De-novo pathway discovery for multi-omics data

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

M. Sc. Sebastian Winkler

aus München

Tübingen

2021

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	10.03.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	Prof. Dr. Daniel Huson

Every passing hour brings the Solar System forty-three thousand miles closer to Globular Cluster M13 in Hercules - and still there are some misfits who insist that there is no such thing as progress.

Ransom K. Ferm

Pfhhh#!Pfuch@BrfhhH)?

Linus Winkler

Declaration

All work presented here was performed by Sebastian Winkler, unless stated otherwise. Colleagues' contributions are listed in details in Contributions. The thesis itself was written by Sebastian Winkler. The work was performed under supervision of Prof. Dr. Oliver Kohlbacher within the Applied Bioinformatics Group, Faculty of Computer Science, University of Tübingen.

Abstract

This thesis presents algorithms and software which allow the extraction of biologically meaningful patterns from high-throughput multi-omics data and biomolecular networks. It describes the concept and implementation of an algorithm which allows the extraction of deregulated subnetworks from large directed molecular interaction networks based on node scores derived from omics data. Statistical underpinnings of the algorithms are derived and the algorithm is benchmarked against its closest methodological relative. Relying on fractional integer programming, the algorithm and its implementation, DeRegNet, allow many flexible modes of application. I demonstrate the application of the algorithm in the context of the public The Cancer Genome Atlas (TCGA) liver cancer dataset, a study investigating the role of folate one-carbon metabolism in liver cancer and a study about the phosphoproteomic regulation of the *Saccharomyces cerevisiae* (budding yeast) cell cycle. Finally, the general architecture and some implementation details of a web-based API for DeRegNet are presented.

Zusammenfassung

In der vorliegenden Arbeit werden Methoden und Software vorgestellt, die es erlauben, aus Hochdurchsatz Omics-Daten und biomolekularen Interaktionsnetzwerken biologisch relevante Muster zu extrahieren. Es wird ein Algorithmus entwickelt, der es ermöglicht, aus großen gerichteten molekularen Interaktionsnetzwerken sog. deregulierte Teilnetzwerke zu extrahieren. Deregulierung wird hierbei über auf die Knoten des Netzwerkes abgebildete Omics-Daten definiert. Es wird eine statistische Grundlage für den vorgestellten Algorithmus diskutiert und eine Evaluierung hinsichtlich methodisch verwandter Verfahren vorgenommen. Der Algorithmus und seine Implementierung, DeRegNet, beruhen auf fraktionaler ganzzahliger Optimierung und erlauben zahlreiche Anwendungsszenarien. Exemplarisch wird die Anwendung auf öffentlich zugängliche Daten des TCGA-Projekts vorgestellt (TCGA: The Cancer Genome Atlas), hier genauer an Hand der Daten zum hepatozellulären Karzinom (Leberkrebs). Weiterhin werden Anwendungen auf eine Studie des Folate One-Carbon Metabolismus im Leberkrebs, als auch auf die phosphoproteomische Regulierung des Saccharomyces cerevisiae (Backhefe) Zellzyklus beschrieben. Abschließend wird auf die allgemeine Architektur und einige Implementationsdetails einer web-basierten API (Application Programming Interface) zur Bereitstellung von DeRegNet eingegangen.

Acknowledgments

I would like to thank Prof. Dr. Kohlbacher and all members of the Applied Bioinformatics Group (ABI). Special gratitude goes to Philip Thiel for sharing his TeX thesis template within ABI. Also I would like to thank my various office co-inhabitants for their geniality: Thorsten Tiede, Mirjam Figaschewski, Matthew Divine, Lukas Zimmermann, Eugen Netz and Nico Weber. I would also like to thank the International Max Planck Research School (IMPRS) "From Molecules to Organisms", i.e. its organizers and participants.

Most importantly, for objective and scientific reasons, I like to express elevated gratitude to my wife Ivana and our son Linus. I especially congratulate Linus for his existence! Well done!

Contents

Chapter 1

Introduction

This thesis is about de-novo pathway identification within biomolecular networks based on omics data. I therefore begin by shortly introducing the most common omics technologies in section 1.1 and then proceed to introduce biomolecular networks in the context of de-novo pathway identification in section 1.2. I also conceptually outline the methodological precursors to de-novo pathway identification, namely (topological) pathway enrichment methods. While this chapter will remain fairly conceptual in nature, chapter 2 will then provide a more formal review of functional enrichment in general and de-novo subnetwork enrichment in particular.

1.1 Multi-omics data

Cancer research, as one of the major biomedical challenges of the 21^{st} century, is increasingly being conducted in a genome-wide and personalized fashion by utilizing modern molecular high-throughput technologies, such as DNA sequencing, RNA-Seq [WGS09] and mass spectrometry [AMH13] for proteomic and metabolomic analyses. These technologies enable the experimental probing of multiple molecular organizational levels of biology.

The respectively targeted layers of biological organization are referred to as *omes* (e.g. genome, transcriptome, proteome, metabolome), while the general fields of

study addressing these omes are known as *omics* (e.g. genomics, transcriptomics, proteomics, metabolomics). Experimental technology used to assess genome-wide characteristics of the associated ome is called an *omics technology* and an *omics layer* is an experimental read-out produced with the corresponding omics technology.

Many of of these omics technologies are increasingly applied in clinical settings and publicly available large-scale data resources such as The Cancer Genome Atlas (TCGA) [TCW15] provide ample opportunity to leverage the vast amounts of available data for methodological and ultimately biomedical progress. These resources can provide valuable reference datasets in the analysis of molecular profiles of individual patients and patient groups. However, one of the biggest challenges in the analysis of omics data remains *functional annotation/interpretation*: The interpretation of the experimental read-outs with the goal of understanding the underlying known or unknown biological processes and functions, is a vital step in providing personalized, precise and focused molecular therapies.

1.2 From pathway enrichment to de-novo pathway discovery in biomolecular networks

1.2.1 Functional annotation via pathway enrichment methods

One of the most widely used approaches for functional annotation is Gene Set Enrichment (GSE) [Mac14]. In its most basic form, GSE entails hypergeometric and Fisher test-based approaches to detect the overrepresentation of differentially expressed genes. Input information for GSE are a set of predefined gene sets (e.g. representing the gene contents of various pathways from pathway databases [D'E13] such as KEGG [KFT+17], WikiPathways [KRN+16] or Reactome [FJM+18]) and a measure of "deregulation" (e.g. binary indication of differential gene expression, expression fold changes, etc.). The goal of the GSE analysis is to identify those gene sets from the collection which show "high" deregulation. Here, the term "high" is defined by the method's spe-

cific underlying statistical model. In the simplest case, the method examines if a gene set contains a higher number of differentially expressed genes than would be expected by chance, under the assumption that genes are differentially expressed independently of each other with a uniform probability. Many adaptations and variations of GSE exist [Mac14], among them Gene set enrichment analysis (GSEA) [STM+05]. See figure 1.1 for a conceptual depiction of basic GSE.

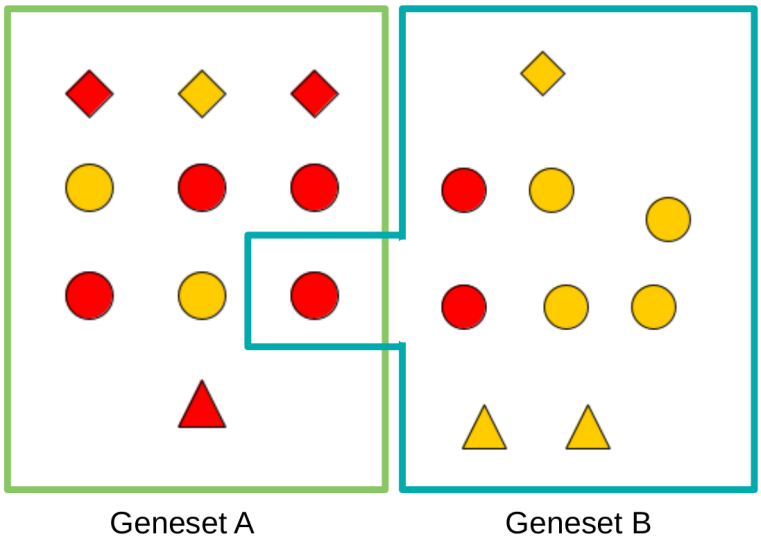


Figure 1.1: Conceptual view of classical pathway/gene set analysis. Gene sets/pathways are considered merely as sets of genes ignoring any explicit biomolecular interactions between the elements of a gene set/pathway. Here red nodes represent differentially regulated genes and a basic GSE analysis employing hypergeometric Over-representation analysis (ORA) would test for more red nodes than expected, given a method-specific statistical model, in any given gene set. Rhombic elements represent *receptors* while triangle elements represent *targets*/*terminals* which could correspond to membrane receptors and transcription factors in a biochemical signal transduction pathway. Elements encircled by the green and blue boxes respectively denote predefined gene sets.

1.2.2 Functional annotation via topology-aware pathway enrichment

Classical GSE methods treat pathways as an unstructured collection of genes[1] and do not explicitly take into account the biomolecular connections between the genes/proteins. Explicit network representations of the interactions between genes/proteins play an important role in systems biology [BBI07, KHT09, KP12, BGL11, QZ14, WHFL13, BSG16, Fur13, CRH+15] and exist in various forms. Genes/proteins can be connected in a network to represent signaling pathways, metabolic networks [CDK13], gene regulatory networks [Big11] or protein-protein interaction networks [LWH+17, SMC+17] which can convey more fine-grained views into biological systems functions than just unstructured functionally grouped sets of genes (gene sets, pathways). Another aspect in this context is *pathway crosstalk*, i.e. the interconnections of pathways by virtue of shared biological agents between two gene sets or a known connection between two biological agents each being in different gene sets. Both situations, i.e. that a gene/protein is part of multiple pathways or that genes/proteins in different pathways are known to interact, are common in practice.

There has been extensive research into the possibility of designing enrichment methods which take into account the topology of pathways [JE16, MTB+13, IPB18]. An example of such an approach is the calculation of topology-dependent perturbation scores for each gene [TDK+09]. See figure 1.2 for a conceptual representation of topological GSE. For a conceptual depiction of topological GSE which allows for pathway crosstalk, see figure 1.3.

1.2.3 De-novo pathway enrichment

While historically defined pathways have a solid base in biological findings and can provide useful guidance for functional interpretation of omics experiments, molecular and cellular events are often more complicated and involve the direct interaction of molec-

[1] See figure 1.1.

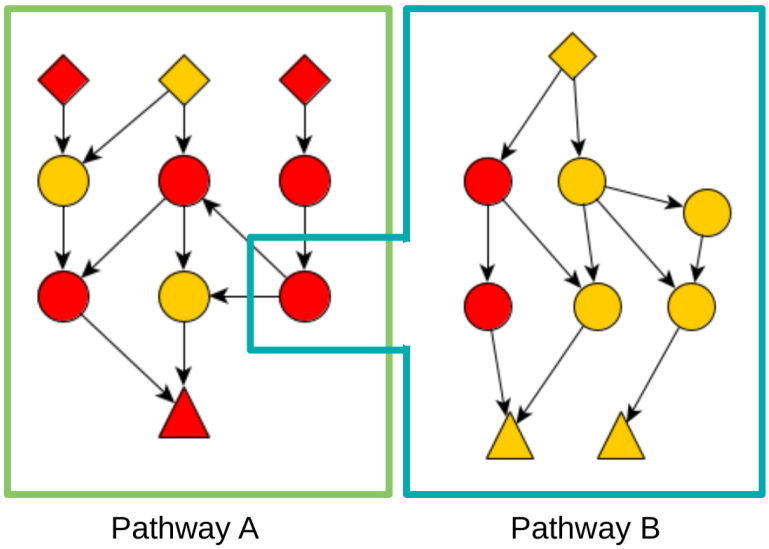


Figure 1.2: Conceptual view of topological pathway/analysis. Biomolecular interactions are taken into account when calculating enrichment for any given pathway. Gene sets/pathways are still predefined though and interactions between pathways are usually not taken into account, compare figure 1.3. Again, elements colored in red correspond to *deregulated* nodes. Rhombic elements represent *receptors* while triangle elements represent *targets*/*terminals* which could correspond to membrane receptors and transcription factors in a biochemical signal transduction pathway. Elements encircled by the green and blue boxes respectively denote predefined gene sets.

ular entities across predefined pathway boundaries. Correspondingly, a wide range of methods were proposed which aim to extract "deregulated" patterns from larger regulatory networks without relying on predefined pathways [MCRI13, BAG+17]. These methods are often referred to as *de-novo* pathway enrichment methods, emphasizing that the pathways are defined/extracted from the data itself and are not given as fixed gene sets a priori. In this thesis I often use the term deregulated subnetwork, dysregulated subnetwork or de-novo subnetwork synonymously with *de-novo pathway*.

Some de-novo methods are tailored to the characteristics of a particular data type. An example are methods attempting to find significantly mutated pathways/networks [VRU16, VUR12a, ZZ18, CDS+10, HSC+13, VUR12b], trying to factor in the pecularities of mutation data in a network context. Another way to categorize these methods is based on how they handle undirected or directed interaction networks.

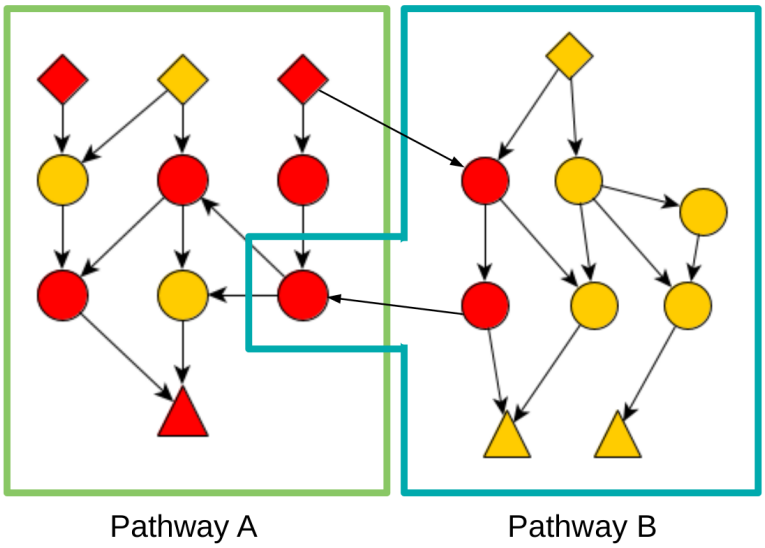


Figure 1.3: Conceptual view of topological pathway/analysis with pathway crosstalks. Pathway crosstalks happen when genes are part of multiple pathways or genes in different pathways are known to interact. Even with pathway crosstalks accounted for, the gene sets/pathways as such are still predetermined. Again, elements colored in red correspond to *deregulated* nodes. Rhombic elements represent *receptors* while triangle elements represent *targets*/*terminals* which could correspond to membrane receptors and transcription factors in a biochemical signal transduction pathway. Elements encircled by the green and blue boxes respectively denote predefined gene sets.

A lot of biomolecular interactions are directed in nature, e.g. protein A phosphorylates protein B, enzyme A precedes enzyme B in a metabolic pathway in contrast to symmetric interactions such as physical interactions of proteins in protein complexes. Some methods designed for undirected networks are described for example in the following studies: [IOSS02, PN05, US07, DKR+08, ZWCA08, US09, UKKS10, DWC+11, BBBB+11, AFK+12, APB+14, ALDH+16]. More detailed description of these methods and further extensions is available in [MCRI13, BAG+17] and in chapter 2 of this thesis. While methods which work natively with directed networks are rarer [DKR+08, KBG+09, BRK+12, AS13, GSH+13, MSI+15, LTG+19], it is instrumental to be able to capture the effects of directed biomolecular interactions in the process of discovering dysregulated networks. One particular approach is the one described in [BRK+12] which utilized an integer programming approach in order to find deregu-

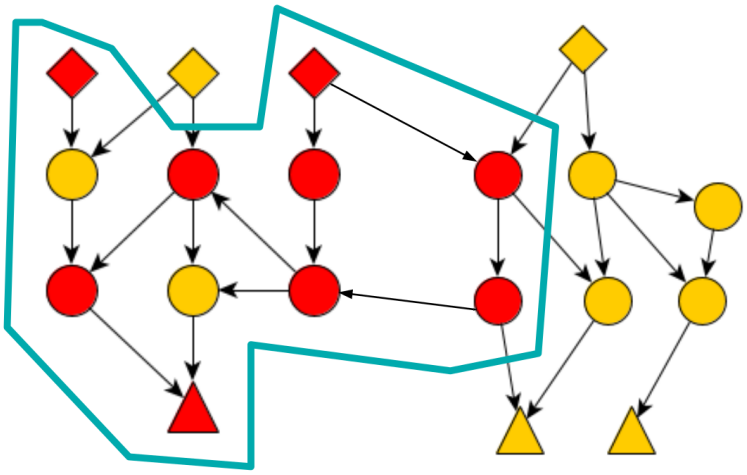


Figure 1.4: Conceptual view of de-novo pathway analysis. De-novo pathway identification / deregulated subnetwork discovery drops the predetermined pathways and defines enriched subnetworks/pathways from the omics data itself. Again, elements colored in red correspond to *deregulated* nodes. Rhombic elements represent *receptors* while triangle elements represent *targets*/*terminals* which could correspond to membrane receptors and transcription factors in a biochemical signal transduction pathway.

lated subnetworks. These subnetworks show deregulated subnetworks downstream or upstream of a so called root node where the latter can be fixed *a priori* or determined by the algorithm itself. Further approaches include the prize-collecting Steiner tree methods proposed in [HF09, HCG+13, TBP+13, TGK+16] which allow for flexible identification of subnetworks which connect certain types of nodes (so called *sources* with *terminals*). Chapter 2 provides a more detailed view of de-novo pathway enrichment in general and (exact) methods in particular. See figure 1.4 for a conceptual depiction of de-novo pathway enrichment.

1.3 Outlook

After a technical overview on existing functional enrichment methods with a focus on exact methods in chapter 2, the remainder of this thesis will introduce the de-novo pathway enrichment algorithm *DeRegNet* in chapter 3, outline several applications

of DeRegNet to omics datasets in chapter 4 and concludes with a description of a web-based Application Programming Interface (API) for DeRegNet in chapter 5.

Chapter 2

Functional enrichment: a methodological overview

This chapter provides an overview of the main methodological principles involved in the progression from gene set enrichment to de-novo pathway/subnetwork detection algorithms. I start by outlining some terminology and notation to be used in the following. The terminology and notation is explicitly referring to graph concepts throughout, even for methods which could be formulated without any reference to such concepts. The following sections then outline gene set enrichment, topological extensions to gene set enrichment and finally de-novo subnetwork/pathway enrichment methods, focusing on exact directed methods since the algorithms proposed in subsequent chapters of this thesis fall into this category.

2.1 Terminology and notation

Formally, it is given a directed graph $G = (V, E)$, i.e. $E \subset V \times V$, representing knowledge about biomolecular interactions in some way. To avoid certain pathologies in the models defined below, it is assumed that G has no self-loops, i.e. $(v, v) \notin E \ \forall v \in V$. For a subset $S \subset V$, one defines $\delta^+(S) = \{u \in V \backslash S : \exists v \in S : (v, u) \in E\}$ and $\delta^-(S) = \{u \in V \backslash S : \exists v \in S : (u, v) \in E\}$, i.e. the sets of outgoing nodes into and incoming nodes

from a set of nodes S. For a node $v \in V$ one writes $\delta^{\pm}(v) := \delta^{\pm}(\{v\})$. Furthermore, it is given a score function $s : V \to \mathbb{R}$, describing some summary of experimental data available for the biomolecular entities represented by the nodes. For a given graph $G = (V, E)$ any node labeling function $f : V \to \mathbb{R}$ is implicitly implied to be a vector $f \in \mathbb{R}^{|V|}$, subject to an arbitrary but fixed ordering of the nodes (shared across all node labeling functions). In particular, with $f_v := f(v)$ for $v \in V$, given $f, g : V \to \mathbb{R}$, one can write $f^T g = \sum_{v \in V} f_v g_v$. For $S \subset V$ and $f : V \to \mathbb{R}$ one defines $f_S : V \to \mathbb{R}$ via $f_S(v) := 0$ for all $v \in V \setminus S$ and $f_S(v) := f(v)$ for all $v \in S$. Defining $e : V \to \mathbb{R}$ with $e(v) := 1$ for all $v \in V$, one further can write $e_S^T f = \sum_{v \in S} f_v$ for $S \subset V$ and $f : V \to \mathbb{R}$. Comparison of node labeling functions f, g are meant to be understood element-wise, e.g. $f \leq g$ means $f_v \leq g_v$ for all $v \in V$. An edge weight (function) is a function $w : E \to \mathbb{R}$.

2.2 Gene set enrichment

Under gene set enrichment I subsume all methods which try to identify pre-defined pathway/gene sets which are *enriched* with members (genes, proteins, ...) which are *deregulated* according so some measure of deregulation *without* taking topological interactions between members and/or pathways/gene sets into account. See figure 1.1 for a conceptual depiction of that general setting. The measure of deregulation often is an indicator of differential expression, but this is flexible due to the nature of most enrichment methods and can also correspond to more complex measures such as continuous measurements of some omics modality and also is not restricted to gene expression. In the following I outline the basic application of standard statistical tests based on the hypergeometric distribution characteristics of the enrichment problem. While many extensions and modifications of basic gene set enrichment exist [Mac14, SLM08], discussion of the most basic setting will suffice as a proxy for the purposes of this thesis as it captures the essential characteristics of the classical gene set enrichment problem. Further methods for gene set enrichment are GSEA

[MLE+03, STM+05], LRpath based on logistic regression [SLM08, KKM+12], the random set approach [NQdB+07] or gene set analysis (GSA) [ET06]. For a comparison and classification of these and further methods it is referred to [Mac14].

Hypergeometric enrichment tests

A **pathway/gene set database** is represented by a set $\mathscr{P} \subset 2^V$, i.e. a set $\mathscr{P} = \{P_1, ..., P_n\}$ with $P_i \subset V$ for all $i = 1, ..., n$ some $n \in \mathbb{N}$. A **deregulation set** is a subset $D \subset V$ of nodes which is indicated to be deregulated according to some measure. One common deregulation set is the set obtained by all genes which are differentially expressed between two given conditions. Hypergeometric tests of gene set enrichment (also sometimes referred to as overrepresentation analysis (ORA)) test whether a given pathway P_i contains significantly more (or less) nodes from the deregulation set as would be expected by chance. These tests correspond to one-sided versions of the Fisher exact test (χ^2 test for contingency tables). See figure 2.1 for a canonical depiction of the setting. Under the null hypothesis that each node/gene has the same probability of being deregulated regardless of it being in P_i or not the counts of deregulated nodes/genes in a given pathway follow a hypergeometric distribution, i.e.

$$P(|P_i \cap D| = k) = \frac{\begin{pmatrix} |D| \\ k \end{pmatrix}\begin{pmatrix} |V|-|D| \\ |P_i|-k \end{pmatrix}}{\begin{pmatrix} |V| \\ |P_i| \end{pmatrix}} = \frac{\begin{pmatrix} e_D^T e \\ e_D^T e_{P_i} \end{pmatrix}\begin{pmatrix} e^T(e-e_D) \\ e^T(e_{P_i}-e_D) \end{pmatrix}}{\begin{pmatrix} e^T e \\ e^T e_{P_i} \end{pmatrix}}$$

and enrichment (of nodes from D in a given pathway P_i) can then be calculated by calculating a one-sided p-value under that null hypothesis by summing all probabilities $P(|P_i \cap D| = k')$ for all k' which are at least as extreme as the observed k. For

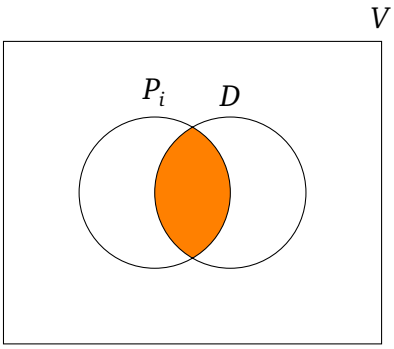


Figure 2.1: Conceptual depiction of a pathway/gene set P_i overlapping with a deregulation set D. Both sets are subsets of the set of all nodes/genes V which are the nodes of an underlying regulatory network $G = (V, E)$. The topological interaction encoded by E are ignored in standard gene set enrichment methods.

enrichment/overrepresentation[1] this means calculating

$$p := P(|P_i \cap D| \geq k) = \sum_{k' \geq k} P(|P_i \cap D| = k').$$

Note, that after each P_i is associated with its enrichment p value, multiple testing correction should be carried out due to testing multiple pathways simultaneously for enrichment. It may seem trivial, but I stress here that the just outlined enrichment method (and related methods [Mac14]) do not make any explicit use of E, i.e. the interactions between genes, given an underlying regulatory network $G = (V, E)$. Also the identified enriched gene sets are given a priori via the pathway database.

2.3 Topological pathway enrichment

As conceptually motivated in subsection 1.2.2, **topological gene set enrichment** refers to pathway enrichment methods which make use of the connections between genes in the pathways. Formally, a **topological pathway database** is a set $\{G_1, ..., G_n\}$ of subgraphs $G_i = (V_i, E_i)$ of G such that G_i is the subgraph of G induced by $V_i \subset V$, i.e.

[1] Note, that one can also test for depletion/underrepresentation of deregulated nodes/genes by calculating $P(|P_i \cap D| \leq k) = \sum_{k' \leq k} P(|P_i \cap D| = k')$.

$E_i = \{(u,v) \in E | u,v \in V_i\}$. In the context of topological pathway analysis, each G_i is referred to as *a pathway*. See figure 1.2 for conceptual depiction corresponding to that definition. The result of a topological pathway enrichment analysis is a p-value for every pathway indicating whether that pathway is enriched with respect to some measure of deregulation (like gene expression fold changes) under the constraints imposed by the topology of each pathway. How the connections between the genes in each pathway, i.e. its topology, are taken into account when determining a pathway's significance for enrichment is characteristic for a given topological pathway enrichment method. For a review on topological pathway enrichment in general and existing methods in particular I refer to [JE16, MTB+13, IPB18]. In the following I provide an overview of Signaling Pathway Impact Analysis (SPIA) [TDK+09] as a representative example for a topological path way enrichment algorithm.

Signaling Pathway Impact Analysis (SPIA): [TDK+09]

Signaling Pathway Impact Analysis [TDK+09] was one of the first topological pathway enrichment methods. Given a node score $s : V \to \mathbb{R}$ representing the difference of some omics measure, such as a gene expression fold change between two conditions, SPIA is based on a pathway-specific derived perturbation score $p^{(i)} : V_i \to \mathbb{R}$ which captures the interplay of pathway topology and the mapped deregulation score s as $p_v^{(i)} = s_v + \beta_i^{(v)} \tilde{p}^{(i)}_{\delta_i^-(v)}$[2]. Here, $\delta_i^{\pm}(v) := \delta^{\pm}(v) \cap V_i$. Furthermore, $\beta_i^{(v)}$ denotes the row corresponding to node $v \in V_i$ of a pathway-specific parameter matrix $\beta_i = (\beta_{uv}^{(i)}) \in \mathbb{R}^{V_i \times V_i}$ and $\tilde{p}_v^{(i)} = \frac{p_v^{(i)}}{|\delta_i^+(v)|}$ is the perturbation score of v normalized by the number of downstream genes $\delta_i^+(v)$ of v (in pathway G_i). Intuitively, the perturbation score of gene $v \in V_i$ in pathway G_i is determined by its general deregulation s_v and the combined contributions of its upstream regulators. The latter in turn is quantified as the weighted sum of the upstream regulators' perturbation scores normalized with each regulator's number of targeted genes. The normalization serves to down-weigh

[2] $p_v^{(i)} = s_v + \sum_{u \in \delta^-(v)} \beta_{vu}^{(i)} \frac{p_u^{(i)}}{|\delta_i^+(u)|}$

contributions from nodes which regulate many other nodes relative to nodes which may only regulate very few nodes what can then be seen as a potentially more significant influence. [TDK+09] define a node's perturbation accumulation (for pathway G_i) as $a_v^{(i)} := p_v^{(i)} - s_v$ and show that given above model, $a^{(i)}$ can be determined as $a^{(i)} = B^{(i)}(I - B^{(i)})^{-1} s_{V_i}$ where $B^{(i)} = (\frac{\beta_{uv}}{|\delta_i^+(v)|}) \in \mathbb{R}^{V_i \times V_i}$. To arrive at a p value indicating enrichment, SPIA also assumes the classical null hypothesis H_0 that deregulated genes are completely random [TDK+09][3]. The topological perturbation p-value is given by $p_{\text{PERT}} = P(e^T A^{(i)} \geq e^T a^{(i)} | H_0)$ and can be evaluated by simulating perturbation accumulation scores $A^{(i)}$ by bootstrapping as described in [TDK+09]. p_{PERT} is then combined with a standard ORA p-value (see section 2.2) by classical p-value aggregation via Fisher's method [Fis92] to arrive at final p-value for the significance of pathway enrichment.

2.4 De-novo subnetwork enrichment

In this section I outline existing *de-novo* subnetwork enrichment approaches whose rationale was introduced in the introduction and is conceptually depicted in figure 1.4. For dedicated reviews, it is referred to [MCRI13] and [BAG+17]. De-novo subnetwork enrichment acquired many names during the last two decades, such as active subnetwork/module detection or deregulated subnetwork/module detection. I try to restrict myself to *de-novo subnetwork/pathway enrichment/detection/discovery* and *deregulated subnetwork/subgraph enrichment/detection/discovery* throughout this thesis. The discussion here is stratified according to whether the algorithms are heuristic or exact in nature. The next subsection outlines heuristic approaches while the following subsection gives some details on exact methods. The overview on exact methods is my primary focus due to the fact, that this thesis develops such an exact approach to de-novo enrichment starting in the next chapter.

[3] See previous section 2.2

2.4.1 Heuristic methods for de-novo subnetwork detection

One of the first papers describing a de-novo method for detecting deregulated subnetwork was presented by [IOSS02][4]. I sketch the methods of [IOSS02] which is based on simulated annealing in the next subsection.

Starting with the Simulated Annealing approach of [IOSS02] many further heuristic methods were proposed [IOSS02, PN05, US07, US09, UKKS10, DWC+11, BBBB+11, AFK+12, APB+14, ALDH+16]. These methods, while achieving similar end results on an abstract level, vary vastly in terms of suitable underlying networks, interpretation of outcomes and algorithmic strategies employed. Algorithmic approaches employed range from ant colony optimization [ALDH+16], dynamic programming [DWC+11], Markov random fields [VBS+10] to message passing approaches [BBBB+11]. See [MCRI13, BAG+17] for further references.

Ideker et al. 2002 [IOSS02]: de-novo enrichment via simulated annealing

One of the first studies examining the possibilities of de-novo pathway/subnetwork enrichment was the work of Ideker et al. from 2002 [IOSS02] who referred to the deregulated subnetworks inferred by their algorithms as *active modules*. The algorithm itself and its most commonly used implementation [IOSS02] is known as *jActiveModules*. *jActiveModules* applies the metaheuristic principle of simulated annealing to find deregulated subnetworks within a larger regulatory network. The algorithm is outlined as algorithm 1 in more detail to give a somewhat representative example on how heuristic optimization approaches can be applied to find de-novo pathways.

2.4.2 Exact methods for de-novo subnetwork detection

This subsection gives an overview on exact methods for de-novo enrichment. Here, the term *exact* is understood in terms of how the underlying optimization problem for finding deregulated subgraphs is solved, namely provably exact. This is in contrast to

[4] Who called the resulting subnetworks *active modules*

Data: A undirected graph $G = (V, E)$, node score $s : V \to \mathbb{R}$, start and end temperature $T_0, T_{\text{end}} > 0$, maximal number of iterations $N \in \mathbb{N}$
Result: $V' \subset V$ and the deregulated subnetwork $G' = (V', E')$ induced by V'
Set $x_v = 1$ (otherwise $x_v = 0$) with probability $\frac{1}{2}$ for each $v \in V$
$s_0 := s^T x$

(Annealing phase)
for $i = 1, ..., N$ **do**
  Pick $v \in V$ at random and set $x_v = |1 - x_v|$ (Toggle)
  $s_i = s^T x$ (Toggled score)
  if $s_i \leq s_{i-1}$ **then**
    $x_v = |1 - x_v|$ (toggle back) with probability $1 - e^{\frac{s_i - s_{i-1}}{T_i}}$
  end
  $T_i = \sqrt[N]{\frac{T_{\text{end}}}{T_0}} \cdot T_{i-1}$ (Update temperature)
end

(Quenching phase)
$\mathscr{B}(x) := \{\tilde{x} \in \{0,1\}^V | \exists v \in V : |x_v| = |\tilde{x}_v - 1|, s^T x < s^T \tilde{x}\}$
while $\mathscr{B}(x) \neq \emptyset$ **do**
  Set $x = \tilde{x}$ for some $\tilde{x} \in \mathscr{B}(x)$
end
Let $G' = (V', E')$ be the subgraph of G induced by $V' = \{v \in V | x_v = 1\}$
return (Connected component $\mathscr{C} \subset V'$ of G' with highest score $s^T e_{\mathscr{C}}$)

Algorithm 1: Simulated annealing for de-novo enrichment [IOSS02]. The algorithm applies simulated annealing to find subnetworks by iteratively toggling a node's membership in the current subnetwork and evaluating the resulting total score of the subnetwork. According to the defining principle of simulated annealing, new solutions are accepted with a certain probability even if they worsen the score of the resulting subnetwork. Once the annealing stopped, local search is used to find a locally optimal solution. This is called *quenching* in the context of simulated annealing. Finally the best scoring connected component of the subgraph induced by the selected nodes is returned as the deregulated subgraph.

the heuristic methods introduced in the previous subsection which find solutions by means of heuristic approximations to a solution. Often one also cannot provide any guarantees on how close an approximate heuristic solution may be to an optimal one. This drawback gave rise to so called exact methods for de-novo pathway enrichment which are are based on mixed-integer programming. Before reviewing existing exact methods, I give an overview of the so called (directed) maximum weight connected subgraph problem on which many exact methods are based on. In particular, the algorithms introduced in this thesis solve certain extended and generalized versions of that problem as outlined in chapter 3.

(Directed) Maximum Weight Connected Subgraph Problem (MWCSP)

In terms of mathematical optimization and subject to minor modifications[5], most exact (and some heuristic) de-novo subnetwork detection methods solve instances of the so called (directed) Maximum Weight Connected Subgraph Problem[6]. In the following I provide formal definitions of that problem.

In the setting of an undirected graph one defines [DKR+08]:

Definition 1 (Maximum Weight Connected Subgraph Problem (MWCSP))
Given an undirected graph $G = (V, E)$ and node scores $s : V \to \mathbb{R}$, find a set of nodes $V' \subset V$ whose induced subgraph (V', E') maximizes $e_{V'}^T s$ such that (V', E') is connected.

For directed graphs, the definition largely stays the same except for the notion of connectivity:

Definition 2 (Directed Maximum Weight Connected Subgraph Problem (DMWCSP))
Given a directed graph $G = (V, E)$ and node scores $s : V \to \mathbb{R}$, find a set of nodes $V' \subset V$ whose induced subgraph (V', E') maximizes $e_{V'}^T s$ such that there is a node $r \in V'$ such that there is a directed path from r to every other node $v \in V'$.

[5] For example the requirement of the subgraphs to be of a certain predefined size $k \in \mathbb{N}$.
[6] Or its rooted variant.

In the directed setting, by fixing the root node in the DMWCSP to a particular node in the underlying graph one arrives at the so called **Rooted Maximum Weight Connected Subgraph Problem (RMWCSP)**:

Definition 3 (Rooted Maximum Weight Connected Subgraph Problem (RMWCSP))
Given a directed graph $G = (V, E)$, node scores $s : V \to \mathbb{R}$, and a node $r \in V$ called the root node, find a set of nodes $V' \subset V$ with $r \in V'$ such that the induced subgraph (V', E') maximizes $e_{V'}^T s$ such that there is a directed path from r to every other node $v \in V'$.

