investigation and cannot definitively identify driver genes that are mutated at relatively low frequencies [34].

2.1.3 Focus on pathways

An alternative method to assessing the recurrence of individual mutations or genes is to examine mutations within the context of cellular signaling and regulatory pathways, and biological considerations further support this approach. In particular, multiple alternative driver mutations in different genes can lead to similar downstream effects, hence the selective advantage is distributed across the frequencies of these gene alterations, which means that different mutations can affect the same pathway across various samples [2, 22]. This suggests that the focus should be on driver pathways rather than on individual driver mutations. Indeed, most recent cancer genome sequencing studies analyze known pathways for enrichment of somatic mutations, and methods have been developed to identify pathways that are significantly mutated

across multiple patients. Additionally, new algorithms have extended pathway analysis to genome-scale gene interaction networks [33].

However, pathway analysis relies on the prior identification of gene groups within known pathways. While some of them are well-documented and cataloged in multiple databases, our understanding remains incomplete. In particular, many databases aggregate all elements of pathways but often lack details on which specific components are active in particular cell types.

These limitations, along with the increasing number of sequenced cancer genomes, raise the question of whether it is possible to automatically identify groups of genes with driver mutations or mutated driver pathways directly from somatic mutation data, collected from a large number of patients [33]. This topic will be explored in the following sections, which will discuss the various techniques developed for identifiying driver pathways based on mutation data.

2.1.4 Searching for driver pathways

Finding mutated driver pathways may seem implausible, because of the enormous number of possible gene sets to test, e.g. there are more than 10^{26} sets of 7 human genes. This makes it necessary to find specific properties or characteristics to guide the search efficiently. Fortunately, our current understanding of the somatic mutational processes in cancer suggests constraints on the expected patterns of mutations, which considerably narrow down the number of gene sets that need to be considered [33].

First, studies suggest that a major cancer pathway should be disrupted in a substantial number of patients, thus it is expected that most patients will exhibit aberrations in some gene within this pathway. Therefore, it is assumed that driver genes constituting a driver pathway are frequently mutated across many samples, a property that is referred to as **coverage** [33].

Second, while this feature is useful for identifying driver pathways, most techniques developed in recent years for recognizing driver pathways leverage a much stronger statistical property observed in cancer patient data: each patient typically has a relatively small number of mutations that affect multiple pathways, thus each pathway will contain *1 driver mutation on average* per sample. This concept of **mutual exclusivity** among driver mutations within the same pathway, as statistically observed in patient samples, is then axiomatized and employed by research algorithms designed to identify driver mutations and pathways [22]. Note that mutual exclusivity *does not affect different pathways*, as it occurs exclusively within a single pathway.

Therefore, *a driver pathway consists of genes that are mutated in numerous patients, with mutations being approximately mutually exclusive.* It is also observed that pathways exhibiting these characteristics are generally shorter and comprised of fewer genes on average [22].

While the precise explanation for this phenomenon is not yet fully understood, several hypotheses appear plausible [10, 7, 33]:

- one hypothesis is that mutually exclusive genes are functionally connected within a common pathway, acting on the same downstream effectors and creating functional redundancy; consequently, they would share the same selective advantage, meaning that the alteration of one mutually exclusive gene would be sufficient to disrupt their shared pathway, thereby removing the selective pressure to alter the others; this explanation, however, does not fully account for the phenomenon because the co-alteration of mutually exclusive genes should not result in negative effects on the cell;

- an alternative explanation is that the co-occurrence of mutually exclusive alterations is detrimental to cancer survival, leading to the elimination of cells that harbor such co-occurrences; moreover, some pairs of mutually exclusive genes could be *synthetic lethal*, meaning that while the alteration of one gene may be compatible with cell survival, the simultaneous aberration of both genes would be lethal to the cell.

An example of the latter is provided by the gene pair ERG and SPOP, which are commonly overexpressed in patients with prostate cancer, but they are mutually exclusive due to their *synthetic lethality*. Wild-type SPOP facilitates the degradation of various proteins, including ZMYND11, which regulates androgen receptor (AR) signaling. Tumors with mutant ERG require reduced AR signaling to sustain their cancerous effects; therefore, mutant ERG upregulates WT SPOP to enhance the degradation of ZMYND11 and lower AR signaling. In contrast, when SPOP is mutated, it loses the ability to degrade ZMYND11, leading to its accumulation and increased AR signaling. This amplified AR signaling is incompatible with the function of mutant ERG, which relies on low AR signaling. Consequently, while ERG and SPOP mutations can each support oncogenic activity individually, their simultaneous aberration is not viable due to the conflicting requirements for AR signaling [3].

The next sections will focus on the algorithms and techniques developed to quantify levels of mutual exclusivity within gene groups.

2.2 Assessing mutual exclusivity

2.2.1 Challenges in quantifying mutual exclusivity

Finding an effective method to appropriately quantify the level of mutual exclusivity is not straightforward. In the statistical literature, two types of mutual exclusivity are defined: *hard* and *soft*. Hard mutual exclusivity describes events that are presumed to be strictly mutually exclusive, with the null hypothesis being that any observed overlap is due to random errors. However, in this context, it is not feasible to test for hard mutual exclusivity, as this is a property observed statistically from patient data. Therefore, it is necessary to relax the constraint to soft mutual exclusivity, where two otherwise independent events overlap less than expected by chance due to some statistical interaction.

Moreover, while soft mutual exclusivity of a pair of genes can be assessed using the Fisher's exact test, there is no agreed-upon method for analytically testing mutual exclusivity among more than two genes. For instance, one intuitive approach could involve checking whether each pair of genes within the group exhibits mutual exclusivity; this method, however, may be overly strict, as a gene set can exhibit a strong mutual exclusivity pattern as a whole even if no individual pairs show any [2].

Due to the complexity of measuring mutual exclusivity, recent papers have proposed various approaches, based on different assumptions, which will be discussed in later sections.

2.2.2 A deterministic formalization of mutual exclusivity

One of the earliest [10] and most widely used mathematical formalizations for modeling and quantifying mutual exclusivity was introduced by Vandin et al. [33], the authors of an algorithm called Dendrix. But, before discussing it, some preliminary definitions are needed to provide context. In fact, all papers explored in this work will reference the following definitions.

Definition 2.1 (Mutation matrix)**.** A **mutation matrix** is a matrix with m rows and n columns, where each row represents a patient and each column represents a gene, and the entry $a_{i,j}$ is equal to 1 if and only if gene j is mutated in patient i.

Example 2.1 (Mutation matrix)**.** An example of a mutation matrix is the following:

	g_1	g_2	g_3
p_1	0	1	0
p_2	1	1	0
p_3	0	0	1

Table 2.1. A mutation matrix.

Definition 2.2 (Coverage of a gene)**.** Given a gene g, the **coverage of** g denotes the set of patients which have g mutated, and it is defined as follows

$$\Gamma(g) := \{i \mid a_{i,g} = 1\}$$

Definition 2.3 (Mutual exclusivity)**.** A set M of genes is **mutually exclusive** if no patient has more than one mutated gene of M, formally

$$\forall g, g' \in M \quad \Gamma(g) \cap \Gamma(g') = \varnothing$$

Definition 2.4 (Coverage of a set)**.** Given a set M of genes, the **coverage of** M denotes the set of patients in which at least one of the genes of M is mutated, and it is defined as follows

$$\Gamma(M) := \bigcup_{g \in M} \Gamma(g)$$

Note that any gene set M of size k can be thought of as an $m \times k$ column submatrix of a mutation matrix A of size $m \times n$, up to rearranging A's columns (their order does

not matter since they represent genes). Accordingly, such a submatrix is said to be **mutually exclusive** if each row contains at most one 1. These two representations will be used interchangeably throughout this work.

To define an equation that mathematically assesses mutual exclusivity and coverage of a given gene set M, the formalization should reflect the following properties (discussed in previous sections):

i) high coverage: most patients have at least one mutation in M;

ii) high approximate exclusivity: most patients have exactly one mutation in M.

To evaluate these two properties, Vandin et al. [33] introduced a measure that quantifies the trade-off between the two. First, they define the following formula, which measures M's *coverage overlap*.

Definition 2.5 (Coverage overlap)**.** Given a set M of genes, the **coverage overlap of** M is defined as follows:

$$\omega(M) := \sum_{g \in M} |\Gamma(g)| - |\Gamma(M)|$$

Note that the sum in Definition 2.5 is the number of 1s in M's corresponding submatrix.

Example 2.2 (Coverage overlap)**.** Considering the mutation matrix in Example 2.1; if $M = \{g_1, g_2\}$, then

$$\omega(M) = |\Gamma(g_1)| + |\Gamma(g_2)| - |\Gamma(\{g_1, g_2\})| = |\{p_2\}| + |\{p_1, p_2\}| - |\{p_1, p_2\}| = 1 + 2 - 2 = 1$$

Indeed, $\omega(M)$ is the number of patients that are counted more than once in the sum, i.e. *the number of patients that have more than one mutation in* M. Note that $\omega(M) \geq 0$, with equality holding only if no patient has more than one mutated gene of M.

Definition 2.6 (Mutual exclusivity)**.** A gene set M is considered to be **mutually exclusive** if $\omega(M) = 0$.

Note that this definition matches the one given in Definition 2.3.

Finally, the equation developed by Vandin et al. [33] can be described.

Definition 2.7 (Weight of gene set)**.** Given a set of genes M, to take into account both coverage and coverage overlap, the following measure is introduced:

$$W(M) := |\Gamma(M)| - \omega(M) = 2\,|\Gamma(M)| - \sum_{g \in M} |\Gamma(g)|$$

This weight assesses the degree of mutual exclusivity among M's genes, and the extent to which their mutations cover the patient data. It does this by calculating

M's coverage and subtracting M's coverage overlap. Indeed, $W(M) = \Gamma(M)$ when M is mutually exclusive, because it has no coverage overlap. Therefore, the *optimal gene set* is the one that **maximizes** its weight, as a higher weight value indicates greater levels of coverage and mutual exclusivity.

As previously mentioned, a gene set M can be represented as a column submatrix of a mutation matrix A. Thus, finding the *optimal gene set* is equivalent to identifying the *optimal A's column submatrix*, which means that the following problem has to be solved.

> **Maximum Weight Submatrix Problem** (MWSP): Given an $m \times n$ mutation matrix A, and an integer $k > 0$, find a $m \times k$ submatrix of A that maximizes $W(M)$.

Finding the solution to this problem is computationally difficult, even for small values of k (e.g. there are $\approx 10^{23}$ subsets of size $k = 6$ of 20,000 genes). In fact, the following proof (provided by Vandin et al. [33]) shows that this problem is NP-Complete.

Theorem 2.1 (The MWSP is NP-Complete)**.** The Maximum Weight Submatrix Problem is NP-Complete.

Proof. The associated decision problem of the MWSP is formulated as follows:

$$h{-}MWSP := \{\langle A, h\rangle \mid \exists M \text{ column submatrix of } A : W(M) = h\}$$

It can be shown that this problem is in NP, as follows. Consider the following verifier V, which takes an input $\langle w, c\rangle$, and computes as described below:

- interpret w as $\langle A, h\rangle$, where A is a $m \times n$ matrix, and c as a matrix M with m rows; if the encoding is not correct, *reject*;
- if M is not a submatrix of A, *reject*;
- evaluate $W(M)$; *accept* if and only if $W(M) = h$.

It follows that

$$\begin{aligned} \langle A, h\rangle \in h{-}MWSP &\implies \exists M \text{ submatrix of } A \mid W(M) = h \\ &\implies \exists c = M \mid \langle \langle A, h\rangle, M\rangle \in L(V) \end{aligned}$$

$$\begin{aligned} \langle A, h\rangle \notin h{-}MWSP &\implies \text{incorrect coding} \vee \nexists M \text{ submatrix of } A \mid W(M) = h \\ &\implies \nexists c = M \mid \langle \langle A, h\rangle, M\rangle \in L(V) \end{aligned}$$

therefore

$$\langle A, h\rangle \in h{-}MWSP \iff \exists M \text{ submatrix of } A \mid \langle \langle A, h\rangle, M\rangle \in L(V)$$

hence V verifies $h{-}MWSP$. Note that V operates in polynomial time, as each of its computations can be completed in polynomial time; in particular, computing $W(M)$ for a submatrix M of dimensions $m \times k$ requires $O(m \cdot k)$ time.

The proof of NP-Hard-ness is by reduction from the Independent Set Problem (ISP), which is known to be NP-Hard [15]. In the ISP, it is asked whether there is an independent set of size k in a given graph G. An independent set for $G = (V, E)$ is a set of vertices $I \subseteq V(G)$ such that there is no edge among the vertices of I, i.e.

$$\forall u, v \in I \mid u \neq v \quad (u, v) \notin E(G)$$

Given an instance of the ISP, a mutation matrix representing an instance of the MWSP is built in polynomial time as follows:

- let $\Delta := \max_{v \in G} \deg(v)$, and for each $v \in V(G)$ let $\mathcal{S}_v := \left\{ s_v^{(1)}, \ldots, s_v^{(\Delta - \deg(v))} \right\}$ be a set of variables; also, consider the following set

$$\mathcal{S} := \{ s_e \mid e \in E(G) \} \cup \left(\bigcup_{v \in V(G)} \mathcal{S}_v \right)$$

- build a matrix A of size $|\mathcal{S}| \times |V(G)|$, as illustrated below

	v_1	$\ldots$	v_n
s_{e_1}	$\ddots$		
$\vdots$		$\ddots$	
s_{e_m}			$\ddots$
$s_{v_1}^{(1)}$	$\ddots$		
$\vdots$		$\vdots$	
$s_{v_1}^{(\Delta - \deg(v_1))}$			
$\vdots$		$\vdots$	
$s_{v_n}^{(1)}$			
$\vdots$		$\vdots$	
$s_{v_n}^{(\Delta - \deg(v_n))}$			$\ddots$

Table 2.2. The described matrix.

- define A's cells as follows:

$$a_{s,v} = 1 \iff s = s_{(u,v)}, u \in V(G) \vee s \in \mathcal{S}_v$$

which means that $a_{s,v}$ will be 1 if and only if s is either a variable from the set $\{ s_e \mid e \in E(G) \}$ where the edge e has v as endpoint, or s is a variable defined in $\mathcal{S}_v$.

Note that this matrix can be built in polynomial time, because

- its first half contains $m \cdot n$ cells
- the vertex $\hat{v}$ that maximizes $|\mathcal{S}_{\hat{v}}|$ is such that

  $$\deg(\hat{v}) = 1 \implies |\mathcal{S}_{\hat{v}}| = \Delta - \deg(\hat{v}) = \Delta - 1 = O(\Delta) = O(n)$$

  and since there are n sets, the matrix's second half contains $(n \cdot n) \cdot n = n^3$ cells

therefore the time complexity to create it is $O\left(n^3 + nm\right)$. Moreover, note that:

i) $\forall v \in V(G) \quad |\Gamma(v)| = \Delta$ due to the added variables at the end of each column;

ii) $\forall u, v \in V(G) \quad \Gamma(u) \cap \Gamma(v) \neq \varnothing \iff (u, v) \in E$, since no pair of columns can have a 1 in the same row in the second half of A by definition of the sets $\mathcal{S}_{v_1}, \ldots, \mathcal{S}_{v_n}$, therefore $\Gamma(u)$ and $\Gamma(v)$ can have an intersection if and only if there is an edge $(u, v) \in E(G)$.

Hence, consider a set $M = \{v_1, \ldots, v_k\}$ of k columns of A. Note that:

- from (i) it follows that

  $$\sum_{i=1}^{k} |\Gamma\left(v_i\right)| = k\Delta$$

  and consequently $|\Gamma(M)| \leq k\Delta$, meaning that the largest value $|\Gamma(M)|$ can have is $k\Delta$; thus, from the equation in Definition 2.7, it follows that the maximum value $W(M)$ can reach is

  $$\begin{aligned} W(M) &= 2\,|\Gamma(M)| - \sum_{i=1}^{k} |\Gamma\left(v_i\right)| \\ &= 2k\Delta - k\Delta \\ &= k\Delta \end{aligned}$$

- from (ii) it follows that $|\Gamma(M)| = k\Delta \iff \forall u, v \in V(G) \quad \Gamma(u) \cap \Gamma(v) = \varnothing \iff \forall u, v \in V(G) \quad (u, v) \notin E(G) \iff M$ is an independent set, by definition.

This means that $W(M) = k\Delta$, i.e. is maximized, if and only if M is an independent set; therefore, the MWSP can be solved on A if and only if the ISP can be solved on G. □

The approach developed by Vandin et al. [33] will be discussed in the next chapter.

2.2.3 A statistical approach

Given that exact mutual exclusivity in real somatic data is unlikely, a common approach in this field is to rely on statistical methods. The following section will describe the metric developed by Babur et al. [2], that employs statistical analysis to identify driver pathways.

To begin with, Babur et al. [2] criticize the metric developed by Vandin et al. [33], because it has a strong bias toward highly mutated genes, and in some instances the excessive emphasis on coverage leads to false positives and negatives. Therefore, they propose a metric that extends Fisher's exact test — also known as *hypergeometric test* — to quantify the mutual exclusivity within a gene set.

Before exploring their metric, it is important to note that a uniform alteration frequency across samples may not always hold, particularly for hyper-mutated samples, often resulting from prior mutations in DNA repair mechanisms. Addressing this heterogeneity is challenging, as each overlap in the null model has a different probability. This remains an open problem, and to partially mitigate it, Babur et al. [2] decided to exclude hyper-altered samples from the analysis.

The following definitions will introduce the metric they developed. Consider the following null hypothesis:

> H_0: *Given a group of genes, a member gene is altered independently from the union of the other alterations in the group.*

Using Dendrix's notation, H_0 states that for a given gene set M, for every gene $g \in M$, mutations in $\Gamma(g)$ are independent from alterations in $\Gamma(M - \{g\})$. In brief, H_0 states that any observed pattern among gene alterations is due to *random chance*, not due to any underlying biological or oncogenic mechanism. Given a single gene g, H_0 can be tested by evaluating g's co-distribution with the union of the others, through an *hypergeometric test*, which is performed as described below.

Definition 2.8 (Notation)**.** Let M be a gene set, and let $g \in M$; define the following variables:

- $\gamma(g) := |\Gamma(g)|$
- $\gamma(M) := |\Gamma(M)|$
- $M_g := M - \{g\}$
- $\gamma(g, M_g) := |\Gamma(g) \cap \Gamma(M_g)|$

To test H_0 for the gene $g \in M$, it is necessary to quantify *the probability that there are $\gamma(g, M_g)$ patients who have both gene g and any gene in M mutated*; let this probability be represented by the random variable X. Since X follows an hypergeometric distribution, denoted as

$$X \sim H(m, \gamma(g), \gamma(M_g))$$

this probability can be assessed by using the PMF of the hypergeometric distribution, namely:

$$P(X = \gamma(g, M_g)) = \frac{\binom{\gamma(g)}{\gamma(g, M_g)}\binom{m - \gamma(g)}{\gamma(M_g) - \gamma(g, M_g)}}{\binom{m}{\gamma(M_g)}}$$

Note that, by using the inclusion-exclusion principle

$$|\Gamma(M)| = |\Gamma(g)| + |\Gamma(M - \{g\})| - |\Gamma(g) \cap \Gamma(M - \{g\})|$$

this probability can also be evaluated using Fisher's exact test, by employing the following contingency table:

	alterations in $\Gamma(g)$	alterations *not* in $\Gamma(g)$	
alterations in $\Gamma(M - \{g\})$	$\gamma(g, M_g)$	$\gamma(M_g) - \gamma(g, M_g)$	$\gamma(M_g)$
alterations *not* in $\Gamma(M - \{g\})$	$\gamma(g) - \gamma(g, M_g)$	$m - \gamma(M)$	$m - \gamma(M_g)$
	$\gamma(g)$	$m - \gamma(g)$	m

Table 2.3. Fisher's exact test

This probability is used to determine the mutual exclusivity score of M, but further details will be discussed in the next chapter, as the scoring method involves multiple functions that are integral to running the algorithm itself. The following algorithm will be used to evaluate M's score.

Algorithm 2.1 *p-values procedure*: given a gene set M, derived from a mutation matrix A, the algorithm returns the p-values of each gene $g \in M$.

```
1: function pValues(M, A)
2:     P := {}
3:     for g ∈ M do
4:         P.add_entry(g, P(X ≤ γ(g, M_g)))        ▷ this is g's p-value
5:     end for
6:     return P
7: end function
```

2.2.4 Extending the deterministic equation

While identifying individual driver pathways is crucial for cancer research and treatment, most cancer patients are likely to have driver mutations across multiple pathways. The metrics discussed so far do not account for multiple pathways simultaneously. As will be shown in the next chapter, these formalizations can be applied iteratively to identify multiple driver pathways, though this may not be the most precise approach. Leiserson et al. [22] refine the weight function of Vandin et al. [33] to extend their metric, enabling the assessment of mutual exclusivity across multiple driver pathways.

To effectively identify multiple pathways, it is necessary to establish criteria for evaluating potential *collections of gene sets*. Based on the same biological reasoning mentioned earlier, it is expected that each pathway will contain approximately one driver mutation. Furthermore, since each driver pathway is crucial for cancer development, it is expected that most patients will harbor a driver mutation in most driver pathways. Consequently, high exclusivity is predicted within the genes of each pathway, along with high coverage of each pathway individually. One metric that meets these criteria is the sum of individual weights of a given collection of gene sets, as defined below.

Definition 2.9 (Weight of a collection)**.** Given a collection of t gene sets $M = \{M_1, \ldots, M_t\}$, the **weight of** M is defined as follows:

$$W'(M) := \sum_{\rho=1}^{t} W(M_\rho)$$

This metric is employed by Leiserson et al. [22] in their algorithm, which will be explored in the next chapter.

2.2.5 A clustering method

Another notable approach used in several studies to find mutually exclusive modules involves constructing gene graphs and identifying clusters based on specific criteria. This method is demonstrated by Hou et al. [16], who propose an algorithm designed to address the limitations of previous techniques. They argue that earlier approaches are generally inefficient for large datasets, lack consistency in results due to their randomized nature, and can only identify a few small modules. Additionally, these methods require restructuring whenever new biological information is added, whereas their approach has a notable degree of flexibility, as its objective function does not need to change with the addition of new data sources. Moreover, it has low computational cost, an important consideration in this context, as previously discussed. The following equations will describe how Hou et al. [16] formalized biological assumptions to define a technique able to achieve these results.

First, the structure of the graph will be described. Let $G = (V, E)$ be a *complete graph* of genes, thus an edge exists between any pair of vertices. Each edge $(u, v) \in E(G)$ is assigned two weights:

- a **positive weight** w^+_{uv}, which represents *the cost of placing u and v in different clusters*; thus, by making w^+_{uv} large, placing u and v in different clusters is discouraged, and viceversa;
- a **negative weight** w^-_{uv}, which represents *the cost of placing u and v in the same cluster*; thus, by making w^-_{uv} large, placing u and v in the same cluster is discouraged, and viceversa.

The clustering algorithm aims to identify clusters of vertices (i.e. genes) that exhibit both *high coverage and mutual exclusivity within clusters.*

As mentioned earlier, this algorithm is quite flexible and allows the integration of weight values with information derived from external data. Specifically, Hou et al. [16] present multiple versions of their algorithm, depending on the type of information used to define the weights between edges. The following labels will be used to distinguish the components of the weights:

- the (e) label refers to *exclusivity*;
- the (c) label refers to *coverage*;
- the (n) label refers to *network information*;
- the (x) label refers to *expression data*.

To define the weights, linear combinations of these components are utilized. The different versions of this algorithm, and the various definitions of the weights, will be discussed in the following chapter. This secion will specifically focus on how Hou et al. [16] defined mutual exclusivity and coverage.

Let A be an $m \times n$ mutation matrix, as described in Definition 2.1. In addition, let C be an $m \times n$ matrix representing the CNV data, where $c_{i,j} = 0$ means that there is no change in the copy number of gene j in sample i, otherwise, the corresponding number reflects the deviation of the CNV number from its baseline — hence, C contains both positive and negative values. Following this, a binary matrix M is constructed combining A and C, as follows:

$$m_{i,j} = 0 \iff \begin{cases} a_{i,j} = 0 \\ l_{\text{cnv}} < c_{i,j} < h_{\text{cnv}} \end{cases} \tag{2.1}$$

where l_{cnv} and h_{cnv} are lower and upper bounds on copy numbers that determine the significance level. Thus, if $m_{i,j} = 0$, no mutation of gene j is recorded in sample i, otherwise gene j is *deemed mutated*.

Definition 2.10 (Coverage of a vertex)**.** Given a vertex $u \in V(G)$, i.e. a gene, the **coverage of** u is defined as follows

$$\mathscr{S}(u) := \{i \mid m_{i,u} = 1\}$$

and it denotes the set of patients in which u is altered.

Note that $\mathscr{S}(u)$ corresponds to $\Gamma(u)$ under Dendrix's notation, but is defined through the M matrix respectively.

Now that the preliminaries have been covered, the following definitions will outline how mutual exclsuivity and coverage are defined.

Definition 2.11 (Mutual exclusivity component)**.** The **mutual exclusivity component** between two genes $u, v \in V(G)$ is defined as follows:

$$w^{-}_{uv}(\text{e}) := a \cdot \frac{|\mathscr{S}(u) \cap \mathscr{S}(v)|}{\min(|\mathscr{S}(u)|, |\mathscr{S}(v)|)}$$

where a is a user-defined scaling parameter.

This ratio is often referred to as *Intersection over Minimum* (IoM), and suits the criteria of mutual exclusivity because the fewer patients who have both u and v mutated, the smaller the weight, making it more plausible that u and v are mutually exclusive, therefore the cost of placing them in the same cluster should be low. Note that

$$\forall u, v \in V(G) \quad a = 1 \implies 0 \leq w^{-}_{uv}(\mathrm{e}) \leq 1 \tag{2.2}$$

Focusing on **coverage**, if two genes u and v increase the coverage of the set significantly, $w^{+}_{uv}(\mathrm{c})$ should be large such that they are encouraged to be placed in the same cluster. Let

$$D(u, v) := |\mathscr{S}(u) \Delta \mathscr{S}(v)| \tag{2.3}$$

where Δ denotes the symmetric difference of two sets; a large value of $D(u, v)$ suggests that u and v should be placed in the same cluster. Also, let

$$\mathscr{D} := \{D(u, v) \mid u, v \in V(G)\} \tag{2.4}$$

and let $T(J)$ be the J-th percentile of the values in $\mathscr{D}$.

Definition 2.12 (Coverage component)**.** The **coverage component** is defined as follows:

$$w^{+}_{uv}(\mathrm{c}) := \begin{cases} 1 & D(u, v) > T(J) \\ \dfrac{D(u, v)}{T(J)} & D(u, v) \leq T(J) \end{cases}$$

Note that, similar to Equation 2.2

$$\forall u, v \in V(G) \quad 0 \leq w^{+}_{uv}(\mathrm{c}) \leq 1 \tag{2.5}$$

The next chapter will illustrate the various versions of the algorithm developed by Hou et al. [16].

Chapter 3

Finding driver pathways

In the previous chapter, various studies were discussed in terms of how they formalized biological assumptions, with particular emphasis on the metrics developed to assess *coverage* and *mutual exclusivity* within gene groups. This chapter will delve deeper into the algorithms employed by these studies to identify driver pathways using their respective metrics and hypotheses.

Existing approaches can be categorized into two types: ***de novo*** approaches, which identify mutually exclusive patterns using only genomic data from patients, and ***knowledge-based*** methods, which integrate the analysis with external *a priori* information. *De novo* approaches might lack sufficient information as they do not utilize pre-existing pathway databases, protein-protein interaction (PPI) networks or phenotype data. Conversely, given that our understanding of gene and protein interactions in humans is still incomplete, and many pathway databases fail to accurately represent the specific pathways and interactions present in cancer cells, *knowledge-based* approaches may be limited by their dependence on existing data sources. Consequently, *de novo* methods might yield new but potentially less accurate results, while *knowledge-based* approaches may limit the discovery of novel biological insights [10, 22].

3.1 Dendrix

3.1.1 A greedy approach

Vandin et al. [33] introduced the most widely adopted metric in pathway discovery research, namely $W(M)$ (presented in Definition 2.7). In addition to this, they defined the Maximum Weight Submatrix Problem (MWSP), discussed in Section 2.2.2, and proposed the following *greedy algorithm*, called Dendrix (*de novo* [10]), to solve it.

Algorithm 3.1 *Greedy Dendrix*: given the set of all genes $\mathcal{G}$, and an integer k, the algorithm finds the set of genes M of size k that maximizes $W(M)$.

```
1: function GREEDYDENDRIX(G, k)
2:     M := {g1, g2} such that M maximizes W(M)      ▷ pick the best gene pair
3:     for i ∈ [3, k] do
4:         Choose ĝ ∈ arg max_{g∈G} W(M ∪ {g})
5:         M = M ∪ {ĝ}
6:     end for
7:     return M
8: end function
```

Clearly, the time complexity of the algorithm is $O\left(n^2 + kn\right) = O\left(n^2\right)$ — where $n = |\mathcal{G}|$, therefore $k \leq n$ — because finding $\{g_1, g_2\}$ in line 2 requires $O\left(n^2\right)$ and the arg max in line 4 has cost $O(n)$.

While this algorithm is efficient, there is generally no guarantee that it will identify the optimal set $\hat{M}$ that maximizes $W(\hat{M})$. However, Vandin et al. [33] prove that Algorithm 3.1 can correctly identify $\hat{M}$ with high probability when the mutation data come from the *Gene Independence Model* (GIM), which is described below

Definition 3.1 (Gene Independence Model)**.** Let A be an $m \times n$ mutation matrix, such that $\hat{M}$ is the *maximum weight submatrix* of A, and $|\hat{M}| = k$; the matrix A satisfies the **Gene Independence Model** (GIM) if and only if:

i) each gene $g \notin \hat{M}$ is mutated in each patient with probability $p_g \in [p_L, p_U]$, independently of all other events, for some $0 \leq p_L, p_U \leq 1$;

ii) $W(\hat{M}) = rm$ for some $0 < r \leq 1$;

iii) for all $M \subset \hat{M}$ of cardinality $l := |M|$, it exists $0 \leq d < 1$ such that

$$W(M) \leq \frac{l+d}{k} W(\hat{M})$$

Note that:

- condition (i) reflects the independence of mutations for genes outside the mutated pathway, a standard assumption for somatic single-nucleotide mutations, according to Vandin et al. [33];
- condition (ii) ensures that mutations in $\hat{M}$ cover a large number of patients and are mostly exclusive;
- condition (iii) means that each gene in $\hat{M}$ is important, so there are no subset of $\hat{M}$ that predominantly contributes to $W(\hat{M})$.