As indicated, the (R)MWCSP has found explicit applications in network biology [DKR+08], [BRK+12] in the context of de-novo subnetwork detection which will be outlined in more detail in the following subsections. The problem also attracted general computational and theoretical research in recent years [BWB17], [LAS16], from different integer programming formulations and problem-specific branch-and-cut strategies [EKK14], [ÁMLM13a], [ÁMLM13b], [AB11], to more recent research on computational strategies for addressing large-scale instances [AMS17] and problem reduction techniques and heuristics [RKM19], [RK19].

Dittrich et al. 2008 [DKR+08]: Prize-collecting Steiner tree problem

[DKR+08] presented a solution to the (undirected) de-novo subnetwork detection problem by formulating it as an undirected MWCSP (see 2.4.2) and showing that the problem can be transformed to a so called Prize-collecting Steiner tree (PCST) problem.

Definition 4 (Prize-collecting Steiner tree (PCST) problem)
Given an undirected graph $G = (V, E)$, node scores $p : V \to [0, \infty)$ and edge scores $c : E \to [0, \infty)$, find a set of nodes $V' \subset V$ and edges $E' \subset E$ such that the subgraph (V', E') is connected[7] *and maximizes $e_{V'}^T p - e_{E'}^T c$.*

[7] Note that the requirement of *connectedness could also be replaced by the requirement of (V', E') being a tree since from any connected subgraph one can remove edges such that the result is a tree while only improving the objective in the process due to $c(e) \geq 0$ for all $e \in E$.*

In above definition of the PCST problem, the node score p can still be interpreted as a measure of deregulation while the edge score c can be interpreted as the cost of including an edge into the solution and hence a solution of the PCST problem tries to maximize deregulation while minimizing the cost of edge inclusion. Given an undirected MWCSP instance (see definition 1), [DKR+08] achieve an equivalent PCST formulation by defining $p_v := s_v - \max_{u \in V} s_u$ for every $v \in V$ and $c_e := -\max_{u \in V} s_u$ for every $e \in E$. I refer to [DKR+08] for the proof of equivalence. In order to solve the resulting PCST problem [DKR+08] employ the published solution strategy by [LWP+06] which solves the problem via integer programming with lazy constraints.

[DKR+08] also contains a methodology with which their exact algorithm can produce *suboptimal* additional solutions (next to the optimal one) by re-solving under additional constraints which force the subnetwork to only share a certain percentage with the already found subnetworks. See subsection 3.1.6 in chapter 3 for an adaptation to the algorithms developed in this thesis. [DKR+08] furthermore provide a detailed statistical framework which explicitly spells out the underlying statistical assumptions of their model and its application. I refer to [DKR+08] for further details.

Zhao et al. 2008 [ZWCA08]: From receptor to target

Another early and interesting exact de-novo enrichment approach was formulated by [ZWCA08]. Given an undirected regulatory network $G = (V, E)$ the authors do not formulate a model in terms of node scores but instead work with edge scores $w : E \to \mathbb{R}$ which have to be engineered to reflect node-level changes. The primary example provided in [ZWCA08] is an edge score based on gene expression correlation between any two given genes. The other essential idea of [ZWCA08] in the context of exact de-novo enrichment is the idea of receptors and targets in biological pathways, i.e. the notion that many biological pathways implement a directed signal transmission capability from some kind of receptor to some kind of target. The most prominent example would be a signaling cascade, relaying signals from membrane receptors to transcription factors and their targets via its constituting molecular interactions.

Receptors and targets are called starting and ending nodes/genes in [ZWCA08] and given via sets $R \subset V$ and $T \subset V$. Note however, that the model of [ZWCA08] was formulated for undirected networks and thus cannot make optimal use of the implied directionality of the receptor/start and target/ending nodes in a network context. Additionally, the method of [ZWCA08] allows to specify a set $K \subset V$ of genes known to be contained in the to be identified de-novo subnetwork. [ZWCA08]'s model then introduces indicator variables for both nodes and edges, i.e. $x = (x_v)_{v \in V}$ and $Y = Y^T = (y_{uv})_{\{u,v\} \in V \times V}$. $x_v = 1$ means v is included in the de-novo subnetwork and $y_{uv} = 1$ means that $(u,v) \in E$ is included in the subnetwork. With edge score matrix $W = W^T = (w_{uv})_{(u,v) \in V \times V} \subset \mathbb{R}^{|V| \times |V|}$ the model can be formulated as follows:

$$\min_{x \in \{0,1\}^V, y \in \{0,1\}^{V \times V}} \quad -W \odot Y + \lambda ||Y||_F^2 \tag{2.1a}$$

$$\text{s.t.} \quad y_v^T e \leq x_v \quad \forall v \in V \tag{2.1b}$$

$$y^{(v)} e \leq x_v \quad \forall v \in V \tag{2.1c}$$

$$y^{(v)} e \geq 1 \quad \forall v \in R \cup T \tag{2.1d}$$

$$y^{(v)} e \geq 2x_v \quad \forall v \in V \setminus (R \cup T) \tag{2.1e}$$

$$x_v = 1 \quad \forall v \in R \cup T \cup K \tag{2.1f}$$

Here, $y^{(v)}$ denotes the row of Y which corresponds to $v \in V$ (as a row vector) and y_v denotes the column of Y corresponding to $v \in V$ (as a column vector) and $||Y||_F := \left(\sum_{v \in V} ||y_v||_2^2 \right)^{\frac{1}{2}}$ denotes the Frobenius norm of Y. $\lambda > 0$ denotes a penalty factor which penalizes edge inclusion. Furthermore, $W \odot Y := \sum_{v \in V} \sum_{u \in V} w_{uv} y_{uv}$ denotes the Hadamard product of of W and Y. The special role of receptors and targets in the subnetwork is now achieved by constraints (2.1d) and (2.1e) which simply enforce receptors and targets to be connected while requiring of all other included nodes to have at least two connections within the subnetwork. Note however that

the approach cannot really distinguish between receptors on one side and targets on the other. Additionally it requires receptors and targets to be predetermined and does not allow to select receptors and targets from a given set of potential receptors and targets. Regardless, the approach [ZWCA08] was the first to put forward the idea of semantically relevant special nodes (receptors/targets) in the context of exact de-novo enrichment methods.

From Steiner trees to forests [HF09, HCG+13, TBP+13, TGK+16]

[HF09, HCG+13] introduce de-novo subnetwork enrichment methods which are based on the Prize-collecting Steiner tree (PCST) problem as in [DKR+08]. While technically also based on the PCST the modeling angle is slightly different compared to [DKR+08]. Formally, [HF09, HCG+13] use the so called Goemans-Williamson formulation of the PCST problem [HCG+13] with an additional penalty factor $\beta > 0$[8]. This formulation is technically equivalent to the PCST problem from [DKR+08] up to the additional penalty factor β.

Definition 5 (PCST problem: extended Goemans-Williamson formulation [HCG+13]) *Given an undirected graph $G = (V, E)$, node scores $s : V \to [0, \infty)$ and edge scores $c : E \to (0, \infty)$, find a set of nodes $V' \subset V$ and edges $E' \subset E$ such that the subgraph (V', E') is connected and minimizes $e_{E'}^T c + \beta e_{V \setminus V'}^T s$.*

While the PCST problem which is solved in [DKR+08] stems from the formal reformulation of the MWCSP the PCST formulation is used by [HF09, HCG+13] as an explicit modelling approach. Note, that while c still represents *costs* associated with edge inclusion, the node score s corresponds to a penalty of not including a node into the solution and hence is semantically equivalent with the usual deregulation score. The difference here is that in contrast to maximizing the deregulation within subnetwork, [HF09, HCG+13] try to minimize the deregulation score of nodes which do not end up in the subgraph. The edge score c is modelled as a interaction confidence, i.e.

[8] [HF09] use the formulation without additional penalty factor β which is introduced by [HCG+13].

$c_e > c_{e'}$ means the confidence of edge $e' \in E$ is higher than that of $e \in E$ and hence inclusion of e in the solution is more costly. In terms of solution technology [HF09, HCG+13] also utilize the [LWP+06]. Akin to the strategy of [DKR+08] to find suboptimal subnetworks which could still carry biological significance, [TBP+13] extend the methods of [HF09, HCG+13] to the Prize-collecting Steiner forest (PCSF) approach which is able to find multiple deregulated de-novo subnetworks simultaneously.

Backes et al. 2012 [BRK+12]: Directed networks

[BRK+12] were one of the first to explicitly tackle the de-novo subnetwork problem for directed networks by solving extended versions of the (rooted) Directed Maximum Weight Connected Subgraph Problem (DMWCSP) and their work constitutes the most direct predecessor of the work presented in this thesis. Another important feature of the [BRK+12] formulation is the restriction to node-level decision variables in the integer programming model while most other approaches rely on both node and edge variables[9] like for example all exact methods summarized in the previous subsections [DKR+08, HF09, HCG+13, TBP+13, ZWCA08]. Introducing binary decision variables x and y the [BRK+12] model can then be formulated as follows:

$$\max_{x,y \in \{0,1\}^V} \quad s^T x \tag{2.2a}$$

$$\text{s.t.} \quad y \leq x \tag{2.2b}$$

$$e^T y = 1 \tag{2.2c}$$

$$e^T x = k \tag{2.2d}$$

$$x_v - y_v - e^T_{\delta^-(v)} x \leq 0 \quad \forall v \in V \tag{2.2e}$$

$$e^T_C (x - y) - e^T_{\delta^-(C)} x \leq |C| - 1 \quad \forall C \subset V \; ic, \; |C| > 1 \tag{2.2f}$$

[9] For most biomolecular networks the number of nodes is significantly less than the number of edges and hence integer programming formulations avoiding decision variables corresponding to edges can potentially lead to significantly smaller models.

Here, for subset of nodes $C \subset V$ I write $C\ ic$ if the subgraph induced by C contains a cycle with all nodes from C. While x corresponds to inclusion of a node in the subgraph, y decides on a particular *root node*. See the definition of Rooted Maximum Weight Connected Subgraph Problem (RMWCSP), definition 3. Constraints (2.2e) and (2.2f) together achieve the connectedness of the resulting subgraph. Since the constraints (2.2f) are exponentially many in terms of the size of the underlying graph [BRK+12] take these constraints into account as lazy constraints during the branch-and-cut procedure which solves the integer program. In order to alleviate the rather strict constraint of a fixed subgraph size $k \in \mathbb{N}$, [BRK+12] employ the strategy to find deregulated subgraphs for a range of subgraph sizes and take the union graph[10] as the resulting de-novo subnetwork.

Further exact methods

While the exact methods presented so far constitute the set of methods most relevant to the methods developed in this thesis, this short section collects some minimal review on further exact methods found in the literature. Gosline et al. 2012 [GSUF12] proposed *SAMNet*, while Atias and Sharan, 2013 [AS13] introduced *iPoint*, both exact methods based on network flow interpretations and formulations of the de-novo subnetwork detection problem.

An interesting aspect of the method of Gaire et al. 2013 [GSH+13], called *CASNet*, is its emphasis on the consistency of the deregulation measures mapped to the nodes of the network and the semantic meaning of the edges connecting deregulated nodes.

Building on the work of Melas et al. 2015 [MSI+15], Liu et al. 2019 [LTG+19] presented *CARNIVAL*, an ambitious model which not only tries to infer de-novo subnetworks but tries to do so by simultaneously inferring activation state of reactions (edges), whether a given interaction is inhibitory or activating given mapped deregulation scores, and more, while also being able to take into consideration experimentally

[10] Given a list of $n \in \mathbb{N}$ subgraphs $G_1 = (V_1, E_1), ..., G_n = (V_n, E_n)$ of some graph $G = (V, E)$ (i.e. $V_i \subset V$ and $E_i \subset E$ for all $i = 1, ..., n$) the *union graph* $\cup_{i=1}^{n} G_i$ of $(G_i)_i$ is defined as $(\cup_{i=1}^{n} V_i, \cup_{i=1}^{n} E_i)$.

applied perturbations. Note also, that CARNIVAL is based on a hypergraph model for the underlying regulatory network.

2.5 Summary and Discussion

This chapter reviewed the progression of methods for functional enrichment from gene set enrichment and topological pathway enrichment to de-novo subnetwork enrichment with a focus on exact approaches based on integer programming for the latter methodological paradigm. Table 2.1 summarizes some characteristics of the exact algorithms reviewed in this chapter. The next chapter introduces DeRegNet, an exact method for de-novo subnetwork enrichment and the main algorithmic framework developed in this thesis.

Ref.	Node var.	Edge var.	# variables	Stat. model	Lazy constr.	Directed
[DKR+08]	Yes	Yes	$\|V\|+\|E\|$	Yes	Yes	No
[HF09]	Yes	Yes	$\|V\|+\|E\|$	No	Yes	No
[HCG+13]	Yes	Yes	$\|V\|+\|E\|$	No	Yes	No
[TBP+13]	Yes	Yes	$\|V\|+\|E\|$	No	Yes	No
[ZWCA08]	Yes	Yes	$\|V\|+\|E\|$	No	No	No
[GSUF12]	No	Yes	$O(\|E\|+2\|V\|)$	No	No	Yes
[AS13]	Yes	Yes	$\|V\|+2\|E\|$	No	No	Yes
[GSH+13]	Yes	Yes	$\|V\|+2\|E\|$	No	No	Yes
[MSI+15]	Yes	Yes	$8\|V\|+3\|E\|$	No	No	Yes
[LTG+19]	Yes	Yes	$8\|V\|+3\|E\|$	No	No	Yes
[BRK+12]	Yes	No	$\|V\|$ or $2\|V\|$	Yes (*)	Yes	Yes
DeRegNet	Yes	No	$\|V\|$ or $2\|V\|$	Yes	Yes	Yes

Table 2.1: Exact de-novo subnetwork enrichment methods. The table shows some characteristics of the methods introduced in this chapter, as well as of DeRegNet, the method developed in this thesis starting with the next chapter. The columns *Node var.* and *Edge var.* denote whether a method employs decision variables corresponding to nodes and edges respectively. *# variables* gives the number of variables of the respective model formulations, while *Stat. model* logs whether the model has an accompanying statistical model or interpretation. *Lazy constr.* documents whether lazy constraints are necessary for the solution of the model and *Directed* finally summarizes whether the model makes/can make explicit use of the directionality of the interactions in given underlying regulatory network. (*): Introduced and specialized to [BRK+12] in chapter 3 of this thesis.

Chapter 3

A *de-novo* pathway discovery algorithm for omics data

This chapter develops an algorithm for de-novo pathway/subnetwork identification and describes its technical basis and general implementation. The algorithm and/or the implemented software are referred to as **DeRegNet**. For a very high-level overview, see figure 3.1. Methodologically, DeRegNet builds mainly upon the work of [BRK+12] and the prize-collecting Steiner tree methods proposed in [HF09, HCG+13, TBP+13, TGK+16] and also in [ZWCA08]. DeRegNet handles directed interaction networks and adapts from [BRK+12] the general integer programming approach in such a way that it can encapsulate the general idea of sources and targets as put forward in the prize-collecting Steiner tree/forest (PCST/PCSF) approaches [HF09, HCG+13, TBP+13, TGK+16]. The idea of receptors and terminals captures the idea of deregulated networks starting or ending at certain types of nodes, for example membrane receptors and transcription factors. More specifically, I extend the integer programming approach of [BRK+12] to fractional integer programming to allow for the necessary flexibility to incorporate sources and targets by means of variable subgraph size. I also show that DeRegNet can be seen as general maximum likelihood estimation with respect to a statistical model introduced below. The chapter now proceeds to describe a mathematical

optimization model which is at the heart of DeRegNet together with a motivating statistical model. Various extensions and application modes of the algorithm are explained. Naturally, the chapter continues detailing the solution methods for the DeRegNet model and finally closes with a description of a benchmark of DeRegNet comparing it to its closest methodological relative [BRK+12]. Applications of DeRegNet to actual omics data are presented in chapter 4.

3.1 Fractional integer-programming for finding deregulated subnetworks

3.1.1 The Maximum Average Weight Connected Subgraph Problem (MAWCSP)

Analogously to the directed (R)MWCSP (see definitions 2, 3 in chapter 2) one can define versions which strive to optimize the average score in the subgraph.

Definition 6 (Maximum Average Weight Connected Subgraph Problem (MAWCSP))
Given a directed graph $G = (V, E)$ and node scores $s : V \to \mathbb{R}$, find a set of nodes $V' \subset V$ whose induced subgraph (V', E') maximizes $\frac{e_{V'}^T s}{e^T e_{V'}}$ such that there is a node $r \in V'$ such that there is a directed path from r to every other node $v \in V'$.

Definition 7 (Rooted Maximum Average Weight Connected Subgraph Problem (RMAWCSP))
Given a directed graph $G = (V, E)$, node scores $s : V \to \mathbb{R}$, and a node $r \in V$ called the root node, find a set of nodes $V' \subset V$ with $r \in V'$ such that the induced subgraph (V', E') maximizes $\frac{e_{V'}^T s}{e^T e_{V'}}$ such that there is a directed path from r to every other node $v \in V'$.

The next subsection introduces the DeRegNet model for finding deregulated subnetworks in biomolecular networks. Mathematically speaking, the proposed model is an extended version of the just defined (R)MAWCS problem.

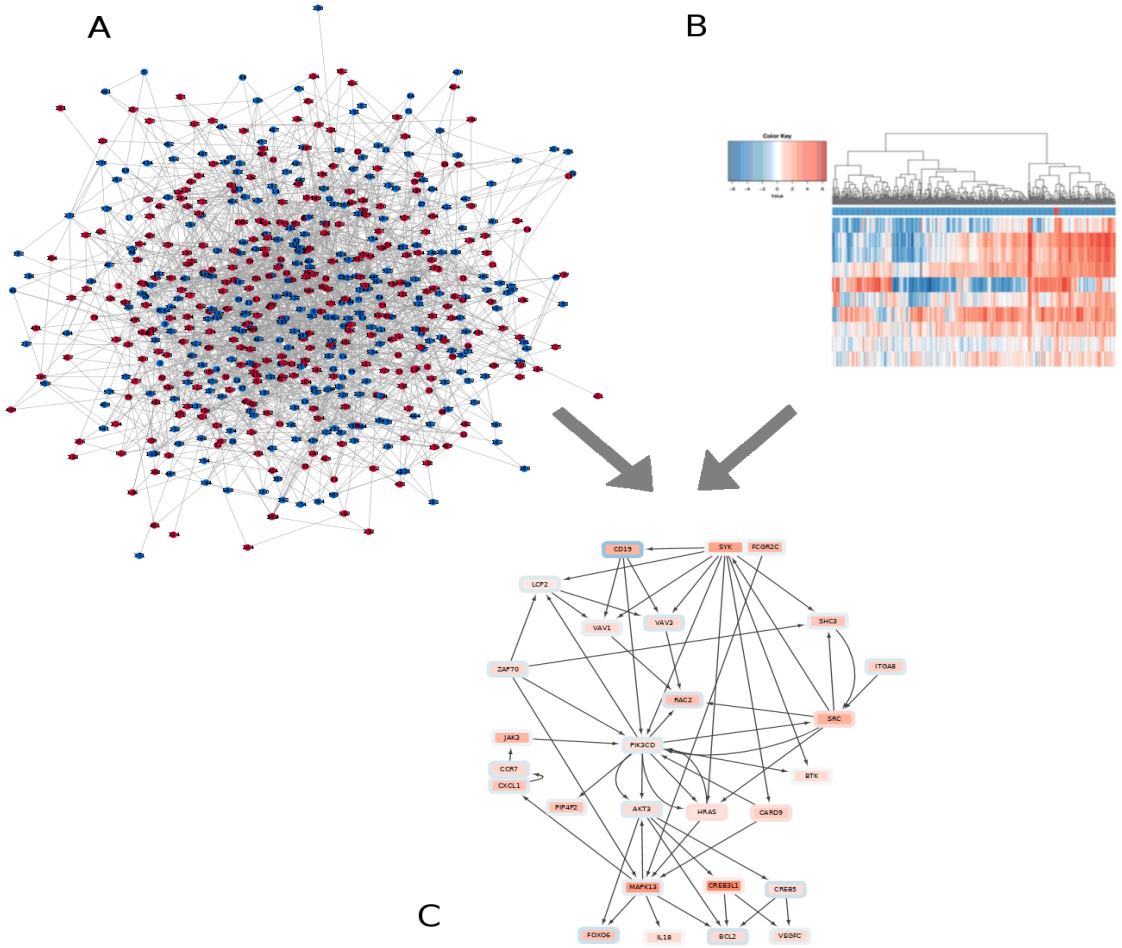


Figure 3.1: DeRegNet's inputs are a biomolecular network (A), such as a signaling or gene regulatory network, and omics measurements (B), such as gene expression data. The latter are mapped onto the nodes of the network acting as node-level measures of deregulation. This mapping is reflected here in (A) and (C) as the color of the nodes. (C) DeRegNet then extracts the most deregulated subnetwork from the larger regulatory network according to some definition of *most deregulated*

.

3.1.2 The DeRegNet fractional integer programming model

Given the generic setting and notation introduced in the previous chapter, I now formulate the main model of DeRegNet. Apart from the *directed* graph G and node scores s, there are given possibly empty subsets of nodes $S \subset V$ and $T \subset V$. It is referred to S as *sources* and to T as *terminals*, independent of the biological semantics underlying the definition of these sets (see below). The model is an extended version of the MAWCSP introduced above. Note, that in the following I formulate all problems as maximization problems while there are situations where minimization may, depending on the semantics of the node score[1], be the proper choice. As in [BRK+12], the problem of finding deregulated subnetworks is modelled in terms of indicator variables $x_v = \mathbf{I}(v \in V')$[2] and $y_v = \mathbf{I}(v \text{ is the root node})$ where $V' \subset V$ is a set of nodes inducing a subgraph such that one can reach every node in that subgraph by means of a directed path from the root node. In addition, the root is supposed to be a source node and all nodes in the subgraph with no outgoing edges are supposed to be terminal nodes. Furthermore, a subset of nodes $V' \subset V$ induces a strongly connected subgraph (V' *iscs*, for short) if

[1] Minimization may for example be prudent in case the node scores represent p values originating from some statistical significance test. Compare section 4.3.

[2] $\mathbf{I}(P) = 1$ if P, $\mathbf{I}(P) = 0$ if not P for some predicate P.

the subgraph induced by V' is strongly connected. The proposed model then reads as follows:

$$\max_{x,y \in \{0,1\}^V} \quad \frac{s^T x}{e^T x} \tag{3.1a}$$

$$\text{s.t.} \quad y \leq x \tag{3.1b}$$

$$e^T y = 1 \tag{3.1c}$$

$$k_{min} \leq e^T x \leq k_{max} \tag{3.1d}$$

$$x_v - y_v - e^T_{\delta^-(v)} x \leq 0 \quad \forall v \in V \tag{3.1e}$$

$$e^T_S (x - y) - e^T_{\delta^-(S)} x \leq |S| - 1 \quad \forall S \subset V \; iscs, \; |S| > 1 \tag{3.1f}$$

$$y_v = 0 \quad \forall v \in V \setminus R \quad \text{if } R \neq \varnothing \tag{3.1g}$$

$$x_v - e^T_{\delta^+(v)} x \leq 0 \quad \forall v \in V \setminus T \quad \text{if } T \neq \varnothing \tag{3.1h}$$

$$e^T_{\mathbf{Inc}} x = |\mathbf{Inc}| \tag{3.1i}$$

$$e^T_{\mathbf{Ex}} x = 0 \tag{3.1j}$$

The model derives from the corresponding integer linear programming model in [BRK+12] and adapts it for the fractional case, most notably here are the constraints involving the the receptors R (3.1g) and the terminals T (3.1h). (3.1g) ensures that the root node is a receptor[3], while (3.1h) ensures that any node in the subgraph with no outgoing edges is a terminal node. (3.1b) means that a node can only be the root if it is included in the subgraph. (3.1c) means that there is exactly one root. (3.1d) means that the size of subgraph has to be within the bound given by $k_{min}, k_{max} \in \mathbb{N}$. (3.1e) says that a node $v \in V$ in the subgraph is either the root node or there is another node $u \in V$ in the subgraph such that there is an edge $(u, v) \in E$. Moreover, the the constraints (3.1i) and (3.1j) trivially allow to include and exclude specific nodes from

[3]Of course, for practical implementation one can also just introduce variables $y_v \in \{0, 1\}$ only for nodes $v \in T$ in the first place. In terms of formulation one would need to make a difference for constraints (3.1e,f) as well and formulate them differently (with or without y) for nodes in R on the one hand and for nodes not in R on the other.

given sets **Inc**, **Ex** $\subset V$ respectively [4]. The constraint (1f) is the most involved one and describes (potentially) exponentially many constraints[5] which ensure that there are no disconnected directed circles by requiring that any strongly connected component in the subgraph either contains the root node or has an incoming edge from another node which is part of the subgraph but not part the given strongly connected component. Note, that one could merge (3.1e) and (3.1f) in terms of formulation by requiring $e_S^T(x-y)-e_{\delta^-(S)}^T x \leq |S|-1$ for all $S \subset V$ *iscs*, $|S| \geq 1$. Finally, the objective (3.1a) describes the notion of maximizing the average score of the subgraph. This is crucial for allowing the model the flexibility to connect source nodes to target nodes. I also give a probabilistic motivation of the the just described model, in particular its objective function, in the next subsection.

For the rest of this thesis I will use the following terminology in the context of the mathematical programming problem (3.1):

Definition 8 (DeRegNet instances, data, and subgraphs)
A tuple $(G, R, T, \textbf{\textit{Ex}}, \textbf{\textit{Inc}}, s)$ *is called an* ***instance of DeRegNet*** *(a* ***DeRegNet instance****, an* ***instance of the DeRegNet model****). Here,* $G = (V, E)$ *is the* ***underlying graph****,* $R \subset V$ *is the* ***receptor set****,* $T \subset V$ *is the* ***terminal set****,* ***Ex*** $\subset V$ *is the* ***exclude set****,* ***Inc*** $\subset V$ *is the* ***include set*** *and* $s : V \to \mathbb{R}$ *is the* ***node score*** *(the* ***score****). Further,* $x_v : V \to \{0,1\}$ *is called a* ***subgraph*** *with the understanding that it is referred to the subgraph of* G *induced by* $V^* = \{v \in V : x_v = 1\}$*. Equivalently to* $x_v : V \to \{0,1\}$*, it is also referred to the corresponding* $V^* = \{v \in V : x_v = 1\}$ *as a subgraph. A subgraph is* ***feasible for DeRegNet*** *(for the DeRegNet instance), if it satisfies DeRegNet's constraints (3.1b-j). A subgraph satisfying these constraints is called a* ***feasible subgraph****. A feasible subgraph which optimizes problem (3.1) is called an* ***optimal subgraph****.*

[4] In many situations, specific nodes, i.e. genes in the case of gene regulatory networks, may be of interest in other topological positions than in a receptor or terminal role. In that case just requiring a certain gene to be part of the subgraph without any special constraints on its inclusion in topological terms can be of value.

[5] In terms of the size of the underlying graph G.

In terms of the notation and exact formulation provided above, I will now proceed to formally specify certain topological characteristics of solutions of the above model which were casually asserted before. For similar proofs and also alternative formulations for the MWCSP it is referred to [BRK+12], [ÁMLM13b], [ÁMLM13a], [EKK14], [AB11]. I first formally recapture the defining topological feature of problems of (R)M(A)WCS flavour for DeRegNet.

Proposition 1

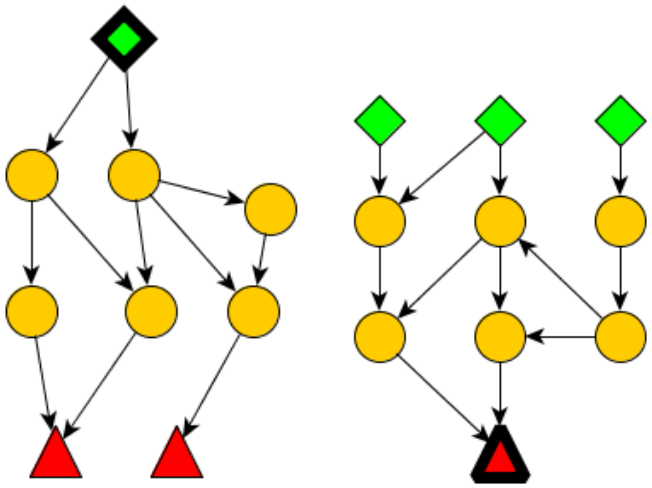
A feasible subgraph V^* of a DeRegNet instance has the property that any node in the subgraph can be reached from the root of the subgraph.

Proof. Any given node $v \in V^*$ of the subgraph is contained in a strongly connected component. By constraints (3.1e) and (3.1f) this strongly connected component either contains the root node or is reachable from some node $u \in V^*$ in the subgraph which is not in that strongly connected component: Let $S \subset V$ be the vertex set inducing the strongly connected component. If the root is not in S we have $e_S^T(x-y) = |S|$ and hence one has $e_{\delta^-(S)}^T x \geq 1$, otherwise one would have $e_S^T(x-y) - e_{\delta^-(S)}^T x \geq |S|$ in violation of constraints (3.1e) and (3.1f). If the root node is in S, it holds that $e_S^T(x-y) = |S|-1$ and hence constraints (3.1e) and (3.1f) always hold due to $e_{\delta^-(S)}^T x \geq 0$. In the case, that the root node is in v's component, v is reachable from the root node. In the case the component does not contain the root, repeat the argument with u instead of v. Again, the root is in the strongly connected component of u or the component is reachable from some $u' \in V^*$, and so on. Since the subgraph has a finite number of strongly connected components, one ultimately will encounter the component containing the root in the above argument which establishes the existence of a path to any arbitrary $v \in V^*$ from the root node. ■

The terminals from the terminal set T represent terminals of a subgraph in the following sense.

Proposition 2

A feasible subgraph V^* of a DeRegNet instance has the property that a node $v \in V^*$ in the



(a) **(b)**

Figure 3.2: Conceptual view of subgraphs extracted by DeRegNet. (a) From a receptor node/root node (green cube) one can reach any node in the subnetwork. Nodes without any edges leading to other nodes (red triangles) of the subnetwork need to be elements of the so called terminal nodes. Generally, all nodes in the subgraph can be reached from the root node. **(b)** By reversing the the orientation of the underlying network before applying DeRegNet, one can find subgraphs with only one terminal "root" node and multiple receptor nodes such that the terminal node can be reached from any other node in the subgraph. See subsection 3.1.4.

subgraph with $v \notin T$ has to have an outgoing edge into the subgraph, i.e. only terminal nodes are allowed to have no outgoing edges within the subgraph.

Proof. Given a non-terminal node $v \notin T$ one has constraint (3.1h): $x_v - e^T_{\delta^+(v)} x \leq 0$, i.e. if $x_v = 1$ it has to hold that $e^T_{\delta^+(v)} x \geq 1$. The latter inequality means that there exists another node $u \in V^*$ such that $(v,u) \in E$, E being the edge set of the underlying graph. ■

For a high-level conceptual depiction of the topology of subgraphs encapsulated in DeRegNet's constraints, see figure 3.2 (a).

3.1.3 Statistical interpretation for binary node scores

This subsection formalizes the notion that a *deregulated* subgraph satisfying given topological constraints should have higher/maximal probability of deregulation with respect to all possible subgraphs of that particular topological class. I present a basic probabilistic model yielding one possible formal probabilistic rationale for optimizing a model of the above form 3.1.2. Furthermore I provide a suitable interpretation of the model proposed in [BRK+12] in terms of that model, showing that DeRegNet solves a more general problem in the statistical sense necessitated by the probabilistic model introduced.

The model assumes binary node scores $s : V \to \{0,1\}$ which are realizations of random variables $\mathbf{S} = (S_v)_{v \in V}$. Further it is assumed the existence of a subset of vertices $V' \subset V$ such that $S_v | v \in V' \sim Ber(p')$ and $S_v | v \in V \backslash V' \sim Ber(p)$ with $p, p' \in (0,1)$ denoting probabilities of deregulation outside and inside of the deregulated subgraph respectively. It is assumed that $p' > p$ to reflect the idea of *higher* deregulation (probability) in the *deregulated* subgraph. The network context (dependency) is introduced via the restriction that $V' \in \mathscr{C}(V) \subset \mathscr{P}(V)$. Here, $\mathscr{C}(V)$ denotes the set of feasible substructures and should (can) reflect topologies inspired by known biomolecular pathway topologies like the one described in [BRK+12] and the last subsection. Furthermore it is assumed, that the (S_v), given a network context and deregulation probabilities p, p', are independent. Introducing the notation $\alpha(\tilde{V}) := |\{v \in \tilde{V} : S_v = 1\}|$ and considering V', p, p' to be parameters, and a subgraph determined by indicator variables x as outlined in the previous subsection, we can state:

Proposition 3

The log-likelihood $\mathscr{L}_s(\tilde{V}, p, p') = \log \mathbf{P}(\mathbf{S} = \mathbf{s} | V' = \tilde{V}, p, p')$ *under above model is given by:*

$$s^T x \log \frac{p'(1-p)}{p(1-p')} - e^T x \log \frac{1-p}{1-p'} + s^T e \log p + (e-s)^T e \log(1-p).$$

Proof.

$$\begin{aligned}\mathbf{P}(\mathbf{S}=\mathbf{s}|V'=\tilde{V},p,p') &= \prod_{v\in\tilde{V}}\mathbf{P}(S_v=s_v|V'=\tilde{V},p')\cdot\prod_{v\in V\setminus\tilde{V}}\mathbf{P}(S_v=s_v|V'=\tilde{V},p)\\ &= p'^{\alpha(\tilde{V})}(1-p')^{|\tilde{V}|-\alpha(\tilde{V})}p^{\alpha(V\setminus\tilde{V})}(1-p)^{|V\setminus\tilde{V}|-\alpha(V\setminus\tilde{V})}\end{aligned}$$

Employing decision variables $x_v = \mathbf{I}(v \in \tilde{V})$, we can write $\alpha(\tilde{V}) = s^T x, |\tilde{V}| = e^T x, \alpha(V\setminus\tilde{V}) = s^T(e-x)$ and $|V\setminus\tilde{V}| = e^T(e-x)$. It follows that the log-likelihood $\mathscr{L}_s(x,p,p') = \mathscr{L}_s(\tilde{V},p,p') = \log\mathbf{P}(\mathbf{S}=\mathbf{s}|V'=\tilde{V},p,p')$ can be written as:

$$\begin{aligned}\mathscr{L}_s(\tilde{V},p,p') &= s^T x\log p' + (e-s)^T x\log(1-p')\\ &\quad + s^T(e-x)\log p + (e-s)^T(e-x)\log(1-p)\\ &= s^T x\log\frac{p'(1-p)}{(1-p')p} - e^T x\log\frac{1-p}{1-p'} + s^T e\log p + (e-s)^T e\log(1-p)\end{aligned}$$

■

I call an optimization model maximizing the objective $s^T x$ subject to any constraints on x (the subgraph topology) a *model of Backes-type* [BRK+12]. Note that the DeRegNet model reduces to a Backes-type model in case of $k_{min} = k_{max}$[6].

Proposition 4

Any subgraph model of Backes-type enforcing a fixed subgraph size can be interpreted as maximum likelihood estimation with respect to subgraph structure given the above model.

Proof. Given the log-likelihood as determined by proposition 3, ignoring the constant term with respect to x, a maximum likelihood estimator V^* with respect to subgraph structure can be determined as follows:

[6] See DeRegNet model formulation 3.1.2

$$V^* \in \underset{\tilde{V}\subset\mathscr{C}(V)}{\operatorname{argmax}} \mathscr{L}_s(\tilde{V}, p, p') \tag{3.2}$$

$$= \underset{\tilde{V}\subset\mathscr{C}(V)}{\operatorname{argmax}} \left\{ s^T x \log \frac{p'(1-p)}{p(1-p')} - e^T x \log \frac{1-p}{1-p'} \right\} \tag{3.3}$$

$$= \underset{\tilde{V}\subset\mathscr{C}(V)}{\operatorname{argmax}} \left\{ s^T x \log \frac{p'(1-p)}{p(1-p')} \right\} \tag{3.4}$$

$$= \underset{\tilde{V}\subset\mathscr{C}(V)}{\operatorname{argmax}} s^T x \tag{3.5}$$

Here, equality (3 .4) follows from the assumption that the topological constraints of the optimization model enforce a constant subgraphs size (i.e. $e^T x = k$ for some fixed $k \in \mathbb{N}$). The last equality follows (by assumption $p' > p$) because $\log \frac{p'(1-p)}{p(1-p')} > 0$. Overall, a maximum likelihood estimator is given by a solution to a given Backes-type optimization model max $s^T x$ with subgraph topology restricted to subgraphs from $\mathscr{C}(V)$.

■

In particular, the specific model proposed by [BRK+12] lends itself to the just justified interpretation:

Corollary 1

The optimization model suggested by [BRK+12] can be interpreted as maximum likelihood estimation with respect to a subgraph structure of fixed size given the above probabilistic model.

I now proceed to provide a maximum likelihood interpretation for the DeRegNet model. Since the DeRegNet model does not assume a fixed subgraph size, above conclusions do not apply. Under the assumption that the parameter p is estimated external to the model and represents some general base level of deregulation one can by (conceptual) reduction from the full log-likelihood $\mathscr{L}_s(\tilde{V}, p, p')$ to $\mathscr{L}_s(\tilde{V}, p')$ state the following proposition.

Proposition 5

Solving a DeRegNet instance (see 3.1.2) amounts to maximum likelihood estimation under above model with respect to subgraph structure and deregulation probability p' (assuming $p' > 0$).

Proof. Given the log-likelihood as in proposition 3, one can differentiate with respect to p':

$$\frac{\partial}{\partial p'} \mathscr{L}_s(\tilde{V}, p, p') = \frac{\partial}{\partial p'} \mathscr{L}_s(\tilde{V}, p') \tag{3.6}$$

$$= \frac{\partial}{\partial p'} s^T x \log \frac{p'(1-p)}{(1-p')p} - \frac{\partial}{\partial p'} e^T x \log \frac{1-p}{1-p'} \tag{3.7}$$

By computing

$$\frac{\partial}{\partial p'} \log \frac{p'(1-p)}{(1-p')p} = \frac{\partial}{\partial p'} \log \frac{p'}{p} - \frac{\partial}{\partial p'} \log \frac{1-p'}{1-p} \tag{3.8}$$

$$= \frac{p}{p'} \cdot \frac{1}{p} - \frac{1-p}{1-p'} \cdot \frac{-1}{1-p} \tag{3.9}$$

$$= \frac{1}{p'} + \frac{1}{1-p'} \tag{3.10}$$

and

$$\frac{\partial}{\partial p'} \log \frac{1-p'}{1-p} = -\frac{1}{1-p'} \tag{3.11}$$

one obtains

$$\frac{\partial}{\partial p'} \mathscr{L}_s(\tilde{V}, p') = s^T x \frac{1}{p'} + s^T x \frac{1}{1-p'} - e^T x \frac{1}{1-p'} \tag{3.12}$$

Requiring $\frac{\partial}{\partial p'} \mathscr{L}_s(\tilde{V}, p'^*) = 0$ and with

$$\frac{\partial}{\partial p'} \mathscr{L}_s(\tilde{V}, p'^*) = 0 \Leftrightarrow \frac{1-p'^*}{p'^*} + 1 = \frac{e^T x}{s^T x} \tag{3.13}$$

$$\Leftrightarrow p'^* = \frac{s^T x}{e^T x} \tag{3.14}$$

and[7]

$$\frac{\partial^2}{\partial p'^2} \mathscr{L}_s(\tilde{V}, p') = s^T x \frac{-1}{p'^2} + s^T x \frac{1}{(1-p')^2} - e^T x \frac{1}{(1-p')^2} \leq -\frac{s^T x}{p'^2} < 0 \tag{3.15}$$

one arrives at

$$V^*_{MLE} \in \underset{\tilde{V} \subset \mathscr{C}(V)}{\operatorname{argmax}}\ p'^* = \underset{\tilde{V} \subset \mathscr{C}(V)}{\operatorname{argmax}}\ \frac{s^T x}{e^T x} \tag{3.16}$$

since no terms involving x were dropped in the derivation for p'^*.

■

The propositions of this subsection show, that, given the introduced statistical model, solving a DeRegNet instance instead of an instance of the optimization model proposed in [BRK+12] allows to carry out maximum likelihood estimation without the need to fix the subgraph size in advance. Given the assumptions of the model, these results hold regardless of further topological constraints and only relate to the respective objective functions.

3.1.4 Fixing the root node

Instead of the *root* being determined by the algorithm as outlined in the previous paragraph, one can also specify a given node $r \in V$ as root [BRK+12]. In this case, one

[7] Since $s^T x \leq e^T x$ and $e^T x > 0$ under the assumption that the subgraphs are constrained to have at least one node and $p' > 0$.

does not need the y variables anymore and, since the constraint logic can be carried over analogously, we can write the corresponding fractional integer problem as:

$$\max_{x \in \{0,1\}^V} \quad \frac{s^T x}{e^T x} \tag{3.17a}$$

$$\text{s.t.} \quad x_r = 0 \tag{3.17b}$$

$$k_{min} \leq e^T x \leq k_{max} \tag{3.17c}$$

$$x_v - e^T_{\delta^-(v)} x \leq 0 \quad \forall v \in V \setminus \{r\} \tag{3.17d}$$

$$e^T_S x - e^T_{\delta^-(S)} x \leq |S| - 1 \quad \forall S \subset V \; iscs, \; |S| > 1 \tag{3.17e}$$

$$x_v - e^T_{\delta^+(v)} x \leq 0 \quad \forall v \in V \setminus T \quad \text{if } T \neq \varnothing \tag{3.17f}$$

$$e^T_{\mathbf{Inc}} x = |\mathbf{Inc}| \tag{3.17g}$$

$$e^T_{\mathbf{Ex}} x = 0 \tag{3.17h}$$

Note, that the above formulation is a special case of the more general formulation of the previous section, namely $R = \{r\}$. It is nonetheless convenient to sometimes refer to the tuple $(G, r, T, \mathbf{Ex}, \mathbf{Inc}, s)$ as a *rooted DeRegNet instance*. All other terminology from the previous section carries over without modification.

3.1.5 Reversing the orientation

The default version of the just outlined algorithm will find subnetworks which possess a *root* node from which one can reach any other node in the subnetwork. This can be interpreted as the subnetwork being deregulated downstream of that root. As outlined in the previous sections, this root can either be determined by the algorithm or predetermined by biological curiosity or insight. By reversing the orientation of the graph one can easily obtain subnetworks where the *root* can be reached from any node in the subnetwork [BRK+12]. Such a subgraph can be interpreted as deregulated upstream of the either algorithmically determined or user-defined *root* node. In that case a

more intuitive name for the "root" is "terminal" or "destination". See Figure 3.2 for a visual comparison of the two scenarios. Formally, this difference in the structure of the output can be achieved by substituting the original graph G with the transposed graph $\tilde{G} = (V, \tilde{E})$, $\tilde{E} = \{(u, v) \in V \times V : (v, u) \in E\}$, and defining the models as before with the roles of receptors and terminals exchanged.

Definition 9

*A **reverse solution** of a DeRegNet instance $I = (G, R, T, \boldsymbol{Ex}, \boldsymbol{Inc}, s)$ with underlying graph $G = (V, E)$ is the (graph) transpose of an optimal subgraph of the DeRegNet instance $\tilde{\mathbf{I}} = (\tilde{G}, T, R, \boldsymbol{Ex}, \boldsymbol{Inc}, s)$. The latter is called the reverse instance of* $\mathbf{I}$*. Here, $\tilde{G}$ denotes the transposed graph of G, i.e. $\tilde{G} = (V, \tilde{E})$, $\tilde{E} = \{(u, v) \in V \times V : (v, u) \in E\}$.*

After the algorithm found subnetworks with respect to the reversed graph the resulting subnetworks have to be re-reversed to reflect physical reality. Also note, that the reversed instance exchanges the roles of receptors and terminal nodes to keep the intuitive and semantic notions associated with these terms in line with the topology of the just defined reverse solutions.

3.1.6 Extracting suboptimal subnetworks

Although the strategy to optimize seems like a sensible heuristic, it is nonetheless just an heuristic. There is no intrinsic need for a biological system at hand to behave consistently with this optimization objective in the sense that it is not granted that the patterns found by the algorithm actually correspond to what is biologically important in the given situation, even despite the outlined statistical model. Vice versa, something (nodes; a particular pattern of nodes) not showing up in any subgraph does not mean that they may not be important in the given context. While this cannot be mediated completely, it is sensible to find at least possible suboptimal patterns along with the optimal one. This can be seen as a step to capture mathematically speaking slightly less optimal but biologically potentially similarly or even more important patterns. I implement this notion by following the approach found in [DKR+08] and adapt it to

DeRegNet. Given a specified *maximal overlap* $\alpha \in [0,1)$ and a (induced) subgraph $V^* \subset V$ one adds to the models (3.1) or (3.2) the suboptimality constraint $e_{V^*}^T x \leq \alpha \cdot e^T x$ and reoptimizes, forcing any corresponding subgraph to be found to maximally have $100 \cdot \alpha$ % node overlap with the the nodes of the previously found subgraph. One can iterate this theme. For example, given a set of subgraphs $V^{(1)}, ..., V^{(k)}$ for some $k \in \mathbb{N}$ one can add the constraints $e_{V^{(j)}}^T x \leq \alpha \cdot e^T x$ for all $j = 1, ..., k$ to the DeRegNet instance to obtain a optimal subgraph of that modified DeRegNet instance which is guaranteed to have node overlap $\leq \alpha$ with any of the $V^{(j)}$. With $V^{(1)} = V^*$ being the original optimal subgraph of a DeRegNet instance one thus obtains a series of suboptimal subgraphs $V^{(2)}, ..., V^{(k)}$. The question which k to choose can be for example decided such that one chooses the k for which $\frac{e_{V^{(k+1)}}^T s}{|V^{(k+1)}|} < \beta \cdot \frac{e_{V^*}^T s}{|V^*|}$ for the first time for some $\beta \in [0,1]$. Here, β quantifies the degree of suboptimality one is willing to accept.

3.2 Solving the DeRegNet model

The model outlined in the preceding section are fractional integer programming problems. Generally, a **Fractional mixed-integer linear program (FMILP)** is an optimization problem of the following form:

$$\max \quad \frac{c^T x + d}{p^T x + q} \tag{3.18a}$$

$$\text{s.t.} \quad x \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} \tag{3.18b}$$

$$Ax \leq b \tag{3.18c}$$

Here, $c, p \in \mathbb{R}^n$, $d, q \in \mathbb{R}$ define the objective, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ define $m \in \mathbb{N}$ linear constraints and $n_c \in \mathbb{N}$, $n_i \in \mathbb{N}$ denote the number of continuous and discrete (integer) variables respectively. I assume w.l.o.g. $p^T x + q > 0$ for all feasible x[8].

[8]This condition (or that $p^T x + q < 0$ for all feasible x has to be decided for every instance. For DeRegNet instances the objective denominator is $e^T x > 0$ for a reasonable setting of $k_{min} > 0$, see 3.1.2. In general, the issue can also be decided by solving one of the MILPs 2.22 and 2.23 for determining lower and upper bounds on the objective denominator.

DeRegNet solves its integer fractional linear programming problems introduced in the previous sections by one out of two implemented methods. Firstly, a generalization of the Charnes-Cooper transformation [CC62] for fractional linear programs described by [YGGY13] and secondly an iterative scheme as introduced generally by Dinkelbach [Din62, Din67] and subsequently applied in the context of integer fractional programming by [Anz74, YCG09]. While the Dinkelbach-type algorithm solves the problem by iteratively solving certain non-fractional versions of the original problem until some convergence criterion is met, the generalization of the Charnes-Cooper method requires linearization of artificially introduced quadratic constraints through model reformulation. The linearization of quadratic constraints is implemented in terms of the methods described by [Glo75, AF05, AFG04].

As in [BRK+12] the exponentially many constraints forbidding any strongly connected components not containing the root and with no incoming edges from the subgraph are handled by lazy constraints. Every time a feasible integer solution which beats the current best lower bound of the optimal value is found, the Kosaraju–Sharir algorithm [Sha81] is employed (as implemented by the Lemon graph library [Lem]) to check for violating components and, in the case of violating components, the corresponding constraints are added to the model. Both solution approaches, the generalized Charnes-Cooper method and the Dinkelbach-type algorithm, allow for the lazy constraints to be handled in terms of the original model formulation since both retain the relevant variables of the model within the transformed model(s). In the following subsections I provide details on the relevant bits and pieces of the solution technology just outlined.

3.2.1 Dinkelbach-type algorithm

Originating in the 1960's [Din62, Din67] and studied more specifically in the context of FMILP problems [Anz74, YCG09] later on, the Dinkelbach algorithm relies on the iterative solution of linear problems only containing the original variables and an

auxiliary iteration parameter. *Algorithm 11* details the procedure. In the following, as well as in the entire thesis, *Dinkelbach algorithm* and *Dinkelbach-type algorithm* are used synonymously to refer to *Algorithm 2*.

Data: FMILP with feasible set $\mathscr{S}$
Result: solution x^* of FMILP
Initialization:
$\pi = \pi_0$ (parameter, has to be a lower bound of the optimal objective value)
$\epsilon > 0$ (termination tolerance)
$F = \infty$
while $F > \epsilon$ **do**
  $x^* = arg\,max\,\{c^T x + d - \pi\,(p^T x + q) : x \in \mathscr{S}\}$
  $F = c^T x^* + d - \pi\,(p^T x^* + q)$
  $\pi = \dfrac{c^T x^* + d}{p^T x^* + q}$
end
return x^*

Algorithm 2: Dinkelbach-type algorithm

The mixed-integer linear program appearing in the *while*-loop of *algorithm 2* is called a *Dinkelbach iteration problem*. Dinkelbach's algorithm iteratively solves a sequence Dinkelbach iteration problems until some convergence criterion is met. See Appendix A for more details on the general theory of Dinkelbach's scheme.

I now proceed proving that the fractional integer programming model for finding deregulated subgraphs proposed in the main (3.1.2) text can be solved via Dinkelbach's algorithm. The only points to clarify are the suitability of Dinkelbach's algorithm for models with lazy constraints, the suitability of an initial value for π of 0 and the positivity of the objective denominator (see Appendix A for an exposition of these technical requirements in general form).

Proposition 6 (Dinkelbach-type algorithm for DeRegNet)
The Dinkelbach algorithm is correct for the fractional integer programming problem of DeRegNet.

Proof. The first point to observe is that the objective of DeRegNet is always ≥ 0 hence the initialization condition of the iteration parameter $\pi = 0$ satisfies $\pi \leq \pi^*$. Fur-

thermore, for subgraphs which are constrained to contain at least one node, the denominator of the objective is strictly positive. These two properties are enough to guarantee convergence of Dinkelbach's algorithm as detailed in Appendix A. Also since the original decision variables are also part of the parameterized Dinkelbach iteration problems introducing lazy constraints is technically feasible. Since lazy constraints can only decrease the maximum objective, after every iteration $\pi \leq \pi^*$ where π^* is the optimal objective determined by the current constraints and hence lazy constraints do not interfere with the correctness of Dinkelbach's algorithm since it requires a starting value of π which is a lower bound of the optimal objective value (see appendix A). ■

Note that lazy constraints effectively amount to restarting Dinkelbach's algorithm (in a valid initialization state) every time a lazy constraint is added. Hence, convergence can also only be considered superlinear (see appendix A) with respect to the current optimal objective determined by the lazy constraints.

For more details on the theoretical underpinnings of Dinkelbach's method the reader is referred to appendix A.

3.2.2 Generalized Charnes-Cooper transformation

The so called Generalized Charnes-Cooper transformation [YGGY13] described in this subsection derives its name and general idea from the classical Charnes-Cooper transformation [CC62] used to solve continuous fractional linear problems (FLPs). Consider the above general form of a FMILP in the following slightly more detailed format:

$$\max \quad \frac{c_c^T x_c + c_i^T x_i + d}{p_c^T x_c + p_i^T x_i + q} \tag{3.19a}$$

$$\text{s.t.} \quad x = \begin{pmatrix} x_c \\ x_i \end{pmatrix} \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} \tag{3.19b}$$

$$A_c x_c + A_i x_i \leq b \tag{3.19c}$$

where we explicitly decomposed the variable x into its continuous and integer parts $x_c \in \mathbb{R}^{n_c}$ and $x_i \in \mathbb{Z}^{n_i}$. Analogously we have $c = \begin{pmatrix} c_c \\ c_i \end{pmatrix}$ with $c_c \in \mathbb{R}^{n_c}$, $c_i \in \mathbb{R}^{n_i}$, and $p = \begin{pmatrix} p_c \\ p_i \end{pmatrix}$ with $p_c \in \mathbb{R}^{n_c}$, $p_i \in \mathbb{R}^{n_i}$, and $A = \begin{pmatrix} A_c & A_i \end{pmatrix}$ with $A_c \in \mathbb{R}^{m \times n_c}$, $A_i \in \mathbb{R}^{m \times n_i}$. As detailed in [YGGY13] one can now define additional variables $u := \frac{1}{p_c^T x_c + p_i^T x_i + q}$ and $z := \frac{x_c}{p_c^T x_c + p_i^T x_i + q} = ux$. Note, that $u > 0$ by assumption. After incorporating the definition of u as a further constraint and multiplying all original constraints with u one arrives at the following quadratic mixed-integer problem:

$$\max \quad c_c^T z + c_i^T (u \cdot x_i) + d \tag{3.20a}$$

$$\text{s.t.} \quad x_i \in \mathbb{Z}^{n_i},\ z \in \mathbb{R}^{n_c},\ u \in \mathbb{R}_+ \tag{3.20b}$$

$$p_c^T z + p_i^T (u \cdot x_i) + qu = 1 \tag{3.20c}$$

$$A_c z + A_i (u \cdot x_i) - bu \leq 0 \tag{3.20d}$$

Note that the above problem is not a mixed-integer *linear* program (MILP) but a quadratic mixed-integer problem due to the terms ux_i in the transformed constraints. This is addressed in the next subsection. With the notation of this subsection one can formulate the following propositions formalizing the equivalence of the two model formulations [YGGY13][9]:

Proposition 7 (Feasible points of the generalized Charnes-Cooper transform)
A point (x_c, x_i) is a feasible solution of problem (A.30) if and only if (z, x_i, u) is a feasible solution of problem (A.31).

Proposition 8 (Equivalence of solutions of the generalized Charnes-Cooper transform)
An feasible point (x_c^*, x_i^*) of (A.30) is optimal if and only if (z^*, x_i^*, u^*) is optimal for (A.31). It holds that $z^* = u^* x_c^*$ and $u^* = \frac{1}{p_c^T x_c^* + p_i^T x_i^* + q}$.

[9] Their proofs follow directly from the definition of the transformation.

With respect to lazy constraints involving the integer variables x_i there do not arise any complications since they are part of both problem formulations. Lazy constraints for the continuous variables x_c require more care due to the necessity to transform the constraints correspondingly. The DeRegNet model does only contain integer (in fact, binary) variables and hence it is straight-forward to incorporate lazy constraints in the solution process in terms of the original model formulation.

Linearization of binary-continuous quadratic constraints

In contrast to the iterative Dinkelbach scheme, the reformulation-linearization method described in the last section relies on the linearization of products of integer and continuous variables. Since we only deal with binary variables in this paper, we assume from now on that all integer variables are in fact binary.[10] While there exist variations on the theme of linearization [AFG04], [AF05], I here present the implemented version going back to [Glo75].

Given a continuous variable $v \in \mathbb{R}$ and a binary variable $x \in \{0,1\}$ one introduces a third (continuous) variable $z \in \mathbb{R}$ corresponding to $z = vx$ and substitutes any appearance of the product vx with z. Along with z one introduces the following constraints to ensure equivalence:

$$\begin{aligned} z &\leq Ux \\ z &\geq Lx \\ v - U(1-x) &\leq z \\ v - L(1-x) &\geq z \end{aligned} \tag{3.21}$$

[10] In case of a proper integer variable $x \in D \subset \mathbb{Z}$, one can introduce auxiliary binary variables $x'_d \in \{0,1\}, d \in D$ with $x = \sum_{d \in D} d \cdot x'_d$ and $\sum_{d \in D} x'_d = 1$ in order to transform its product with continuous variables into a sum of products between binary and continuous variables.

Here, $U \in \mathbb{R}$ is an upper and $L \in \mathbb{R}$ is a lower bound of v which are either given by the problem formulation itself, can be inferred from manual insight into the problem or by solving a certain MILP in some cases.

Proposition 9 (Linearization binary-continuous products)
Let $v \in \mathscr{S} \subset \mathbb{R}$ with bounded $\mathscr{S}$ and let $x \in \{0,1\}$ and $z \in \mathbb{R}$. Furthermore $U \geq \sup \mathscr{S}$ and $L \leq \inf \mathscr{S}$. Then, the constraints 3.21 are satisfied if and only if $z = vx$.

Proof. Let $z = vx$, then $z = vx \leq Ux$ since U is an upper bound of v and $z = vx \geq Lx$ since L is a lower bound of v. Also for the case $x = 1$ one has $v - U(1-x) = v = vx = z$ and $v - L(1-x) = v = vx = z$ and for the case $x = 0$ the two constraints $v - U(1-x) \leq z$ and $v - L(1-x) \geq z$ reduce to $v \leq U$ and $v \geq L$ respectively which is true by assumption. Conversely, let the constraints in (A.15) be satisfied. Then in the case $x = 1$, the constraints $v - U(1-x) \leq z$ and $v - L(1-x) \geq z$ imply $v \leq z \leq v$ and hence $z = v = vx$. In the case $x = 0$ the first two constraints of (A.15) imply $z = 0 = vx$. ■

The lower bound L and the upper bound U can generally be obtained by solving suitable MILPs [YGGY13] involving the denominator of the original objective. To obtain the (tightest possible) lower bound one can solve the following problem:

$$\max \quad p_c^T x_c + p_i^T x_i + q \tag{3.22a}$$

$$\text{s.t.} \quad x = \begin{pmatrix} x_c \\ x_i \end{pmatrix} \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} \tag{3.22b}$$

$$A_c x_x + A_i x_i \leq b \tag{3.22c}$$

Analogously, to obtain the (tightest possible) upper bound one can solve the following minimization problem:

$$\min \quad p_c^T x_c + p_i^T x_i + q \tag{3.23a}$$

$$\text{s.t.} \quad x = \begin{pmatrix} x_c \\ x_i \end{pmatrix} \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} \tag{3.23b}$$

$$A_c x_x + A_i x_i \leq b \tag{3.23c}$$

Note however, that any lower and upper bound would work. The trade-off between less tight bounds on the denominator variable and the necessity of solving up to two MILPs up front has to be decided for every FMILP model to be solved with the generalized Charnes-Cooper transformation.

In case of DeRegNet, lower and upper bound on the objective denominator are explicitly set in the problem formulation in the form of minimal and maximal subgraph size. Hence one does not have to solve any MILPs up front and has (optimal) lower and upper bounds for the inverse denominator readily available due to the problem formulation.

Caching transformed model formulations

For DeRegNet's use cases it is quite common to optimize DeRegNet instances which just differ in terms of their node scores, i.e. share the same underlying graph. For example, finding deregulated subgraphs for individual cases in a TCGA cohort with a fixed regulatory network derived from KEGG will require to solve a model with the same structural properties but with differing score data[11]. In particular, in such a situation the reformulation and linearization procedure of the generalized Charnes-Cooper transform only has to be carried out once and can be reused across cases

[11] For example a omics-readout for every case in the cohort. See chapter 4.

since it does not depend structurally on the objective data vector s. While solution time of a DeRegNet instance with the generalized Charnes-Cooper transform tends to be dominated by the time to solve the resulting integer linear program, reuse of the transformed model structure can nonetheless result in significant computational savings.

3.2.3 Lazy constraints in branch-and-cut MILP solvers

For reference, this section contains an high-level outline of how lazy constraints fit into branch-and-cut algorithms for solving mixed-integer programs. For a general introduction to branch-and-cut for MILPs it is referred to [CCZ14] and [CBD10].
Let a MILP with $n_c \in \mathbb{N}$ continuous and $n_i \in \mathbb{N}$ integer variables of the following form be given:

$$\begin{aligned}
\max \quad & c^T x + d^T y && \text{(3.24a)}\\
\text{s.t.} \quad & x \in \mathbb{R}^{n_c} && \text{(3.24b)}\\
& y \in \mathbb{Z}^{n_i} && \text{(3.24c)}\\
& Ax + By \leq b && \text{(3.24d)}\\
& x, y \geq 0 && \text{(3.24e)}
\end{aligned}$$

Here $c \in \mathbb{R}^{n_c}$, $d \in \mathbb{R}^{n_i}$, $A \in \mathbb{R}^{m \times n_c}$ and $B \in \mathbb{R}^{m \times n_i}$ for some $m \in \mathbb{N}$. The *(natural) linear programming relaxation* of a MILP of the above form is the following:

$$\begin{aligned} \max \quad & c^T x + d^T y && \text{(3.25a)} \\ \text{s.t.} \quad & x \in \mathbb{R}^{n_c} && \text{(3.25b)} \\ & y \in \mathbb{R}^{n_i} && \text{(3.25c)} \\ & Ax + By \leq b && \text{(3.25d)} \\ & x, y \geq 0 && \text{(3.25e)} \end{aligned}$$

Lazy constraints are constraints which are not initially explicitly part of the model formulation, the reason usually being that it would require an computationally infeasible exponential number of constraints (with respect to the number of variables). On a relatively high level of abstraction, the classical branch-and-cut strategy for solving MILPs with lazy constraints can be formulated as detailed in algorithm 3.
In order for branch-and-cut with lazy constraints to be computationally at least potentially feasible, the determination of the violated lazy constraints for any feasible solution should be a efficient/polynomial time subroutine. The details for the separation routine for the DeRegNet model are provided in the next section.

3.2.4 Lazy constraints for the DeRegNet model

For DeRegNet the lazy constraint separation subroutine centers around finding the strongly connected components of the given solution. This is generally considered an efficiently solvable problem.

Interlude: Strongly connected components

Given a directed graph $G = (V, E)$ one says that G is strongly connected if and only if there is a directed path from every node $v \in V$ to every other node $u \in V$. A

Data: MILP and lazy constraints
Result: Solution of MILP satisfying any lazy constraint
Initialization:
$\mathscr{L} = \{\text{MILP}\}$ (Set of MILP problems in search tree)
$\underline{z} = -\infty$ (Current best lower bound for optimal objective)
$(x^*, y^*) = (null, null)$ (Current best feasible solution)
while $\mathscr{L} \neq \emptyset$ **do**
  Choose P from $\mathscr{L}$ and remove P from $\mathscr{L}$
  (*) Solve linear programming relaxation of P
  Let z be the solution value and (x, y) be the solution of the relaxation
  if $z > \underline{z}$ **then**
    if (x, y) *feasible for* P **then**
      Find the set **V** of violated lazy constraints by *seperation subroutine* (Δ)
      if $\mathbf{V} = \emptyset$ **then**
        $(x^*, y^*) := (x, y)$
        $\underline{z} := z$
        Remove all subproblems from $\mathscr{L}$ with optimal solution value $< \underline{z}$
      end
      else
        Insert P back into $\mathscr{L}$
        Add lazy constraints from **V** to models in $\mathscr{L}$
      end
    end
    else
      if *you want to add cuts* **then**
        Add cuts and GOTO (*)
      end
      else
        Branch and add created subproblems to $\mathscr{L}$
      end
    end
  end
end
return (x^*, y^*), $\underline{z}$

Algorithm 3: Branch-and-cut for MILPs with lazy constraints [CCZ14, BRK+12]

strongly connected component of a directed graph is any maximal subgraph which is strongly connected[12]. Sometimes one refers to $V' \subset V$ as inducing a strongly connected component if the subgraph induced by V' is a strongly connected component. One denotes the set of node sets inducing all strongly connected components of a graph $G = (V, E)$ by $\mathbf{SCC}(G) \subset \mathscr{P}(V)$. The three classical algorithms which can be used to solve the problem of finding a directed graph's strongly connected components in $O(|V|+|E|)$ time are the Kosarju-Sharir algorithm [Sha81], Tarjan's algorithm [Tar72] and variants of the path-based strong component algorithm [Dij72]. A strongly connected *subgraph* (in contrast to *component*) is a subgraph of a graph which is strongly connected.

Lazy constraint separation subroutine of DeRegNet

This subsection and algorithm 4 provide the details on the lazy constraint separation subroutine employed for the solution of the DeRegNet model. The formal details are given as algorithm 4. In short, given a (potential) incumbent solution to a DeRegNet instance not containing all strong-component constraints 3.1f, the subroutine finds the strongly connected components of the corresponding subgraph and checks whether any such component either contains the root node itself or has at least one incoming edge from within the subgraph but from outside the component. If so, the (potential) incumbent is feasible, hence an actual incumbent solution. Otherwise the violated constraint is added to the model in while the (potential) incumbent is declared infeasible.

3.2.5 Primal heuristics for the DeRegNet model

Every feasible solution of a mixed-integer program provides a lower bound on the optimal solution value (for maximization problems). The feasible solution which currently gives the best lower bound on the optimal value during a branch-and-bound procedure is called the *incumbent (solution)*. Branch-and-bound (and hence branch-and-cut) for mixed-integer programs relies on pruning parts of the search tree of LP

[12] I.e. adding any node not in the subgraph would render the resulting subgraph to be not strongly connected anymore.

Data: DeRegNet instance and $x, y : V \to \{0,1\}$
Result: *True* if x and y do not violate any lazy constraints, *false* otherwise
$V^* = \{v \in V : x_v = 1\}$ (nodes implied by x)
$G^* = (V^*, E^*)$ the subgraph induced by V^*
$\mathscr{C} = \mathbf{SCC}(G^*)$, $\mathscr{C} \in \mathscr{P}(V^*)$ (Find strongly connected components)
for *C in $\mathscr{C}$ with $|C| > 1$* **do**
  if $e_C^T(x-y) - e_{\delta^-(C)}^T x > |C| - 1$ **then**
    return *false*
  end
end
return *true*

Algorithm 4: Lazy constraint subroutine for DeRegNet. In case a potential incumbent is found all strongly connected components are checked to assess feasibility. In case any strongly connected component does not contain the root node and has no incoming edges from another component, a (lazy) constraint enforcing the requirement is added. $\mathbf{SCC}(G)$ denotes the set of all strongly connected components of a given graph G.

relaxation subproblems by assessing whether the optimal solution value of a given LP relaxation is less than the best lower bound provided by the incumbent. See algorithm 3 as a reference. Primal heuristics [Ber06] aim at finding and/or improving feasible solutions during a branch-and-bound procedure. While some generic methods for primal heuristics exist [GM97a], [GM97b], [FGA05], [BSW04], [BM80], they tend to be highly problem-specific [Ber06]. Of special interest in the context of DeRegNet are primal heuristics for the MWCSP [RK19], [ÁMLM13a], [ÁMLM13b]. In the following I describe start and improvement heuristics useful during the solution of DeRegNet instances.

Start heuristics

A priori there is no feasible solution known at the beginning of a branch-and-bound procedure for solving a mixed-integer program. Heuristics which try to find initial feasible solutions are called *start heuristics*. I outline two start heuristics which can be employed at the beginning of the branch-and-bound search for the solution of the

DeRegNet model.

Greedy start heuristic. The first start heuristic is called *greedy start heuristic* and basically starts with the highest scoring node and greedily adds neighbors of already added nodes until the average score of the thus defined subgraph starts decreasing. If the currently selected subgraph is feasible upon termination, one has found a feasible solution. The formal procedure is outlined in algorithm 5. There are a number subtleties attached to this start heuristic. First and foremost the procedure only assures the reachability constraints regarding the root node. Most other constraints may or may not be satisfied at any given time during the procedure, mostly: subgraph size constraints and constraints ensuring the necessity of leaf nodes to be from the subset of terminal nodes. While the subgraph size constraint is relatively easily manageable by stopping the procedure when the maximal subgraph size is reached and by restarting in case the minimal subgraph size can not be achieved in the first place. In the latter case, one can restart the procedure from the best scoring node not already selected during earlier attempts of the greedy start heuristic. The issue of the terminal node constraints is not easily handled and hence the greedy start heuristic is in effect only usable in case $T = \emptyset$. Also instances with $\mathbf{Inc} \neq \emptyset$ cannot be handled by this heuristic.

Receptor-terminal shortest path heuristic. The second start heuristic is more suitable in situations where there is a non-empty terminal set T. In short, it finds the shortest path between a pair of receptor and terminal nodes with high node scores. The **SHORTEST_PATH** subroutine referenced in algorithm 6 can be an implementation of any of the canonical algorithms to find single-source shortest paths with unit edge weights in directed graphs in polynomial time [Dij59], [Joh77], [AMOT90]. Subject to $\mathbf{Ex} = \mathbf{Inc} = \emptyset$ all connectivity constraints will be satisfied by construction. If the subgraph size constraints are met is up to chance however. Again, running multiple times with the, say K, highest scoring pairs of receptors and terminals, can help in this situation. Note, that the restriction of $\mathbf{Ex} = \mathbf{Inc} = \emptyset$ could be lifted by formulat-

Data: DeRegNet instance with $T = \mathbf{Inc} = \emptyset$
Result: Feasible solution of DeRegNet instance **or** *null*
if $R \neq \emptyset$ **then**
  $V_I = R$
end
else
  $V_I = V \setminus \mathbf{Ex}$
end
$v^* = \text{argmax}_{v \in V_I} s_v$ (Select feasible root with highest score)
$V^* = \{v^*\}$ (Selected DeRegNet solution)
$N = \delta^+(v^*) \setminus \mathbf{Ex}$ (Candidate nodes to be potentially added next)
$A^* = s_{v^*}$ (Current average score of selected subgraph)
CONTINUE = **true**
while *CONTINUE* ***and*** $|V^*| < k_{max}$ **do**
  $v^* = \text{argmax}_{v \in N} s_v$ (Highest scoring node in candidate set)
  if $s_{v^*} \geq A^*$ **then**
    $A^* = \frac{|V^*|A^* + s_{v^*}}{|V^*|+1}$ (Update average score of selected subgraph)
    $V^* = V^* \cup \{v^*\}$ (Update current subgraphs)
    $N^* = (\delta^+(v^*) \setminus V^*) \setminus \mathbf{Ex}$ (New candidate nodes)
    $N = (N \setminus \{v^*\}) \cup N^*$ (Update candidate nodes)
  end
  else
    CONTINUE = **false**
  end
end
if V^* *feasible* **then**
  return V^* (Return feasible solution of DeRegNet instance)
end
else
  return *null* (Return nothing to indicate failure to find feasible solution)
end

Algorithm 5: Greedy start heuristic for the DeRegNet model

ing the corresponding shortest path problem by canonical means in terms of integer programming problems [Tac16]. This possibility is not explored further however since solving integer programs to get initial feasible solutions to integer program may be a slippery slope. Especially in the case of DeRegNet where the main problem to solve is formulated in terms of decision variables corresponding to nodes while shortest path integer programming formulations usually introduce decision variables corresponding to the edges of the graph.

Data: DeRegNet instance with $\mathbf{Ex} = \mathbf{Inc} = \emptyset$
Result: Feasible solution of DeRegNet instance **or** *null*
if $R \neq \emptyset$ **then**
  $V_R = R$
end
else
  $V_R = V$
end
$r^* = \text{argmax}_{v \in V_R} s_v$ (Receptor with highest score)
if $T \neq \emptyset$ **then**
  $V_T = T$
end
else
  $V_T = V$
end
$t^* = \text{argmax}_{v \in V_T} s_v$ (Terminal with highest score)
$V^* = \{r^*, t^*\}$ (Selected DeRegNet solution)
$P =$ **SHORTEST_PATH**(G, r^*, t^*) (Find shortest path between receptor and terminal)
$V^* = V^* \cup P$ (Add nodes from shortest path)
if $k_{min} \leq |V^*| \leq k_{max}$ **then**
  return V^* (Return solution if it satisfies the subgraph size constraints)
end
else
  return *null* (Return nothing if size constraints are not met)
end

Algorithm 6: Receptor-terminal shortest path start heuristic for the DeRegNet model

Improvement heuristics

In case a feasible solution is found at a particular branch-and-bound node (which may be a new incumbent or not), heuristics which try to improve that given feasible solution are called *improvement heuristics*. Here I describe a simple greedy improvement heuristic which can be applied to any feasible solution, either found during the branch-and-cut procedure or otherwise. It works analogously to the greedy start heuristic (algorithm 5), the only difference being that one is already starting with a feasible solution. In particular, the heuristic can be applied to solutions constructed by the receptor-terminal shortest path start heuristic (algorithm 6) described in the previous section. Trying to improve the greedy start heuristic (algorithm 5) with the improvement strategy outlined below is futile however since by construction the former already added all potential subgraph nodes in a greedy fashion. During a branch-and-cut run any new feasible solution can potentially be improved by the heuristic. In case of an incumbent one can hope for an even better incumbent, in case of a feasible solution one can hope to improve it up to a point where it actually becomes a new incumbent. The description of the heuristic is provided as algorithm 7.