Although it is possible to efficiently obtain accurate results with high probability under the GIM, genes within $\hat{M}$ may exhibit *observed mutation frequencies* similar

to those of genes outside $\hat{M}$. This similarity can make it challenging to distinguish between them based solely on mutation frequency, regardless of the number of patients.

To illustrate this, consider the following scenario: observed gene mutation frequencies fall within the range of $[3 \times 10^{-5}, 0.13]$ — based on a background mutation rate of $\approx 10^{-6}$ [8]. If somatic mutations are measured in $n = 20,000$ human genes, and the size of $\hat{M}$ is 10, then approximately 2,400 patients are needed for the greedy algorithm to identify $\hat{M}$ with a probability of at least $1 - 10^{-4}$. While this number of patients is expected to be available from large-scale cancer sequencing projects [17], it exceeds current availability.

Therefore, in practical applications, the effectivness of this greedy algorithm depends on having mutation data from a sufficiently large number of patients. Moreover, the GIM model may be appropriate for certain types of somatic mutations, such as single-nucleotide aberrations, but may not be suitable for others. To address these limitations, Vandin et al. [33] developed an alternative approach, which will be detailed in the following section.

3.1.2 Using MCMC

To overcome the drawbacks of the aforementioned greedy algorithm, Vandin et al. [33] developed a *Markov Chain Monte Carlo* (MCMC) approach, which does not rely on assumptions about the distribution of mutation data or the number of patients. In particular, this MCMC method does not assume independence among mutations in different genes, making it particularly useful for analyzing copy-number aberrations (CNAs), which often involve correlated mutations due to amplification or deletion of adjacent genes.

Vandin et al. [33] employed a Metropolis-Hastings algorithm to sample sets $M \subseteq \mathcal{G}$ of k genes, with the following stationary distribution, proportional to $e^{cW(M)}$, for some constant $c > 0$

$$\pi(M) = \frac{e^{cW(M)}}{\sum_{R \in \mathcal{M}_k} e^{cW(R)}}$$

where $\mathcal{M}_k := \{M \subset \mathcal{G} : |M| = k\}$. While there are no guarantees on the rate of convergence of the Metropolis-Hasting algorithm to the stationary distribution, Vandin et al. [33] prove that their MCMC is rapidly mixing, therefore the stationary distribution is effectively reached in a practical number of steps.

The main idea of the MCMC algorithm involves constructing a Markov chain, where each state represents a collection of k columns from a given mutation matrix A, and transitions between these states occur by swapping one gene. Further details of the algorithm are discussed below.

Definition 3.2 (Dendrix's MCMC)**.** The **MCMC's procedure** of Dendrix is defined through the following steps:

1. *initialization*: given the set of all genes $\mathcal{G}$, choose an arbitrary subset $M_0 \subseteq \mathcal{G}$ of k genes;

2. *iteration*: for $t = 1, 2, \ldots$ derive M_{t+1} from M_t as follows:
   (a) choose a gene w uniformly, at random, from $\mathcal{G}$;
   (b) choose a gene v uniformly, at random, from M_t;
   (c) let $M'_t := (M_t - \{v\}) \cup \{w\}$;
   (d) let $\Delta_W := W(M'_t) - W(M_t)$;
   (e) let $P(M_t, w, v) := \min\left[1, e^{c\Delta_W}\right]$, where $c > 0$;
   (f) set $M_{t+1} := M'_t$ with probability $P(M_t, w, v)$, else $M_{t+1} := M_t$.

Note that:

- w is choosen from $\mathcal{G}$, thus if $w \in M_t$ then
$$M'_t = (M_t - \{v\}) \cup \{w\} = M_t - \{v\}$$
which means that no genes were added to W'_t; this must be allowed because maximizing the weight may require removing genes already present in the current set, without adding new ones;
- Δ_W measures the change in the weight function with the new set, and the constant c is a scaling factor that adjusts the importance of this difference; note that
$$\Delta_W \geq 0 \implies e^{c\Delta_W} \geq 1 \implies P(M_t, w, v) = 1 \implies M_{t+1} = M'_t$$
in fact when $\Delta_W > 0$ the weight has improved, therefore the next iteration should start from M'_t; conversely
$$\Delta_W < 0 \implies e^{c\Delta_W} < 1 \implies P(M_t, w, v) = e^{c\Delta_W}$$
which means that when $\Delta_W < 0$ (i.e., the weight has decreased) the change will be perfomed with probability $e^{c\Delta_W}$ — note that this value is still close to 1 if the weight has not decreased significantly.

Vandin et al. [33] prove that their MCMC is rapidly mixing for some $c > 0$, but details of this proof are beyond the scope of this work. The following sections will briefly discuss some of the results obtained with their algorithm.

3.1.3 Results

This section will briefly discuss the results reported by Vandin et al. [33] from running the MCMC algorithm on real data. To improve the efficiency of the algorithm, the mutation matrix was optimized by merging genes $T = \{g_1, \ldots, g_h\}$ that were mutated in the same patients into larger *metagenes* g_T, where each metagene represents the combined mutations occurring in those same patients. The MCMC algorithm samples gene sets with a frequency proportional to their weights, thus in order to focus on high-weight sets, only those with a frequency of at least 1% were reported.

Vandin et al. [33] applied their MCMC algorithm (with the constant c set to $c = 0.5$) to analyze somatic mutations obtained from high-throughput genotyping of 238 oncogenes across 1,000 patients, spanning 17 cancer types (a study conducted by Thomas et al. [32]). A mutation matrix was constructed with 298 patients and 18 mutation groups, based on the groupings from Thomas et al. [32]. They ran the MCMC algorithm on gene sets ranging in size from 2 to 10, sampling every 10,000 iterations after running the algorithm for 10 million iterations. The analysis identified a set of 8 mutation groups, altered in 94% of patients with at least one mutation, totaling 295 mutations ($p < 0.01$). They state that these mutated genes are linked to well-known cancer pathways. Additionally, two sets of size 10, which included the initial 8 mutation groups, were found in 95% of patients, accounting for 302 mutations ($p < 0.01$).

They also applied their algorithm to somatic mutations in lung adenocarcinoma, using data from The Cancer Genome Atlas (TCGA) [8], which included 188 patients and 623 genes, of which 356 were found to be mutated in at least one patient. For gene sets of size $k = 2$, the pair (EGFR, KRAS) was identified in 99% of samples, covering 90 patients, with no overlap, indicating high mutual exclusivity; additionally, when analyzing sets of size $k = 3$, the algorithm uncovered a novel triplet (EGFR, KRAS, STK11), appearing in 8.4% of samples; the significance of both the pair and the triplet was confirmed through permutation tests. Vandin et al. [33] highlight that all three genes are involved in regulating the mTOR pathway, which is known to be crucial in lung adenocarcinoma. To identify additional sets, the MCMC algorithm was rerun after removing the triplet, revealing the pair (ATM, TP53) with a sampling frequency of 56%, covering 76 patients. Although these reported sets showed high exclusivity, their relatively low coverage suggests that they may not represent complete driver pathways. Indeed, Vandin et al. [33] propose that this could be due to the limited number of genes analyzed or the focus on specific mutation types. Moreover, there was no significant overlap between patients with mutations in (ATM, TP53) and those with mutations in (EGFR, KRAS, STK11), suggesting that these gene sets likely belong to distinct biological pathways.

The MCMC algorithm was also applied to mutation data from 84 patients with glioblastoma multiforme (GBM), analyzing somatic mutations across 601 genes, resulting in a total of 453 mutations, with 223 genes found to be mutated in at least one patient (filtered using CNAs). For gene sets of size $k = 2$, the most frequently sampled pair was (CDKN2B, CYP27B1), appearing 18% of the time, and for $k = 3$, the triplet (CDKN2B, RB1, CYP27B1) was sampled 10% of the time — a permutation test confirmed the significance of this triplet. The analysis revealed that CYP27B1 had a nearly identical mutation profile to a metagene composed of six adjacent genes, but was excluded due to an additional mutation in a single patient. The metagene's amplification likely targeted CDK4, suggesting that the key triplet of interest was (CDKN2B, RB1, CDK4), which is part of the RB1 signaling pathway, associated with shorter survival in GBM patients. After removing this triplet, the pair (TP53, CDKN2A) was sampled with 30% frequency, linked to the p53 tumor suppression pathway. Further analysis, after removing both sets, revealed the pair (NF1, EGFR), which was sampled 44% of the time and is part of the RTK pathway, crucial for cell proliferation and survival.

All these findings highlight the ability of the MCMC algorithm to identify significant gene sets related to known cancer pathways.

The following section will introduce an extension of the MWSP, along with a modification to the weight function $W(M)$.

3.2 Multi-Dendrix

3.2.1 An alternative approach to the MWSP

Vandin et al. [33] propose a solution to the MWSP that may not appear immediately intuitive, given that the problem itself resembles an optimization problem. Indeed, Leiserson et al. [22], the authors of Multi-Dendrix (*de novo* [10]), present an alternative approach by formulating the problem as an *Integer Linear Program* (ILP), called Dendrix$_{ILP}(k)$, which is described below.

To begin, it is necessary to define two sets of indicator variables: consider a gene set M, described by a set of variables, one for each gene $j \in M$, defined as follows

$$I_M(j) = 1 \iff j \in M \tag{3.1}$$

and a set of indicator variables, one for each patient i, expressed in this form

$$C_i(M) = 1 \iff \exists g \in M \mid i \in \Gamma(g) \tag{3.2}$$

therefore $C_i(M)$ is equal to 1 if and only if M *covers* the i-th patient.

The ILP formulation provided by Leiserson et al. [22] is illustrated below.

Definition 3.3 (Dendrix$_{ILP}(k)$)**.** Dendrix$_{ILP}(k)$ is defined by the following ILP:

$$\text{maximize} \sum_{i=1}^{m} \left(2 \cdot C_i(M) - \sum_{j=1}^{n} I_M(j) \cdot a_{i,j} \right), \tag{3.3}$$

$$\text{subject to} \sum_{j=1}^{n} I_M(j) = k, \tag{3.4}$$

$$\sum_{j=1}^{n} I_M(j) \cdot a_{i,j} \geq C_i(M), \tag{3.5}$$

$$\text{for } 1 \leq i \leq m.$$

Note that Equation 3.3 uses the second version of the definition provided in Definition 2.7, and Equation 3.4 limits the size of M to be exactly k. Moreover, note that Equation 3.5 only forces $C_i(M) = 0$ when the i-th patient has no mutated genes in M, but does not force $C_i(M) = 1$ when the patient has at least one, as required by

Equation 3.2. However, the objective function will be maximized when $C_i(M) = 1$, thus Equation 3.2 is satisfied.

Lemma 3.1 (Correctness of Dendrix$_{ILP}(k)$)**.** *Given a gene set M, the sum in Equation 3.3 correctly evaluates $W(M)$.*

Proof. Rearranging the terms in Equation 3.3

$$\sum_{i=1}^{m}\left(2 \cdot C_i(M) - \sum_{j=1}^{n} I_M(j) \cdot a_{i,j}\right) = 2\sum_{i=1}^{m} C_i(M) - \sum_{i=1}^{m}\sum_{j=1}^{n} I_M(j) \cdot a_{i,j}$$

and it is trivial to check that

$$|\Gamma(M)| = \sum_{i=1}^{m} C_i(M)$$

since it it true by definition, and

$$\sum_{g \in M} |\Gamma(g)| = \sum_{i=1}^{m}\sum_{j=1}^{n} I_M(j) \cdot a_{i,j}$$

because the RHS counts the number of cells of A such that $a_{i,j} = 1$ for every $j \in M$. □

The next section will discuss how Leiserson et al. [22] extended this ILP formulation to enable the search for multiple driver pathways.

3.2.2 The ILP

As outlined in Section 2.2.4, Leiserson et al. [22] propose that the most effective approach for this research is to identify multiple driver pathways rather than focusing on a single one. To accomplish this, they extended the weight metric introduced by Vandin et al. [33] to find a collection of gene sets that maximizes the sum of their individual weights. Specifically, they extended the MWSP as follows:

> **Multiple Maximum Weight Submatrices Problem** (MMWSP): Given an $m \times n$ mutation matrix A, and integer $t > 0$, and two integers $k_{\min}, k_{\max} \geq 0$, find a collection $M = \{M_1, \ldots, M_t\}$ of column submatrices that maximizes $W'(M)$, where each submatrix M_ρ — for $1 \leq \rho \leq t$ — has size $m \times k_\rho$ for some $k_{\min} \leq k_\rho \leq k_{\max}$.

Note that the sets in the optimal collection may vary in size, as different pathways are likely to have different lengths; additionally, note that this problem is NP-Complete, as for the case where $t = 1$ (proof provided in Theorem 2.1). Furthermore, Leiserson et al. [22] state that collections M with a large value of $W'(M)$ are also likely to exhibit higher coverage $\Gamma(M_\rho)$, for each individual gene set M_ρ. Consequently, optimal solutions tend to produce collections where many patients have mutations

in more than one gene set, which may involve pairs or larger groups of co-occurring mutations — a phenomenon observed in real cancer data.

The ILP developed by Leiserson et al. [22] to simultaneously search for multiple driver pathways is described below.

Definition 3.4 (Multi-Dendrix)**.** Multi-Dendrix is defined by the following ILP:

$$\text{maximize} \sum_{\rho=1}^{t} \sum_{i=1}^{m} \left(2 \cdot C_i(M_\rho) - \sum_{j=1}^{n} I_{M_\rho}(j) \cdot a_{i,j} \right), \tag{3.6}$$

$$\text{subject to} \sum_{j=1}^{n} I_{M_\rho}(j) \cdot a_{i,j} \geq C_i(M_\rho), \tag{3.7}$$

$$k_{\min} \leq \sum_{j=1}^{n} I_{M_\rho}(j) \leq k_{\max}, \tag{3.8}$$

$$\text{for } 1 \leq i \leq m, \quad 1 \leq \rho \leq t,$$

$$\sum_{\rho=1}^{t} I_{M_\rho}(j) \leq 1, \quad 1 \leq j \leq n. \tag{3.9}$$

Note that:

- Equation 3.6 and Equation 3.7 extend Equation 3.3 and Equation 3.4 respectively;
- Equation 3.8 allows each gene group to have a size between $k_{\min}$ and $k_{\max}$;
- Equation 3.9 forces each gene to appear in *at most 1 set* within the collection.

Moreover, Leiserson et al. [22] state that this ILP can be extended to allow the gene sets of the collection to overlap, since the genes in the intersection may be involved in multiple biological processes. Hence, Equation 3.9 is replaced with the following equation:

$$\sum_{\rho=1}^{t} I_{M_\rho}(j) \leq \Delta, \quad 1 \leq j \leq n \tag{3.10}$$

where Δ is the maximum number of gene sets a gene can be a member of, and the following constraint is added:

$$\sum_{j=1}^{n} \sum_{\substack{\rho'=1 \\ \rho \neq \rho'}}^{t} I_{M_\rho}(j) \cdot I_{M_{\rho'}}(j) \leq \tau, \quad 1 \leq \rho \leq t \tag{3.11}$$

where τ is the maximum size of the intersection between two gene sets.