3.2.6 Approximate solutions via branch-and-bound gap cut

One can use a mixed-integer programming solver generically to obtain suboptimal solutions to a given (maximization) MILP with optimal objective value z^*. During the branch-and-cut search one obtains lower bounds on the optimal value by feasible solutions to the problem and an upper bound by the solution value of the initial LP relaxation of the problem. Let $\underline{z} \leq z^*$ be the best available lower bound and let $\bar{z} \geq z^*$ be the upper bound obtained by the relaxed problem. The *relative gap* λ_{rel} during a branch-and-cut search is defined as $\lambda_{rel} := \frac{z^*}{\bar{z}}$.[13] With the upper bound $\hat{\lambda}_{rel} := \frac{\bar{z}}{\underline{z}} \geq \frac{z^*}{\bar{z}}$ on the gap it follows that $z^* \leq \hat{\lambda}_{rel}\underline{z}$ and hence $\alpha z^* \leq \underline{z}$ with $\alpha := \hat{\lambda}_{rel}^{-1}$. Stopping the branch-and-cut procedure at the given gap upper bound value hence provides an

[13] While the *(absolute) gap* λ_{abs} is defined as $\lambda_{abs} := \bar{z} - z^*$.

Data: Feasible solution of a DeRegNet instance
Result: Another feasible solution of (the same) DeRegNet instance
$V^* = \{v \in V : x_v = 1\}$ (Selected DeRegNet solution)
$N = (\bigcup_{v \in V^*} \delta^+(v)) \setminus (V^* \cup \mathbf{Ex})$ (Candidate nodes to be added next)
$A^* = \frac{s^T x}{e^T x}$ (Current average score of selected subgraph)
CONTINUE = **true**
while *CONTINUE* ***and*** $|V^*| < k_{max}$ **do**
  $v^* = \text{argmax}_{v \in N} s_v$ (Highest scoring node in candidate set)
  if $s_{v^*} \geq A^*$ **then**
    $A^* = \frac{|V^*|A^* + s_{v^*}}{|V^*|+1}$ (Update average score of selected subgraph)
    $V^* = V^* \cup \{v^*\}$ (Update current subgraphs)
    $N^* = (\delta^+(v^*) \setminus V^*) \setminus \mathbf{Ex}$ (New candidate nodes)
    $N = (N \setminus \{v^*\}) \cup N^*$ (Update candidate nodes)
  end
  else
    CONTINUE = **false**
  end
end
return V^*

Algorithm 7: Greedy improvement heuristic for the DeRegNet model

approximate solution of a posteriori approximation guarantee of $\hat{\lambda}_{rel}^{-1} \leq 1$. I refer to the strategy of stopping the branch-and-cut search once the gap upper bound is below a certain threshold as *gap cut* or *gap (cut) thresholding*. Employing the gap cut strategy can be useful in situations where the MILP solver can find reasonably good solutions in reasonable time but would take significantly more time to find the optimal solution. The option of to carry out gap cut thresholding is incorporated in the implementation of DeRegNet for this very reason.

3.2.7 Software for solving fractional integer programs: libgrbfrc

In order to solve the fractional integer programs formulated above, a C++ library based on the commercial Gurobi solver [Gurb] was implemented. *libgrbfrc* [Lib] implements the two solution methods from above: Dinkelbach's algorithm and the generalized Charnes-Cooper transform. Due to the requirements of the developed optimization

models the implementations support lazy constraints. Academic licenses for Gurobi are readily obtained [Gura].

3.2.8 Implementation and availability of DeRegNet

DeRegNet's implementation is written in C++ and Python and utilizes the Gurobi optimization library [Gurb] and the Lemon graph library [Lem]. The software is open source under BSD-3-Clause OSI-approved license [BSD] and is available at [DeRc] where you can also find installation instructions and usage examples. The algorithm can be run either by using a Python package or a command line tool. Furthermore, I implemented an easy-to-use Docker image (*sebwink/deregnet* available at Docker Hub [Docc]) which bundles all necessary dependencies. Currently, in order to run DeRegNet, a license for the Gurobi optimization library is needed. For academic purposes these licences are readily obtained at [Gura].

3.3 Benchmarking DeRegNet

The evaluation and benchmarking of *de-novo* pathway enrichment or deregulated subnetwork detection algorithms and implementations remains a big challenge. While many of the methods cited in the introduction can be applied to reveal useful biological insight, there are limited studies concerning the comparison of formal and statistical properties of the proposed methods. The two main obstacles are a lack of well-defined gold standard datasets as well as the differences concerning the exact input and/or output of the methods. For example, it is not immediately clear how to compare algorithms which produce undirected subnetworks to those which elicit directed networks of a certain structure. An important first step toward atoning the issue in general is described in [BAG+17] which focuses on benchmarking approaches for undirected networks. For the purposes of this thesis, I designed and performed benchmarks of DeRegNet relative to its closest methodological relative, namely the algorithm described in [BRK+12].

Note however, while I am comparing the integer programming based algorithm of [BRK+12] to the fractional integer programming algorithm of DeRegNet, I am using the former as implemented in the DeRegNet software package [DeRc]. This renders the benchmark less dependent on implementation technology since both algorithms have been implemented with the same general stack of languages and libraries. For the benchmark I always utilize the human KEGG network as the underlying regulatory network (see subsection 4.1.1 in chapter 4). I then repeatedly simulate subgraphs which match the structure of both models (DeRegNet and [BRK+12]). The formal simulation procedure is described in algorithm 8. Initially, the simulated subgraph consists of one randomly selected root node, to which we iteratively add a random "outgoing" neighbor of a randomly selected current node in the subgraph until the size of the subgraph matches a randomly chosen value. The latter is uniformly chosen to be an integer between a given lower and an upper bound. "Outgoing" neighbors of $v \in V$ are any node from the set $\delta^+(v) = \{u \in V \backslash \{v\} : (v,u) \in E\}$. All nodes in the simulated "real" subgraph are assigned a node score of 1, while all nodes which are not contained in the subgraph are assigned a node score of 0. These node scores are then flipped with a certain probability p_f to emulate noise in the measurements of deregulation. In summary, we obtain random "real" subgraphs and simulated scores. In terms of the probabilistic interpretation of DeRegNet presented in section 3.1.3 the above scheme corresponds to a deregulation probability of $1 - p_f$ for nodes in the "real" subgraph and of p_f for nodes not part of the "real" subgraph. Hence, under the assumptions outlined in section 3.1.3 the simulations are restricted to values $p_f \in [0, \frac{1}{2})$.
Given a sequence of $N \in \mathbb{N}$ of these simulated instances, both subnetwork identification algorithms are run in order to find subgraphs which can then be compared to the known simulated real subgraph. Here, a *hit* (*true positive*, *tp*) is defined as a node appearing in a subgraph calculated by some algorithm which is also an element of the real subgraph. A *false positive* (*fp*) is a node which appears in a subgraph calculated by an algorithm but is not part of the real subgraph. A *false negative* is defined as a node which is part of the true subgraph but not part of the subgraph detected by an

Data: A directed graph $G = (V, E)$, receptor set $R \subset V$, $p_f \in [0, \frac{1}{2})$
Result: A DeRegNet instance, a simulated *true* optimal subgraph $V' \subset V$ and the simulated root node $r \in V' \cap R$
Choose $r \in R$ with probability $\frac{1}{|R|}$ (Choose root node)
$V' := \{r\}$ (Initialize subgraph with root)
Choose $k \in [k_{min}, k_{max}]$ with probability $\frac{1}{k_{max} - k_{min} + 1}$ (Choose subgraph size)
while $|V'| \neq k$ **do**
    if $(\bigcup_{v' \in V'} \delta^+(v)) \setminus V' = \emptyset$ **then**
        RESTART Algorithm 8
    end
    Choose $v \in (\bigcup_{v' \in V'} \delta^+(v)) \setminus V'$ with probability $|(\bigcup_{v' \in V'} \delta^+(v)) \setminus V')|^{-1}$
    $V' = V' \cup \{v\}$
end
for $v \in V'$ **do**
    Sample $s(v) \sim$ **Ber**$(1 - p_f)$
end
for $v \in V \setminus V'$ **do**
    Sample $s(v) \sim$ **Ber**(p_f)
end
return $(G, R, \emptyset, \emptyset, \emptyset, s)$, V', r

Algorithm 8: Simulating DeRegNet instances with known "optimal" subgraph. Here, **Ber**(p_f) denotes a Bernoulli random variable with parameter p_f.

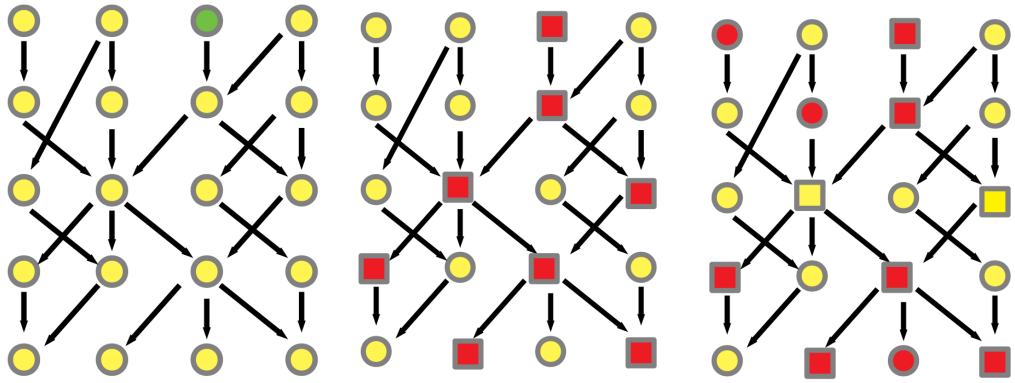


Figure 3.3: Simulating DeRegNet instances. See algorithm 8 for a formal outline of the simulation procedure. On a high level it proceeds like this: **(a)** Choose root node randomly. **(b)** Simulate a feasible subgraph by randomly choosing nodes maintaining the topological constraints of the model. Set the deregulation score of nodes in the subgraph to one. **(c)** Introduce noise by flipping node deregulation scores randomly.

algorithm. Furthermore, we can compare the sizes of the calculated subgraphs with the size of the real subgraph. In more formal terms, given an algorithm $\mathscr{A}$, which on a given instance with true subgraph size $|V'|$ finds a subgraph $V_{\mathscr{A}}$, one defines:

- *true positive rate* **TPR** $:= \frac{|V' \cap V_{\mathscr{A}}|}{|V'|}$, i.e. the number of actual hits divided by the number of possible hits, i.e. true subgraph size
- *false positive rate* **FPR** $:= \frac{|V_{\mathscr{A}} \setminus V'|}{|V'|}$, i.e. the proportion of nodes in the subgraph found by the algorithms which are not part of the true subgraph relative to true subgraph size
- *size efficiency* **SE** $:= \frac{|V_{\mathscr{A}}|}{|V'|}$, i.e. the proportion of algorithm subgraph size to true subgraph size

Another comparison metric is the running time of the algorithms. Further, the benchmark is based on the realistic assumption that we do not know the exact size of the real subgraph and that one can only assume lower and upper bounds on the subgraph size instead. Since the [BRK+12] algorithm does need an a fixed priori specified subgraph

size we employ a strategy suggested by [BRK+12] to circumvent that fact. Namely, to iterate from the lower to the upper bound, find a subgraph for each size and then regard the union graph of all found subgraphs as the one subgraph emitted by the algorithm. The strategy is summarized as algorithm 9. DeRegNet natively requires only a lower and an upper bound on subgraph size as parameters.

Data: A DeRegNet instance with underlying graph $G = (V, E)$
Result: A set $V' \subset V$ (inducing a subgraph)
$V' := \emptyset$ (Initialize the final subgraph)
for $k = k_{min}$; $k \leq k_{max}$; $k{+}{+}$ **do**
  $V' = V' \cup$ **APPLY_BACKES**(k)
end
return V'

Algorithm 9: Applying [BRK+12] for benchmarking DeRegNet. Here, **APPLY_BACKES**(k) refers to applying the algorithm of [BRK+12] with fixed subgraph size k, understood to return a set of nodes corresponding to the induced subgraph found by the run.

Figure 3.4 shows results of simulation runs carried out according to the described procedure.

As can be seen in Figure 3.4, outperforms [BRK+12] in terms of false positive rate (FPR), runtime and size efficiency, while the true positive rate of [BRK+12] is hard to beat. Nonetheless DeRegNet achieves solid performance also in terms of TPR.

Less quantitatively, note that DeRegNet allows for subgraphs which originate from so called source (root, receptor) nodes and *end* at so called terminal nodes. This is not readily possible with the [BRK+12] algorithm due to the necessity to specify a fixed subgraph size *a priori* and the resulting lack of flexibility to connect receptors to targets.

Lastly, note that DeRegNet actually provides an open-source implementation of the [BRK+12] algorithm[14]. All benchmarks have been carried out with the following setup: software: Ubuntu 18.04, Gurobi 8.1.1, hardware: 12x Intel i7-8750H @ 4.100 GHz, 32 GB RAM, Samsung SSD 970 EVO Plus.

[14] Currently the implementation only supports the commercial Gurobi MILP solver as a solver backend.

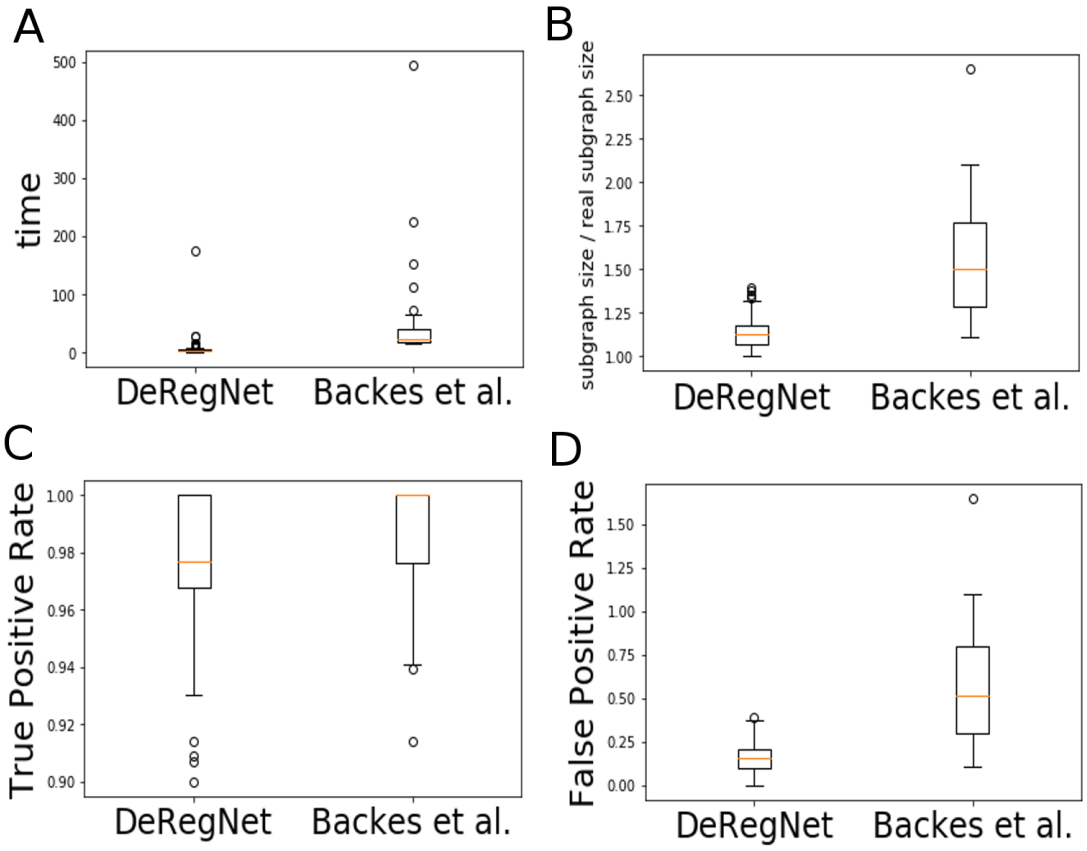


Figure 3.4: Benchmark patterns for DeRegNet and [BRK+12]. (A) Running time (in seconds) of DeRegNet (Dinkelbach algorithm) and $k_{max} - k_{min} + 1$ runs of the [BRK+12] algorithm: DeRegNet at least matches beats the performance of [BRK+12] on our test instances **(B)** Size efficiency (SE): the size DeRegNet subgraphs is closer to the true size of the subgraph **(C)** TPR: [BRK+12] finds more of the true subgraph nodes than DeRegNet with a mean of 100 % possible hits, while DeRegNet still achieves always more than 90 % of possible hits with a mean well above 95 % **(D)** FPR: DeRegNet is less noisy than [BRK+12] in that it finds less false positive nodes. Moreover, DeRegNet is more consistent with respect to that metric while the variance of [BRK+12]'s FPR is considerable.

3.4 Summary and Discussion

This chapter presented the technical details of the de-novo pathway identification framework DeRegNet, i.e. the mathematical optimization problem which serves as its basis and the various methods employed to allow or improve the solution of that problem. Furthermore, a statistical model for DeRegNet was provided, also explicitly outlining DeRegNet's formal relationship to related work. DeRegNet's concrete implementation was described, providing links to the open source software implemented by the author. Finally, a simulation-based benchmark was carried out and DeRegNet manages to compare favourably to it closed methodological relative [BRK+12].

While the methodological development and concrete implementation of DeRegNet as outlined in this chapter is complete to a large degree, there always exists room for improvements. In particular, although the commercial Gurobi solver [Gurb] allows for a relatively flexible academic licensing scheme [Gura], it would benefit DeRegNet to be able to work with further commercial but especially open source MILP solvers. This becomes especially apparent with respect to potential cloud or HPC deployments (compare chapter 5). On a methodological level, more complex statistical models for models like DeRegNet and similar methods would benefit the field of de-novo pathway enrichment as a whole. This would likely also pave the way to more rigorous/well-defined benchmarking approaches/principles. The application of DeRegNet to actual omics data is outlined in chapter 4, demonstrating its suitability as a freely available [DeRc] heuristic hypothesis generation tool.

Chapter 4

Applications of *de-novo* pathway discovery

This chapter highlights three application of the DeRegNet framework introduced on a technical level in chapter 3 to several omics datasets. Section 4.1 describes various applications to the public TCGA liver cancer dataset. Section 4.2 show the how DeRegNet subgraphs can shed light onto the folate one-carbon metabolism (1C metabolism) in the context of another liver cancer study. Finally, section 4.3 explains the application of DeRegNet to phosphoproteomic regulation of the *S. cerevisiae* cell cycle.

4.1 Application to TCGA liver cancer data

While subsection 4.1.1 introduces the underlying biomolecular network used to find subgraphs and the node scores derived from the TCGA dataset, the following subsections detail mainly two application modes of DeRegNet to TCGA liver cancer omics data. First, subsection 4.1.2 contains results concerning the *global* application of DeRegNet, i.e. use of DeRegNet to find subgraphs based on node scores which describe the dataset as a whole. Secondly, subsection 4.1.3 describes a personalized approach for finding subgraphs based on node scores for every case/patient/participant of the TCGA study at hand. Finally, based on this personalized application of DeRegNet, subsection

4.1.4 outlines an approach for survival prediction based on features derived from the determined subgraphs.

4.1.1 Network and omics data

KEGG network

While many sources for directed biomolecular networks are available, e.g. [CGD+11], in this paper I exclusively utilize a directed gene-level network constructed from the KEGG database with the KEGGgraph R-package [ZW09]. The script used to generate the network as well as the network itself can be found in the DeRegNet GitHub repository [DeRc]. The constructed directed KEGG network has 5522 nodes and 58295 edges.

TCGA-LIHC data and RNA-Seq derived node scores

Gene expression data was downloaded for hepatocellular carcinoma TCGA project from the Genomic Data Commons Portal [tcg]. Raw quantified RNA-Seq counts were normalized with DESeq2 [LHA14] which was also used for calculating log2 fold changes for every gene between cancer and control tissue. Personalized log2 fold changes were calculated by dividing a patients tumor sample expression by the mean of all available control samples (adding a pseudo count of 1) before taking the log. The following node scores are defined.

- *Global RNA-Seq score s*: s_v = RNASeq log2-fold change for a gene $v \in V$ as calculated by DESeq2 for the TCGA-LIHC cohort
- *Trinary personalized RNA-Seq score* s^c for case c:

$$s_v^c = \begin{cases} +1 & \text{if personalized log2 fold} > 2 \\ -1 & \text{if personalized log2 fold} < -2 \\ 0 & \text{else} \end{cases} \tag{4.1}$$

We refer to subgraphs found with the global RNA-Seq score s as *global subgraphs*. A global subgraph can further be subdivided as being *upregulated* or *downregulated* depending on whether the subgraphs were found by employing a maximization or minimization objective respectively. For (any) node score $s : V \to \mathbb{R}$ we define $|s| : V \to \mathbb{R}$ by $|s|(v) := |s(v)|$ for all $v \in V$. *Dysregulated* global subgraphs are those which were found by using the score $|s|$ under a maximization objective. Similarly subgraphs found with any of the scores s^c with a maximization objective are called *upregulated* while those found with minimization objective are called *downregulated* (personalized subgraphs for case/patient c). Subgraphs found with a $|s^c|$ score under maximization are called *dysregulated* (personalized subgraphs for case/patient c). Any of the above subgraph types is called a *deregulated* subgraph.

4.1.2 Global deregulated subgraphs for TCGA-LIHC

Using the DeRegNet algorithm I determined the deregulated global subgraphs obtained from running the algorithm with the global score defined in the previous section. The optimal and four next best suboptimal subgraphs were calculated for every modality. The subgraphs were then summarized as a subgraph of the union graph of optimal and suboptimal subgraphs in order to allow streamlined interpretation[1].

Reconstruction of transcriptional activation of *WNT* signaling

The summarized global upregulated subgraph is shown in Figure 4.1.

The subgraphs shows the activation of the *WNT* signaling pathway by means of over-expressed Glypican-3 (*GPC3*), which represents a membrane-bound heparin sulphate proteoglycan [ARF13]. *GPC3* has been extensively researched as an early biomarker and potential therapy target in HCC [ZSYT18, WLD16, FH14, FC13, HK11, BAM+12] (See figure 4.2).

[1]For the specific subgraphs determined by DeRegNet, see figures A.1, A.2, A.3, A.4, A.5 for the upregulated subgraphs and A.6, A.7, A.8 , A.9, A.10 for the downregulated subgraphs.

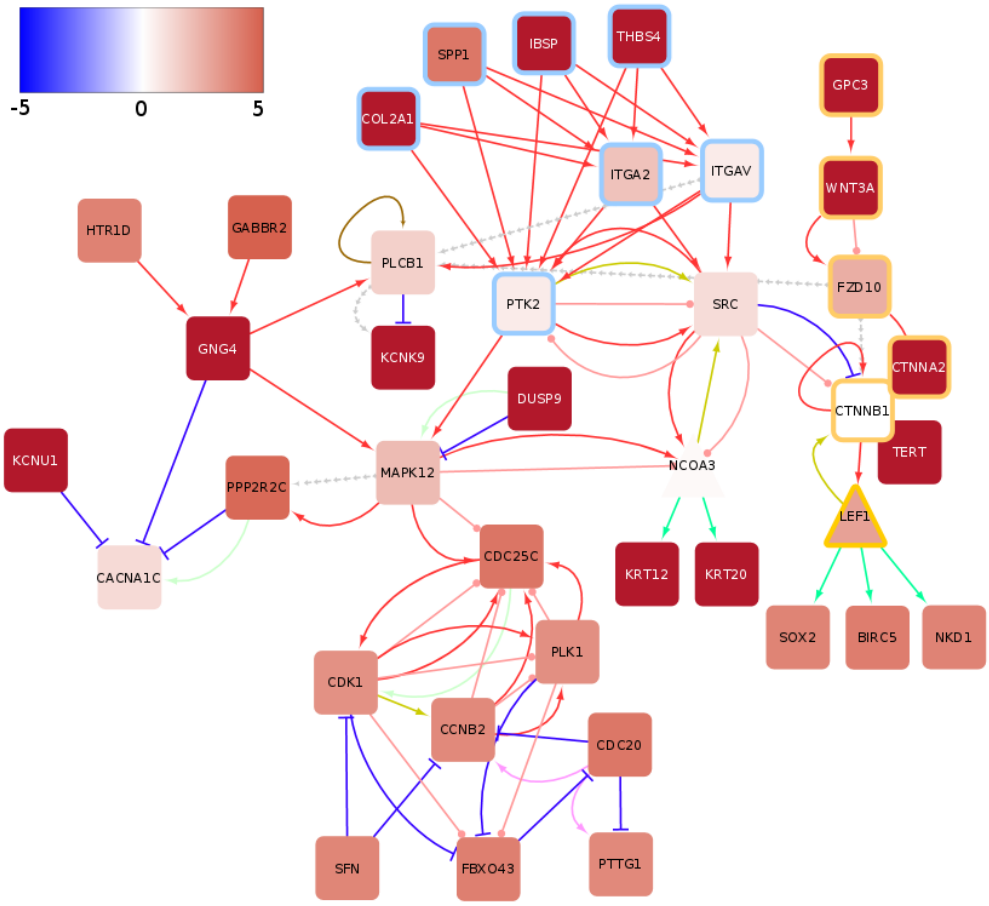


Figure 4.1: Global upregulated subgraph for TCGA-LIHC reconstructs transcriptional activation of *WNT* signaling. The Color of nodes indicates the average $\log_2$ fold change of tumor samples compared to controls as represented in the color bar. The color of rims around nodes indicates genes contained in the integrin pathway (blue), the *WNT* pathway (yellow) and diverse other pathways (no rim). The color of edges indicates following interactions: activation (red), inhibition (dark blue), compound (brown), binding/association (yellow), indirect effect (dashed grey), phosphorylation (pink), dephosphorylation (light green), expression (green) and ubiquitination (light purple).

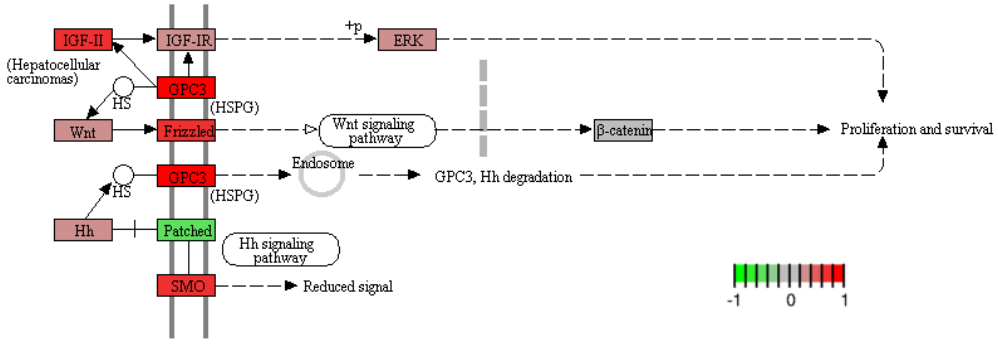


Figure 4.2: *GPC3*-mediated activation of *WNT* signaling is a well-documented process in liver cancer. The figure shows the relevant KEGG map (Proteoglycans in cancer: hsa05205) with TCGA-LIHC min-max-scaled $\log_2$ fold changes mapped onto the genes. This process was automatically recaptured by the upregulated subgraphs for TCGA-LIHC. See Figure 4.1.

Genomic analysis conducted over the past decade have identified mutations affecting Telomere Reverse Transcriptase (*TERT*), *β*-catenin (*CTNNB1*) and cellular tumor antigen *p53 (TP53)* [LZRP+16] as common driver mutations in HCC. Mutations in the *TERT* promoter are a well-studied factor in liver cancer development [NZR16, QOT+14] and lead to *TERT* overexpression while mutations in *CTNNB1*, activate *CTNNB1* and result in activation of *WNT* signaling. Previous studies have determined that *TERT* promoter mutations significantly co-occur with *CTNNB1* alternation and both mutations represent events in early HCC malignant transformation [TTC+14]. In agreement, the DeRegNet algorithm recaptures the importance of a *CTNNB1*:*TERT* connection on a transcriptional level.

The subgraphs further show a possible alternative mechanism of *CTNNB1* activation through upregulated *GPC3*, an early marker of HCC, as well as Wnt Family member 3a (*WNT3A*) and Frizzled 10 (*FZD10*). *WNT3A* promotes the stabilization of *CTNNB1* and consequently expression of genes that are important for growth, proliferation and survival [AM13] through activity of transcription factor Lymphoid Enhancer-Binding Factor 1 (*LEF1*). As shown in the subgraph figure 4.1, *LEF1*'s known targets SRY-box 2 (*SOX2*)[2] and Baculoviral IAP Repeat Containing 5 (*BIRC5*) are likely important contrib-

[2] Sex-Determining Region Y (*SRY*)

utors to *WNT* pathway driven *WNT* proliferation. *SOX2* is a pluripotency-associated transcription factor with known role in HCC development [SSL+13, WHC+13, LLZ+16] and *BIRC5* (survinin) is an anti-apoptotic factor often implicated in chronic liver disease and liver cancer [MJB+12, MMF+07, Su16].

In summary, the algorithm reconstructed important components of the canonical *WNT* signaling pathway activation in liver cancer [TB08, LXC+16, VTMG16, CN12, NC17] from TCGA-LIHC RNA-Seq data and pairwise gene-gene interaction information from KEGG.

Crosstalk between integrin and *WNT* signaling

Another interesting pattern emerging in the upregulated subgraphs is the crosstalk between the *WNT* signaling cascade and integrin signaling. Over-expression of Secreted Phosphoprotein 1 (*SPP1*) has been shown to be a common feature for most known human malignancies and it is commonly associated with poor overall survival [BCO+08]. The binding of *SPP1* to integrins (e.g. integrin $\alpha V\beta 3$) leads to further activation of kinases associated with proliferation, epithelial-mesenchymal-transition, migration and invasion in HCC, such as Mitogen Activated Kinase-like Protein (*MAPK*), Phosphatidylinositol-4,5-bisphosphate 3-kinase (*PI3K*), Protein Tyrosine Kinase (*PTK2*), and SRC proto-oncogene/Non-receptor tyrosine kinase (*SRC*) [WJXK16]. Further captured by the subgraphs is that elevated expression of *PTK2* and *MAPK12* are accompanied with elevated expression of cell cycle related genes (Cell Division Cycle 25 Homolog C / M-phase inducer phosphatase 1 (*CDC25C*), Cyclin-dependent Kinase 1 (*CDK1*) and Polo-like Kinase 1 (*PLK1*)), thus connecting over-expression of kinases with cell proliferation.

Although KEGG lists the interaction between *SRC* and *CTNNB1* as inhibitory in nature, other studies have concluded that activated Src enhances the accumulation of nuclear beta-catenin and therefore through their interaction contributes to an oncogenic phenotype [KGD+05].

In conclusion, the upregulated subgraphs capture the interaction of *SPP1* with integrin and consequent activation of *PTK2* and *SRC* together with their connection to the *WNT* signaling pathway (via *CTNNB1*) and cell cycle genes.

Downregulated oncogenes *FOS* and *JUN* and drug metabolism

The global downregulated subgraphs are centered around down-regulation of transcription factors *FOS* and *JUN*. The subgraph summary is depicted in figure 4.3. *FOS* and *JUN*, which form *AP-1* the transcription factor complex, are considered to be oncogenic factors and necessary for development of liver tumors [EW03]. Considering their prominent role in liver tumorigenesis, further experimental study of the significance of Jun and Fos downregulation on HCC development could be of great interest. Interestingly, RNA-seq data show that all *FOS* (*FOS*, *FOSB*, *FOSL1*, *FOSL2*) and *JUN* (*JUN*, *JUNB*, *JUND*) isoforms are downregulated in a majority of liver tumors of the TCGA cohort (See figure 4.4).

Furthermore, the subgraphs show a number of downregulated Cytochrome P450 (*CYP*) enzymes as part of the most downregulated network of genes. *CYP3A4* is mainly expressed in the liver and has an important role in the conversion of carcinogens, such as aflatoxin B_1 toward their ultimate DNA-reactive metabolites [Luc05], as well as, in detoxification of anticancer drugs [UGAR05]. Although the downregulation of *CYP* enzymes could potentially render HCC tumors sensitive to chemotherapy, liver tumors are notoriously unresponsive to chemotherapy [LZRP+16]. Therefore, it is unclear how the gene pattern of *CYP* enzymes captured by the presented subgraphs could influence the HCC response to therapy and which compensatory mechanism is employed to counteract *CYP* downregulation.

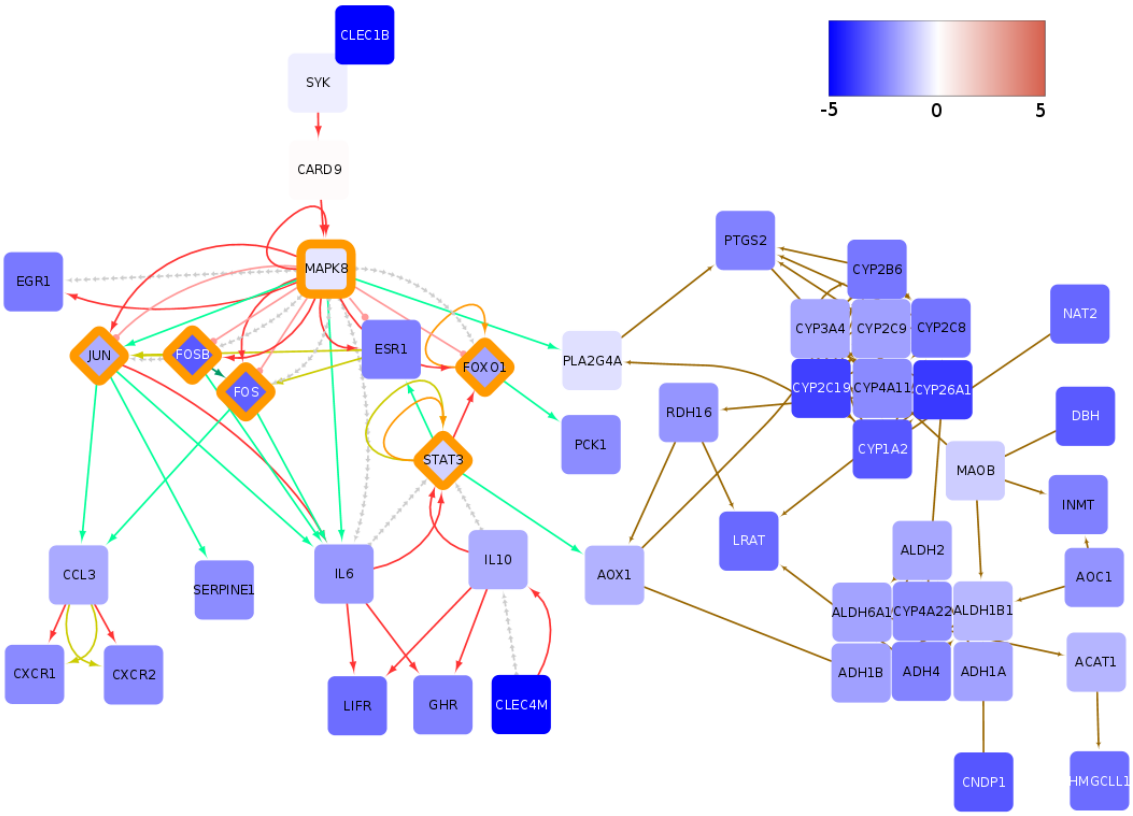


Figure 4.3: Global downregulated subgraph for TCGA-LIHC are centered on *FOS* and *JUN* transcription factors and drug metabolism. Color of nodes indicates the average $\log_2$ fold change of tumor samples compared to controls as represented by the color bar. The color of edges indicates the following interactions: activation (red), compound (brown), binding/association (yellow), indirect effect (dashed grey) and expression (green). Also noteworthy it the general connection of transcriptional activators and inhibitors to signaling as well as metabolic networks. Transcription regulators have been highlighted with an orange rim.

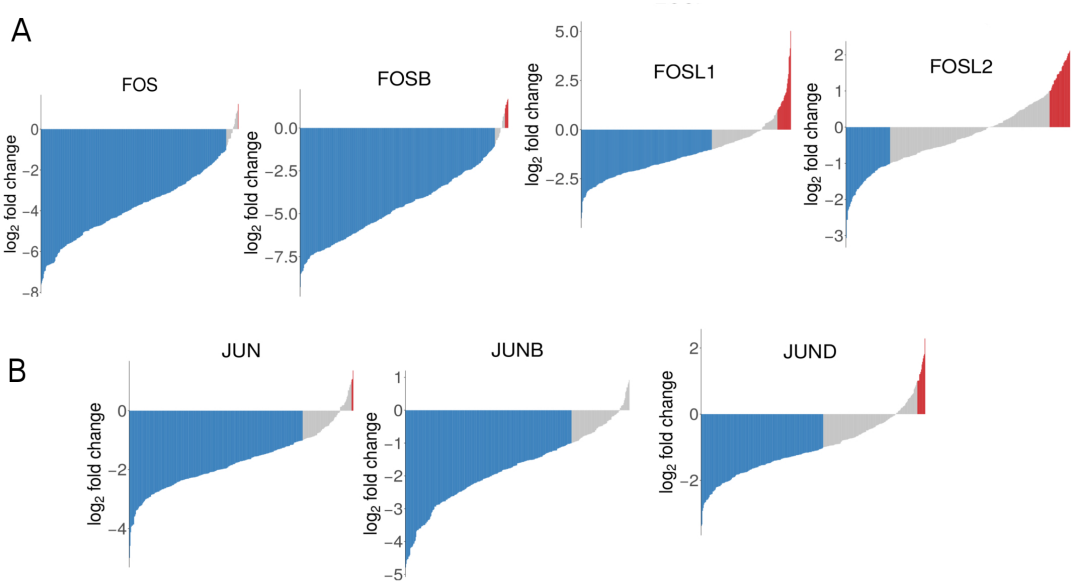


Figure 4.4: Expression of *FOS* and *JUN* isoforms in tumor of TCGA-LIHC cohort. (A) Log_2-fold changes of FOS isoforms in individual tumors compared to the mean control value in the TCGA-LIHC dataset. (B) Log_2 fold changes of *JUN* isoforms in individual tumors compared to the mean control expression value in the TCGA-LIHC dataset. The color of the bars in the waterfall plot indicate mRNA downregulation $\geq$ 1.5-fold (blue), mRNA upregulation $\geq$ 1.5-fold (red). Related to Figure 4.3.

4.1.3 Personalized deregulated subgraphs for TCGA-LIHC

Finding deregulated subgraphs in a patient-resolved manner enables steps toward personalized medicine. In this section I describe a case study where DeRegNet was employed to find an upregulated subgraph for every TCGA-LIHC patient. By stratifying patients according to whether their subgraph contains a gene or not, one can identify genes whose inclusion into a patient's inferred subgraph provides a survival handicap or advantage. I first detail the *subnetwork-defined cancer gene* approach and thereafter highlight one particular such gene, namely Spleen Tyrosine Kinase (*SYK*), in the context of its defining subgraphs.

Subnetwork-defined cancer genes

Genes, gene products or biomolecular agents are likely to bring about their various phenotypic effects only in conjunction with other agents via their shared biomolecular network context. By that token, one can search for genes which convey phenotypic differences by means of some defined network context. Here, I propose DeRegNet subgraphs as network context for a given case/patient in order to find genes whose inclusion into a case's subgraph associates with a significant difference in overall survival. Algorithm 10 describes the procedure more formally. Genes implicated by the outlined procedure are termed *network-defined cancer genes*. The next section provides details on a specific network-defined cancer gene obtained by application of the procedure to personalized upregulated subgraphs in the TCGA-LIHC cohort. Figure 4.6 shows the survival effect for some other *subnetwork-defined cancer genes* determined based on the personalized subgraphs.

Spleen tyrosine kinase (*SYK*) as a network-defined cancer gene

Patients whose subgraph contain the spleen tyrosine kinase (*SYK*) showed comparatively bad survival outlook (see Figures 4.5, 4.7).

Data: A set of cases $\mathscr{C}$, DeRegNet instances $I_c = (G = (V, E), R, T, \mathbf{Ex}, \mathbf{Inc}, s^c)$ for every $c \in \mathscr{C}$, a subset of *nodes of interest* $V_I \subset V$ and a *survival mapping* $p : \mathscr{C} \to [0, \infty)$.
Result: A mapping $pval : V_I \to [0, 1]$ associating each $v \in V_I$ with a p-value.
for $c \in \mathscr{C}$ **do**
  Solve the DeRegNet instance I_c to obtain the nodes V_c contained in c's subgraph
end
for $v \in V_I$ **do**
  $\mathscr{C}_v := \{c \in \mathscr{C} : v \in V_c\}$
  Obtain the Kaplan-Meier estimate [KM58] for p w.r.t groups $\mathscr{C}_v$ and $\mathscr{C} \setminus \mathscr{C}_v$.
  $pval(v) :=$ p-value of log rank test [ABG08] between groups $\mathscr{C}_v$ and $\mathscr{C} \setminus \mathscr{C}_v$
end
Carry out multiple testing correction of *pval*
return *pval*

Algorithm 10: Finding subnetwork-defined cancer genes. After finding subgraphs for individual cases/patients the procedure partitions a set of cases/patients according to whether they contain a given gene in their determined subnetwork and tests whether the thus defined partition conveys a significant survival difference. Note, that in the described setting, the DeRegNet instances only differ in terms of their case-dependent node score s^c.

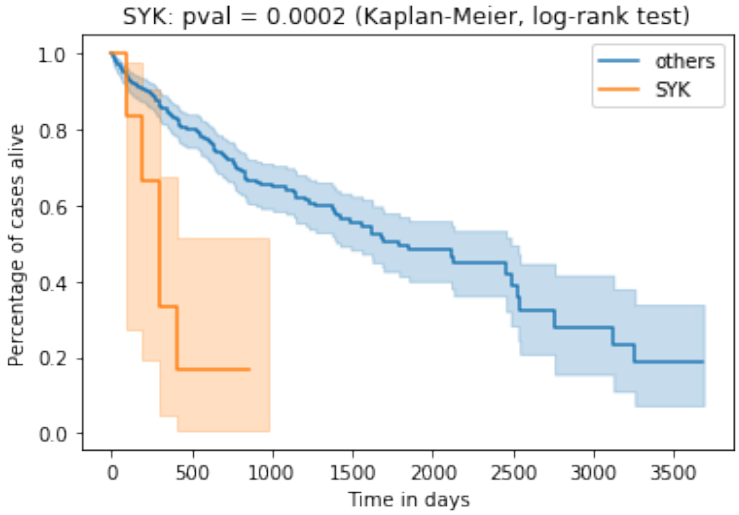


Figure 4.5: *SYK* signaling indicates poor survival. TCGA-LIHC cases TCGA-5C-AAPD, TCGA-CC-A3MA, TCGA-ED-A5KG, TCGA-DD-AACH, TCGA-YA-A8S7, TCGA-CC-5261, TCGA-CC-A3M9 show activated *SYK* signaling and poor survival. Survival difference is significant with $p = 0.2 \cdot 10^{-3}$ (Kaplan-Meier curve estimates [KM58, ABG08] and log-rank test [ABG08]).

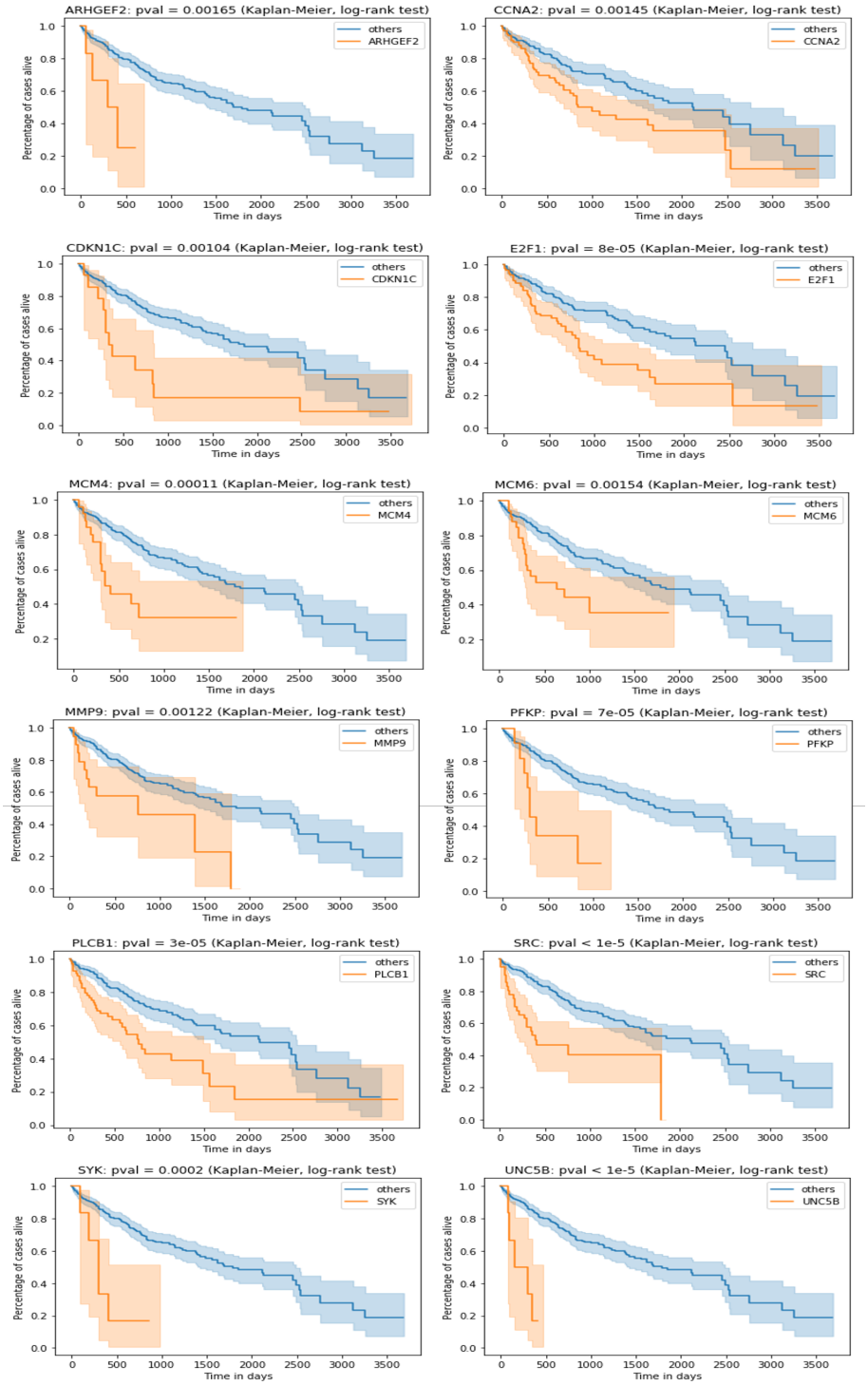


Figure 4.6: Subset of genes whose inclusion into a patient's inferred subgraph indicates poor survival. Survival difference is calculated using Kaplan-Meier estimates [KM58, ABG08] and log-rank test [ABG08]. Related to Figure 4.5.

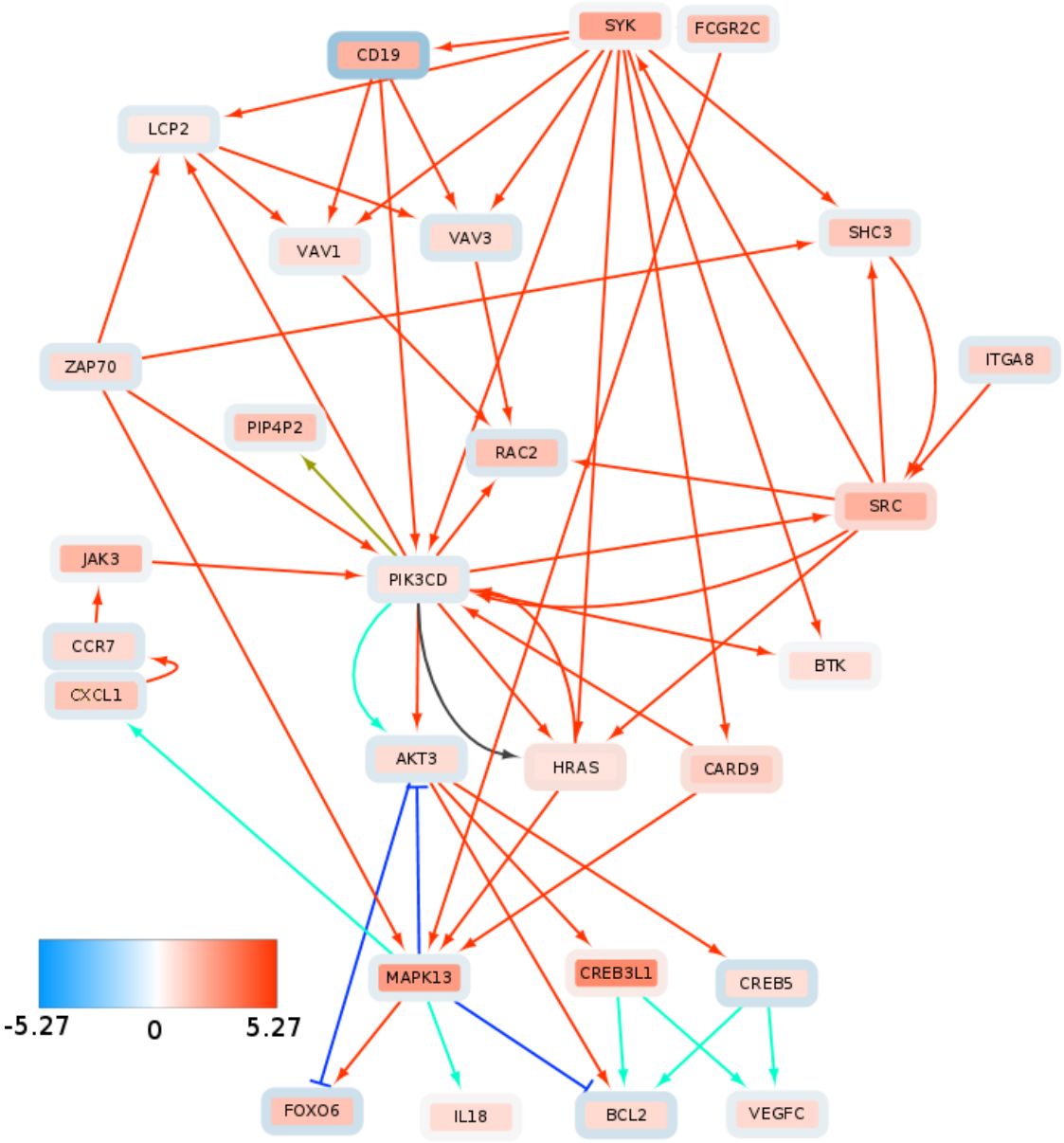


Figure 4.7: Consistent upregulation of *SYK* signaling components and downstream targets in subgraph of patients with poor survival. Inner color represents the average $\log_2$ fold change across the "*SYK*-positive" patients and rim color represent average log2 fold change across the rest of the TCGA-LIHC cohort. Color of edges indicates following interactions: activation (red), inhibition (dark blue), compound (brown), indirect effect (dark grey) and expression (blue green).

SYK is most commonly expressed in immune cells and its deregulation has been originally associated with hematopoietic cancers [Low11, KG15, MRT10]. However, it has been shown that *SYK* plays a role in various other cancer types and its respective roles seem to vary significantly depending on the molecular (i.e. ultimately network) context [KG15]. *SYK* comes in the form of two splice variants, *SYK*(*L*) and *SYK*(*S*) [HYW+14]. In the context of liver cancer, *SYK* promoter hypermethylation and corresponding *SYK* downregulation has been associated with poor survival [SLK+14]. Furthermore, Checkpoint Kinase 1 (*CHK1*) mediated phosphorylation of *SYK*(*L*) and associated *SYK* degradation has been considered an oncogenic process [HHY+12], associating low levels of *SYK* as a factor of poor survival. On the other hand, [HYW+14] *SYK*(*S*) expression promotes metastasis development in HCC and thus leads to poor survival outcome. Furthermore, high *SYK* expression has been shown to promote liver fibrosis [QZL+18]. The development of HCC is closely related to formation and progression of fibrosis. Fibrosis represents excessive accumulation of extracellular matrix (ECM) and scarring tissue in an organ. A fibrotic environment promotes development of dysplastic nodules which can gradually progress to liver tumors [BB05]. In short, a somewhat inconsistent role of *SYK* as a tumor suppressor or oncogene can be observed in many cancers [KG15], including liver cancer.

By employing DeRegNet, I identified by means of the approach defined as algorithm 10 a subgroup of HCC patients from the TCGA-LIHC cohort which show poor survival and a distinguished *SYK*-signaling pattern shown in Figure 4.7. The depicted network is manually extracted from the union graph of all the patient's subgraphs which contained *SYK*. The network shows *SRC*-*SYK*-mediated/enabled activation of PI3K-Akt signaling via B-lymphocyte antigen CD19 (*CD19*) and Phosphatidylinositol 4,5-bisphosphate 3-kinase catalytic subunit delta (*PI3KCD*)[3] [TYZ15]. Furthermore, *SYK* also feeds into mitogen-activated protein kinase 11-13 (p38) signaling (only *MAPK*13 shown) through GTPase Hras (*HRAS*) and aspase recruitment domain-containing protein 9 (*CASP9*). p38 signaling promotes cytokine expression via Growth-regulated

[3] p110δ

alphaprotein (*CXCL1*). Increased cytokine expression and activation is another canonical effect of *SYK* signaling [MRT10]. This in turn, activates JAK signaling through Januskinase 3 (*JAK3*) activity, thereby reinforcing PI3K activation. Interestingly, *SYK* signaling is consistently linked to the upregulation of the guanine nucleotide exchange factors *VAV1* and *VAV3* [MRT10, Low11][4]. The proto-oncogene *VAV3* is associated to adverse outcomes in colorectal [UFH+15] and breast cancer [CMMGE+12, CCL+15]. Furthermore *VAV3* mutations have been profiled to be potential drivers for liver cancer [LXK+18]. *VAV* signaling is mediated by forming a complex with Lymphocyte cytosolic protein 2 (*LCP2*)[5] upon activation of *SYK* signaling. *VAV*-meditated Ras-related C3 botulinum toxin substrate 2 (*RAC2*) activation may play a role in intravastation and motility [RCP11]. Additionally, the subgraph shows upregulation of the B-cell lymphoma 2 (*BCL2*) gene, a known regulator of apoptosis [HS13], and vascular endothelial growth factor-C (*VEGGC*) which can promote metastasis [MJJ+01] and angiogenesis [TZW+08, TAZ+10].

4.1.4 Subgraph features for predicting survival

Predicting phenotypes based on clinical and molecular data is one of the big challenge on the road to personalized medicine. A frequently readily available measure of phenotype for cancer patients is survival (i.e. the time from disease onset/diagnosis to (possibly disease induced) death). Improving upon clinical predictors with molecular data often still poses significant challenges [YVAO+14]. Here, I provide an example of the suitability of deregulated subgraph-derived features for predicting survival in the TCGA-LIHC dataset. In particular, we demonstrate that predictions based on subgraphs is at least as good GSEA-based predictions obtained in a comparable manner. Furthermore, subgraph derived features can improve upon predictions based on clinical features alone.

[4] Guanine nucleotide exchange factor (*VAV*)
[5] SLP-76

Data preparation and feature engineering

Survival times were binarized by labeling all patients with survival less than three years (1095 days) as bad outlook patients ($y = 0$) and all patients with last follow-up time larger than three years as good outlook patients ($y = 1$). The resulting dataset consisted of 198 patients from the TCGA-LIHC cohort[6]. For every case the following features are derived:

- **clinical**: Features from clinical data comprising *age*, *gender*, *body mass index (BMI)*, *tumor stage* (!) and *tumor morphology*. Age (in years) and BMI were scaled via z-scores. Tumor stage and morphology where one-hot encoded.

- **gsea**: Features derived from (single sample) Gene Set Enrichment Analysis (GSEA) [STM+05]. Two lists of significantly enriched pathways w.r.t good outcomes vs. bad outcomes and vice versa were computed by (standard) GSEA. From every list I retained pathways with adjusted p-value less than 0.1, which resulted in a total of 14 KEGG pathways. After performing ssGSEA, every sample received the corresponding personalized ssGSEA enrichment scores for these pathways as a 14-dimensional feature vector. The above steps were carried out with *gseapy* [gse]. For more information on single-sample GSEA, see [FBL+18]. The obtained features were scaled via z-scores.

- **subgraph_overlap**: Features based on up- and downregulated subgraphs for the good and bad outcome subgroups. Subgraphs were computed based on the global deregulation score for the good outcome and bad outcome patients respectively (on the respective training sets only, see below). Every sample is then associated with the regulation-aware node overlap[7] between its personalized de-, up- and downregulated subgraphs and up- and downregulated global subgraphs

[6] Some cases dropped out due to incomplete or missing survival data.

[7] Given two (induced) subgraphs $V', V'' \subset V$ and node scores $s', s'' : V \to \{-1, 0, 1\}$ the deregulation-aware node overlap is defined as $\sum_{v \in V} (\mathbb{I}(v \in V')\ s'_v) \cdot (\mathbb{I}(v \in V'')\ s''_v)$.

for the good and bad outcome subgroups respectively. This amounts to a 12-dimensional feature vector. Again, z-scores were applied.

- **ndcg**: Subgraph features derived from network-defined cancer genes. After identifying network-defined cancer genes (see previous subsection) for de-, up- and downregulated subgraphs one obtains a binary indicator for every case representing whether it contains any given such gene or not, leading to 15-dimensional feature vectors corresponding to 15 network-defined cancer genes.
- **subgraph**: *subgraph_overlap* and *ndcg* combined (concatenated).

Under a *feature combination* it is understood the combination of two or more of the just defined features. In the following, I use a plus sign to indicate feature combinations, e.g. *subgraph* = *subgraph_overlap* + *ndcg*. As another example, *subgraph* + *clinical* then denotes *subgraph* features combined with *clinical* features.

Survival prediction with clinical, pathway and subgraph features

The experiments described in the following were carried out with scikit-learn [sci]. Every feature/feature combination was tested by training a Support Vector Machine, a simple artificial neural network, a random forest and a logistic regression. For every algorithm we performed an algorithm-specific grid search for model selection. The grid search was equivalent for different feature combinations in order to be able to assess the comparative suitability of the features. The grid search was conducted with 6-fold cross validation estimating mean Receiver Operating Characteristic (ROC) curves and Area under the curve (AUC) scores.

Features *gsea* and *subgraph_overlap* are roughly equivalent with respect to the underlying logic, with subgraphs or pathways as contextual data inputs respectively. Hence, comparing these two features may give an indication of the suitability of subgraph vs. pathway methods for feature engineering for survival prediction. Figure 4.8 shows that the *subgraph_overlap* features hold promise w.r.t *gsea* features.

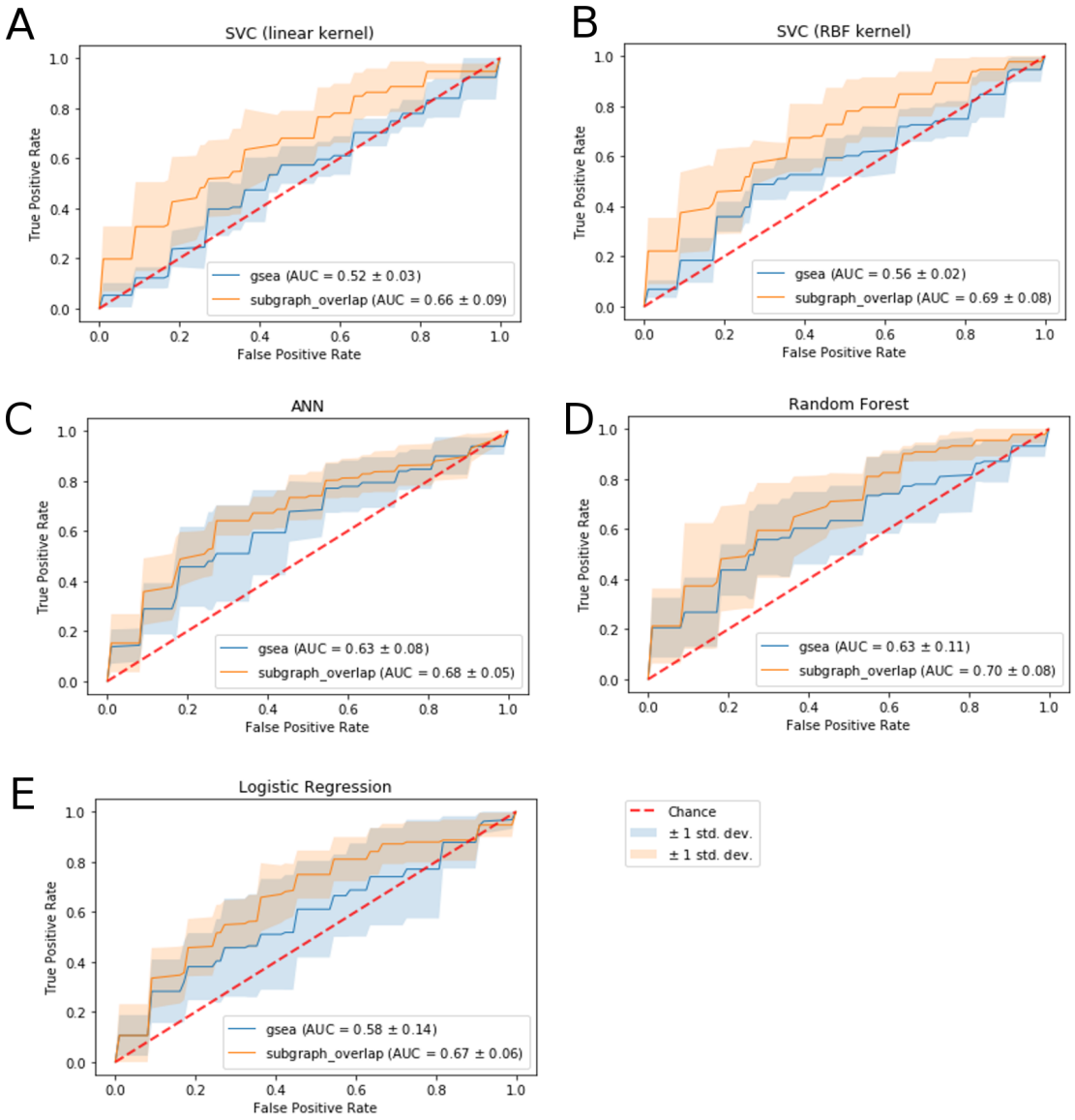


Figure 4.8: Subgraph features vs. GSEA features across models. (A) Support Vector Classifier (SVC) with linear kernel (B) Support Vector Classifier (SVC) with radial basis function (RBF) kernel (C) Artificial Neural Network (ANN) (D) Random forest (E) Logistic Regression.

Furthermore, it has been shown that improving upon clinical features with molecular features for survival prediction is not an easy task [YVAO+14]. The experiments conducted here show that for the given setting, prediction models combining clinical and subgraph features (based on molecular interactions and data) provide performance gains compared to a purely clinical model. Also, the subgraph features achieve parity with classifiers based on clinical data alone. Figure 4.9 represents these findings.

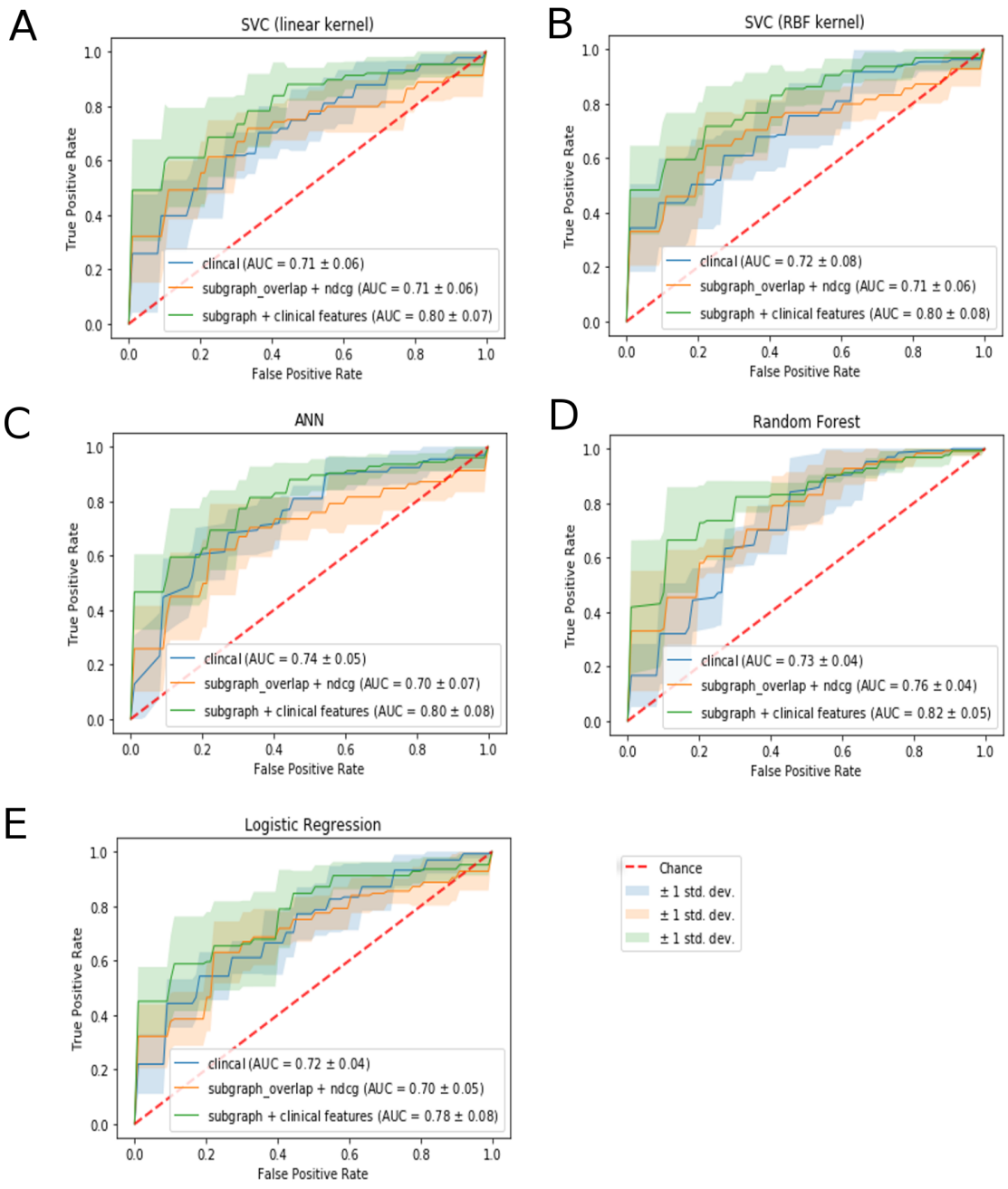


Figure 4.9: Subgraph features vs. clinical features across models. Clinical + subgraph features together outperform either in isolation. Subgraph features perform comparably to clinical features. (A) Support Vector Classifier (SVC) with linear kernel (B) Support Vector Classifier (SVC) with radial basis function (RBF) kernel (C) Artificial Neural Network (ANN) (D) Random forest (E) Logistic Regression.

4.2 Application to 1C metabolism in liver cancer

Folate one-carbon metabolism (1C metabolism) produces many metabolites which serve as intermediate compounds that are channeled in production of key metabolites such as nucleotides and amino acids. Additionally, this pathway plays an important role in methylation as well as free radicals control [Loc13]. In a study examining the dysregulation of metabolic pathways in liver cancer [WTD$^+$] a deregulated subgraph based on liver cancer RNA-Seq data[8] showed decisive network effects for altered 1C metabolism and associated pathways, helping to guide research into further dissection of the importance of this pathway in liver cancer development [WTD$^+$]. The subgraph is shown in figure 4.10. Subsequent metabolic flux analysis performed on cell lines derived from the tumors has further supported the insight into importance of 1C pathway in liver carcinogenesis.

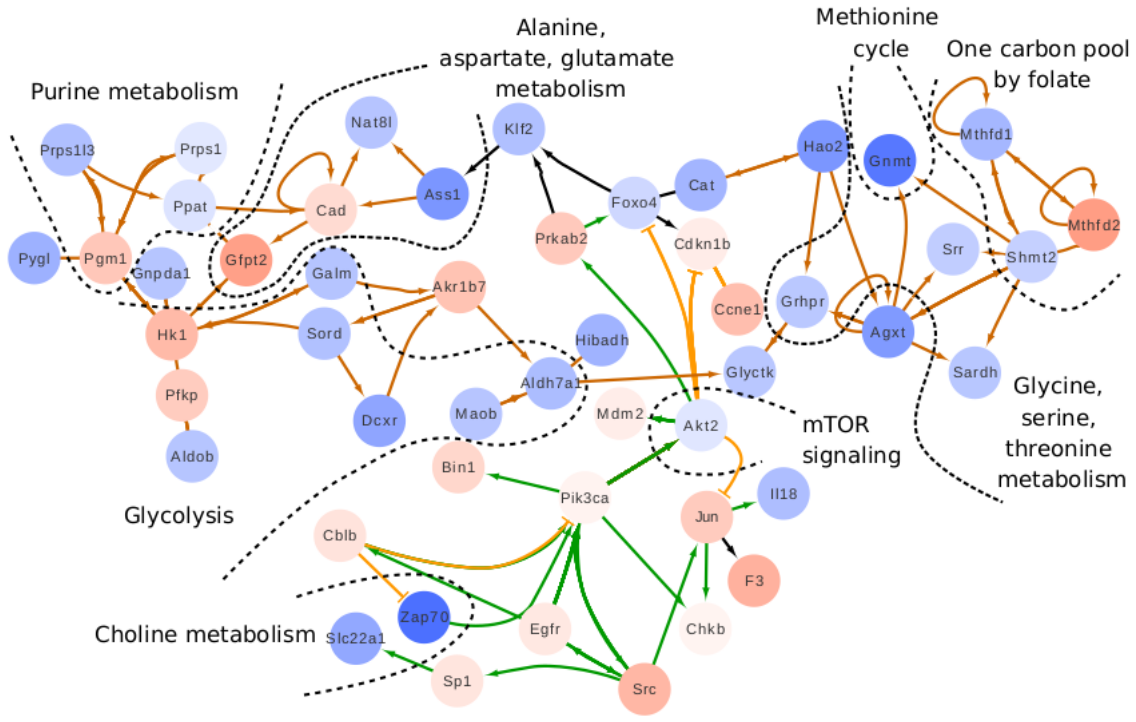


Figure 4.10: Folate one-carbon metabolism and its deregulated network context. See main text, section 4.2 and [WTD$^+$].

[8] Log_2 node scores similar to section 4.1 and the KEGG network from subsection 4.1.1.