3.2.3 Comparing Multi-Dendrix with Iter-Dendrix

Since the greedy algorithm of Dendrix can identify a single driver pathway, finding multiple pathways could be achieved by running the algorithm iteratively. Leiserson et al. [22] provide a detailed explanation of this approach, referred to as Iter-Dendrix, with the pseudocode shown below.

Algorithm 3.2 *Iter-Dendrix*: given the set of all genes $\mathcal{G}$, an integer k, and an integer t, the algorithm finds the collection M of t gene sets of size k that maximizes $W'(M)$.

```
1: function ITERDENDRIX(𝒢, k, t)
2:     M := ∅
3:     for i ∈ [1, t] do
4:         M_i := greedyDendrix(𝒢, k)        ▷ procedure defined in Algorithm 3.1
5:         M = M ∪ {M_i}
6:         𝒢 = 𝒢 − M_i
7:     end for
8:     return M
9: end function
```

This procedure runs the greedy algorithm iteratively, removing the chosen set from $\mathcal{G}$ after each iteration. Vandin et al. [33] discussed this approach toward the end of their work, highlighting certain limitations. In particular, if the gene sets corresponding to each pathway are disjoint, Iter-Dendrix can be effective in identifying these sets, successfully finding disjoint sets M_1 and M_2 with high weight, as exclusivity is only evaluated within sets, not between them. However, if M_1 and M_2 share genes, removing one set could also remove part of the other. In cases where the overlap is minimal, this approach may still identify the remaining portion of the second set. However, if the sets significantly intersect, Iter-Dendrix is likely to fail [33].

Leiserson et al. [22] compare the outputs of their ILP with Iter-Dendrix: denoting with M and I the collections of gene sets obtained from Multi-Dendrix and Iter-Dendrix respectively, they state that $W'(M) \geq W'(I)$. They also argue that M could contain sets with strictly greater weight than the ones comprising I, due to several factors:

- there may be multiple gene sets I_ρ that maximize $W(I_\rho)$ on the ρ-th iteration of Iter-Dendrix, and this version of Dendrix can only extend one of them;
- the gene set I_ρ that maximizes $W(I_\rho)$ selected by Iter-Dendrix in the ρ-th iteration may not be a member of M, since M could include gene sets that are suboptimal when considered in isolation;
- when $k_{\min} < k_{\max}$, Multi-Dendrix may choose gene sets with fewer than $k_{\max}$ genes, if doing so maximizes the overall weight $W'(M)$.

Leiserson et al. [22] state that all of these scenarios occur when analyzing real mutation data.

3.2.4 Results

Leiserson et al. [22] applied Multi-Dendrix and Iter-Dendrix to four somatic mutation datasets: GBM, lung adenocarcinoma, a newer GBM dataset, and BRCA; these datasets were processed to remove low-frequency mutations and outliers. After processing, the GBM dataset included 46 genes from 84 patients, the lung dataset had 190 genes from 163 patients, the newer GBM dataset contained 398 genes from 261 patients, and the BRCA dataset included 375 genes from 507 patients. They focused on results from the GBM and BRCA datasets, as they are more representative of modern genomic data, and the data was obtained from computing collections of sizes ranging between $2 \leq t \leq 4$, with a minimum size $k_{\min} = 3$, and a maximum size ranging between $3 \leq k_{\max} \leq 5$.

In the GBM analysis, both algorithms produced similar results, except Iter-Dendrix identified the IRF5 gene in one case, though Multi-Dendrix ran significantly faster (142 seconds compared to *over 10 hours*). They identified four main modules in the data, corresponding to key signaling pathways related to cancer, with mutations affecting a large proportion of samples:

- RB signaling pathway: this module, including genes such as CDK4, RB1 and CDKN2A/B, was mutated in 87.7% of samples, and it also included mutations in MSL3, a gene with a potential role in cancer that merits further investigation;
- RTK/RAS/PI(3)K pathway: this module included PTEN, PIK3CA, PIK3R1, and IDH1, among others; mutations in this module were present in 62.8% of samples, and while IDH1 is not a known member of this pathway, its mutual exclusivity with other genes suggests complex interactions;
- p53 signaling pathway: this module featured TP53, MDM2, MDM4, and NLRP3, affecting 57.8% of samples; this module highlights critical interactions in cancer progression, and it includes NPAS3, which has emerging links to GBM.
- RTK/RAS/PI(3)K and RB pathways: this module, involving EGFR, PDGFRA, and RB1, appeared in 45.6% of samples; while EGFR and PDGFRA are part of the RTK/RAS/PI(3)K pathway and RB1 is in the RB pathway, the mutual exclusivity here may be influenced by subtype-specific mutations.

When applying the two algorithms to the BRCA dataset, at first the algorithms grouped frequently mutated genes into single sets, despite their high coverage overlap. This was due to the weight function outweighing coverage $|\Gamma(M)|$ over overlap $\omega(M)$. To enhance mutual exclusivity, Leiserson et al. [22] increased the overlap penalty, by using the following modifed weight function:

$$W(M) = |\Gamma(M)| - \alpha\omega(M)$$

and a value of $\alpha = 2.5$. With this adjustment, Multi-Dendrix identified four distinct modules:

- PI(3)K/AKT pathway: this module contains genes such as PTEN, PIK3CA, PIK3R1, AKT1, and HIF3A, and an amplification at 12p13.33; it is mutated in 61% of samples, and it includes not only key genes in this pathway, but also the 12p amplification, though its target remains unclear;
- p53 signaling pathway: this module includes mutations in TP53, CDH1, GATA3, CTCF, and GPRIN2, affecting 56% of samples; this module relates to known breast cancer-related genes involved in metastasis and proliferation, but it does not have any known interactions;
- p38-JNK1 stress kinase pathway: this module features mutations in MAP2K4, MAP3K1, PPEF1, SMARCA4, and WWP2, present in 44.4% of samples; it includes both kinases and a phosphatase, though interactions within this module are minimal;
- cell cycle progression: this module comprises CCND1 amplification and mutations in MAP2K4, RB1, and GRID1, found in 36.3% of samples; it includes mutations in MAP2K4, with limited interaction evidence.

To summarize, despite some differences in specific results, Multi-Dendrix and Iter-Dendrix produced largely consistent findings; Multi-Dendrix, however, was *significantly* faster. Both methods successfully identified key gene modules across the GBM and BRCA datasets, uncovering important cancer-related pathways and patterns of mutual exclusivity.

The following section will discuss a method that utilizes the same scoring function $W(M)$, but is based on a genetic algorithm.

3.3 MDPFinder

3.3.1 The genetic algorithm

As outlined in the previous chapter, the weight function $W(M)$, has been widely adopted across multiple studies, due to its intuitive nature and its suitability for formalizing mutual exclusivity and coverage. One study that employed this metric — though not previously discussed, because its approach to mutual exclusivity mirrors that of Vandin et al. [33] — is the work by Zhao et al. [39], which introduced an algorithm called MDPFinder (*knowledge-based* [10]).

Their method utilizes a *Genetic Algorithm* (GA), a flexible and adaptable approach capable of optimizing a wide range of scoring functions. It models genetic variation within a population, evolving through a process of random selection, thereby avoiding the need to enumerate all possible solutions.

Before detailing the genetic procedure, it is necessary to first define the hypothesis space and the genetic operators.

Definition 3.5 (Hypothesis space)**.** A **member** of the population is defined by a binary string of length n, i.e. the number of genes. Given a gene set M, the value of

the i-th position of an individual represents the membership of the i-th gene in M. Therefore, if the target gene set has size k, the **hypothesis space** is constituted by all the possible binary strings with length n that have k 1s, namely

$$\mathcal{H} := \left\{ (x_1, \ldots, x_n) \mid x_i \in \{0,1\}, i \in [1,n], \sum_{j=1}^{n} x_j = k \right\}$$

An individual of the population is denoted as $h_i \in \mathcal{H}$, and its corresponding gene set as M_i.

Definition 3.6 (Fitness function)**.** The **fitness** f_i of each individual $h_i \in \mathcal{H}$ is defined as the rank r_i of the score $W(M_i)$, in *ascending order*:

$$\forall h_i \in \mathcal{H} \quad f_i := r_i$$

Note that it is used an *ascending order* because $W(M_i)$ has to be *maximized*, and *higher-ranking individuals* are favored in selection for the next generation, such that fitness increases with rank.

Definition 3.7 (Selection probability)**.** Given the rank r_i of an individual h_i, the **selection probability** is defined as follows:

$$p_i = \frac{2r_i}{P(P+1)}$$

where P is the population size.

Therefore, individual with the highest fitness value (i.e., *highest ranking*) is most likely to be transferred into the next generation.

This selection operator is based on the roulette wheel selection, which states that the probability of choosing an individual is equal to

$$p_i = \frac{f_i}{\sum_{j=1}^{P} f_j} = \frac{r_i}{\frac{P(P+1)}{2}} = \frac{2r_i}{P(P+1)}$$

which is precisely the equation in Definition 3.7.

Definition 3.8 (Crossover operator)**.** The **crossover operator** specifies the breeding process as follows: the offspring inherits the variables shared by both parents, while the non-shared ones are selected from the symmetric difference of the parents' genetic makeup.

Definition 3.9 (Mutation operator)**.** The **mutation operator** randomly sets the value of one variable from 1 to 0, and changes another variable value from 0 to 1, ensuring the feasibility of every offspring.

To prevent premature convergence and enhance the accuracy of the algorithm, Zhao et al. [39] employ a local search strategy to improve search performance, which is described below.

Definition 3.10 (Local search)**.** The **local search** procedure is defined as follows: the values of two variables are randomly altered, as the mutation operator; if this adjustment improves the current solution, it is accepted. The search is terminated once all variables have been tested with this routine.

Definition 3.11 (GA procedure)**.** The following are the details of the **GA procedure**:

1. *population generation*: a random population of size P and mutation rate p_m is generated, where $P = n$ (i.e. the number of available genes);
2. *breeding*: for each iteration, P couples are selected from the current population, based on p_i, and each couple generates an offspring;
3. *mutation*: each offspring may optionally receive a mutation with probability p_m;
4. *selection*: all parents and offspring are ranked based on their scoring values, and the top P individuals are selected to form the next generation (this is commonly referred to as truncation selection);
5. *local search*: verify if the iteration is stuck in a local solution (e.g. if the maximum scoring value does not improve over two consecutive iterations); if this is the case, perform a local search;
6. *termination*: proceed as such until the termination criterion is met (e.g. if the current maximum scoring value does not improve over 10 consecutive iterations); if this occurs, then end the procedure.

Note that the algorithm is independent of how $W(M)$ is defined, offering significant versatility in its application. The following section will describe an integration procedure, employed by Zhao et al. [39], to improve the results.

3.3.2 The integration procedure

In practical applications, multiple optimal solutions may exist. Additionally, due to data noise and other factors, the solutions considered optimal — i.e. the ones with the highest $W(M)$ — may not necessarily be the most relevant in a biological context. To identify the most biologically meaningful solutions, Zhao et al. [39] integrate other types of data, to refine the results. Specifically, the GA procedure is extended by incorporating gene expression data to enhance its performance.

The integrative model is developed based on the observation that genes within the same pathway typically collaborate to perform a specific function. Consequently, the expression profiles of gene pairs within the same pathway often exhibit higher correlations than those in different pathways. This characteristic can be leveraged to distinguish between gene sets that have the same score: the model focuses on detecting gene sets whose scores $W(M)$ are close to the optimal solution, but whose member genes display stronger correlations with each other.

Definition 3.12 (Integrative measure)**.** Given an $m \times n$ mutation matrix A, an expression matrix E with the same dimensions, and an A's submatrix M of size $m \times k$, the integrative model is defined by the following **measure**:

$$F_{ME} := W(M) + \lambda \cdot R(E_M)$$

where E_M is E's expression submatrix that corresponds to M, and $R(E_M)$ is described by the following equation:

$$R(E_M) := \sum_{j_1 \neq j_2} \frac{|\text{pcc}(x_{j_1}, x_{j_2})|}{\frac{k(k-1)}{2}}$$

where $\text{pcc}(\cdot)$ is the Pearson correlation coefficient, and x_j is the expression profile of gene j.

In other words, $R(E_M)$ represents the sum of the correlation coefficients for each pair of genes (note that $j_1 \neq j_2$ means that the pair j_1, j_2 is counted only once), normalized by the total number of possible gene pairs in M.

Moreover, note that

$$-1 \leq \text{pcc}(x_{j_1}, x_{j_2}) \leq 1 \implies 0 \leq |\text{pcc}(x_{j_1}, x_{j_2})| \leq 1 \implies 0 \leq R(E_M) \leq 1$$

therefore, when $\lambda = 1$ the value of F_{ME} can be used to discriminate the gene sets with the same $W(M)$. Moreover, for values of $\lambda \geq 1$, the gene set with strongest correlation and approximate exclusivity can be identified. The next section will describe the results obtained by the GA algorithm using the integrative model, with λ values set to 1 and 10.

3.3.3 Results

Zhao et al. [39] compared the results of three algorithms:

- the findings from their GA approach;
- the outcomes from an ILP, identical to the one developed by Leiserson et al. [22], which they refer to as $\text{Dendrix}_{ILP}(k)$, discussed in Section 3.2.1;
- the MCMC algorithm, developed by Vandin et al. [33], discussed in Section 3.1.2.

They grouped genes mutated in the same patients into *metagenes*, similar to the approach of Leiserson et al. [22] (details in Section 3.2.4). To evaluate the significance of the identified gene patterns, a permutation test was employed. Results were reported for gene sets with sizes ranging from $2 \leq k \leq 10$, including an analysis of second optimal patterns by removing the initially identified gene set, as was done by Vandin et al. [33] (details in Section 3.1.3).

The three algorithms were initially tested on the dataset used by Vandin et al. [33] to compare their performance: the ILP produced exact results in *less than 1 second*,

while the GA and MCMC algorithms took over 60 and 5 seconds, respectively. Despite the differences in runtime, all three methods identified the same gene sets, such as the set (EGFR, KRAS, STK11) in the lung adenocarcinoma dataset when $k = 3$, as also reported by Vandin et al. [33]. Additionally, the ILP and GA methods were applied to three datasets not used by Vandin et al. [33]. As before, the ILP consistently obtained exact results in *under 1 second*, demonstrating its efficiency across multiple datasets. Details of all findings are presented below.

The first dataset used by Zhao et al. [39] was created by Stransky et al. [29], who performed whole-exome sequencing on 74 tumor-normal pairs, revealing previously unimplicated genes in head and neck squamous cell carcinoma (HNSCC). The mutation dataset includes 4920 genes, with an average of 130 coding mutations per sample. The mutation matrix is sparse, with only TP53 and TTN mutated in more than 20 samples, affecting 46 and 23 samples, respectively. To explore other pathways, Zhao et al. [39] removed these two genes from the dataset and applied the three algorithms to the remaining genes. For gene sets with $k = 6$, a unique optimal set (ANO4, CDKN2A, NFE2L2, NOTCH1, SYNE1, TP63) was identified, altered in 60.8% of the samples, with a p-value of 0.01. For $k < 6$, the optimal solutions were subsets of these six genes, while for $k > 6$ multiple optimal solutions were found. Zhao et al. [39] suggest that mutations in CDKN2A, NOTCH1, TP63, and SYNE1 are linked to terminal differentiation in squamous epithelia.

The second dataset used by Zhao et al. [39] was sourced from the TCGA [8] and includes data on DNA CNAs, gene expression profiles from 206 glioblastoma samples, with sequence data available for 91 of these. After preprocessing, mutation and expression matrices were constructed using 90 samples and 1126 genes. For mutation patterns, when $k = 2$ two key gene sets were identified: (CDKN2A, TP53), which are involved in the p53 signaling pathway, and (CDKN2B, CDK4-TSPAN31). Analysis of the expression data showed that CDK4 has a stronger correlation with CDKN2B than TSPAN31, highlighting CDK4's greater importance. When $k = 3$, the optimal gene set included CDK4, CDKN2B, and RB1, confirming findings from other studies. After removing these five genes, an additional optimal set was identified at $k = 5$: (PTEN, EGFR, PIK3R1, PIK3CA, GRIA2). Most of these genes are part of the RTK/RAS/PI(3)K signaling pathway, which is critical in glioblastoma.

The final dataset reported by Zhao et al. [39] is the ovarian carcinoma dataset from a recent TCGA study. This dataset includes mRNA expression, microRNA expression, promoter methylation, DNA CNAs from 489 samples, and exon DNA sequences from 316 tumors. After preprocessing, mutation and expression matrices were created, covering 313 samples and 6108 genes. The mutation distribution was uneven as in the first database, with TP53 mutations prevalent in most samples, while TTN mutations were considered artifacts and were removed from the analysis. For $k = 2$, the gene pair (CCNE1, MYC), involved in cell cycle progression, was identified in 135 samples. At $k = 3$, NINJ2 was added to this optimal gene set. For $k = 4$, the ILP model identified a set of four genes: (KRAS, PPP2R2A, PRPF6, RYR2). In contrast, the integrative model identified an alternative set: (KRAS, MAPK8IP2, NF1, STMN3), which showed stronger correlations among the genes. KRAS, NF1, and MAPK8IP2 are part of the MAPK signaling pathway, while STMN3 is associated with cancer

progression. Zhao et al. [39] highlight that these findings demonstrate the advantage of the integrative model in identifying gene sets with functional relationships, even when mutation-based scoring results in suboptimal solutions.

In summary, the comparison performed by Zhao et al. [39] demonstrated the ILP's efficiency in identifying significant gene sets across various cancer datasets, and their findings showcased their integrative model's strength in revealing functional relationships among genes.

The following section will explore the specifics of a very distinct algorithm, that employs statistical approach and a different scoring function.

3.4 Mutex

3.4.1 A different greedy method

The approach developed by Vandin et al. [33] employed a greedy algorithm to search for the most mutually exclusive driver pathway (discussed in Section 3.1.1); this technique is highly versatile, as it can be adapted to a broad range of scoring functions. Similarly, Babur et al. [2] also utilized a greedy algorithm in their study, developing an algorithm called Mutex (*knowledge-based* [10]), which employs a fundamentally different scoring function than $W(M)$, incorporating statistical and probabilistic elements, as mentioned in the previous chapter.

The following algorithms will provide a *partial* description of their approach; first, the greedy procedure is described below.

Algorithm 3.3 *Greedy Mutex*: given a gene g, an integer $k_{\max}$, a directed gene graph G, a mutation matrix A, and a boolean variable `final`, the algorithm returns the gene set M, of size $|M| < k_{\max}$, that maximizes the Mutex's scoring function — which will be described later — using g as the starting gene.

```
1: function GREEDYMUTEX(g, k_max, G, A, final)
2:     M := {g}
3:     do
4:         M_p := M
5:         M := expandGroup(M, G, A, final)        ▷ refer to Algorithm 3.4
6:     while M ≠ M_p ∧ |M| < k_max                 ▷ if new genes were added to M
7:     return M
8: end function
```

The algorithm proceeds as follows:

- first, M is initialized with only g;
- in each iteration, the algorithm saves M in a temporary variable M_p, and computes `expandGroup` (which will be described later);

- if $M = M_p$ (i.e., M was not expanded), or if $|M|$ exceeds the maximum size, the algorithm terminates and returns M.

Note that the roles of G, A and `final` will be clarified in the following pseudocodes.

The next algorithm illustrates the `expandGroup` procedure in detail.

Algorithm 3.4 *Group expansion procedure*: given a gene set M, a directed gene graph G, a mutation matrix A, and a boolean variable `final`, the algorithm expands M, if possible.

```
1:  function expandGroup(M, G, A, final)
2:      b := NULL                                   ▷ b is the current best candidate
3:      b_s := 1                                    ▷ b_v is b's associated score
4:      N := {}
5:      if final then
6:          P := pValues(M, A)                      ▷ refer to Algorithm 2.1
7:          for g ∈ M do
8:              N.add_entry(g, N_g)                 ▷ N_g is g's null distribution, based on P[g]
9:          end for
10:     end if
11:     m_s := score(M, A, final, N)                ▷ current M's score
12:     for c ∈ δ(M) do                             ▷ the set of candidates (refer to Definition 3.13)
13:         if final then
14:             P' := pValues(M ∪ {c}, A)
15:             for g ∈ (M ∪ {c}) do
16:                 N[g] = N'_g                     ▷ N'_g is based both on P'[g] and N_g
17:             end for
18:         end if
19:         c_s := score(M ∪ {c}, A, final, N)      ▷ c's associated score
20:         if c_s < b_s ∧ c_s < m_s then           ▷ c discarded if c_s does not improve M
21:             b := c
22:             b_s := c_s
23:         end if
24:     end for
25:     if b ≠ NULL then
26:         return M ∪ {b}
27:     end if
28:     return M
29: end function
```

This procedure is extensive, but it can be divided into smaller sections. However, before delving into the details, it is necessary to introduce and discuss some key pieces of information.

First, the algorithm expects a directed gene graph G as input. This graph is constructed by Babur et al. [2] using data from the Pathway Commons [6], SPIKE [26], and SignaLink [13] databases. In this graph, the vertices represent genes, and

the directed edges indicate signaling relationships between genes or proteins — the authors describe the generation of this graph in another work [1].

Given this directed gene graph G, consider the following definition.

Definition 3.13 (Proximity)**.** Given a directed gene graph G, and a gene set $M \subseteq V(G)$, the **proximity** of M is defined as the set of vertices v such that, when added to M, there exists a gene $s \in V(G)$ for which all genes in the augmented set still share a *common downstream target* that can be reached without traversing any genes outside of the augmented gene set. Using symbols

$$\delta(M) := \{v \in V(G) \mid \exists s \in V(G) : \forall u \in M \cup \{v\} \quad \exists u \to s \text{ only traversing } M \cup \{s, u\}\}$$

where $u \to s$ is a path that starts in u and ends in s.

This definition is dense, but the following example [2] can help clarify its meaning.

Example 3.1 (Proximity)**.** The figure below shows an example of the expansion of an initial gene set $\{A\}$, using the Algorithm 3.3. Vertices with bold borders represent the elements of the current set M, while the grey-colored ones are the current candidates in the *proximity* $\delta(M)$ of M.

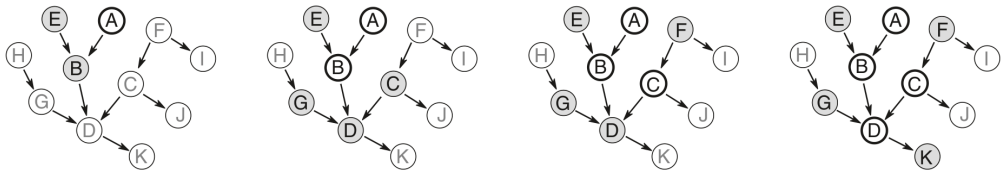

Figure 3.1. Greedy expansion of $\{A\}$ (left to right).

The *proximity* $\delta(M)$ is utilized in line 2 of the algorithm to define the set of potential candidates that could expand M. Indeed, the goal of Babur et al. [2] is to identify mutually exclusive altered groups, where members share a **common downstream signaling target**. They state that this strategy narrows the search space to areas with a higher concentration of true positives. While it may slightly reduce recall, it also lessens the loss of statistical power associated with multiple hypothesis testing. Additionally, this approach provides an initial explanation for the observed mutual exclusivity, i.e. *through a shared effect on a downstream gene.*

Another aspect to discuss is the *null distribution* of a given gene $g \in M$, which represents g's distribution under H_0 (defined in Section 2.2.3), denoted as $\mathcal{N}_g$ in the algorithm. Given a gene $g \in M$, Babur et al. [2] estimate g's null distribution through a procedure that approximately involves the following steps:

1. *initialization*: define $\mathcal{N}_g$ as an empty list;
2. *iteration*: for $i = 1, \ldots, i_{\max}$ derive $\mathcal{N}_g[i]$ as follows:
   (a) randomly permute g's alterations — i.e., permute g's column in A randomly;

(b) let $M_g := \texttt{greedyMutex}(g, k_{\max}, G, A, \texttt{false})$;

(c) let $\mathcal{P}_g := \texttt{pValues}(M_g, A)$;

(d) let $\mathcal{N}_g\texttt{[}i\texttt{]} := \mathcal{P}_g\texttt{[}g\texttt{]}$.

This is a *sketch* of the complete procedure, and the omitted details extend well beyond the scope of this discussion. The main conclusion from this algorithm is that g's *null distribution* is derived from its p-value (namely $\mathcal{P}_g\texttt{[}g\texttt{]}$), computed when its alterations are randomly permuted. This randomization aims to modify $\gamma(g, M_g)$, simulating a scenario in which H_0 holds true, i.e. alterations in $\Gamma(g)$ are independent from mutations in $\Gamma(M - \{g\})$. Note that $\mathcal{P}_g\texttt{[}g\texttt{]}$ is computed through the `greedyMutex` function, with `final` set to `false` to prevent infinite recursion and avoid recomputing the *null distributions*. In the `expandGroup` procedure, it will be assumed that for any $g \in M$, $\mathcal{N}_g$ can be computed as described.

With these definitions established, the `expandGroup` algorithm can be explained in detail. Specifically, it can be broken down into the following sections:

1. if $\texttt{final} = \texttt{true}$, fill $\mathcal{N}$ such that

$$\forall g \in M \quad \mathcal{N}\texttt{[}g\texttt{]} = \mathcal{N}_g$$

   where $\mathcal{N}_g$ is g's *null distribution*, based on g's p-value computed on M;

2. *initialization*: let m_s be M's *score* (refer to Algorithm 3.5);

3. *iteration*: for each candidate c in $\delta(M)$, compute as follows:

   (a) if $\texttt{final} = \texttt{true}$, update $\mathcal{N}$ such that

$$\forall g \in (M \cup \{c\}) \quad \mathcal{N}\texttt{[}g\texttt{]} = \mathcal{N}'_g$$

   where $\mathcal{N}'_g$ is g's *null distribution*, based both on g's p-value computed on $M \cup \{c\}$, and on $\mathcal{N}_g$;

   (b) let c_s be $(M \cup \{c\})$'s score;

   (c) if c_s improves *both* the current best score, and m_s, update the current best score with c_s.

4. if the best candidate b could be determined, return $M \cup \{b\}$; otherwise, return M.

Note that this procedure requires c_s to improve *both* the current best score and M's base score, meaning that suboptimal solutions are not explored by the algorithm.

Finally, the last procedure, which computes the mutual exclusivity score of a given gene set, can be introduced. Although this score is a crucial part of the metric developed by Babur et al. [2] to assess mutual exclusivity within a gene set, this algorithm could not be introduced in the previous chapter because it relies on the *null distribution* dictionary $\mathcal{N}$, defined in Algorithm 3.4, which in turn depends on the *null distribution* estimation, based on Algorithm 3.3 (with $\texttt{final} = \texttt{false}$).

Algorithm 3.5 *Scoring procedure*: given a gene set M, a mutation matrix A, a boolean variable `final`, and a *null distribution* dictionary $\mathcal{N}$, the algorithm computes M's mutual exclusivity score.

1: **function** SCORE(M, A, `final`, $\mathcal{N}$)
2: $\quad\mathcal{P} := \texttt{pValues}(M, A)$
3: $\quad$**if** `final` **then** ▷ initial p-values correction
4: $\quad\quad$**for** $g \in M$ **do**
5: $\quad\quad\quad\mathcal{N}_g := \mathcal{N}[g]$
6: $\quad\quad\quad c := |\{i \mid \mathcal{N}_g[i] \leq \mathcal{P}[g]\}|$
7: $\quad\quad\quad\mathcal{P}[g] := \max\left(\mathcal{P}[g], \dfrac{c}{\mathcal{N}_g.\texttt{len}()}\right)$
8: $\quad\quad$**end for**
9: $\quad$**end if**
10: $\quad$**return** $\max_{g \in M} \mathcal{P}[g]$ ▷ the *least* significant is the *largest*
11: **end function**

First, the algorithm evaluates $\mathcal{P}$, the p-value dictionary; then, if `final` = `true`, $\mathcal{P}$ is corrected for *multiple hypothesis testing*. Finally, the largest value in $\mathcal{P}$ is returned. Note that both in line 7 and line 10, the largest value is chosen: this is because the *larger* the p-value, the *less significant* it is. Therefore, in line 7, the focus is on being as cautious as possible, while in line 10, the aim is to ensure that each member of M contributes to the pattern. Lastly, the boolean variable `final` ensures that, during the evaluation of the *null distributions*, the scores being used remain uncorrected.

The complete algorithm works by calling `greedyMutex` for each possible $g \in \mathcal{G}$ with `final` set to `true`, and then comparing the resulting sets. Finally, these sets may optionally undergo an FDR control procedure, which lies outside the scope of this work. Note that many details from the original code have been removed for brevity, in each described algorithm.

3.4.2 Results

Babur et al. [2] applied their algorithm to identify mutual exclusion patterns in mutation and copy number profiles from multiple TCGA studies. The gene network they employed was cropped to the *proximity* of significantly mutated genes (derived from MutSig [21]) and significantly altered genes (provided by GISTIC [24]). Lastly, to reduce noise, genes with low alteration rates were filtered out in each study, and groups of up to 5 genes were examined.

They identified a total of 199 genes in their results, with 31 appearing in at least two studies. Notably, TP53 was the most recurrent gene, followed by well-known tumor suppressors and oncogenes such as PTEN, KRAS, and MYC. Among less recognized genes, OBSCN and ARID1A are highlighted for their potential roles in cancer — the latter has previously been shown to act as a tumor suppressor in gastrointestinal cancers (GI). The most frequent common targets among the result groups include PIK3R1, HRAS, BRAF, MYC, RAC1, and RHOC, with mutually exclusive alterations observed upstream of RHOC in five datasets. Although RHOC

alterations are infrequent in TCGA samples, its overexpression is associated with cancer cell metastasis, suggesting that its activation may represent a significant downstream effect of driver alterations.