4.3 Application to S. cerevisiae Cell cycle regulation

DeRegNet was applied in the context of a study of the regulation of the S. cerevisiae cell cycle by metabolic processes [ZWS+19a, ZWS+19b]. The application of DeRegNet is based on phosphoproteomic time series spanning the cell cycle. Based on these measurements, time-course patterns were defined representing meaningful changes of phosphorylation [ZWS+19b, ZWS+19b]. These abstract patterns were then used to statistically test for their occurrence for any given phosphorylation site in any given protein covered by the dataset [ZWS+19b, ZWS+19a]. The minimum of the p-values of all sites for a given protein was then taken as a node score for DeRegNet where a node represents a protein in a regulatory network derived from KEGG[9] [ZWS+19a]. As shown in figure 4.11, the obtained deregulated subgraphs (minimizing the average p-value score) could capture essential parts of the yeast cell cycle [ZWS+19a]. The subgraph is shown in figure 4.11.

4.4 Summary and Discussion

This chapter described various applications of the DeRegNet algorithm as introduced in chapter 3. In the context of the TCGA-LIHC dataset, it was outlined in section 4.1 how the application of DeRegNet in a global fashion could identify driving factors of liver cancer such as a transcriptionally activated WNT-pathway. Another example of the insights DeRegNet can provide is interaction of integrin and WNT signaling, as well as drug metabolism in liver cancer. In fact, profiling of such interactions between pathways is one of the main strengths of the algorithm over classical gene enrichment methods. Additionally, the application of DeRegNet in a patient-specific manner could identify a consistent subgroup of patients showing poor prognosis potentially due to aberrant *SYK* signaling and therefore can generate meaningful hypotheses suitable for further experimental follow-up. Given that the *SYK* example is just one example case

[9] Analogously as described in subsection 4.1.1 but with the S. cerevisiae version of KEGG instead.

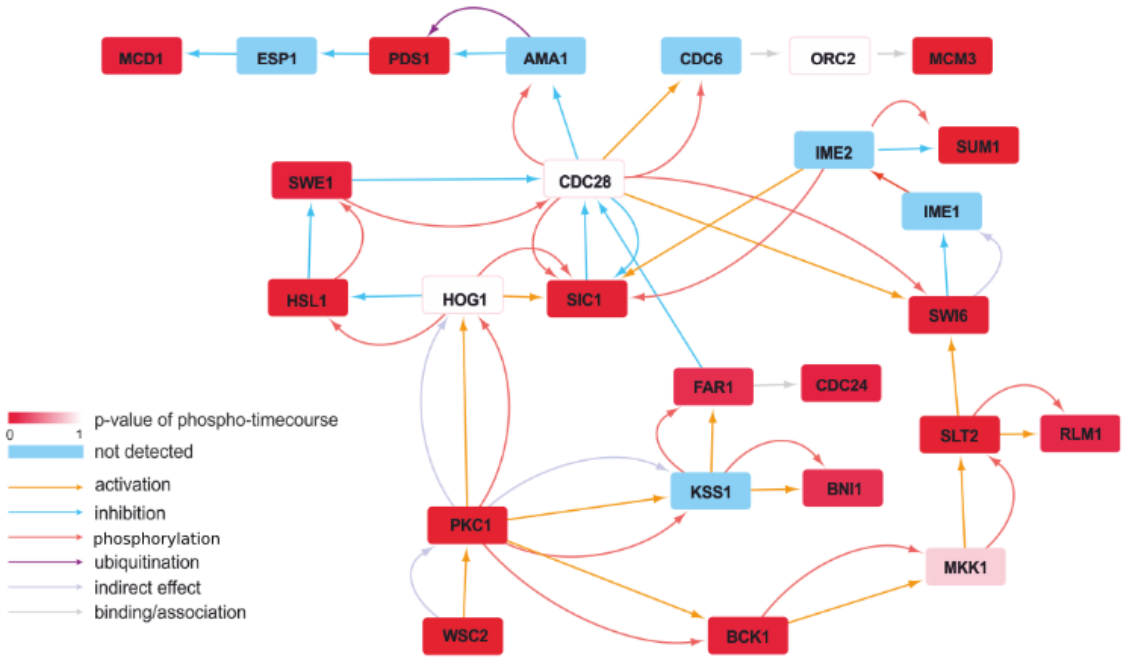


Figure 4.11: Cell cycle regulation in yeast on a time-resolved phosphoproteomic level. The DeRegNet subgraph captures important parts of the yeast cell cycle [ZWS+19a]. See main text, section 4.3. One conceptually important aspect of DeRegNet becomes especially apparent in the depicted subgraph: The subgraphs uncovered can gracefully deal with non-activated proteins not showing any significant phosphorylation pattern and even proteins not detected in the phosphoproteomic data by means of their network context.

of a network-defined cancer gene, this indicates that DeRegNet is a useful hypothesis generation tool for network-guided personalized cancer research. In summary, DeRegNet can provide sensible insight into a given omics experiment and may lead to novel and so far uncharacterized discoveries of gene/pathways involved in carcinogenesis. [10] Furthermore, I demonstrated the usefulness of subgraph-derived features for Machine Learning approaches to Personalized Medicine. Two further independent applications of DeRegNet, one in the context of the folate one-carbon metabolism in liver cancer cell lines in section 4.2 and another on the phosphoproteomic regulation of the yeast cell

[10] Note, for example, that we only presented and discussed network-defined cancer genes (e.g. *SYK*) for upregulated subgraphs, while I did not present the results of an analysis based on downregulated or generically deregulated (either up- or downregulated) subgraphs which would lead to similar opportunities [dere].

cycle in section 4.3, show further realized possibilities of DeRegNet. Together with the documented open-source software [DeRc] implementing the presented algorithm in a readily accessible manner, I hope that DeRegNet is a viable option for any researcher interested in network interactions in a high-throughput omics context.

Chapter 5

A REST-style API for *de-novo* pathway discovery

5.1 Introduction and Context

I claim that science[1] generally benefits from principled ways of enabling access to the generic components of scientific methodology, e.g. conceptual description or specification of methods, experimental protocols, research software, etc.. The most prominent way of distributing certain aspects of scientific research is certainly via adequate publication of that work [BF14]. In the field of Bioinformatics it is quite common that a body of scientific work is supported by, enabled by or even to some large extent consists itself of software components which are reusable to a certain degree. Common mechanisms of software distribution, research software or otherwise, are the distribution of the corresponding source code and/or precompiled binaries/packages. In principle, this enables anybody to install and run the software on a suitable system of their own. A common additional/alternative step is to provide access to the software's functionality remotely via computer networking mechanisms, one wide-spread system in use being the public internet and the world-wide-web [RFC]. The field of

[1] The societal/economic process, as well as the generated body of work/knowledge itself

Bioinformatics has a long history making methods and algorithms available via web services and webpages, see for example the annual special edition of the journal *Nucleic Acids Research* [nar20]. Embedded in the general theme of the increasing adoption of cloud computing in industry and research [BZdlP+10, LMS+14, LN18, BFK+19], recent[2] technological trends in implementing web-based services include containerization[3], container orchestration[4] and microservices [Ric18][5],[6]. This chapter describes the logic and implementation of a web-based API for the DeRegNet algorithm and associated functionality. All components of the API are containerized by means of Docker [Doca] images and orchestrated via *docker-compose* [Docb]. Finally, the API is meant to be embedded as one microservice out of many within a larger system of cooperating services. Nonetheless it provides all functionality necessary to find de-novo pathways by means of DeRegNet (see next section). The design of the API itself (in contrast to its implementation) is loosely guided by the principles of Representational State Transfer (REST) introduced by Fielding [Fie00].

5.2 The DeRegNet API

The DeRegNet API exposes the de-novo subgraph inference functionality of DeRegNet via a set of so called *resource types*. A concrete instance of a certain resource type is called a *resource (instance)* and is uniquely identifiable by an identifier (ID) generated and provided by the API on creation of the resource. Resources or collections of resources of a given resource type are addressable through a dedicated HTTP [Htt][7]

[2] The time-scale here is roughly 5 years with a reference of 2020; it is quite common to come across different time scales when witnessing discussions on the latest cloud technology, ranging from days to minutes.

[3] Containerization: The packaging of software by means of (Linux) container technology

[4] Container orchestration: The process to run and enable interactions between multiple software components which interact and are packaged as containers, e.g. Docker images

[5] Microservices: A set of smaller (network-based) software components with defined functionality which can interact to achieve higher order functionality

[6] There is certainly no shortage of other discussion threads, tools and attached keywords. For a plethora of relatively concrete entrypoints one can consult the CNCF's (Cloud Native Computing Foundation) cloud-native landscape [CNCa]

[7] HTTP: the Hyper-Text Transfer Protocol

endpoint which defines the possible queries for resources of a particular type. Furthermore resources can be referenced (by their respective ID) in queries and actions involving the HTTP endpoint of other resource types which provides the means to carry out actions involving resources of multiple different types. For the exact specification we refer the reader to [DeRd][8]. An interactive documentation of the API can be found here: [DeRa][9].

5.2.1 Resource types, resources and endpoints

As outlined, resources represent the data associated with the DeRegNet API and any resource is of a particular resource type. The DeRegNet API defines six resource types: *graphs*, *node sets*, *parameter sets*, *node scores*, *runs* and *subgraphs*. In the following I provide details on the resource types of the DeRegNet API and sketch the respective queries enabled by their respective endpoints. Every HTTP endpoint associated with a particular resource type allows for certain queries supported by canonical usage of HTTP verbs, request data and status codes [Htt]. For the complete and formal documentation of the endpoints provided by the DeRegNet API it is referred to [DeRa] and [DeRd]. For graphical overview on defined resource types and their basic relationships see figure 5.1.

Graphs. Resources of resource type *graph* represent a biomolecular network with respect to which DeRegNet can find subgraphs. This can for example correspond to a network derived from KEGG (see previous chapter) for a particular organism. A KEGG network restricted to just interactions of a certain type (for example phosphorylation/dephosphorylation) could correspond to another graph resource. A graph resource is meant to be reused across multiple runs (see below) of DeRegNet. The graph endpoint allows to define and upload new graphs which can then later be used as a base graph for finding subgraphs. This can canonically be achieved via POST

[8] https://sebwink.github.io/deregnet-rest/server/swagger/swagger.yaml
[9] https://sebwink.github.io/deregnet-rest/docs/index.html

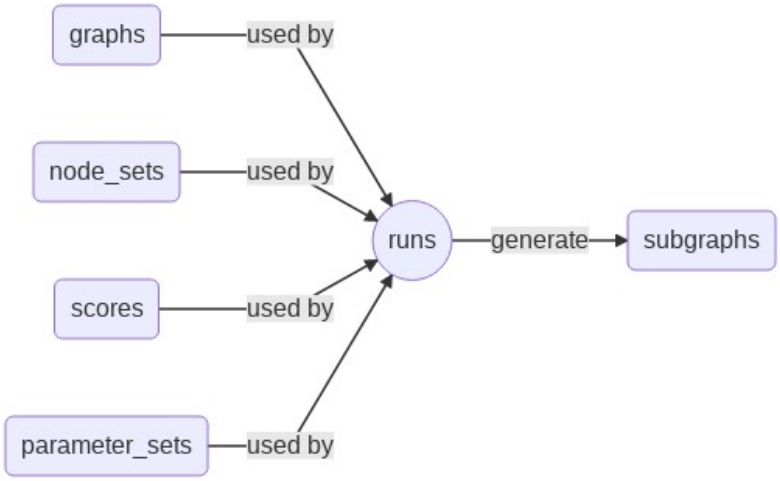


Figure 5.1: Conceptual dependencies between resource types in the DeReg-Net API. The API defines six resource types as described in more detail in the main text: graphs, node sets, (node) scores, parameter sets, runs and subgraphs. The four resource types on the left of the figure can be considered representing the data needed to run DeRegNet, namely an underlying *graph*, node *scores* (derived for example from gene expression data), different *node sets* and *parameters* to configure the algorithm. The *run* resource type then represents an actual run of the DeRegNet algorithm in order to generate *subgraphs* using the relevant data.

requests. GET requests allow to retrieve information on already registered graphs (number of nodes/edges etc.). Finally, graphs can be deleted. The data necessary for initial graph definition (besides the actual GraphML data representing the graph) is displayed in listing 5.1 while its representation once uploaded is shown in listing 5.2.

```
{
    "node_id_attr": <Node attribute with default IDs>,
    "name": <Name of graph>,
    "description": <Description of graph>
}
```

Listing 5.1: Initial graph info. The data necessary to define a graph initially (besides the GraphML representation of the network). node_id_attr represents the default node attribute used for mapping node score IDs to nodes in a given graph.

```
{
    "time_of_upload": <Timestamp of graph upload>,
    "name": <Name of graph>,
    "description": <Description of graph>,
    "id": <ID of graph>,
    "num_nodes": <Number of nodes in the graph>,
    "num_edges": <Number of edges in the graph>
}
```

Listing 5.2: General graph info. Once uploaded the API creates an ID, logs the timestamp of the update and registers some basic statistics on the graph.

Node sets. Resources of resource type *node set* represent sets of nodes which appear in some *graph*-type resources and are meant to be sets of biologically meaningful nodes which can be used as receptor and/or terminal nodes when running DeRegNet. One example would be to register a list of known oncogenes as a node set. The latter can then be referenced and used as receptor or terminal set in a given DeRegNet run. Nodes from a given node set can also be forced to be included or excluded during DeRegNet's search for optimal and suboptimal subgraphs. Just like a graph resource, a node set resource is meant to be used across many DeRegNet runs. The node set endpoint allows node set resources to be created, queried for information or deleted. The data necessary to define a node set is schematically shown in listing 5.3.

```
{
    "nodes": [
        <node 1>,
        <node 2>,
        ...,
        <node N>
    ],
    "name": <name of node set>,
    "description": <description of node set>
}
```

Listing 5.3: Node set data. A node set consists of a list of node labels.

Node scores. Node score resources represent the scores for the nodes/vertices of a given graph with respect to which DeRegNet should find subgraphs. For instance, the RNA-Seq profile of a TCGA case could represent a node score while the binary mutation indicator of a patient could be another node score resource. Note, that each TCGA case

would give rise to separate node score resources. Nodes in a node score resources have to be matched with node identifiers in the *graph* resource they are meant to be used with. Note, that the definition of the nodes scores as such is external of the DeRegNet API. Given a node score, the API allows the creation, retrieval and deletion of node scores as API resources, as well as their use in DeRegNet subgraph finding runs. The data necessary to define a node score is schematically shown in listing 5.4.

```
{
    "score_values": [
        <score of node 1>,
        <score of node 2>,
        ...,
        <score of node N>
    ],
    "node_ids": [
        <node 1>,
        <node 2>,
        ...,
        <node N>
    ],
    "name": <name of node score>
    "description": <description of node score>
}
```

Listing 5.4: Node score data. A node score consists of a list of node labels and a list of scalars with matching dimension containing the scores for the nodes referenced by the labels.

Parameter sets. DeRegNet allows for configuration parameters besides the biologically relevant data captured in the preceding resource types (*graph*, *node set* and *node score*). These parameters are represented by resources of type *parameter set*. Parameters which can be defined in a parameter set are for example the lower and upper bounds on subgraph size, the algorithm to be used (i.e. Dinkelbach's algorithm or the generalized Charnes-Cooper transform) or whether one wants to find maximal or minimal subgraphs with respect to the node score. The parameter set endpoint allows to organize parameter settings for DeRegNet runs, i.e. the type of algorithm used, minimal/maximal number of nodes in subgraphs, etc. The endpoint supports the canonical actions of creation, querying and deletion. Parameter sets are referenced by

IDs created and returned during creation of a parameter set. Additionally the endpoint allows to query for the default parameter setting used for DeRegNet runs. An example parameter set is shown in listing 5.5

```
{
    "model_sense": "max",
    "default_score": 0.0,
    "num_suboptimal": 1,
    "flip_orientation": false,
    "max_overlap": 10.0,
    "gap_cut": 0.05,
    "min_size": 15,
    "abs_values": true,
    "max_size": 50,
    "algorithm": "gcc"
}
```

Listing 5.5: An example parameter set. A parameter set allows to configure run resources. E.g. that optimization should stop once the relative gap is less than 0.05 ("gap_cut": 0.05) or to use Generalized Charnes-Cooper method ("algorithm": "gcc").

Runs. Given a *graph*, optionally one or more *node sets* as receptors, terminals, included or excluded nodes, a *node score* and a *parameter set* a *run* resource represents an actual run of DeRegNet in order to find subgraphs. The *run endpoint* allows to define runs of DeRegNet in order to find subgraphs relative to data defined and registered via the previous endpoints. With a POST request to the *run endpoint* one can instruct the API to find subgraphs within a given *graph* using a given *node score*. Optionally one can specify *node sets* as receptor or terminal nodes for the subgraphs. Another use of *node sets* is to instruct the run to find subgraphs which include (or exclude) nodes from a node set by force in any subgraph. Existing runs can be queried for their status by GET requests. Information returned are for example whether the run finished, whether it successfully found subgraphs and if so the IDs of these subgraphs. Each run also keeps track of its input resources, i.e. underlying graph resource, node score resource, potential node set resources and parameter set resources. The data necessary to define a subgraph run is schematically shown in listing 5.6 while the information structure concerning a run which is retrievable via a GET request to the run endpoint is shown

schematically in listing 5.7.

```
{
    "name": <Name of run resource>,
    "description": <Description of run resource>,
    "include_id": <ID of include node set resource>,
    "terminals_id": <ID of terminals node set resource>,
    "parameter_set_id": <ID of parameter set resource>,
    "score_id": <ID of node score resource>,
    "root": <Node ID of root node>,
    "receptors_id": <ID of receptors node set resource>,
    "exclude_id": <ID of exclude node set resource>,
    "graph_id": <ID of graph resource>
}
```

Listing 5.6: Data to define a run. In order to define a run, one needs to (optionally) reference node set resources for exclude, include, receptor and terminal set, optionally a fixed root node, a parameter set resource, graph resource and node score resource.

```
{
    "subgraph_ids": [
        <ID of subgraph resource 1>,
        <ID of subgraph resource 2>
    ],
    "description": "Description of run resource",
    "started": true,
    "id": <ID of run resource>,
    "done": true,
    "post_time": <timestamp>,
    "run_input": <See listing 5.6>
}
```

Listing 5.7: Basic run information. With a GET request to the run endpoint one can retrieve information on created runs. Among other things, it details its status in terms of completion ("done") and references the subgraphs found by means of subgraph resource IDs ("subgraph_ids"). Also, it logs all the input data used to define the run ("run_input"). This is highly useful for tracking data dependencies of generated subgraphs.

Subgraphs. Finally, subgraphs are the outcome of a DeRegNet run and are represented by resources of *subgraph* type. Besides some general information (Is the subgraph optimal or suboptimal? How many nodes and edges does it have? etc.) it also references the *run* resource which produced it. This allows to keep track of the context of the subgraph (What is the underlying graph? With which parameters was the DeRegNet run performed?). The subgraph endpoint also enables to download different representations of available subgraphs, namely in Simple Interaction Format (SIF) [Sim] and

GraphML format [Gra].

```
{
    "score": <Node score of subgraph>,
    "run_id": <ID of run resource which found the subgraph>,
    "optimal": <Optimality status of subgraph>,
    "root": <Node ID of determined root node>,
    "id": <ID of subgraph resource>,
    "num_nodes": <Number of nodes in subgraph>,
    "num_edges": <Number of edges in subgraph,
    "optimality_type": <Optimality type of subgraph>
}
```

Listing 5.8: Basic subgraph information. The score of the subgraph is listed, which run generated it, whether the model representing the subgraph was solved to optimality and also if it is the optimal subgraph or a suboptimal one ("optimality_type").

Example workflow

Figure 5.2 shows a typical example workflow of an API user interacting with the API[10]. It details the upload of a custom graph to find subnetworks in, the registration of a node score and a node set. Finally the user makes a request to find subgraphs referencing data uploaded in the previous steps. Upon success of the subgraph finding run, the user can retrieve the found subnetworks by means of the respective endpoint.

5.2.2 Architecture and implementation

General architecture

The architecture of the DeRegNet API is based mostly on a straightforward variation on standard multi-tier architecture [Sch09] for client-server systems as ubiquitously applied in industry and academia. It is based on three layers/tiers, namely a *server layer*, a *data layer* and a *worker layer*[11]. The architecture is presented graphically in figure 5.3. The server layer provides the entry point for users of the API who are not meant to interact directly with either the data or the worker layer. It provides the

[10] *Interaction* can of course also mean automated programmatic access in this context.
[11] One can consider any HTTP client a presentation layer of the API.

interface for all API functionality and implements it by addressing the data and worker layers. The Data layer consists of three components: 1) a database, keeping track of the API's resources, 2) a cache, keeping track of frequently accessed information and making it accessible without the need to address the database directly and finally 3) a job queue which registers requested subgraph runs. The latter are carried out by the worker layer which listens to new tasks in the job queue and then extracts all necessary information for the subgraph request from the database upon the decision to carry out a particular run in the queue. The main reason of this three layer architecture is the possibility to independently scale the the different API components. For example, the worker layers solve the fractional integer linear programs associated with DeRegNet and hence have different hardware requirements than either database or application servers which merely handle a certain amount of CRUD[12] operations. The server as well as the worker layers are stateless in the sense that all data is stored exclusively in the data layer. This makes these components readily scalable within container orchestration platforms like Docker Swarm [Docd] or Kubernetes [Kub]. By choice, the DeRegNet API does not provide/implement standard API features such as authentication/authorization, (user dependent) rate limiting or traffic monitoring by itself. Instead, for these and other features, the API is meant to rely on a so called API gateway [Ric18][13],[14]. An API gateway proxies all requests to a set of microservices which in turn are usually only accessible through the gateway. The API gateway then can be used to implement generic and common functionality for the microservices like, as mentioned, authentication, authorization, rate limits, traffic monitoring, aggregation of service endpoints, etc. [Ric18]. See figure 5.4 for the role of an API gateway for the DeRegNet API in general and for authorization/authentication in particular.

[12] CRUD: Create-Read-Update-Delete

[13] Examples of API gateways are Kong [Kon], Gloo [Glo] or krakend [Kra]. See also [CNCb]. For more information on the API gateway pattern in general and its intended use and enabled possibilities, see [Ric18, ric]

[14] Of course, instead of and off-the-shelf API gateway one can also imagine a custom proxy for the DeRegNet API.

Authentication and Authorization

The currently realized mode of authentication/authorization for the DeRegNet API is as follows. Firstly, the DeRegNet API does not handle authentication at all. Secondly, to enable authorization, every resource is associated to a *user* of the API. In terms of a API resource, a user is nothing more than a *user* field in the actual data representation of the resource. This then allows filtering of resources with respect to a given user name/identifier. The API server itself is agnostic about how these user names relate to any surrounding context of authentication and where they originally come from. User management is meant to be carried out by the API gateway and any associated user management services[15]. Upon handling of any HTTP request, the API server layer tries to extract a user name/identifier provided via a JSON Web Token (JWT)[16] [Jwt] and then returns only those resources matching the given user or associates a newly generated resource to that user. The DeRegNet server layer as such also does not concern itself with whether the provided JWT is validated or the initial creation of the JWT. It only extracts the user information from a provided JWT. The creation/validation of the JWT is also meant to be carried out by the API gateway and associated services. Under the assumption that the DeRegNet server layer is only reachable from the API gateway (servers) the contents of the JWT can be trusted and the business logic can be carried out based on the provided user name/identifier[17]. See figure 5.4 for a schematic workflow encapsulating the just outlined authentication/authorization logic.

The advantage of the outlined mode of handling authentication and authorization is that the DeRegNet API does not need to make any assumptions about any authentication mechanisms and only needs to extract information relevant for authorization

[15] Often user management and authorization and authentication services can be provided by the API gateway itself. Alternatively one can use external commercial services or rely on open source solutions. One open source solution for authorization and authentication is the ORY stack [Ory]

[16] JWT: cryptographically signed JSON data which allows somebody to verify that the JSON data encoded in the token was issued by a certain party.

[17] If no user name/identifier is provided in the request, resources are filtered/associated to a generic anonymous user.

from an externally cryptographically validated JWT. Hence, in conjunction with an API gateway the DeRegNet API is amenable to any authentication mechanism provided by the former. This can be as simple as HTTP Basic Authentication and as complex as fully fledged OAuth2 [oau] and OpenId Connect [Ope] workflows. The factor determined by the implementation of the DeRegNet API itself is of course the mapping of JWT-encoded (user) information provided from the gateway to the authorization logic within the DeRegNet API. As detailed above, this logic is based on a one-to-one mapping of *users* to *resources*.

Example workflow with internal events

The sequence diagram 5.5 details events in a typical workflow of a series of requests by an API user. In contrast to figure 5.2 it shows the internal events the implementation of the API carries out in order to be able to serve the requests. This entails the server layer interacting with the data layer to create and retrieve resources and to queue subgraph jobs into the job queue, as well as the worker layer communicating with the job queue and the data layer in order to be able to start subgraph finding runs. The interaction logic from the users perspective represents the upload of a regulatory network, the upload of a node score and the subsequent definition of a subgraph finding run with respect to the previously defined resources. In the end, subgraph found by the run are retrieved and downloaded in GraphML format. While the interactions displayed between user and server layer are analogous to the sequence diagram 5.2, the interactions between server and data layer, as well as those between data and worker layer are internal to the API's implementation.

Implementation technology

At the time of writing, the server layer is implemented via OpenAPI/Swagger server stub generation [Swa] for the Python Flask [Fla] web framework from the API specification and corresponding implementation of DeRegNet-specific CRUD logic for the

generated endpoints. The data layer consists of MongoDB [Mon] as the database and Redis [Red] serving as a cache and a job queue via Celery [Cela]. The worker layer consists of Celery workers [Celb] implemented in Python utilizing the DeRegNet Python package. The server layer and the worker layer are encapsulated in Docker images which can be run independently given instances of MongoDB and Redis[18].

5.3 Summary and Discussion

The implemented API provides all necessary base functionality needed to find subgraphs by means of the DeRegNet algorithm (see chapter 3). At the same time it is designed and implemented with the embedding into a larger context in mind by not relying on too many assumptions about common interaction factors in a microservice setting (authorization and user identity, authentication, rate limiting, etc.) but also providing a corresponding integration surface for factors which are determined by the API itself (such as resource access authorization logic as such). The API relies on the open source implementation of DeRegNet [DeRc] and is itself open source [DeRb] under the BSD 3-Clause license [BSD]. The associated software components are packaged as Docker images.

Primary extension points for the API would be additional microservices to implement more specialized functionality in general and for upstream and downstream subgraph analysis in particular. In general, the API would benefit from integration with an identifier mapping service for biomolecular identities. Currently, the mapping of identifiers of node scores to identifiers in underlying networks is only supported manually through careful encoding in the definition of resources and manual linking when defining runs. Concerning *upstream services* this mainly relates to services encapsulating specific means of defining and working with underlying regulatory networks, node

[18]Note that MongoDB and Redis can be deployed as high-availability clusters. This is outside the scope of this thesis but reiterates the point that all layers of the DeRegNet API, server, worker, and the components of the data layer can be scaled independently.

scores and node sets. As an example, one can imagine a service which facilitates the direct import of NDEx [PCW+15, PCP+17, Ndeb, Ndea] networks as base networks for use with the DeRegNet API, as well as other means to define suitable networks for use with DeRegNet like *sbml4j* [sbm]. Similarly, services to define node scores would be highly useful, for example to define canonical global as well as personalized node scores based on the omics data in the TCGA datasets [TCW15]. Other possibilities are services to seamlessly link the output of mutational variant annotation pipelines such as ClinVAP [SSD+20] with suitable node scores. Streamlined node set definition based on canonical sources of gene sets are another obvious extension point. Organizing the process of finding personalized subgraphs for many patients for multiple omics node scores with possibly various modes of DeRegNet application modes (known tumor suppressor/oncogenes genes as receptors/terminals, etc.) can become a organizational challenge. Hence, services supporting these and other complex application scenarios can come in handy and would touch upstream as well as downstream services functionality. The most useful candidate for a *downstream service* is visualization and *BioGraphVisArt* [bio] already provides an implementation. Additional tools relate to the comparison of inferred subgraphs, for example in a personalized setting comparing subgraphs of different patients. See also *BioGraphVisArt* [bio]. The finding of *network-defined cancer genes* (see algorithm 10 in chapter 3) also lends itself to implementation in terms of a downstream service with semi-automated input from potential upstream services for TCGA node scores. While there are certainly more possibilities one can think of right off the bat, I leave it at this.

In purely technical terms, the transfer of the implementation to the Kubernetes [Kub] container orchestration platform is desirable.

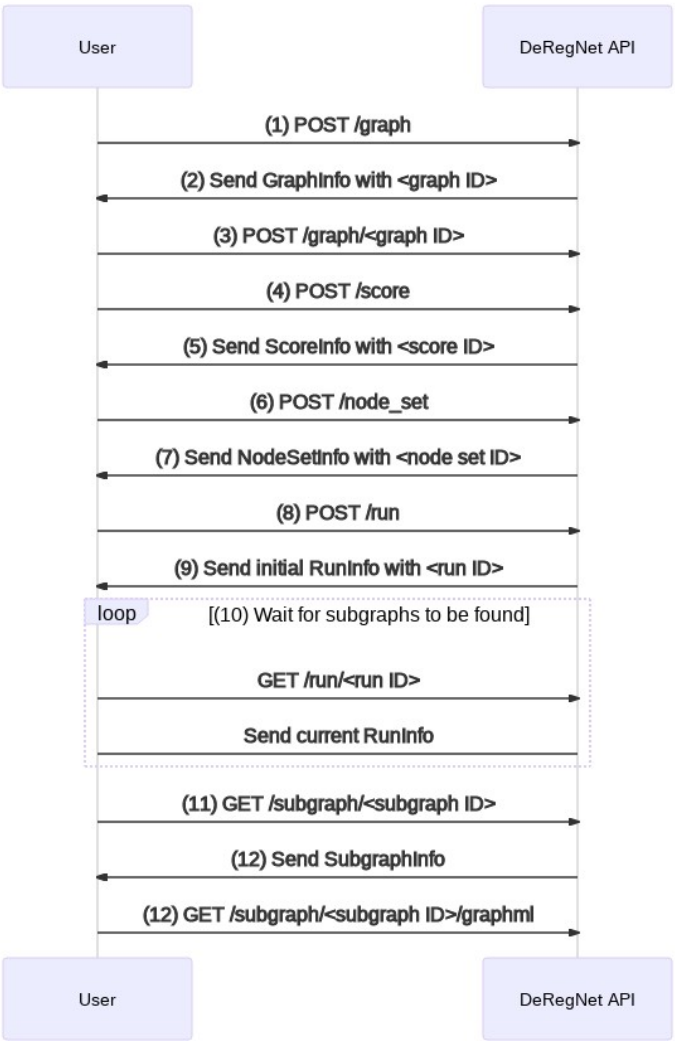


Figure 5.2: Example flow for a user interacting with the DeRegNet API. 1-3) By POSTing a graph a user can register a graph which can then be referenced by an ID. Next (4) and (5) represent the upload of a node score resource and the return of and ID identifying the created node score by the API. Similarly (6) and (7) represent the creation of a node set. The actual command to initiate the finding of subgraphs happens in (8) with a POST request to the run endpoint referencing underlying network, node scores and any other associated resources by their respective IDs. In (9) the user received the ID of the created run which can subsequently be used to query for the status of the run (10). Once the run finished the *SubgraphInfo* response will contain references to IDs of subgraphs found by the run. The subgraphs can be queried for some initial information (11), (12) and finally be downloaded (12) in GraphML format.

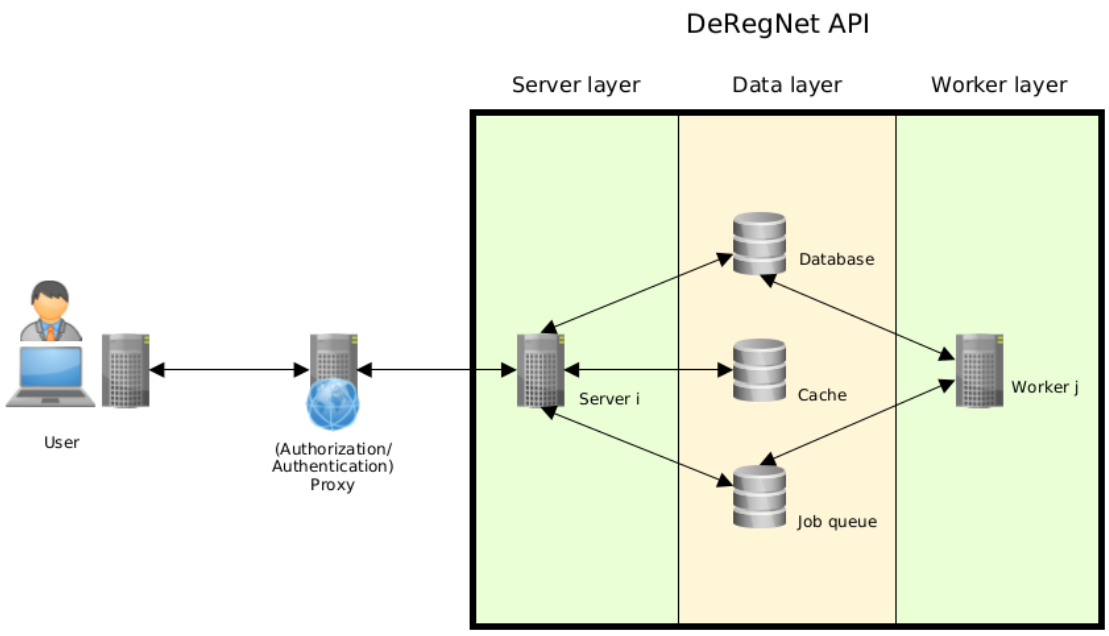


Figure 5.3: Conceptual architecture of the DeRegNet API. The architecture can be roughly divided into three different layers: 1) the *server* layer which provides all user-facing endpoints and implements the CRUD logic of the API 2) the *data* layer which consists of a database, a cache and a job queue and 3) the *worker* layer which is a pool of servers which actually run DeRegNet in order to find subgraphs. Note, that the *server* and *worker* layer are stateless in the sense that they do not themselves store any data and merely carry out functions with data provided externally (to these layers) via the *data* layer. For clarity the stateless layers of the API are colored in pale green while the layer(s) holding persistent data are colored pale yellow.

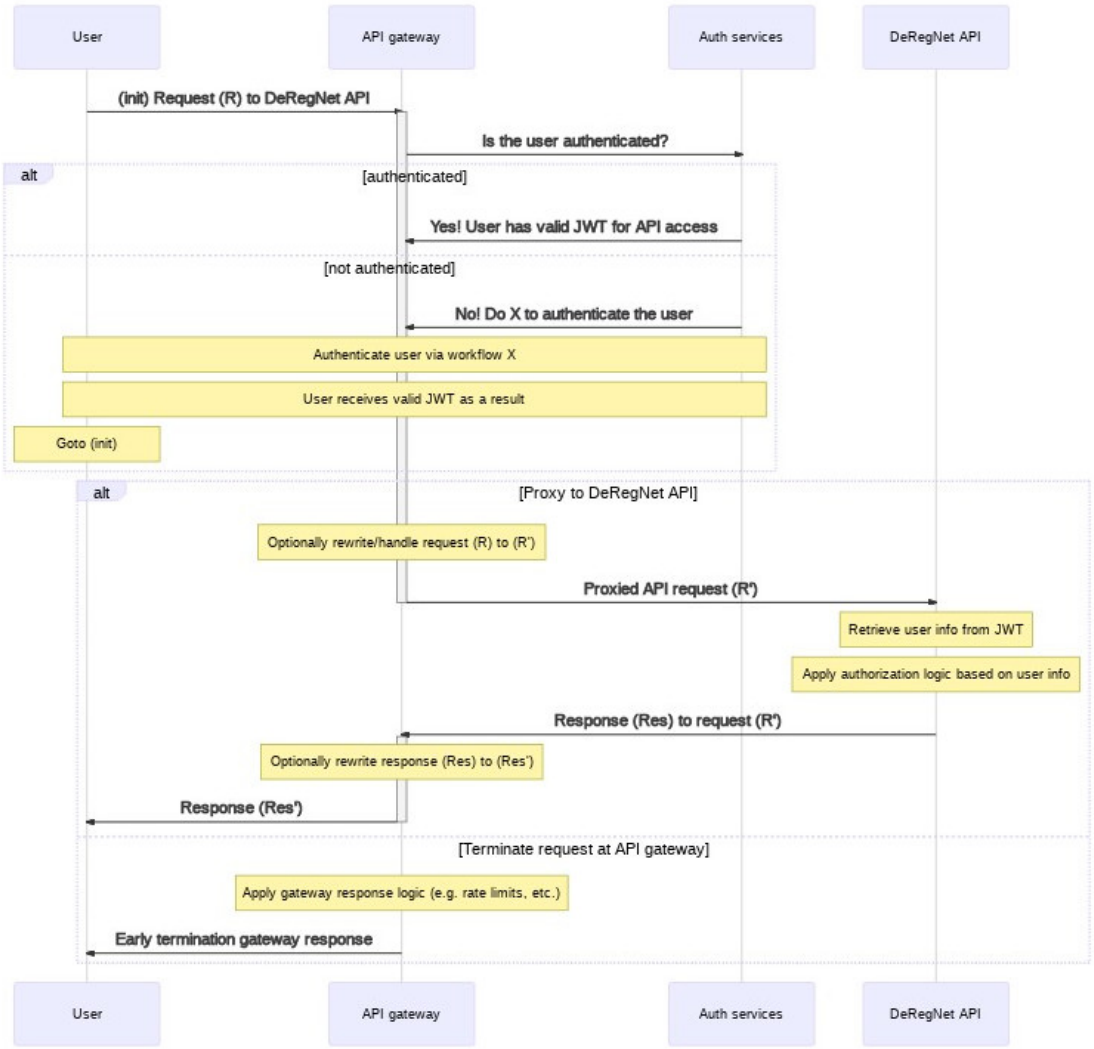


Figure 5.4: The role of the API gateway in general and for authentication/authorization in particular. An API gateway proxies all (end) user requests to a target services/API. By means of *Auth services* it can be used to implement authorization and authentication generically for several services in a compatible way. *Auth services* can be provided by the API gateway itself, a dedicated stack like [Ory] or even third-party identity providers. In addition to the prominent authorization/authentication functionality, an API gateway can also implement various other generic features. One example would be *rate limits*: Based on the user's identity known to the gateway, the gateway can enforce personalized rate limits for the services exposed through the API gateway.