To assess the novelty of the findings, Babur et al. [2] examined co-citations of the recurrent genes with the term "cancer", using CoCiter. The last 10 genes on the list had fewer than 25 co-citations, indicating they are not well-established cancer drivers. However, further investigation revealed that 5 of these genes — TRRAP, OBSCN, RIT1, AGAP2, and RORC — contain so called *mutation hotspots*. Mutation hotspots are DNA segments particularly prone to genetic alterations [25], and are considered indicators of driver mutations, as changes in different regions of a driver gene can confer varying levels of selective advantage to a cancer cell — passenger mutations are typically randomly distributed [2]. Among the remaining 5 genes, the gene pair (CERS2, NCSTN) showed copy number alterations in the results.

Additionally, Babur et al. [2] compared the performance of their method with several previously published studies, including Dendrix [33], MDPFinder [39], and Multi-Dendrix [22]. They derived a large dataset from breast cancer data in cBioPortal, which included 830 genes with an alteration rate of at least 3% across 958 samples. This dataset was constructed through several steps aimed at randomizing gene alterations while preserving alteration ratios. Mutex outperformed the other methods, significantly improving the receiver operating characteristic (ROC) curves; notably, a modified version of Mutex that *did not* use signaling networks showed decreased performance, highlighting the advantages of incorporating pathway information. In contrast, Dendrix, MDPFinder, and Multi-Dendrix performed poorly due to their reliance on the same weight function $W(M)$, which favors noise over signal. Moreover, other methods' generative models also struggled because they assumed equal alteration chances among group members. Mutex demonstrated improved scoring criteria and efficiency, exhibiting strong scalability in terms of memory and runtime, comparable to MDPFinder, and significantly more efficient than Dendrix and similar algorithms.

In conclusion, the greedy algorithm developed by Babur et al. [2] identified known mutually exclusive driver pathways, and highlighted potential roles for lesser-known genes, namely OBSCN, ARID1A and RHOC.

The following section will outline the different versions of the clustering algorithm, anticipated in the previous chapter.

3.5 C^3

3.5.1 Multiple versions

In the final section of the previous chapter (namely, Section 2.2.5), it was mentioned that Hou et al. [16] developed multiple versions of their clustering algorithm; these variants will be explored in detail in the following paragraphs. In particular, they defined three methods for assigning weights to the edges of their gene graph G, to perform their vertex clustering algorithm, called C^3 (*knowledge-based* [10]):

1. **ME-CO**, where w^- depends on *mutual exclusivity* and w^+ depends on *coverage*;
2. **NI-ME-CO**, where w^- depends on *mutual exclusivity* and w^+ depends on *coverage* and *network information*;
3. **EX-ME-CO**, where w^- depends on *mutual exclusivity* and w^+ depends on *coverage*, and *expression data.*

Note that w^- depends solely on the mutual exclusivity component in each version of the algorithm, whereas the value of w^+ depends on the chosen algorithm variant. The following section will introduce the standard version of C^3.

3.5.2 The standard version

The initial version of their clustering algorithm is the standard one, which considers only **mutual exclusivity** and **coverage**, and it is described below.

Definition 3.14 (ME-CO)**.** In the **ME-CO** version of the algorithm, the following definitions apply:

$$\forall u, v \in V(G) \quad w^-_{uv} := w^-_{uv}(\mathrm{e}) \tag{3.12}$$

$$\forall u, v \in V(G) \quad w^+_{uv} := w^+_{uv}(\mathrm{c}) \tag{3.13}$$

The definitions for $w^-_{uv}(\mathrm{e})$ and $w^+_{uv}(\mathrm{c})$ are provided in Definition 2.11 and Definition 2.12 respectively.

Note that in each variation discussed, optional rescaling is applied to ensure that the weight formulas satisfy additional constraints required later in the algorithm, though the specifics are beyond the scope of this analysis.

While this version of the algorithm does not include any external data, the variant outlined in the next section incorporates additional supplementary information into the weights.

3.5.3 Integrating network information

Pan-cancer studies, as reported in multiple papers, have demonstrated a significant relationship between network topology and the distribution patterns of cancer drivers. Specifically, the impact of deleterious mutations on the phenotype can be mitigated by certain configurations of the corresponding protein complexes, while other arrangements can amplify their effect. For example, most variants found in healthy individuals tend to be located at the periphery of the interactome, where they do not affect network connectivity. In contrast, cancer-driver somatic mutations are more likely to occur in central, internal regions of the interactome and within highly integrated components [16]. This suggests that network topology significantly

influences the impact of cancer driver mutations, and to assess the implications for cancer development, Hou et al. [16] analyzed network distances between driver variants to identify patterns.

To precisely quantify the network distances between driver variants, Hou et al. [16] computed the pairwise network distances between genes within a large pathway, comprising 8726 genes, by using an implementation of the standard Dijkstra algorithm. To reduce the computational cost of running Dijkstra's algorithm $O\left(8726^2\right)$ times, 1000 pairs were randomly selected for this test. Using the most comprehensive known driver list from the Cancer Gene Census (CGC) [14], the same distances were calculated for driver genes, this time for all gene pairs. The resulting distribution of shortest paths is shown in Figure 3.2 [16], revealing that the average shortest distance between drivers is *significantly smaller* than that between two randomly selected genes.

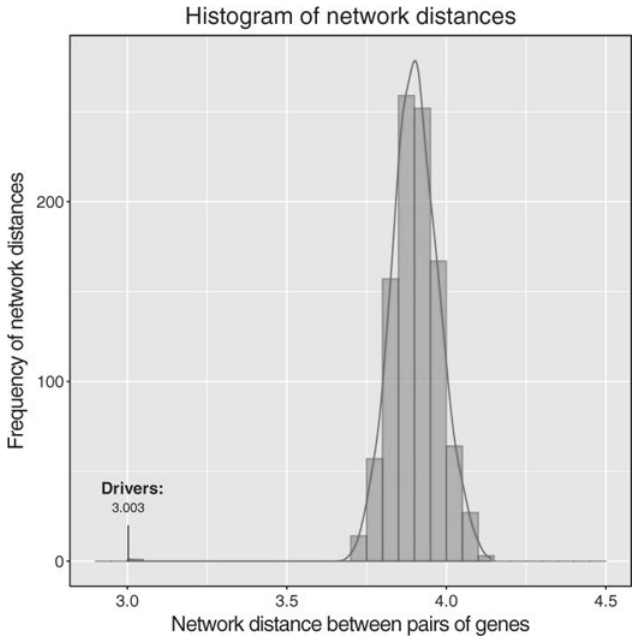

Figure 3.2. Distribution of distances between genes in the network.

These findings indicate that network distance and connectivity information should be considered when identifying potential driver mutations. This can be achieved by adjusting the positive weight of edges connecting two genes: specifically, if both endpoint genes are drivers, they should be sufficiently central within a given pathway, close to other known drivers, or to each other.

Hence, from the KEGG [18] database, Hou et al. [16] built an undirected graph G', where each vertex represents a gene and the edges describe interactions between them — note that $|V(G)| = |V(G')| = n$. For each vertex $u \in V(G')$, let $\mathscr{N}(u)$ denote the set of u's neighbors, and let $\mathscr{N}'(u) := \mathscr{N}(u) \cup \{u\}$. Also, let

$$f(u,v) := \frac{|\mathscr{N}'(u) \cap \mathscr{N}'(v)|}{|\mathscr{N}'(u) \cup \mathscr{N}'(v)|} \tag{3.14}$$

which is referred to as the Jaccard similarity coefficient; a large value of $f(u,v)$ indicates that u and v are well connected in G' and are likely involved in the same pathway, suggesting that they should be clustered together. Furthermore, let

$$\mathscr{F} := \{f(u,v) \mid u,v \in V\,(G')\} \tag{3.15}$$

and let $T'(J')$ be the J'-th percentile of the values in $\mathscr{F}$.

The network information component of the positive weights is described below.

Definition 3.15 (Network information component)**.** The **network information component** is defined as follows:

$$w^+_{uv}(\mathrm{n}) := \begin{cases} 1 & f(u,v) > T'(J') \\ \dfrac{f(u,v)}{T'(J')} & f(u,v) \leq T'(J') \end{cases}$$

Finally, the version of C^3 that incorporates the network information is defined as follows.

Definition 3.16 (NI-ME-CO)**.** The **NI-ME-CO** version of the algorithm is defined by the following equations:

$$\forall u,v \in V(G) \quad w^-_{uv} := w^-_{uv}(\mathrm{e}) \tag{3.16}$$

$$\forall u,v \in V(G) \quad w^+_{uv} := w_1 w^+_{uv}(\mathrm{c}) + w_2 w^+_{uv}(\mathrm{n}) \tag{3.17}$$

where $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

The next section will describe a third variant, that incorporates gene expression data instead of network information.

3.5.4 Integrating expression data

Another valuable type of data that could be integrated into the positive weights for clustering is **gene expression data**. This is based on the assumption that co-expressed genes are likely to be involved in the same function or cancer pathway. Therefore, genes with strong positive or negative co-expression should be clustered together. The following paragrpahs will describe how Hou et al. [16] include expression data into w^+.

Given a vertex $u \in V(G)$, let $\mathbf{z}(\mathrm{u})$ be the vector of the time-evolving expression values of u. Thus, let

$$g(u,v) := \frac{|\langle \mathbf{z}(\mathrm{u}), \mathbf{z}(\mathrm{v}) \rangle|}{||\mathbf{z}(\mathrm{u})||\,||\mathbf{z}(\mathrm{v})||} \tag{3.18}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product of the vectors $\mathbf{a}$ and $\mathbf{b}$, while $||\mathbf{a}||$ stands for its L^2 norm. This equation is known as the cosine similarity, since the ratio that defines $g(u, v)$ is equal to the cosine of the angle between $\mathbf{z}$(u) and $\mathbf{z}$(v) — the only difference being the absolute value in the numerator, to capture both positive and negative correlations. A large value of $g(u, v)$ suggests that the expression vectors of u and v are highly correlated, hence they should be clustered together. Note that

$$\forall u, v \in V(G) \quad 0 \leq g(u, v) \leq 1$$

Moreover, let

$$\mathscr{G} := \{g(u, v) \mid u, v \in V(G)\} \tag{3.19}$$

and let $T''(J'')$ be the J''-th percentile of the values in $\mathscr{G}$.

Hence, the gene expression component of the positive weights can be defined as follows.

Definition 3.17 (Expression data component)**.** The **expression data component** is defined as follows:

$$w^+_{uv}(\mathrm{x}) := \begin{cases} 1 & g(u, v) > T''(J'') \\ \dfrac{g(u, v)}{T''(J'')} & g(u, v) \leq T''(J'') \end{cases}$$

Lastly, the third variant of C^3 is described below.

Definition 3.18 (EX-ME-CO)**.** The **EX-ME-CO** version of the algorithm is defined by the following equations:

$$\forall u, v \in V(G) \quad w^-_{uv} := w^-_{uv}(\mathrm{e}) \tag{3.20}$$

$$\forall w^+_{uv} := w_1 w^+_{uv}(\mathrm{c}) + w_2 w^+_{uv}(\mathrm{x}) \tag{3.21}$$

where $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

3.5.5 Other versions

Hou et al. [16] also mention that other combinations can be used, with appropriate adjustments to the weights, such as the following version, which will be referred to as NI-EX-ME-CO in this work.

Definition 3.19 (NI-EX-ME-CO)**.** The **NI-EX-ME-CO** version of the algorithm is defined by the following equations:

$$\forall u, v \in V(G) \quad w^-_{uv} := w^-_{uv}(\mathrm{e}) \tag{3.22}$$

$$\forall w^+_{uv} := w_1 w^+_{uv}(\mathrm{c}) + w_2 w^+_{uv}(\mathrm{n}) + w_3 w^+_{uv}(\mathrm{x}) \tag{3.23}$$

where $w_1, w_2, w_3 \geq 0$ and $w_1 + w_2 + w_3 = 1$.

The previous sections outlined the definition of the weights for the edges in the gene graph G; instead, the following ones will explain how C^3 operates.

3.5.6 The clustering ILP

Hou et al. [16] opted to use an ILP approach to formulate the clustering algorithm, utilizing the weights defined in the previous sections.

Note that the classical formulation of correlation clustering does not impose any restrictions on cluster sizes. However, most driver identification methods inherently include cluster size limits, as they directly affect the computational complexity of the algorithms — many even fail to operate beyond a certain size. Another reason for imposing a cluster size limit is the expectation that driver genes of specific cancer types will be grouped together, and recent findings indicate that only a small number of drivers are typically present in any given cancer type. Thus, if clusters are too large, they may include drivers from multiple cancer types, hiding this separation of the drivers. Furthermore, introducing cluster size constraints helps to avoid the formation of non-informative *giant clusters* or *singleton clusters* [16].

Therefore, Hou et al. [16] introduce a cluster size constraint by assuming that all clusters are of size k at most; clearly, setting k equal to the total number of vertices effectively removes this constraint, allowing flexibility in cluster size selection.

The ILP of C^3 is defined as follows.

Definition 3.20 (C^3's ILP)**.** The C^3 **algorithm** can be defined by the following ILP:

$$\text{minimize} \sum_{e \in E(G)} (w^+_e x_e + w^-_e (1 - x_e)), \tag{3.24}$$

$$\text{subject to } x_{uv} \leq x_{uz} + x_{zv}, \quad u, v, z \in V(G) \text{ distinct}, \tag{3.25}$$

$$\sum_{\substack{v \in V(G) \\ u \neq v}} (1 - x_{uv}) \leq k, \quad u \in V(G), \tag{3.26}$$

$$x_e \in \{0, 1\}, \quad e \in E(G). \tag{3.27}$$

In this formulation, the variables x_e allow to describe any clustering of the vertices of G, since $x_e \in \{0, 1\}$ for each $e \in E(G)$.

Note that Equation 3.24 aligns with the definition provided in Section 2.2.5, as $x_{uv} = 1$ implies that u and v should belong to different clusters, while $x_{uv} = 0$ implies that the two vertices should be placed into the same cluster.

Furthermore, Equation 3.26 states that for a fixed vertex $u \in V(G)$, the number of variables x_{uv} equal to 0, for any $v \in \mathcal{N}(u)$, must not exceed k — which is the clustering size constraint previously discussed.

Lastly, Equation 3.25 is the triangle inequality, which ensures that if u and z are placed in the same cluster, and z and v are also placed in the same cluster, then u and v will be clustered together. This means that *belonging to the same cluster is a transitive property*, since

$$\begin{cases} x_{uz} = 0 \\ x_{zv} = 0 \\ x_{uv} \leq x_{uz} + x_{zv} \end{cases} \implies x_{uv} = 0$$

The next section will illustrate a relaxation of this ILP.

3.5.7 The rounding procedure

Since solving binary ILPs is NP-Complete [19], Hou et al. [16] relax the problem by changing Equation 3.27 to an interval constraint

$$0 \leq x_e \leq 1$$

leading to an LP program, the solution of which may be fractional. Hence, to obtain a valid clustering, the fractional solutions have to be rounded. Therefore, instead of solving the LP, they remove Equation 3.26 from the linear program, and employ the following rounding procedure to round the fractional values.

Algorithm 3.6 *Rounding procedure*: given a solution $\{x_e\}_{e \in E(G)}$ of the relaxed version of the ILP provided Definition 3.20, a rational value α, and the maximum cluster size k, the algorithm rounds the solution to integer values.

function ROUNDINGPROCEDURE(G, $\{x_e\}_{e \in E(G)}$, α, k)
  $\mathcal{C} := \varnothing$ ▷ the output set of clusters
  $S := V(G)$
  while $S \neq \varnothing$ **do**
    Choose an arbitrary $u \in S$ ▷ this is the *pivot vertex*
    $T := \{w \in S - \{u\} \mid x_{uw} \leq \alpha\}$ ▷ u's neighbors under α's threshold
    if $\sum_{w \in T} x_{uw} \geq \frac{\alpha}{2} |T|$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{\{u\}\}$ ▷ add a singleton cluster $\{u\}$
      $S = S - \{u\}$
    else if $|T| \leq k$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{\{u\} \cup T\}$ ▷ add the cluster $(\{u\} \cup T)$
      $S = S - (\{u\} \cup T)$
    else
      Partition T into $\{T'_0, T_1, \ldots, T_p\}$, such that:
        - $|T'_0| = k$
        - $|T_i| = k + 1$ for each $0 < i < p$
        - $|T_p| \leq k + 1$
      $T_0 := T'_0 \cup \{u\}$ ▷ T_0 has $k + 1$ elements
      for $i \in [0, p]$ **do**
        $\mathcal{C} = \mathcal{C} \cup \{T_i\}$ ▷ add each partition as a cluster
      end for
      $S = S - (\{u\} \cup T)$
    end if
  end while
  return $\mathcal{C}$
end function

The algorithm can be summarized as follows:

1. *initialization*: let $\mathcal{C} := \varnothing$ and $S := V(G)$;
2. *iteration*: while $S \neq \varnothing$, compute as follows:
   (a) choose a *pivot* vertex $u \in S$ arbitrarily;
   (b) let T be the set of u's neighbors $w \in \mathcal{N}(u)$ such that x_{uw} is at most α;
   (c) if $\frac{\alpha}{2} |T|$ is at most $\sum_{w \in T} x_{uv}$, add $\{u\}$ to $\mathcal{C}$ as a *singleton cluster*, and remove u from S;
   (d) otherwise, if $|T|$ is at most k, add $\{u\} \cup T$ to $\mathcal{C}$ as a cluster, and remove it from S;
   (e) otherwise, partition T in multiple subsets, such that each subset contains $k+1$ elements (technically, the last partition will contain only the remain-

ing elements), and the first subset contains u; then, add each partition of T as a separate cluster to $\mathcal{C}$, and remove every partition from S.

The rationale behind the condition in line 7 can be elucidated by examining the meaning of x_{uw}: specifically, if x_{uw} is close to 1, u and w are likely to be placed in different clusters, as previously described Section 3.5.6. Note that, if the sum $\sum_{w \in T} x_{uw}$ is greater than or equal to a value proportional to $|T|$, it indicates that there are numerous edges (u, w) for $w \in T$ with significantly high x_{uw}. Therefore, this suggests that u should likely be placed in a cluster distinct from all its *filtered* neighbors T.

In contrast, if this condition does not hold, it is probable that several variables indicate that u should be clustered with some of its *filtered* neighbors. Therefore, when this condition fails, and $|T| \leq k$, u and T are forced to form a cluster in line 11.

Lastly, if none of the preceding conditions are satisfied, it means that u should not form a *singleton cluster*, but the presence of numerous *filtered* neighbors precludes the formation of a $T \cup \{u\}$ cluster. Therefore, line 14 partitions T into smaller clusters of size $k + 1$.

As a final note, Hou et al. [16] conducted an analysis to determine the optimal value for α, which was found to be $\frac{2}{7}$, but the proof of this value is beyond the scope of this work.

Finally, the complete C^3 algorithm that Hou et al. [16] employed to obtain their results is described below.

Definition 3.21 (C^3)**.** The C^3 **algorithm** is defined as follows: first, the next ILP is solved

$$\text{minimize} \sum_{e \in E(G)} (w_e^+ x_e + w_e^- (1 - x_e)), \tag{3.28}$$

$$\text{subject to } x_{uv} \leq x_{uz} + x_{zv}, \quad u, v, z \in V(G) \text{ distinct}, \tag{3.29}$$

$$0 \leq x_e \leq 1, \quad e \in E(G). \tag{3.30}$$

and then the rounding procedure defined in Algorithm 3.6 is applied.

3.5.8 Results

To perform a comparative analysis with an existing study, Hou et al. [16] selected the CoMEt algorithm — developed by Leiserson et al. [23] — for comparison, and the results of their analysis are detailed below. In particular, they ran both algorithms utilizing mutation and CNV data sourced from the TCGA [8] database, specifically focusing on BRCA and GBM.

Both algorithms were tested on a high-memory server under identical conditions, except when CoMEt encountered memory errors at cluster size $k = 15$, allowing only C^3 to be tested for that case. This highlights the computational flexibility of their algorithm, particularly in terms of adjusting both the cluster size k and the number of clusters formed. In the benchmark Hou et al. [16] focused on the following evaluation criteria:

- to assess the *mutual exclusivity* within a cluster, they evaluated the median pairwise exclusivity for each gene pair (g_1, g_2) of the cluster, utilizing Fisher's exact test — specifically, they used the same contingency table described in Section 2.2.3, but in their case $M = \{g_1, g_2\}$;
- to quantify the *coverage* of a given cluster, they calculated the propotion of patients who exhibited at least one alteration in a gene of the cluster;
- as mentioned in Section 3.5.3, driver genes tend to cluster closer together within biological pathways compared to random gene selections; therefore, to identify potential cancer driver genes, they measured the shortest network distances between genes in the discovered clusters;
- lastly, to determine biological significance based on driver genes, they calculated the proportion of known drivers within the ten most mutually exclusive clusters, using a curated list of driver genes from the CGC [14].

When analyzing *mutual exclusivity*, C^3 demonstrated better performance, particularly for BRCA, where it produced more mutually exclusive clusters with lower p-values across most cluster sizes. Both methods found biologically significant clusters, but C^3's median exclusivity scores were generally stronger, except for cluster size $k = 10$. For GBM, the results were less pronounced due to the smaller dataset, but C^3 still outperformed CoMEt in overall mutual exclusivity.

Regarding *coverage*, CoMEt was superior in GBM, where it achieved a higher median coverage of 0.696 compared to C^3's 0.632. For BRCA, however, both algorithms performed similarly, with no significant difference in coverage. The choice of weights in C^3, which prioritized mutual exclusivity over coverage, likely contributed to its lower coverage performance.

Moreover, in the *pairwise distance* analysis of clustered genes, C^3 and CoMEt performed similarly for BRCA, but C^3 showed a statistically significant improvement in GBM, with smaller average distances between genes in clusters. This indicates that C^3 tends to favor more tightly related clusters in cancer pathways, particularly for GBM.

Lastly, in terms of *driver identification*, C^3 outperformed CoMEt across all cluster sizes. For BRCA, C^3 achieved a median driver proportion of 0.160 in the top ten clusters, while CoMEt reached 0.117. A similar trend was observed for GBM, where C^3 found a median driver proportion of 0.170 compared to CoMEt's 0.120.

In addition to this analysis, Hou et al. [16] tested C^3's ability to identify clusters of genes that may represent novel candidate cancer drivers, focusing on those with

biologically significant interactions and high mutual exclusivity and coverage. The analysis particularly emphasized large cluster sizes, which have not been extensively reported in the literature.

For BRCA, a notable cluster included several potential driver genes such as PTEN, HUWE1, CNTNAP2, GRID2, CACNA1B, CYSLTR2, and MYH1, with a mutual exclusivity p-value of 0.0084. This cluster is primarily influenced by mutations in PTEN and HUWE1, with PTEN being a well-known tumor suppressor gene, and the other genes in the cluster are also potential drivers, with roles in apoptosis, DNA repair, and cell signaling. The tightly interconnected nature of these genes suggests they may collectively define a new driver pathway, supported by the presence of high-impact common drivers like TP53 and MYC, which are critical in cancer pathways such as apoptosis and DNA repair.

In the GBM analysis, a cluster of size 10 genes was identified, containing four known drivers (GLI1, WNT2, BRAF, PLCG1) alongside several potential drivers. Notably, this cluster showed a mutual exclusivity p-value of 0.0901, which is relatively low for GBM. The genes in this cluster are involved in various pathways related to cell growth, apoptosis, and DNA repair, with six of the ten genes forming a compact network community. For instance, GLI1 and GLI2 are key hedgehog signaling genes linked to glioblastoma, playing crucial roles in cell differentiation and stem cell self-renewal.

In conclusion, C^3 outperformed CoMEt in mutual exclusivity, driver gene identification, and cluster tightness, particularly for BRCA. Although CoMEt had better coverage in GBM, C^3 identified more biologically significant clusters and handled larger cluster sizes without errors, making it a more robust tool for cancer research.

Chapter 4

Discussion

The following chapter will provide a discussion of the studies presented, including personal insights into the methodologies they employed and an evaluation of the clarity and quality of their written presentation.

allungare?

4.1 Dendrix

4.1.1 The deterministic formalization

Vandin et al. [33] provided one of the first mathematical formalizations of the phenomena of mutual exclusivity and coverage, in the context of gene mutations. Specifically, the definitions introduced offer a very intuitive approach to formalizing these biological concepts:

- the coverage of a gene is defined as the set of patients exhibiting a mutation of the gene, equivalent to the number of 1s in its column of the mutation matrix;
- a set of genes is defined to be mutually exclusive if no patient has more than one mutated gene in the set, i.e. no row of the set's associated matrix has more than 1 one;
- the coverage of a gene set is the set of patients with at least one mutation in the set;
- the coverage overlap of a gene set is the count of patients who possess more than one mutation within the gene set;
- the weight of a gene set is calculated as the difference between the coverage of the gene set and its coverage overlap.

Consequently, a higher weight for a gene set indicates both greater coverage and mutual exclusivity among its genes. The weight formula suggests that the optimal gene set, i.e. the one that maximazes its weight, is the one where the associated

matrix has a high number of rows with at least one 1, and a minimal number of rows with more than one 1.

In my view, this metric stands out as the most elegant among those discussed in this work: it not only provides a clear and intuitive measure, but also offers a simple and straightforward formula. While this formula may seem to oversimplify the challenge of identifying driver pathways — given that mutual exclusivity alone does not cover all aspects of pathway analysis, and exact mutual exclusivity is rarely observed in real data — it remains a highly regarded deterministic formalization, and numerous studies (some of which are discussed in this work) agree that this metric represents the most refined approach to date.

4.1.2 Additional considerations

I want to commend Vandin et al. [33] for their precise and methodical explanation of their methodology. With only a few minor, negligible details to consider, their work is exceptionally clear regarding their objectives, the methods employed to achieve them, and their actual outcomes. Additionally, their mathematical analysis is thorough and well-supported: in the supplemental material, they provide extensive proofs, including the NP-Hard-ness of the MWSP (which is described in Theorem 2.1), the correctness of their greedy algorithm, and the rapid mixing property of their MCMC approach. I greatly appreciate the clarity of their presentation, which significantly facilitated my understanding of their study. Indeed, this level of clarity is notably superior compared to other papers I have analyzed, which lack such clear explanations, as will be discussed in the subsequent sections.

As a final note, I believe that conducting a comparative analysis between the MCMC approach and a random search method would be interesting. For instance, consider the following algorithm:

1. *initialization*: given the set of all genes $\mathcal{G}$, choose an arbitrary subset $M_0 \subseteq \mathcal{G}$ of k genes;

2. *iteration*: for $t = 1, 2, \ldots$ derive M_{t+1} from M_t as follows:

   (a) define $W \subseteq \mathcal{G}$ and $V \subseteq M_t$ randomly;

   (b) choose $(\hat{w}, \hat{v}) \in \underset{(w,v) \in W \times V}{\arg\max}\, W\left((M_t - \{v\}) \cup \{w\}\right)$;

   (c) set $M_{t+1} := (M_t - \{\hat{v}\}) \cup \{\hat{w}\}$.

At each step, this algorithm selects a predetermined amount of *random adjustments*, choosing the one that maximizes the weight as the base set for the next iteration. It would be interesting to evaluate how this approach performs on real data, and whether the MCMC algorithm outperforms it, particularly when applied to data under the GIM model.

4.2 Multi-Dendrix

4.2.1 The ILP of Dendrix

Leiserson et al. [22] formulated the MWSP as an ILP, which I believe offers a more intuitive and natural approach to the problem. However, as highlighted by several authors, the set M that maximizes $W(M)$ may not always represent an actual biological driver pathway. This limitation stems from the fact that exact mutual exclusivity and coverage are rarely observed in real mutation data, making exact solutions to the MWSP potentially unrealistic. Therefore, I believe a more statistical approach or a probabilistic method, such as the MCMC algorithm used by Vandin et al. [33], may offer more reliable results in certain contexts.

4.2.2 The ILP of Multi-Dendrix

The ILP formulation for the simultaneous identification of multiple driver pathways is a natural extension of Dendrix's ILP. However, it may face the same challenge in that optimal solutions might not correspond to actual biological pathways. Moreover, Leiserson et al. [22] highlight that while the ILP used in Multi-Dendrix effectively finds optimal solutions, it does not rigorously explore suboptimal solutions, in contrast with the MCMC approach, which samples suboptimal solutions based on their weight.

Additionally, the weight function $W'(M)$ in Multi-Dendrix does not explicitly account for the co-occurrence of mutations between genes in different sets. Instead, it prioritizes gene sets with high coverage and approximate exclusivity, which may lead to co-occurrence due to high coverage alone (e.g., when all gene sets have full coverage). Given that co-occurrence is crucial in large biological pathways, algorithms that optimize for gene sets where mutations frequently co-occur might be more effective in identifying key components of these pathways [22].

Lastly, I would like to emphasize that the exposition of their ILP was unclear and somewhat imprecise: in particular, while Leiserson et al. [22] introduced the MMWSP, the ILP they solved appears to address a slightly different version of the problem. In their definition of the MMWSP, each set in the collection has a size of exactly $m \times k$ for a *fixed* k. However, in the definition of the ILP for Multi-Dendrix, there is no equation defining the sizes of the sets in the collection. Based on the context, it is likely that they intended for each gene set to have a different size, *varying* between some fixed $k_{\min}$ and $k_{\max}$. This discrepancy could potentially raise concerns about time complexity, an important consideration given the large scale of data in this field. Moreover, their definitions of some indicator variable sets lacked precision. Nevertheless, it is important to highlight that, according to survey studies [10], this approach is not only among the fastest — thanks to the efficiency of ILP solvers — but also performs exceptionally well in terms of both precision and recall.

4.3 MDPFinder

4.3.1 The ILP of Dendrix

To solve the MWSP, Zhao et al. [39] formulated an ILP that is both identical in formulation and constraints to the one proposed by Leiserson et al. [22]. Although the MDPFinder paper was published in 2012 and the Multi-Dendrix paper in 2013, the latter does not reference the work of Zhao et al. [39], and both papers present the ILP formulation of Dendrix as their own innovation.

For clarity and detailed definitions of the indicator variables, I have chosen to attribute the formulation to Leiserson et al. [22]. In fact, despite previous criticisms of their MMWSP's ILP exposition, their presentation of MWSP's ILP was clearer compared to that of Zhao et al. [39], which would have been challenging to comprehend without additional explanations.

4.3.2 The genetic algorithm

I find the use of genetic algorithms particularly appealing as a conceptual approach for optimizing a given score function, especially when a computationally efficient optimization method is not readily available. Indeed, genetic algorithms offer a flexible and intuitive framework for exploration and optimization. However, a significant drawback is their relatively slow performance compared to other methods. As noted by Zhao et al. [39], their genetic algorithm is considerably slower than the alternatives they tested, being *12 times slower* than the MCMC algorithm and *over 60 times slower* than the ILP.