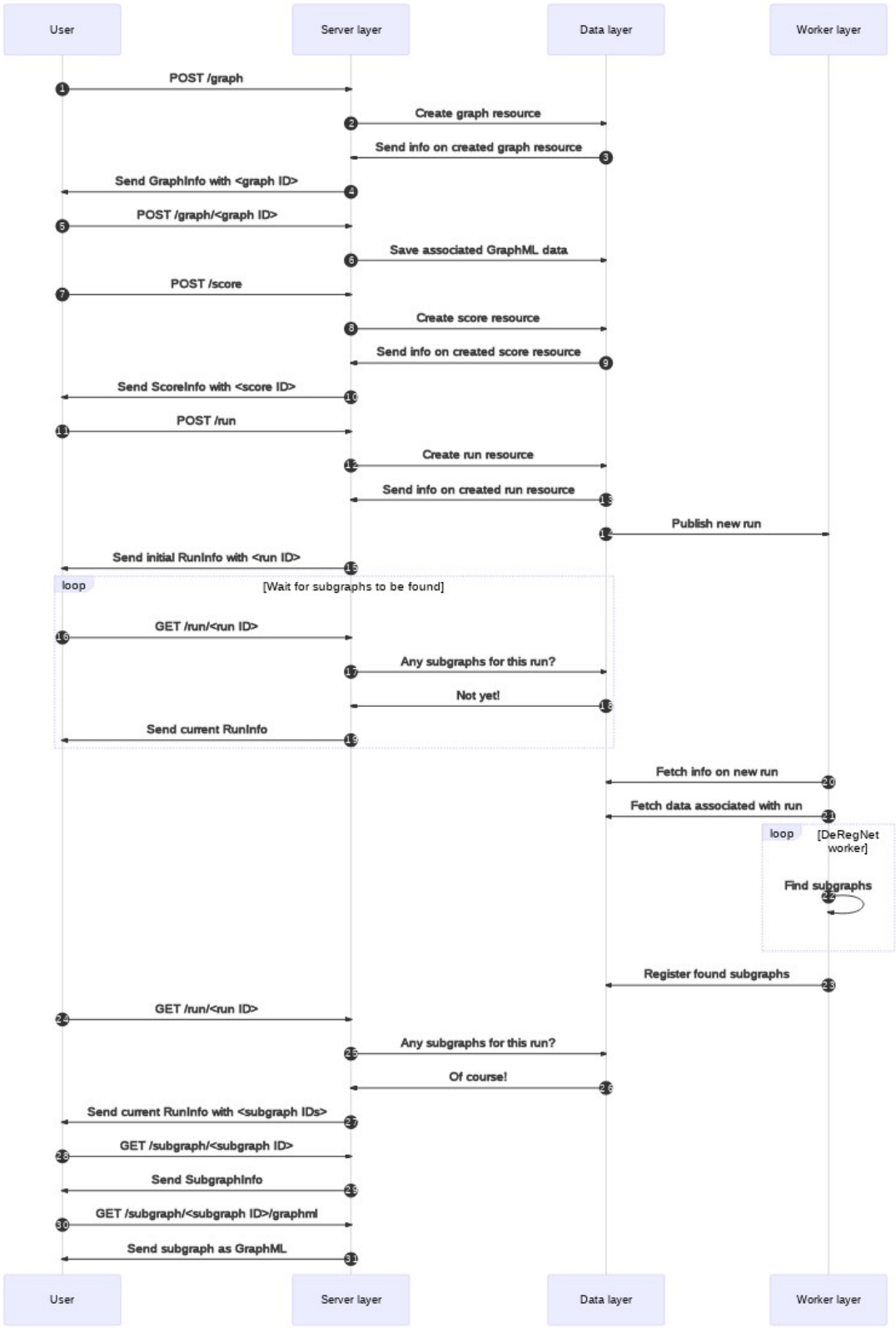


Figure 5.5: Example flow for a user interacting with the DeRegNet API. See main text.

Chapter 6

Conclusion

The preceding chapters introduced an algorithm for de-novo pathway identification / deregulated subnetwork detection. The mathematical underpinnings, technical solution procedures and implementation aspects the algorithm were discussed in chapter 3, as well as a statistical model for proposed method and a simulation-based benchmark with respect to its closest methodological relative [BRK+12]. Special attention was given to the methods employed to solve the fractional integer programming model which constitutes the heart of the algorithm, including a description of the general solution technology as well as problem-specific heuristics. Chapter 4 then detailed concrete applications of the proposed algorithm to omics datasets. Application to the hepatocellular carcinoma dataset of the The Cancer Genome Atlas (TCGA) showed the ability of the algorithm to extract regulatory patterns. In particular, it could reconstruct the transcriptional activation of the *WNT* pathway in the context of liver cancer. Furthermore, it proves its value as an heuristic hypothesis generation tool by uncovering various interesting pathway crosstalks in term of omics deregulation. In the context of the application to the liver cancer TCGA dataset, this thesis outlined approaches to uncover personalized molecular patterns based on a patient-specific application of the algorithm. It was shown how these personalized patterns can then in turn be used to create hypotheses concerning genes which provide phenotypic effect by means of their network context in conjunction with its deregulation. I detailed one

such gene and the associated pattern, namely Spleen Tyrosine Kinase (*SYK*). Also it was shown that predictive features derived from patient-specific subgraphs can help construct Machine Learning models. In particular, the proposed features improve upon just clinical features, which is known to remain challenging problem [YVAO+14]. Two further applications of the proposed algorithm were highlighted, underlining its various application scenarios. One application could provide useful network insights concerning folate one-carbon metabolism in hepatocellular carcinoma [WTD+], while another case study could identify regulatory patterns of the *S. cerevisiae* cell cycle based on phosphoproteomic time series data [ZWS+19a]. The final chapter 5 described the design and implementation of a web-based Application Programming Interface (API) for the outlined algorithm, making a web-based deployment of the methods described in the preceding chapters readily achievable for any interested party.

Bibliography

[AB11] E. Althaus and M. Blumenstock. Algorithms for the maximum weight connected subgraph and prize-collecting steiner tree problems. *11th DIMACS Implementation Challenge in Collaboration with ICERM*, 2011. 18, 31

[ABG08] O. Aalen, O. Borgan, and H. Gjessing. *Survival and Event History Analysis: A Process Point of View*. Springer, 2008. 75, 76

[AF05] W.P. Adams and R.J. Forrester. A simple recipe for concise mixed 0-1 linearizations. *Operations Research Letters*, 33:55–61, 2005. 41, 45

[AFG04] W.P. Adams, R.J. Forrester, and F. Glover. Comparison and enhancement strategies for linearizing mixed 0-1 quadratic programs. *Discrete Optimization*, 1:99–120, 2004. 41, 45

[AFK+12] N. Alcaraz, T. Friedrich, T. Kotzing, A. Krohmer, J. Muller, J. Pauling, and J. Baumbach. Efficient key pathway mining: combining networks and OMICS data. *Integr Biol (Camb)*, 4(7):756–764, Jul 2012. 6, 15

[ALDH+16] N. Alcaraz, M. List, M. Dissing-Hansen, M. Rehmsmeier, Q. Tan, J. Mollenhauer, H. J. Ditzel, and J. Baumbach. Robust de novo pathway enrichment with KeyPathwayMiner 5. *F1000Res*, 5:1531, 2016. 6, 15

[AM13] J. N. Anastas and R. T. Moon. WNT signalling pathways as therapeutic targets in cancer. *Nat. Rev. Cancer*, 13(1):11–26, Jan 2013. 69

[AMH13] A.F.M. Altelaar, J. Munoz, and A.J.R. Heck. Next-generation proteomics: towards an integrative view of proteome dynamics. *Nat. Rev. Genet.*, 14:35–48, 2013. 1

[ÁMLM13a] Eduardo Álvarez-Miranda, Ivana Ljubić, and Petra Mutzel. *The Maximum Weight Connected Subgraph Problem*, pages 245–270. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 18, 31, 52

[ÁMLM13b] Eduardo Álvarez-Miranda, Ivana Ljubić, and Petra Mutzel. The rooted maximum node-weight connected subgraph problem. In Carla Gomes and Meinolf Sellmann, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 300–315, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 18, 31, 52

[AMOT90] R. K. Ahuja, K. Mehlhorn, J. Orlin, and R. E. Tarjan. Faster algorithms for the shortest path problem. *Journal of the ACM*, 37(2), 1990. 53

[AMS17] Eduardo Alvarez-Miranda and Markus Sinnl. A relax-and-cut framework for large-scale maximum weight connected subgraph problems. *Computers & Operations Research*, 87:63 – 82, 2017. 18

[Anz74] Y. Anzai. On integer fractional programming. *J. Operations Research Soc. of Japan*, 17(1):49–66, March 1974. 41

[APB+14] N. Alcaraz, J. Pauling, R. Batra, E. Barbosa, A. Junge, A. G. Christensen, V. Azevedo, H. J. Ditzel, and J. Baumbach. KeyPathwayMiner 4.0: condition-specific pathway analysis by combining multiple omics studies and networks with Cytoscape. *BMC Syst Biol*, 8:99, Aug 2014. 6, 15

[ARF13] A. Arzumanyan, H. M. Reis, and M. A. Feitelson. Pathogenic mechanisms in HBV- and HCV-associated hepatocellular carcinoma. *Nat. Rev. Cancer*, 13(2):123–135, Feb 2013. 67

[AS13] N. Atias and R. Sharan. iPoint: an integer programming based algorithm for inferring protein subnetworks. *Mol Biosyst*, 9(7):1662–1669, Jul 2013. 6, 23, 24

[BAG+17] R. Batra, N. Alcaraz, K. Gitzhofer, J. Pauling, H. J. Ditzel, M. Hellmuth, J. Baumbach, and M. List. On the performance of de novo pathway enrichment. *NPJ Syst Biol Appl*, 3:6, 2017. 5, 6, 14, 15, 58

[BAM+12] G. Bertino, A. Ardiri, M. Malaguarnera, G. Malaguarnera, N. Bertino, and G. S. Calvagno. Hepatocellualar carcinoma serum markers. *Semin. Oncol.*, 39(4):410–433, Aug 2012. 67

[BB05] R. Bataller and D. A. Brenner. Liver fibrosis. *J. Clin. Invest.*, 115(2):209–218, Feb 2005. 78

[BBBB+11] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J. M. Francois, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proc. Natl. Acad. Sci. U.S.A.*, 108(2):882–887, Jan 2011. 6, 15

[BBI07] A. Beyer, S. Bandyopadhyay, and T. Ideker. Integrating physical and genetic maps: from genomes to interaction networks. *Nat. Rev. Genet.*, 8(9):699–710, Sep 2007. 4

[BCO+08] A. Bellahcene, V. Castronovo, K. U. Ogbureke, L. W. Fisher, and N. S. Fedarko. Small integrin-binding ligand N-linked glycoproteins (SIBLINGs): multifunctional proteins in cancer. *Nat. Rev. Cancer*, 8(3):212–226, Mar 2008. 70

[Ber06] Tim Berthold. *Primal Heuristics for Mixed Integer Programs*. PhD thesis, Technische Universität Berlin, 2006. 52

[BF14] S Batling. and S. Friesike. *Opening Science - The Evolving Guide on How the Internet is Changing Research, Collaboration and Scholarly Publishing*. Springer, 2014. 89

[BFK+19] P Belmann, B Fischer, J Krüger, M Prochazka, H Rasche, M Prinz, M Hanussek, M Lang, F Bartusch, B Gläßle, J Krüger, A Pöhler, and A Sczyrba. de.nbi cloud federation through elixir aai [version 1; peer review: 2 approved, 1 not approved]. *F1000Research*, 8(842), 2019. 90

[BGL11] A. L. Barabasi, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.*, 12(1):56–68, Jan 2011. 4

[Big11] M. D. Biggin. Animal transcription networks as highly connected, quantitative continua. *Dev. Cell*, 21(4):611–626, Oct 2011. 4

[bio] Biographvisart github repository. https://github.com/MirjamFi/BioGraphVisart. Accessed: 2020-12-31. 102, 155

[BM80] E. Balas and C. H. Martin. Pivot-and-complement: A heuristic for 0-1 programming. *Management science*, 26:86–96, 1980. 52

[BRK+12] C. Backes, A. Rurainski, G. W. Klau, O. Muller, D. Stöckel, A. Gerasch, J. Kuntzer, D. Maisel, N. Ludwig, M. Hein, A. Keller, H. Burtscher, M. Kaufmann, E. Meese, and H. P. Lenhof. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic Acids Res.*, 40(6):e43, Mar 2012. 6, 18, 22, 23, 24, 25, 26, 28, 29, 31, 33, 34, 35, 37, 38, 41, 50, 58, 59, 61, 62, 63, 64, 107

[BSD] Bsd 3-clause open source license. https://opensource.org/licenses/BSD-3-Clause. Accessed: 2020-12-31. 58, 101

[BSG16] C. P. Bracken, H. S. Scott, and G. J. Goodall. A network-biology perspective of microRNA function and dysfunction in cancer. *Nat. Rev. Genet.*, 17(12):719–732, 12 2016. 4

[BSW04] E. Balas, S. Schmieta, and C. Wallace. Pivot and shift - a mixed integer programming heuristic. *Discrete Optimization*, 1:3–12, 2004. 52

[BWB17] Austin Buchanan, Yiming Wang, and Sergiy Butenko. Algorithms for node-weighted steiner tree and maximum-weight connected subgraph. *Networks*, 72, 04 2017. 18

[BZdlP+10] Javier Bajo, Carolina Zato, Fernando de la Prieta, Ana de Luis, and Dante Tapia. Cloud computing in bioinformatics. In Andre Ponce de Leon F. de Carvalho, Sara Rodríguez-González, Juan F. De Paz Santana, and Juan M. Corchado Rodríguez, editors, *Distributed Computing and Artificial Intelligence*, pages 147–155, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 90

[CAC+19] Sarvenaz Choobdar, Mehmet E. Ahsen, Jake Crawford, Mattia Tomasoni, Tao Fang, David Lamparter, Junyuan Lin, Benjamin Hescott, Xiaozhe Hu, Johnathan Mercer, Ted Natoli, Rajiv Narayan, Fabian Aicheler, Nicola Amoroso, Alex Arenas, Karthik Azhagesan, Aaron Baker, Michael Banf, Serafim Batzoglou, Anais Baudot, Roberto Bellotti, Sven Bergmann, Keith A. Boroevich, Christine Brun, Stanley Cai, Michael Caldera, Alberto Calderone, Gianni Cesareni, Weiqi Chen, Christine Chichester, Lenore Cowen, Hongzhu Cui, Phuong Dao, Manlio De Domenico, Andi Dhroso, Gilles Didier, Mathew Divine, Antonio del Sol, Xuyang Feng, Jose C. Flores-Canales, Santo Fortunato, Anthony Gitter, Anna Gorska, Yuanfang Guan, Alain Guénoche, Sergio Gómez, Hatem Hamza, Andras Hartmann, Shan He, Anton Heijs, Julian Heinrich, Ying Hu, Xiaoqing Huang, V. Keith Hughitt, Minji Jeon, Lucas Jeub, Nathan T. Johnson, Keehyoung Joo, InSuk Joung, Sascha Jung, Susana G. Kalko, Piotr J. Kamola, Jaewoo Kang, Benjapun Kaveelerdpotjana, Minjun Kim, Yoo-Ah Kim, Oliver Kohlbacher, Dmitry Korkin, Kiryluk Krzysztof, Khalid Kunji, Zoltan Kutalik, Kasper Lage, Sean Lang-Brown, Thuc Duy Le, Jooyoung Lee, Sunwon Lee, Juyong Lee, Dong Li, Jiuyong Li, Lin Liu, Antonis Loizou, Zhenhua Luo, Artem Lysenko, Tianle Ma, Raghvendra Mall, Daniel Marbach, Tomasoni Mattia, Mario Medvedovic, Jorg Menche, Elisa Micarelli, Alfonso Monaco, Felix Müller, Oleksandr Narykov, Thea Norman, Sungjoon Park, Livia Perfetto, Dimitri Perrin, Stefano Pirro, Teresa M. Przytycka, Xiaoning Qian, Karthik Raman, Daniele Ramazzotti, Emilie Ramsahai, Balaraman Ravindran, Philip Rennert, Julio Saez-Rodriguez, Charlotta Schärfe,

Roded Sharan, Ning Shi, Wonho Shin, Hai Shu, Himanshu Sinha, Donna K. Slonim, Lionel Spinelli, Suhas Srinivasan, Aravind Subramanian, Christine Suver, Damian Szklarczyk, Sabina Tangaro, Suresh Thiagarajan, Laurent Tichit, Thorsten Tiede, Beethika Tripathi, Aviad Tsherniak, Tatsuhiko Tsunoda, Denes Türei, Ehsan Ullah, Golnaz Vahedi, Alberto Valdeolivas, Jayaswal Vivek, Christian von Mering, Andra Waagmeester, Bo Wang, Yijie Wang, Barbara A. Weir, Shana White, Sebastian Winkler, Ke Xu, Taosheng Xu, Chunhua Yan, Liuqing Yang, Kaixian Yu, Xiangtian Yu, Gaia Zaffaroni, Mikhail Zaslavskiy, Tao Zeng, Jitao D. Zhang, Lu Zhang, Weijia Zhang, Lixia Zhang, Xinyu Zhang, Junpeng Zhang, Xin Zhou, Jiarui Zhou, Hongtu Zhu, Junjie Zhu, Guido Zuccon, Gustavo Stolovitzky, Zoltan Kutalik, Lenore J. Cowen, and The DREAM Module Identification Challenge Consortium. Assessment of network module identification across complex diseases. *Nature Methods*, 16(9):843–852, Sep 2019. 157

[CBD10] D. S. Chen, R. G. Batson, and Y. Dang. *Applied Integer Programming*. Wiley, 2010. 48

[CC62] A. Charnes and W.W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quaterly*, 9:181–186, 1962. 41, 43, 136

[CCL+15] X. Chen, S. I. Chen, X. A. Liu, W. B. Zhou, R. R. Ma, and L. Chen. Vav3 oncogene is upregulated and a poor prognostic factor in breast cancer patients. *Oncol Lett*, 9(5):2143–2148, May 2015. 79

[CCZ14] M. Conforti, G. Cornuéjols, and G. Zanbelli. *Integer Programming*. Springer, 2014. 48, 50

[CDK13] R. Caspi, K. Dreher, and P. D. Karp. The challenge of constructing, classifying, and representing metabolic pathways. *FEMS Microbiol. Lett.*, 345(2):85–93, Aug 2013. 4

[CDS+10] E. Cerami, E. Demir, N. Schultz, B. S. Taylor, and C. Sander. Automated network analysis identifies core pathways in glioblastoma. *PLoS ONE*, 5(2):e8918, Feb 2010. 5

[Cela] Celery. https://docs.celeryproject.org/en/stable/getting-started/introduction.html. Accessed: 2020-12-31. 101

[Celb] Celery workers. https://docs.celeryproject.org/en/stable/userguide/workers.html#guide-workers. Accessed: 2020-12-31. 101

[CGD+11] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res.*, 39(Database issue):D685–690, Jan 2011. 66

[CMMGE+12] C. Citterio, M. Menacho-Marquez, R. Garcia-Escudero, R. M. Larive, O. Barreiro, F. Sanchez-Madrid, J. M. Paramio, and X. R. Bustelo. The rho exchange factors VAV2 and VAV3 control a lung metastasis-specific transcriptional program in breast cancer cells. *Sci Signal*, 5(244):ra71, Oct 2012. 79

[CN12] H. Clevers and R. Nusse. Wnt/β-catenin signaling and disease. *Cell*, 149(6):1192–1205, Jun 2012. 70

[CNCa] Cloud native computing foundation (cncf) landscape. https://landscape.cncf.io/. Accessed: 2020-12-31. 90

[CNCb] Cncf api gateways. https://landscape.cncf.io/category=api-gateway&format=card-mode&grouping=category. Accessed: 2020-12-31. 98

[CRH+15] P. Creixell, J. Reimand, S. Haider, G. Wu, T. Shibata, M. Vazquez, V. Mustonen, A. Gonzalez-Perez, J. Pearson, C. Sander, B. J. Raphael, D. S. Marks, B. F. F. Ouellette, A. Valencia, G. D. Bader, P. C. Boutros, J. M. Stuart, R. Linding, N. Lopez-Bigas, and L. D. Stein. Pathway and network analysis of cancer genomes. *Nat. Methods*, 12(7):615–621, Jul 2015. 4

[D'E13] P. D'Eustachio. Pathway databases: making chemical and biological sense of the genomic data flood. *Chem. Biol.*, 20(5):629–635, May 2013. 2

[DeRa] Deregnet api documentation. https://sebwink.github.io/deregnet-rest/docs/index.html. Accessed: 2020-10-14. 91

[DeRb] Deregnet api github repository. https://github.com/sebwink/deregnet-rest. Accessed: 2020-10-14. 101

[DeRc] Deregnet github repository. https://github.com/sebwink/deregnet. Accessed: 2020-10-14. 58, 59, 64, 66, 88, 101

[DeRd] Deregnet openapi specification. https://sebwink.github.io/deregnet-rest/server/swagger/swagger.yaml. Accessed: 2020-10-14. 91

[dere] Deregnet tcga github repo. https://github.com/sebwink/deregnet-tcga. Accessed: 2020-12-31. 87

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 53

[Dij72] E. W. Dijkstra. *A discipline of Programming*. Prentice-Hall, 1972. 51

[Din62] W. Dinkelbach. Die maximierung eines quotienten zweier linearer funktionen unter linearen nebenbedingungen. *Z. Wahrscheinlichkeitstheorie*, 1:141–145, 1962. 41

[Din67] W. Dinkelbach. On nonlinear fractional programming. *Managment Science*, 13(7):492–498, March 1967. 41

[DKR+08] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Muller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–231, Jul 2008. 6, 17, 18, 19, 21, 22, 24, 39

[Doca] Docker. https://www.docker.com/. Accessed: 2020-12-31. 90

[Docb] Docker compose. https://docs.docker.com/compose/. Accessed: 2020-12-31. 90

[Docc] Docker hub. https://hub.docker.com. Accessed: 2020-12-31. 58

[Docd] Docker swarm mode. https://docs.docker.com/engine/swarm/. Accessed: 2020-12-31. 98

[DWC+11] P. Dao, K. Wang, C. Collins, M. Ester, A. Lapuk, and S. C. Sahinalp. Optimally discriminative subnetwork markers predict response to chemotherapy. *Bioinformatics*, 27(13):i205–213, Jul 2011. 6, 15

[EKK14] Mohammed El-Kebir and Gunnar Klau. Solving the maximum-weight connected subgraph problem to optimality. 09 2014. 18, 31

[ET06] Bradley Efron and Robert Tibshirani. On testing the significance of gene sets. *The Annals of Applied Statistics*, 1, 10 2006. 11

[EW03] R. Eferl and E. F. Wagner. AP-1: a double-edged sword in tumorigenesis. *Nat. Rev. Cancer*, 3(11):859–868, Nov 2003. 71

[FBL+18] M. Foroutan, D. D. Bhuva, R. Lyu, K. Horan, J. Cursons, and M. J. Davis. Single sample scoring of molecular phenotypes. *BMC Bioinformatics*, 19(1):404, Nov 2018. 80

[FC13] J. Filmus and M. Capurro. Glypican-3: a marker and a therapeutic target in hepatocellular carcinoma. *FEBS J.*, 280(10):2471–2476, May 2013. 67

[FGA05] M. Fischetti, F. Glover, and Lodi A. The feasibility pump. *Mathematical Programming*, 104:91–104, 2005. 52

[FH14] M. Feng and M. Ho. Glypican-3 antibodies: a new therapeutic target for liver cancer. *FEBS Lett.*, 588(2):377–382, Jan 2014. 67

[Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000. 90

[Fis92] R. A. Fisher. *Statistical Methods for Research Workers*, pages 66–70. Springer New York, New York, NY, 1992. 14

[FJM+18] A. Fabregat, S. Jupe, L. Matthews, K. Sidiropoulos, M. Gillespie, P. Garapati, R. Haw, B. Jassal, F. Korninger, B. May, M. Milacic, C. D. Roca, K. Rothfels, C. Sevilla, V. Shamovsky, S. Shorser, T. Varusai, G. Viteri, J. Weiser, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio. The Reactome Pathway Knowledgebase. *Nucleic Acids Res.*, 46(D1):D649–D655, Jan 2018. 2

[Fla] Flask. https://flask.palletsprojects.com. Accessed: 2020-12-31. 100

[Fur13] L. I. Furlong. Human diseases through the lens of network biology. *Trends Genet.*, 29(3):150–159, Mar 2013. 4

[Glo] Gloo api gateway. https://github.com/solo-io/gloo. Accessed: 2020-12-31. 98

[Glo75] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Managment Science*, 22(4):455–460, December 1975. 41, 45

[GM97a] F. Glover and Laguna M. General Purpose Heuristics for Integer Programming - Part I. *Journal of Heuristics*, 2:343–358, 1997. 52

[GM97b] F. Glover and Laguna M. General Purpose Heuristics for Integer Programming - Part II. *Journal of Heuristics*, 3:161–179, 1997. 52

[Gra] Graphml. http://graphml.graphdrawing.org/. Accessed: 2020-12-31. 97

[gse] gseapy python package. http://gseapy.rtfd.io/. Accessed: 2020-12-31. 80

[GSH+13] R. K. Gaire, L. Smith, P. Humbert, J. Bailey, P. J. Stuckey, and I. Haviv. Discovery and analysis of consistent active sub-networks in cancers. *BMC Bioinformatics*, 14 Suppl 2:S7, 2013. 6, 23, 24

[GSUF12] S. J. Gosline, S. J. Spencer, O. Ursu, and E. Fraenkel. SAMNet: a network-based approach to integrate multi-dimensional high throughput datasets. *Integr Biol (Camb)*, 4(11):1415–1427, Nov 2012. 23, 24

[Gura] Gurobi academic licensing. https://www.gurobi.com/academia/academic-program-and-licenses/. Accessed: 2020-12-31. 58, 64

[Gurb] Gurobi optimization. https://www.gurobi.com/. Accessed: 2020-12-31. 57, 58, 64

[HCG+13] S. S. Huang, D. C. Clarke, S. J. Gosline, A. Labadorf, C. R. Chouinard, W. Gordon, D. A. Lauffenburger, and E. Fraenkel. Linking proteomic and transcriptional data through the interactome and epigenome reveals a map of oncogene-induced signaling. *PLoS Comput. Biol.*, 9(2):e1002887, 2013. 7, 21, 22, 24, 25

[HF09] S. S. Huang and E. Fraenkel. Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Sci Signal*, 2(81):ra40, Jul 2009. 7, 21, 22, 24, 25

[HHY+12] J. Hong, K. Hu, Y. Yuan, Y. Sang, Q. Bu, G. Chen, L. Yang, B. Li, P. Huang, D. Chen, Y. Liang, R. Zhang, J. Pan, Y. X. Zeng, and T. Kang. CHK1 targets spleen tyrosine kinase (L) for proteolysis in hepatocellular carcinoma. *J. Clin. Invest.*, 122(6):2165–2175, Jun 2012. 78

[HK11] M. Ho and H. Kim. Glypican-3: a new target for cancer immunotherapy. *Eur. J. Cancer*, 47(3):333–338, Feb 2011. 67

[HS13] J. M. Hardwick and L. Soane. Multiple functions of BCL-2 family proteins. *Cold Spring Harb Perspect Biol*, 5(2), Feb 2013. 79

[HSC+13] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker. Network-based stratification of tumor mutations. *Nat. Methods*, 10(11):1108–1115, Nov 2013. 5

[Htt] Rfc 7231: Hypertext transfer protocol (http/1.1): Semantics and content. https://tools.ietf.org/html/rfc7231. Accessed: 2020-10-14. 90, 91

[HYW+14] J. Hong, Y. Yuan, J. Wang, Y. Liao, R. Zou, C. Zhu, B. Li, Y. Liang, P. Huang, Z. Wang, W. Lin, Y. Zeng, J. L. Dai, and R. T. Chung. Expression of variant

isoforms of the tyrosine kinase SYK determines the prognosis of hepatocellular carcinoma. *Cancer Res.*, 74(6):1845–1856, Mar 2014. 78

[IOSS02] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–240, 2002. 6, 15, 16

[IPB18] I. Ihnatova, V. Popovici, and E. Budinska. A critical comparison of topology-based pathway analysis methods. *PLoS ONE*, 13(1):e0191154, 2018. 4, 13

[JE16] M. K. Jaakkola and L. L. Elo. Empirical comparison of structure-based pathway methods. *Brief. Bioinformatics*, 17(2):336–345, Mar 2016. 4, 13

[Joh77] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1), 1977. 53

[Jwt] Rfc 7519: Json web token (jwt). https://tools.ietf.org/html/rfc7519. Accessed: 2020-12-31. 99

[KBG+09] A. Keller, C. Backes, A. Gerasch, M. Kaufmann, O. Kohlbacher, E. Meese, and H. P. Lenhof. A novel algorithm for detecting differentially regulated paths based on gene set enrichment analysis. *Bioinformatics*, 25(21):2787–2794, Nov 2009. 6

[KFT+17] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, 45(D1):D353–D361, Jan 2017. 2

[KG15] M. O. Krisenko and R. L. Geahlen. Calling in SYK: SYK's dual role as a tumor promoter and tumor suppressor in cancer. *Biochim. Biophys. Acta*, 1853(1):254–263, Jan 2015. 78

[KGD+05] R. Karni, Y. Gus, Y. Dor, O. Meyuhas, and A. Levitzki. Active Src elevates the expression of beta-catenin by enhancement of cap-dependent translation. *Mol. Cell. Biol.*, 25(12):5031–5039, Jun 2005. 70

[KHT09] S. Klamt, U. U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Comput. Biol.*, 5(5):e1000385, May 2009. 4

[KKM+12] Jung H. Kim, Alla Karnovsky, Vasudeva Mahavisno, Terry Weymouth, Manjusha Pande, Dana C. Dolinoy, Laura S. Rozek, and Maureen A. Sartor. Lrpath analysis reveals common pathways dysregulated via dna methylation across cancer types. *BMC Genomics*, 13(1):526, Oct 2012. 11

[KM58] E. L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958. 75, 76

[Kon] Kong api gateway. https://github.com/Kong/kong. Accessed: 2020-12-31. 98

[KP12] Y. A. Kim and T. M. Przytycka. Bridging the Gap between Genotype and Phenotype via Network Approaches. *Front Genet*, 3:227, 2012. 4

[Kra] Krakend api gateway. https://github.com/devopsfaith/krakend. Accessed: 2020-12-31. 98

[KRN+16] M. Kutmon, A. Riutta, N. Nunes, K. Hanspers, E. L. Willighagen, A. Bohler, J. Melius, A. Waagmeester, S. R. Sinha, R. Miller, S. L. Coort, E. Cirillo, B. Smeets, C. T. Evelo, and A. R. Pico. WikiPathways: capturing the full diversity of pathway knowledge. *Nucleic Acids Res.*, 44(D1):D488–494, Jan 2016. 2

[Kub] Kubernetes. https://kubernetes.io/. Accessed: 2020-12-31. 98, 102

[LAS16] Alexander A. Loboda, Maxim N. Artyomov, and Alexey A. Sergushichev. Solving generalized maximum-weight connected subgraph problem for network enrichment analysis. In Martin Frith and Christian Nørgaard Storm Pedersen, editors, *Algorithms in Bioinformatics*, pages 210–221, Cham, 2016. Springer International Publishing. 18

[Lem] Lemon graph library. https://lemon.cs.elte.hu/trac/lemon. Accessed: 2020-12-31. 41, 58

[LHA14] M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, 15(12):550, 2014. 66

[Lib] libgrbfrc github repo. https://sebwink.github.io/libgrbfrc/. Accessed: 2020-12-31. 57

[LLZ+16] L. Liu, C. Liu, Q. Zhang, J. Shen, H. Zhang, J. Shan, G. Duan, D. Guo, X. Chen, J. Cheng, Y. Xu, Z. Yang, C. Yao, M. Lai, and C. Qian. SIRT1-mediated transcriptional regulation of SOX2 is important for self-renewal of liver cancer stem cells. *Hepatology*, 64(3):814–827, 09 2016. 70

[LMS+14] B. Liu, R. K. Madduri, B. Sotomayor, K. Chard, L. Lacinski, U. J. Dave, J. Li, C. Liu, and I. T. Foster. Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses. *J Biomed Inform*, 49:119–133, Jun 2014. 90

[LN18] B. Langmead and A. Nellore. Cloud computing for genomic data analysis and collaboration. *Nat Rev Genet*, 19(4):208–219, 04 2018. 90

[Loc13] J. W. Locasale. Serine, glycine and one-carbon units: cancer metabolism in full circle. *Nat Rev Cancer*, 13(8):572–583, Aug 2013. 85

[Low11] C. A. Lowell. Src-family and Syk kinases in activating and inhibitory pathways in innate immune cells: signaling cross talk. *Cold Spring Harb Perspect Biol*, 3(3), Mar 2011. 78, 79

[LTG+19] Anika Liu, Panuwat Trairatphisan, Enio Gjerga, Athanasios Didangelos, Jonathan Barratt, and Julio Saez-Rodriguez. From expression footprints to causal pathways: contextualizing large signaling networks with carnival. *npj Systems Biology and Applications*, 5(1):40, Nov 2019. 6, 23, 24

[Luc05] A. Luch. Nature and nurture - lessons from chemical carcinogenesis. *Nat. Rev. Cancer*, 5(2):113–125, Feb 2005. 71

[LWH+17] T. Li, R. Wernersson, R. B. Hansen, H. Horn, J. Mercer, G. Slodkowicz, C. T. Workman, O. Rigina, K. Rapacki, H. H. Staerfeldt, S. Brunak, T. S. Jensen,

and K. Lage. A scored human protein-protein interaction network to catalyze genomic interpretation. *Nat. Methods*, 14(1):61–64, 01 2017. 4

[LWP+06] Ivana Ljubic, René Weiskircher, Ulrich Pferschy, Gunnar Klau, Petra Mutzel, and Matteo Fischetti. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Math. Program.*, 105:427–449, 02 2006. 19, 22