Despite its undeniable slowness, I appreciate the versatility of this approach. It allows for easy modification of the objective function, and offers a well-suited integration procedure that may be challenging to incorporate into other algorithm types, such as the ILP. Additionally, the ability to explore suboptimal solutions is valuable, as shown in many findings reported in this work. Finally, their explanation of the GA is quite comprehensible, and I value their effort to compare the results across three different approaches.

4.4 Mutex

4.4.1 A very complex greedy algorithm

The clarity of the methodology presented in this paper is notably lacking, leading to significant challenges in understanding their approach. The authors provide a preliminary explanation of their algorithm, yet they omit crucial details, making it very difficult to grasp the complete procedure. The algorithm's recursive nature further complicates matters, as a simplified overview fails to convey the intricacies involved. In an effort to comprehend their methodology, I examined the Java source code included in the supplementary materials; this analysis ultimately clarified the

underlying processes and the implementation of their algorithm, and the pseudocodes provided in this work represent a *significantly simplified* version of the complete algorithm, which involves many nuanced aspects.

Particularly perplexing was their description of the *null distribution* estimation, which I found to be very challenging to interpret. However, after reviewing the source code, I recognized that the process is inherently complex: the *null distributions* are constructed recursively, through multiple calls to the `greedyMutex` procedure (described in Algorithm 3.3), and it incorporates various constants that dictate the evaluation methods, among many other factors. This complexity underscores the challenge of succinctly detailing such an intricate algorithm, and an exhaustive description would be beyond the scope of this work.

Furthermore, I found it particularly surprising that the Babur et al. [2] did not mention a crucial detail regarding their approach, which I would not have discovered had I not reviewed their source code: in their greedy algorithm, they require the `expandGroup` procedure (discussed in Algorithm 3.4) to select a candidate that is *not only* the best among those in $\delta(M)$, but *also improves the current score of* M. I believe that it is essential to explain the rationale behind this decision, as it is not immediately apparent. This choice implies that Mutex *does not explore suboptimal solutions*, but incorporating a less optimal candidate may ultimately lead to a better solution at the end of the greedy algorithm.

4.4.2 Additional considerations

Despite the aforementioned challenges in fully understanding their method, I believe that a statistical approach, such as the one they employed, is indeed the most appropriate choice, as exact mutual exclusivity rarely occurs in nature, as already mentioned. In fact, survey studies [10] indicate that this algorithm significantly outperforms others in terms of both precision and recall. Moreover, I found their hypergeometric model particularly intriguing, as it presents a novel way to assign a score to a given group of genes M, i.e. by evaluating the probability of overlap between the alterations of a specific gene $g \in M$, and the alterations of the remaining ones $M - \{g\}$.

Finally, Babur et al. [2] mention in their *Discussion* section that they aim to expand their work by exploring additional topological structures within the biological network. In fact, their current method focuses on genes with a common downstream target and necessitates that all group members are directly connected in the network without intermediary non-member nodes. However, incorporating linker nodes could facilitate the identification of more distant mutual exclusion relationships. I believe that this approach could provide a valuable direction for future research.

4.5 C^3

4.5.1 The clustering approach

The algorithm proposed by Hou et al. [16] presents a particularly intriguing approach, as applying a clustering method to this problem is not an immediately obvious solution. A potential avenue for further exploration could involve a modified version of the algorithm, where a single weight w is assigned to each edge, and a function is introduced to evaluate it, depending on the trade-off between w^- and w^+. Despite this potential variation, the ILP formulation of the clustering problem is well constructed, and the rounding procedure is intuitively compelling. In addition, their method demonstrates considerable versatility, both in terms of the flexibility of cluster sizes and the potential for integration with external data. The framework allows for the incorporation of any external data directly into the edge weights, further enhancing its adaptability to different contexts and datasets.

It is worth noting that several aspects of their exposition were somewhat unclear. For example, they provided minimal detail regarding the dataset and its relative size, and there is no mention of the source for the vectors representing the time-evolving expression values of the genes. Additionally, they included a misleading figure in their work: the figure implies that in the **EX-ME-CO** version of their algorithm, w^+ is derived from *coverage*, *expression data*, and *network information*. However, this contradicts the related paragraph, where w^+ is defined without using *network information*. Although they suggest that the definition could be optionally extended to include *network information* as shown in the figure, this inconsistency creates confusion. As a result, I am uncertain about how **EX-ME-CO** was precisely defined, and to avoid ambiguity, I chose to define **EX-ME-CO** and **NI-EX-ME-CO** as two distinct versions of the algorithm.

Lastly, in their *Evaluation Methods* section, they stated that they assess mutual exclusivity within a cluster using the probability of pairwise mutual exclusivity, defined by a hypergeometric distribution. They also stated that the pairwise Fisher's method was used by the Mutex algorithm, developed by Babur et al. [2], which was discussed earlier in this work. However, this is incorrect. As outlined in Section 2.2.3, Babur et al. [2] did not compute pairwise mutual exclusivity. Instead, they calculated multiple p-values between each single gene and the rest of the group. This discrepancy is unusual and raises concerns about the accuracy of their description.

Conclusions

TODO

Acknowledgements

TODO

Bibliography

[1] Özgün Babur et al. "Pattern search in BioPAX models". In: *Bioinformatics* 30.1 (Sept. 2013), 139–140. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt539. URL: http://dx.doi.org/10.1093/bioinformatics/btt539.

[2] Özgün Babur et al. "Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations". In: *Genome Biology* 16.1 (Feb. 2015). ISSN: 1474-760X. DOI: 10.1186/s13059-015-0612-6. URL: http://dx.doi.org/10.1186/s13059-015-0612-6.

[3] Tiziano Bernasocchi et al. "Dual functions of SPOP and ERG dictate androgen therapy responses in prostate cancer". In: *Nature Communications* 12.1 (Feb. 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-020-20820-x. URL: http://dx.doi.org/10.1038/s41467-020-20820-x.

[4] *Cancer of any site - Cancer Stat Facts*. URL: https://seer.cancer.gov/statfacts/html/all.html.

[5] *Cancro: la cura*. URL: https://www.airc.it/cancro/affronta-la-malattia/guida-alle-terapie/cancro-la-cura.

[6] E. G. Cerami et al. "Pathway Commons, a web resource for biological pathway data". In: *Nucleic Acids Research* 39.Database (Nov. 2010), D685–D690. ISSN: 1362-4962. DOI: 10.1093/nar/gkq1039. URL: http://dx.doi.org/10.1093/nar/gkq1039.

[7] Jaroslaw Cisowski et al. "What makes oncogenes mutually exclusive?" In: *Small GTPases* 8.3 (July 2016), 187–192. ISSN: 2154-1256. DOI: 10.1080/21541248.2016.1212689. URL: http://dx.doi.org/10.1080/21541248.2016.1212689.

[8] "Comprehensive genomic characterization defines human glioblastoma genes and core pathways". In: *Nature* 455.7216 (Sept. 2008), 1061–1068. ISSN: 1476-4687. DOI: 10.1038/nature07385. URL: http://dx.doi.org/10.1038/nature07385.

[9] Geoffrey M Cooper. *The development and causes of cancer*. 2000. URL: https://www.ncbi.nlm.nih.gov/books/NBK9963/.

[10] Yulan Deng et al. "Identifying mutual exclusivity across cancer genomes: computational approaches to discover genetic interaction and reveal tumor vulnerability". In: *Briefings in Bioinformatics* 20.1 (Aug. 2017), 254–266. ISSN: 1477-4054. DOI: 10.1093/bib/bbx109. URL: http://dx.doi.org/10.1093/bib/bbx109.

[11] Máire A. Duggan et al. "The Surveillance, Epidemiology, and End Results (SEER) Program and Pathology: Toward Strengthening the Critical Relationship". In: *American Journal of Surgical Pathology* 40.12 (Dec. 2016), e94–e102. ISSN: 0147-5185. DOI: 10.1097/pas.0000000000000749. URL: http://dx.doi.org/10.1097/PAS.0000000000000749.

[12] P. EHRLICH. "Experimental Researches on Specific Therapy". In: *The Collected Papers of Paul Ehrlich*. Elsevier, 1960, 106–117. ISBN: 9780080090566. DOI: 10.1016/b978-0-08-009056-6.50015-4. URL: http://dx.doi.org/10.1016/b978-0-08-009056-6.50015-4.

[13] Dávid Fazekas et al. "SignaLink 2 – a signaling pathway resource with multi-layered regulatory networks". In: *BMC Systems Biology* 7.1 (Jan. 2013). ISSN: 1752-0509. DOI: 10.1186/1752-0509-7-7. URL: http://dx.doi.org/10.1186/1752-0509-7-7.

[14] P. Andrew Futreal et al. "A census of human cancer genes". In: *Nature Reviews Cancer* 4.3 (Mar. 2004), 177–183. ISSN: 1474-1768. DOI: 10.1038/nrc1299. URL: http://dx.doi.org/10.1038/nrc1299.

[15] Dorit Hochbaum. *Approximation algorithms for NP-hard problems ed. by Dorit S. Hochbaum*. PWS Publ, 1996.

[16] Jack P. Hou et al. "A new correlation clustering method for cancer mutation analysis". In: *Bioinformatics* 32.24 (Aug. 2016), 3717–3728. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btw546. URL: http://dx.doi.org/10.1093/bioinformatics/btw546.

[17] "International network of cancer genome projects". In: *Nature* 464.7291 (Apr. 2010), 993–998. ISSN: 1476-4687. DOI: 10.1038/nature08987. URL: http://dx.doi.org/10.1038/nature08987.

[18] M. Kanehisa. "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1 (Jan. 2000), 27–30. ISSN: 1362-4962. DOI: 10.1093/nar/28.1.27. URL: http://dx.doi.org/10.1093/nar/28.1.27.

[19] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations*. Springer US, 1972, 85–103. ISBN: 9781468420012. DOI: 10.1007/978-1-4684-2001-2_9. URL: http://dx.doi.org/10.1007/978-1-4684-2001-2_9.

[20] Sushant Kumar et al. "Passenger Mutations in More Than 2, 500 Cancer Genomes: Overall Molecular Functional Impact and Consequences". In: *Cell* 180.5 (Mar. 2020), 915–927.e16. ISSN: 0092-8674. DOI: 10.1016/j.cell.2020.01.032. URL: http://dx.doi.org/10.1016/j.cell.2020.01.032.

[21] Michael S. Lawrence et al. "Mutational heterogeneity in cancer and the search for new cancer-associated genes". In: *Nature* 499.7457 (June 2013), 214–218. ISSN: 1476-4687. DOI: 10.1038/nature12213. URL: http://dx.doi.org/10.1038/nature12213.

[22] Mark D. M. Leiserson et al. "Simultaneous Identification of Multiple Driver Pathways in Cancer". In: *PLoS Computational Biology* 9.5 (May 2013). Ed. by Niko Beerenwinkel, e1003054. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003054. URL: http://dx.doi.org/10.1371/journal.pcbi.1003054.

[23] Mark DM Leiserson et al. "CoMEt: a statistical approach to identify combinations of mutually exclusive alterations in cancer". In: *Genome Biology* 16.1 (Aug. 2015). ISSN: 1474-760X. DOI: 10.1186/s13059-015-0700-7. URL: http://dx.doi.org/10.1186/s13059-015-0700-7.

[24] Craig H Mermel et al. "GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers". In: *Genome Biology* 12.4 (Apr. 2011). ISSN: 1474-760X. DOI: 10.1186/gb-2011-12-4-r41. URL: http://dx.doi.org/10.1186/gb-2011-12-4-r41.

[25] Alex V. Nesta et al. "Hotspots of Human Mutation". In: *Trends in Genetics* 37.8 (Aug. 2021), 717–729. ISSN: 0168-9525. DOI: 10.1016/j.tig.2020.10.003. URL: http://dx.doi.org/10.1016/j.tig.2020.10.003.

[26] Arnon Paz et al. "SPIKE: a database of highly curated human signaling pathways". In: *Nucleic Acids Research* 39.suppl_1 (Nov. 2010), D793–D799. ISSN: 1362-4962. DOI: 10.1093/nar/gkq1167. URL: http://dx.doi.org/10.1093/nar/gkq1167.

[27] *Side effects of cancer treatment*. URL: https://www.cancer.gov/about-cancer/treatment/side-effects.

[28] Rebecca L. Siegel et al. "Cancer statistics, 2024". In: *CA: A Cancer Journal for Clinicians* 74.1 (Jan. 2024), 12–49. ISSN: 1542-4863. DOI: 10.3322/caac.21820. URL: http://dx.doi.org/10.3322/caac.21820.

[29] Nicolas Stransky et al. "The Mutational Landscape of Head and Neck Squamous Cell Carcinoma". In: *Science* 333.6046 (Aug. 2011), 1157–1160. ISSN: 1095-9203. DOI: 10.1126/science.1208130. URL: http://dx.doi.org/10.1126/science.1208130.

[30] *Targeted therapy for cancer*. May 2022. URL: https://www.cancer.gov/about-cancer/treatment/types/targeted-therapies.

[31] *The genetics of cancer*. Aug. 2024. URL: https://www.cancer.gov/about-cancer/causes-prevention/genetics.

[32] Roman K Thomas et al. "High-throughput oncogene mutation profiling in human cancer". In: *Nature Genetics* 39.3 (Feb. 2007), 347–351. ISSN: 1546-1718. DOI: 10.1038/ng1975. URL: http://dx.doi.org/10.1038/ng1975.

[33] Fabio Vandin et al. "De novo discovery of mutated driver pathways in cancer". In: *Genome Research* 22.2 (June 2011), 375–385. ISSN: 1088-9051. DOI: 10.1101/gr.120477.111. URL: http://dx.doi.org/10.1101/gr.120477.111.

[34] Bert Vogelstein et al. "Cancer Genome Landscapes". In: *Science* 339.6127 (Mar. 2013), 1546–1558. ISSN: 1095-9203. DOI: 10.1126/science.1235122. URL: http://dx.doi.org/10.1126/science.1235122.

[35] Michael R. Waarts et al. "Targeting mutations in cancer". In: *Journal of Clinical Investigation* 132.8 (Apr. 2022). ISSN: 1558-8238. DOI: 10.1172/jci154943. URL: http://dx.doi.org/10.1172/JCI154943.

[36] Nhs Website. *Signs and symptoms*. June 2024. URL: https://www.nhs.uk/conditions/cancer/symptoms/.

[37] *What is cancer?* Oct. 2021. URL: https://www.cancer.gov/about-cancer/understanding/what-is-cancer.

[38] Christian Widakowich et al. "Review: Side Effects of Approved Molecular Targeted Therapies in Solid Cancers". In: *The Oncologist* 12.12 (Dec. 2007), 1443–1455. ISSN: 1549-490X. DOI: 10.1634/theoncologist.12-12-1443. URL: http://dx.doi.org/10.1634/theoncologist.12-12-1443.

[39] Junfei Zhao et al. "Efficient methods for identifying mutated driver pathways in cancer". In: *Bioinformatics* 28.22 (Sept. 2012), 2940–2947. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts564. URL: http://dx.doi.org/10.1093/bioinformatics/bts564.