[LXC+16] L. J. Liu, S. X. Xie, Y. T. Chen, J. L. Xue, C. J. Zhang, and F. Zhu. Aberrant regulation of Wnt signaling in hepatocellular carcinoma. *World J. Gastroenterol.*, 22(33):7486–7499, Sep 2016. 70

[LXK+18] X. Li, W. Xu, W. Kang, S. H. Wong, M. Wang, Y. Zhou, X. Fang, X. Zhang, H. Yang, C. H. Wong, K. F. To, S. L. Chan, M. T. V. Chan, J. J. Y. Sung, W. K. K. Wu, and J. Yu. Genomic analysis of liver cancer unveils novel driver genes and distinct prognostic features. *Theranostics*, 8(6):1740–1751, 2018. 79

[LZRP+16] J. M. Llovet, J. Zucman-Rossi, E. Pikarsky, B. Sangro, M. Schwartz, M. Sherman, and G. Gores. Hepatocellular carcinoma. *Nat Rev Dis Primers*, 2:16018, 04 2016. 69, 71

[Mac14] H. Maciejewski. Gene set analysis methods: statistical models and methodological differences. *Brief. Bioinformatics*, 15(4):504–518, Jul 2014. 2, 3, 10, 11, 12

[MCRI13] K. Mitra, A. R. Carvunis, S. K. Ramesh, and T. Ideker. Integrative approaches for finding modular structure in biological networks. *Nat. Rev. Genet.*, 14(10):719–732, Oct 2013. 5, 6, 14, 15

[MJB+12] L. Min, Y. Ji, L. Bakiri, Z. Qiu, J. Cen, X. Chen, L. Chen, H. Scheuch, H. Zheng, L. Qin, K. Zatloukal, L. Hui, and E. F. Wagner. Liver cancer initiation is controlled by AP-1 through SIRT6-dependent inhibition of survivin. *Nat. Cell Biol.*, 14(11):1203–1211, Nov 2012. 70

[MJJ+01] S. J. Mandriota, L. Jussila, M. Jeltsch, A. Compagni, D. Baetens, R. Prevo, S. Banerji, J. Huarte, R. Montesano, D. G. Jackson, L. Orci, K. Alitalo, G. Christofori, and M. S. Pepper. Vascular endothelial growth factor-C-mediated lymphangiogenesis promotes tumour metastasis. *EMBO J.*, 20(4):672–682, Feb 2001. 79

[MLE+03] Vamsi K. Mootha, Cecilia M. Lindgren, Karl-Fredrik Eriksson, Aravind Subramanian, Smita Sihag, Joseph Lehar, Pere Puigserver, Emma Carlsson, Martin Ridderstråle, Esa Laurila, Nicholas Houstis, Mark J. Daly, Nick Patterson, Jill P. Mesirov, Todd R. Golub, Pablo Tamayo, Bruce Spiegelman, Eric S. Lander, Joel N. Hirschhorn, David Altshuler, and Leif C. Groop. Pgc-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, Jul 2003. 11

[MMF+07] M. Montorsi, M. Maggioni, M. Falleni, C. Pellegrini, M. Donadon, G. Torzilli, R. Santambrogio, A. Spinelli, G. Coggi, and S. Bosari. Survivin gene expression in chronic liver disease and hepatocellular carcinoma. *Hepatogastroenterology*, 54(79):2040–2044, 2007. 70

[Mon] Mongodb. https://www.mongodb.com/. Accessed: 2020-12-31. 101

[MRT10] A. Mocsai, J. Ruland, and V. L. Tybulewicz. The SYK tyrosine kinase: a crucial player in diverse biological functions. *Nat. Rev. Immunol.*, 10(6):387–402, Jun 2010. 78, 79

[MSI+15] Ioannis N. Melas, Theodore Sakellaropoulos, Francesco Iorio, Leonidas G. Alexopoulos, Wei-Yin Loh, Douglas A. Lauffenburger, Julio Saez-Rodriguez, and Jane P. F. Bai. Identification of drug-specific pathways based on gene expression data: application to drug induced lung injury. *Integrative Biology*, 7(8):904–920, 05 2015. 6, 23, 24

[MTB+13] C. Mitrea, Z. Taghavi, B. Bokanizad, S. Hanoudi, R. Tagett, M. Donato, C. Voichita, and S. Draghici. Methods and approaches in the topology-based analysis of biological pathways. *Front Physiol*, 4:278, Oct 2013. 4, 13

[nar20] Editorial: the 18th annual Nucleic Acids Research web server issue 2020. *Nucleic Acids Research*, 48(W1):W1–W4, 06 2020. 90

[NC17] R. Nusse and H. Clevers. Wnt/β-Catenin Signaling, Disease, and Emerging Therapeutic Modalities. *Cell*, 169(6):985–999, Jun 2017. 70

[Ndea] Ndex integration server. https://github.com/sebwink/ndex-graphml. Accessed: 2020-12-31. 102

[Ndeb] Ndex website. https://home.ndexbio.org/index/. Accessed: 2020-12-31. 102

[NQdB+07] Michael A. Newton, Fernando A. Quintana, Johan A. den Boon, Srikumar Sengupta, and Paul Ahlquist. Random-set methods identify distinct aspects of the enrichment signal in gene-set analysis. *The Annals of Applied Statistics*, 1(1):85 – 106, 2007. 11

[NZR16] J. C. Nault and J. Zucman-Rossi. TERT promoter mutations in primary liver tumors. *Clin Res Hepatol Gastroenterol*, 40(1):9–14, Feb 2016. 69

[oau] Rfc 6749: The oauth 2.0 authorization framework. https://tools.ietf.org/html/rfc6749. Accessed: 2020-12-31. 100

[Ope] Openid connect. https://openid.net/connect/. Accessed: 2020-12-31. 100

[Ory] Ory stack. https://www.ory.sh/. Accessed: 2020-12-31. 99, 105

[PCP+17] D. Pratt, J. Chen, R. Pillich, V. Rynkov, A. Gary, B. Demchak, and T. Ideker. NDEx 2.0: A Clearinghouse for Research on Cancer Pathways. *Cancer Res*, 77(21):e58–e61, 11 2017. 102

[PCW+15] D. Pratt, J. Chen, D. Welker, R. Rivas, R. Pillich, V. Rynkov, K. Ono, C. Miello, L. Hicks, S. Szalma, A. Stojmirovic, R. Dobrin, M. Braxenthaler, J. Kuentzer, B. Demchak, and T. Ideker. NDEx, the Network Data Exchange. *Cell Syst*, 1(4):302–305, Oct 2015. 102

[PN05] K. R. Patil and J. Nielsen. Uncovering transcriptional regulation of metabolism by using metabolic network topology. *Proc. Natl. Acad. Sci. U.S.A.*, 102(8):2685–2689, Feb 2005. 6, 15

[QOT+14] A. Quaas, T. Oldopp, L. Tharun, C. Klingenfeld, T. Krech, G. Sauter, and T. J. Grob. Frequency of TERT promoter mutations in primary tumors of the liver. *Virchows Arch.*, 465(6):673–677, Dec 2014. 69

[QZ14] G. Qin and X. M. Zhao. A survey on computational approaches to identifying disease biomarkers based on molecular networks. *J. Theor. Biol.*, 362:9–16, Dec 2014. 4

[QZL+18] C. Qu, D. Zheng, S. Li, Y. Liu, A. Lidofsky, J. A. Holmes, J. Chen, L. He, L. Wei, Y. Liao, H. Yuan, Q. Jin, Z. Lin, Q. Hu, Y. Jiang, M. Tu, X. Chen, W. Li, W. Lin, B. C. Fuchs, R. T. Chung, and J. Hong. Tyrosine kinase SYK is a potential therapeutic target for liver fibrosis. *Hepatology*, Mar 2018. 78

[RCP11] E. T. Roussos, J. S. Condeelis, and A. Patsialou. Chemotaxis in cancer. *Nat. Rev. Cancer*, 11(8):573–587, Jul 2011. 79

[Red] Redis. https://redis.io/. Accessed: 2020-12-31. 101

[RFC] Request for comments (rfc). https://www.ietf.org/standards/rfcs/. Accessed: 2020-12-31. 89

[ric] Pattern: Api gateway / backends for frontends. https://microservices.io/patterns/apigateway.html. Accessed: 2020-12-31. 98

[Ric18] C. Richardson. *Microservices Patterns: With examples in Java*. Manning Publications, 2018. 90, 98

[RK19] Daniel Rehfeldt and Thorsten Koch. Combining np-hard reduction techniques and strong heuristics in an exact algorithm for the maximum-weight connected subgraph problem. *SIAM Journal on Optimization*, 29(1):369–398, 2019. 18, 52

[RKM19] Daniel Rehfeldt, Thorsten Koch, and Stephen J. Maher. Reduction techniques for the prize collecting steiner tree problem and the maximum-weight connected subgraph problem. *Networks*, 73(2):206–233, 2019. 18

[sbm] sbml4j github repository. https://github.com/thortiede/sbml4j. Accessed: 2020-12-31. 102

[Sch09] Heiko Schuldt. *Multi-Tier Architecture - Encyclopedia of Database Systems*, pages 1862–1865. Springer US, Boston, MA, 2009. 97

[sci] Scikit-learn python package. https://scikit-learn.org. Accessed: 2020-12-31. 81

[Sha81] M. Sharir. A strong-connectivity algorithm and its applications to data flow analysis. *Computers and Mathematics with applications*, 7(1):67–72, 1981. 41, 51

[Sim] Simple interaction format (sif). http://manual.cytoscape.org/en/stable/Supported_Network_File_Formats.html#sif-format. Accessed: 2020-12-31. 96

[SLK+14] S. H. Shin, K. H. Lee, B. H. Kim, S. Lee, H. S. Lee, J. J. Jang, and G. H. Kang. Downregulation of spleen tyrosine kinase in hepatocellular carcinoma by promoter CpG island hypermethylation and its potential role in carcinogenesis. *Lab. Invest.*, 94(12):1396–1405, Dec 2014. 78

[SLM08] Maureen A. Sartor, George D. Leikauf, and Mario Medvedovic. LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data. *Bioinformatics*, 25(2):211–217, 11 2008. 10, 11

[SMC+17] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, and C. von Mering. The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res.*, 45(D1):D362–D368, Jan 2017. 4

[SSD+20] B. Sürun, C. P. I. Schärfe, M. R. Divine, J. Heinrich, N. C. Toussaint, L. Zimmermann, J. Beha, and O. Kohlbacher. ClinVAP: a reporting strategy from variants to therapeutic options. *Bioinformatics*, 36(7):2316–2317, 04 2020. 102

[SSL+13] C. Sun, L. Sun, Y. Li, X. Kang, S. Zhang, and Y. Liu. Sox2 expression predicts poor survival of hepatocellular carcinoma patients and it promotes liver cancer cell invasion by activating Slug. *Med. Oncol.*, 30(2):503, Jun 2013. 70

[STM+05] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 102(43):15545–15550, Oct 2005. 3, 11, 80

[Su16] C. Su. Survivin in survival of hepatocellular carcinoma. *Cancer Lett.*, 379(2):184–190, 09 2016. 70

[Swa] Swagger. https://swagger.io/tools/. Accessed: 2020-12-31. 100

[Tac16] L. Taccari. Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*, 252(1), 2016. 55

[Tar72] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. 51

[TAZ+10] D. Tvorogov, A. Anisimov, W. Zheng, V. M. Leppanen, T. Tammela, S. Laurinavicius, W. Holnthoner, H. Helotera, T. Holopainen, M. Jeltsch, N. Kalkkinen, H. Lankinen, P. M. Ojala, and K. Alitalo. Effective suppression of vascular network formation by combination of antibodies blocking VEGFR ligand binding and receptor dimerization. *Cancer Cell*, 18(6):630–640, Dec 2010. 79

[TB08] Y. Takigawa and A. M. Brown. Wnt signaling in liver cancer. *Curr Drug Targets*, 9(11):1013–1024, Nov 2008. 70

[TBP+13] N. Tuncbag, A. Braunstein, A. Pagnani, S. S. Huang, J. Chayes, C. Borgs, R. Zecchina, and E. Fraenkel. Simultaneous reconstruction of multiple signaling pathways via the prize-collecting steiner forest problem. *J. Comput. Biol.*, 20(2):124–136, Feb 2013. 7, 21, 22, 24, 25

[tcg] Tcga lihc dataset. https://portal.gdc.cancer.gov/projects/TCGA-LIHC. Accessed: 2020-12-31. 66

[TCW15] K. Tomczak, P. Czerwiska, and M. Wiznerowicz. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Contemp Oncol (Pozn)*, 19(1A):68–77, 2015. 2, 102

[TDK+09] A. L. Tarca, S. Draghici, P. Khatri, S. S. Hassan, P. Mittal, J. S. Kim, C. J. Kim, J. P. Kusanovic, and R. Romero. A novel signaling pathway impact analysis. *Bioinformatics*, 25(1):75–82, Jan 2009. 4, 13, 14

[TGK+16] N. Tuncbag, S. J. Gosline, A. Kedaigle, A. R. Soltis, A. Gitter, and E. Fraenkel. Network-Based Interpretation of Diverse High-Throughput Datasets through the Omics Integrator Software Package. *PLoS Comput. Biol.*, 12(4):e1004879, Apr 2016. 7, 21, 25

[TTC+14] Y. Totoki, K. Tatsuno, K. R. Covington, H. Ueda, C. J. Creighton, M. Kato, S. Tsuji, L. A. Donehower, B. L. Slagle, H. Nakamura, S. Yamamoto, E. Shinbrot, N. Hama, M. Lehmkuhl, F. Hosoda, Y. Arai, K. Walker, M. Dahdouli, K. Gotoh, G. Nagae, M. C. Gingras, D. M. Muzny, H. Ojima, K. Shimada, Y. Midorikawa, J. A. Goss, R. Cotton, A. Hayashi, J. Shibahara, S. Ishikawa, J. Guiteau, M. Tanaka, T. Urushidate, S. Ohashi, N. Okada, H. Doddapaneni, M. Wang, Y. Zhu, H. Dinh, T. Okusaka, N. Kokudo, T. Kosuge, T. Takayama, M. Fukayama, R. A. Gibbs, D. A. Wheeler, H. Aburatani, and T. Shibata. Trans-ancestry mutational landscape of hepatocellular carcinoma genomes. *Nat. Genet.*, 46(12):1267–1273, Dec 2014. 69

[TYZ15] L. M. Thorpe, H. Yuzugullu, and J. J. Zhao. PI3K in cancer: divergent roles of isoforms, modes of activation and therapeutic targeting. *Nat. Rev. Cancer*, 15(1):7–24, Jan 2015. 78

[TZW+08] T. Tammela, G. Zarkada, E. Wallgard, A. Murtomaki, S. Suchting, M. Wirzenius, M. Waltari, M. Hellstrom, T. Schomber, R. Peltonen, C. Freitas, A. Duarte, H. Isoniemi, P. Laakkonen, G. Christofori, S. Yla-Herttuala, M. Shibuya, B. Pytowski, A. Eichmann, C. Betsholtz, and K. Alitalo. Blocking VEGFR-3 suppresses angiogenic sprouting and vascular network formation. *Nature*, 454(7204):656–660, Jul 2008. 79

[UFH+15] Y. H. Uen, C. L. Fang, Y. C. Hseu, P. C. Shen, H. L. Yang, K. S. Wen, S. T. Hung, L. H. Wang, and K. Y. Lin. VAV3 oncogene expression in colorectal cancer: clinical aspects and functional characterization. *Sci Rep*, 5:9360, Mar 2015. 79

[UGAR05] S. D. Undevia, G. Gomez-Abuin, and M. J. Ratain. Pharmacokinetic variability of anticancer agents. *Nat. Rev. Cancer*, 5(6):447–458, Jun 2005. 71

[UKKS10] I. Ulitsky, A. Krishnamurthy, R. M. Karp, and R. Shamir. DEGAS: de novo discovery of dysregulated pathways in human diseases. *PLoS ONE*, 5(10):e13367, Oct 2010. 6, 15

[US07] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Syst Biol*, 1:8, Jan 2007. 6, 15

[US09] I. Ulitsky and R. Shamir. Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics*, 25(9):1158–1164, May 2009. 6, 15

[VBS+10] C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler, and J. M. Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics*, 26(12):i237–245, Jun 2010. 15

[VRU16] F. Vandin, B. J. Raphael, and E. Upfal. On the Sample Complexity of Cancer Pathways Identification. *J. Comput. Biol.*, 23(1):30–41, Jan 2016. 5

[VTMG16] V. Vilchez, L. Turcios, F. Marti, and R. Gedaly. Targeting Wnt/β-catenin pathway in hepatocellular carcinoma treatment. *World J. Gastroenterol.*, 22(2):823–832, Jan 2016. 70

[VUR12a] F. Vandin, E. Upfal, and B. J. Raphael. De novo discovery of mutated driver pathways in cancer. *Genome Res.*, 22(2):375–385, Feb 2012. 5

[VUR12b] F. Vandin, E. Upfal, and B. J. Raphael. Finding driver pathways in cancer: models and algorithms. *Algorithms Mol Biol*, 7(1):23, Sep 2012. 5

[WBW+20] Ivana Winkler, Catrin Bitter, Sebastian Winkler, Dieter Weichenhan, Abhishek Thavamani, Jan G. Hengstler, Erawan Borkham-Kamphorst, Oliver Kohlbacher, Christoph Plass, Robert Geffers, Ralf Weiskirchen, and Alfred Nordheim. Identification of pparγ-modulated mirna hubs that target the fibrotic tumor microenvironment. *Proceedings of the National Academy of Sciences*, 117(1):454–463, 2020. 157

[WGS09] Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, 10:57–63, 2009. 1

[WHC+13] W. Wen, T. Han, C. Chen, L. Huang, W. Sun, X. Wang, S. Z. Chen, D. M. Xiang, L. Tang, D. Cao, G. S. Feng, M. C. Wu, J. Ding, and H. Y. Wang. Cyclin G1 expands liver tumor-initiating cells by Sox2 induction via Akt/mTOR signaling. *Mol. Cancer Ther.*, 12(9):1796–1804, Sep 2013. 70

[WHFL13] J. L. Wilson, M. T. Hemann, E. Fraenkel, and D. A. Lauffenburger. Integrated network analyses for functional genomic studies in cancer. *Semin. Cancer Biol.*, 23(4):213–218, Aug 2013. 4

[WJXK16] Y. Wen, S. Jeong, Q. Xia, and X. Kong. Role of Osteopontin in Liver Diseases. *Int. J. Biol. Sci.*, 12(9):1121–1128, 2016. 70

[WLD16] Y. Wu, H. Liu, and H. Ding. GPC-3 in hepatocellular carcinoma: current perspectives. *J Hepatocell Carcinoma*, 3:63–67, 2016. 67

[WTD+] I. Winkler, A. Thavamani, G.S. Ducker, S. Winkler, K. Matic, D. Weichenhan, M. Graeve, S. Pietschmann, R. Geffers, S. Czemmel, M. Codrea, S. Nahnsen, O. Kohlbacher, R. Weiskirchen, C. Plass, B. Macek, J.D. Rabinowitz, and A. Nordheim. Metabolic Reprogramming of Serine Synthesis, Mitochondrial One-Carbon Metabolism, and Methionine Cycle Activity in Hepatocellular Carcinoma. *In Preparation*. 85, 108, 157

[WWF+] S. Winkler, I. Winkler, M. Figaschewski, T. Tiede, A. Nordheim, and O. Kohlbacher. DeRegNet - An approach to *de novo* identification of maximally deregulated subnetworks based on multi-omics data. *Submitted*. 157

[WWF+21] Sebastian Winkler, Ivana Winkler, Mirjam Figaschewski, Thorsten Tiede, Alfred Nordheim, and Oliver Kohlbacher. De novo identification of maximally deregulated subnetworks based on multi-omics data with deregnet. *bioRxiv*, 2021. 157

[YCG09] F. You, P.M. Castro, and I.E. Grossmann. Dinkelbach's algorithm as an efficient method to solve a class of minlp models for large-scale cyclic scheduling problems. *Computers & Chemical Engineering*, 33:1879–1889, 2009. xv, 41, 136, 137

[YGGY13] D. Yue, G. Guillén-Gosálbez, and F. You. Global optimization of large-scale mixed-integer linear fractional programming problems: a reformulation-linearization method and process scheduling applications. *AIChE Journal*, 59(11):4255–4272, 2013. 41, 43, 44, 46, 136, 137, 138, 139, 140, 141

[YVAO+14] Y. Yuan, E. M. Van Allen, L. Omberg, N. Wagle, A. Amin-Mansour, A. Sokolov, L. A. Byers, Y. Xu, K. R. Hess, L. Diao, L. Han, X. Huang, M. S. Lawrence, J. N. Weinstein, J. M. Stuart, G. B. Mills, L. A. Garraway, A. A. Margolin, G. Getz, and H. Liang. Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nat Biotechnol*, 32(7):644–652, Jul 2014. 79, 83, 108

[ZSYT18] F. Zhou, W. Shang, X. Yu, and J. Tian. Glypican-3: A promising biomarker for hepatocellular carcinoma diagnosis and treatment. *Med Res Rev*, 38(2):741–767, 03 2018. 67

[ZW09] J. D. Zhang and S. Wiemann. KEGGgraph: a graph approach to KEGG PATHWAY in R and bioconductor. *Bioinformatics*, 25(11):1470–1471, Jun 2009. 66

[ZWCA08] X. M. Zhao, R. S. Wang, L. Chen, and K. Aihara. Uncovering signal transduction networks from high-throughput data by integer linear programming. *Nucleic Acids Res.*, 36(9):e48, May 2008. 6, 19, 20, 21, 22, 24, 25

[ZWS+19a] Lichao Zhang, Sebastian Winkler, Fabian Schlottmann, Oliver Kohlbacher, Josh E. Elias, Jan M. Skotheim, and Jennifer C. Ewald. Multiple layers of phospho-regulation coordinate metabolism and the cell cycle in budding yeast. *bioRxiv*, 2019. 86, 87, 108, 157

[ZWS+19b] Lichao Zhang, Sebastian Winkler, Fabian P. Schlottmann, Oliver Kohlbacher, Josh E. Elias, Jan M. Skotheim, and Jennifer C. Ewald. Multiple layers of phospho-regulation coordinate metabolism and the cell cycle in budding yeast. *Frontiers in Cell and Developmental Biology*, 7:338, 2019. 86, 157

[ZZ18] J. Zhang and S. Zhang. The Discovery of Mutated Driver Pathways in Cancer: Models and Algorithms. *IEEE/ACM Trans Comput Biol Bioinform*, 15(3):988–998, 2018. 5

Appendix A

Fractional mixed-integer linear programming (FMILP)

For locality of exposition we restate:

Definition 10 (Fractional mixed-integer linear program; FMILP)
*A **Fractional mixed-integer linear program (FMILP)** is an optimization problem of the following structure:*

$$\max \quad \frac{c^T x + d}{p^T x + q} \tag{A.1a}$$

$$\text{s.t.} \quad x \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_i} \tag{A.1b}$$

$$Ax \leq b \tag{A.1c}$$

Here, $c, p \in \mathbb{R}^n$, $d, q \in \mathbb{R}$ define the objective, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ define $m \in \mathbb{N}$ linear constraints and $n_c \in \mathbb{N}$, $n_i \in \mathbb{N}$ denote the number of continuous and discrete (integer) variables.

We assume $\forall x \in \mathscr{F} : p^T x + q > 0$, $\mathscr{F} := \{x \in \mathbb{R}^n : Ax \leq b\}$. Fractional mixed-integer linear problems are hence mixed-integer problems except for the objective which is a rational function with linear enumerator and denominator instead. While a

FMILP is non-convex, it turns out that a FMILP is pseudolinear and hence quasilinear, rendering local optima to be globally optimal [YCG09].

Proposition 10 (You et al. [YCG09])
A FMILP is pseudoconvex and pseudoconcave.

Proposition 11 (You et al. [YCG09])
A FMILP is strictly quasiconvex and strictly quasiconcave.

Proposition 12 (You et al. [YCG09])
A local optimum of a FMILP is also a global optimum.

The latter facts render FMILP solvable by any generic mixed-integer nonlinear programming (MINLP) solver which can handle pseudolinear objective functions [YCG09]. Empirically, it was shown that iterative schemes [YCG09] or linearization-reformulation approaches [YGGY13] outperform generic MINLP solvers with respect to computing time and memory footprint. These approaches rely on a mixed-integer linear programming (MILP) solver as their optimization kernel, hence unlocking the power of modern MILP software, and rely on transforming the original problem into a (sequence of) MILP problem(s). The DeRegNet software package discussed in the main text implements a Dinkelbach-type algorithm [YCG09] and a reformulation-linearization method [YGGY13] resembling the Charnes-Cooper method [CC62] for solving fractional linear programs (FLP). The remainder of this appendix details the proof for the correctness and superlinear convergence of Dinkelbach's iterative algorithm.

A.1 Dinkelbach-type algorithm (Dinkelbach algorithm)

This section details why Dinkelbach's algorithm works for solving fractional integer programming models. Again for locality of exposition we restate:

Data: FMILP with feasible set $\mathscr{S}$
Result: solution x^* of FMILP
Initialization:
$\pi = 0$
$\epsilon > 0$ (termination tolerance)
$F = \infty$
while $F > \epsilon$ **do**
  $x^* = arg\,max\,\{c^T x + d - \pi\,(p^T x + q) : x \in \mathscr{S}\}$
  $F = c^T x^* + d - r\,(p^T x^* + q)$
  $\pi = \dfrac{c^T x^* + d}{p^T x^* + q}$
end
return x^*

Algorithm 11: Dinkelbach-type algorithm

A.1.1 Correctness of Dinkelbach's Algorithm (11) - based on You et al. [YCG09]

In order to facilitate the following exposition the functions $N : \mathscr{F} \to \mathbb{R}, N(x) := c^T x + d$ for the nominator and $D : \mathscr{F} \to \mathbb{R}, D(x) := p^T x + q$ for the denominator of the objective function are introduced. Without loss of generality one can set $d = q = 0$ since one can introduce dummy variables x_d and x_q with linear constraints $x_d = x_q = 1$ and corresponding coefficients $c_d = p_q = 1$ leading to $N(x) = c^T x + c_d x_d$ and $D(x) = p^T x + p_q x_q$. Furthermore, define $L_\pi(x) := N(x) - \pi D(x)$ and $F : \mathbb{R} \to \mathbb{R}, F(\pi) := \max\{L_\pi(x) : x \in \mathscr{F}\}$ be the optimal objective value of a Dinkelbach iteration problem as a function of the auxiliary parameter π. Without loss of generality we assume $D(x) > 0$ for all $x \in \mathscr{F}$.

The two main results concerning Dinkelbach's algorithm are the following:

Proposition 13 (Optimality criterion, Yue et al. [YGGY13] Proposition 1)
$F(\pi^*) = \max\{N(x) - \pi D(x) : x \in \mathscr{F}\} = 0 \iff \pi^* = \frac{N(x^*)}{D(x^*)} = \max\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$
where $x^* = \text{argmax}\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$

Proposition 14 (Convergence (rate), Yue et al. [YGGY13] Proposition 2)
Dinkelbach's algorithm converges superlinearly to π^* *in where* $x^* \in \operatorname{argmax}\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$ *and* $\pi^* = \frac{N(x^*)}{D(x^*)}$.

We follow Yue et al. [YGGY13] in proving the above propositions via a series of lemmas.

Lemma 1 (Yue et al. [YGGY13] Appendix, Lemma 4)
F is convex.

Proof. For $\lambda \in [0,1]$, let $x_\lambda \in \mathscr{F}$ be $x_\lambda \in \operatorname{argmax}\{L_{\lambda\pi'+(1-\lambda)\pi''}(x) : x \in \mathscr{F}\}$ with $\pi', \pi'' \in \mathbb{R}$. Then:

$$\begin{aligned} F(\lambda\pi' + (1-\lambda)\pi'') &= \max\{L_\pi(x) : x \in \mathscr{F}\} && \text{(A.2)} \\ &= N(x_\lambda) - [\lambda\pi' + (1-\lambda)\pi'']D(x) && \text{(A.3)} \\ &= \lambda[N(x_\lambda) - \pi' D(x_\lambda)] + (1-\lambda)[N(x_\lambda) - \pi'' D(x_\lambda)] && \text{(A.4)} \\ &= \lambda F(\pi') + (1-\lambda)F(\pi'') && \text{(A.5)} \end{aligned}$$

■

Lemma 2 (Yue et al. [YGGY13] Appendix, Lemma 5)
F is strictly monotonically increasing, i.e. $\pi' < \pi'' \implies F(\pi') < F(\pi'')$.

Proof. Given $\pi' < \pi''$ one obtains with $x' = \operatorname{argmax}\{L_{\pi'}(x) : x \in \mathscr{F}\}$ and $x'' = \operatorname{argmax}\{L_{\pi''}(x) : x \in \mathscr{F}\}$:

$$\begin{aligned} F(\pi'') &= N(x'') - \pi'' D(x'') && \text{(A.6)} \\ &< N(x'') - \pi' D(x'') && \text{(A.7)} \\ &\leq N(x') - \pi' D(x') && \text{(A.8)} \\ &= F(\pi') && \text{(A.9)} \end{aligned}$$

■

Lemma 3 (Yue et al. [YGGY13] Appendix, Lemma 6)

$F(\pi) = 0$ *has a unique solution.*

Proof. Follows from $\lim_{\pi\to\infty} F(\pi) = -\infty$ and $\lim_{\pi\to-\infty} F(\pi) = \infty$ and F being strictly monotonically increasing (Lemma 2). ■

Lemma 4 (Yue et al. [YGGY13] Appendix, Lemma 7)

$\forall x' \in \mathscr{F} : F(\frac{N(x')}{D(x')}) \geq 0$

Proof. For any $x' \in \mathscr{F}$ one has:

$$F(\frac{N(x')}{D(x')}) = \max\{N(x) - \frac{N(x')}{D(x')}D(x) : x \in \mathscr{F}\} \tag{A.10}$$

$$\geq N(x') - \frac{N(x')}{D(x')}D(x') \tag{A.11}$$

$$= 0 \tag{A.12}$$

■

One can now prove proposition 1:

Proof of proposition 1. We have to show: $F(\pi^*) \iff \pi^* = \frac{N(x^*)}{D(x^*)} = max_{x\in\mathscr{F}}\frac{N(x)}{D(x)}$.

$\Longrightarrow$: Given $F(\pi^*) = max_{x\in\mathbb{F}}N(x) - \pi^*D(x)$ it follows with $x^* := \text{argmax}\{N(x) - \pi^*D(x) : x \in \mathscr{F}\}$ for all $x \in \mathscr{F}$ $0 = N(x^*) - \pi^*D(x^*) \geq N(x) - \pi^*D(x)$. Hence $\frac{N(x)}{D(x)} \leq \pi^* = \frac{N(x^*)}{D(x^*)}$, i.e. $x^* = \text{argmax}\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$.

$\Longleftarrow$: With $x^* = \text{argmax}\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$ one has $\pi^* = \frac{N(x^*)}{D(x^*)} \geq \frac{D(x)}{N(x)}$. Under our general assumption $D(x) > 0$ for all $x \in \mathscr{F}$ it follows $N(x) - \pi^*D(x) \leq 0 = N(x^*) - \pi^*D(x^*)$ for all $x \in \mathscr{F}$ which shows $x^* = \text{argmax}\{N(x) - \pi^*D(x) : x \in \mathscr{F}\}$.

■

From now onward, let π^* be the unique solution of $F(\pi) = 0$ and let $x^* \in \text{argmax}\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$ with $\pi^* = \frac{N(x^*)}{D(x^*)}$.

Lemma 5 (Yue et al. [YGGY13] Appendix, Lemma 8)

Let $x' \in \text{argmax}\{N(x) - \pi' D(x)\}$ *and* $x'' \in \text{argmax}\{N(x) - \pi'' D(x) : x \in \mathscr{F}\}$ *with* $\pi' < \pi''$, *then* $D(x') \geq D(x'')$.

Proof. Adding the inequalities $N(x')-\pi' D(x') \geq N(x'')-\pi' D(x')$ and $N(x'')-\pi'' D(x'') \geq N(x')-\pi'' D(x')$ leads to $(\pi''-\pi')D(x') \geq (\pi''-\pi')D(x'')$, i.e. $D(x') \geq D(x'')$ since $\pi'' \geq \pi'$ by assumption. ■

Lemma 6 (Yue et al. [YGGY13] Appendix, Lemma 9)

Let $x' \in \operatorname{argmax}\{N(x)-\pi' D(x)\}$ *and* $x'' \in \operatorname{argmax}\{N(x)-\pi'' D(x) : x \in \mathscr{F}\}$, *then* $f(x'')-f(x') \geq \frac{F(\pi'')}{D(x'')} - \frac{F(\pi')}{D(x')}$

Proof. From $F(\pi'') = N(x'')-\pi'' D(x'') \geq N(x')-\pi'' D(x'')$ it follows $\frac{N(x'')}{D(x')} - \pi'' \frac{D(x'')}{D(x')} \geq \frac{N(x')}{D(x')} - \pi''$. This implies:

$$\frac{N(x'')}{D(x'')} - \frac{N(x')}{D(x')} \geq \frac{N(x'')}{D(x'')} + (-\pi'' + \frac{D(x'')}{D(x')}\pi'' - \frac{N(x'')}{D(x')}) \tag{A.13}$$

$$= \frac{N(x'')}{D(x'')} - \frac{N(x'')}{D(x')} + \pi''(\frac{D(x'')}{D(x')} - \frac{D(x'')}{D(x'')}) \tag{A.14}$$

$$= N(x'')(\frac{1}{D(x'')} - \frac{1}{D(x')}) + \pi'' D(x'')(\frac{1}{D(x'')} - \frac{1}{D(x'')}) \tag{A.15}$$

$$= -F(\pi'')(\frac{1}{D(x')} - \frac{1}{D(x'')}) \tag{A.16}$$

$$= \frac{F(\pi'')}{D(x'')} - \frac{F(\pi'')}{D(x')} \tag{A.17}$$

■

Lemma 7 (Yue et al. [YGGY13] Appendix, Lemma 10)

Let $x' \in \operatorname{argmax}\{N(x)-\pi' D(x)\}$ *and* $x'' \in \operatorname{argmax}\{N(x)-\pi'' D(x) : x \in \mathscr{F}\}$ *and* $F(\pi^*) = 0$, *then if follows for* $\pi' \leq \pi'' \leq \pi^*$, *that* $\frac{N(x')}{D(x')} \leq \frac{N(x'')}{D(x'')}$.

Proof.

$$\frac{N(x'')}{D(x'')} - \frac{N(x')}{D(x')} \geq \frac{F(\pi'')}{D(x'')} - \frac{F(\pi')}{D(x')} \tag{A.18}$$

$$\geq \frac{F(\pi'')}{D(x')} - \frac{F(\pi'')}{D(x')} \tag{A.19}$$

$$= 0 \tag{A.20}$$

The first inequality follows from lemma 9, the second from lemma 7 and 8. ■

Lemma 8 (Yue et al. [YGGY13] Appendix, Lemma 11)
Let $x' \in \operatorname{argmax}\{N(x) - \pi' D(x)\}$ *and* $x'' \in \operatorname{argmax}\{N(x) - \pi'' D(x) : x \in \mathscr{F}\}$, *then* $f(x'') - f(x') \leq (-F(\pi'') + (\pi' - \pi'')D(x''))(\frac{1}{D(x')} - \frac{1}{D(x'')})$.

Proof. From $N(x') - \pi' D(x') \geq N(x'') - \pi'' D(x'')$ it follows $\frac{N(x')}{D(x')} - \pi' \geq \frac{N(x'')}{D(x')} - \pi' \frac{N(x'')}{D(x')}$ by dividing by $D(x') > 0$. It then follows:

$$
\begin{aligned}
f(x'') - f(x') &= \frac{N(x'')}{D(x'')} - \frac{N(x')}{D(x')} && \text{(A.21)}\\
&\leq \frac{N(x'')}{D(x'')} - \pi' - \frac{N(x'')}{D(x')} + \pi' \frac{D(x'')}{D(x')} && \text{(A.22)}\\
&= \frac{N(x'')}{D(x'')} - \frac{N(x'')}{D(x')} - \pi'(\frac{D(x'')}{D(x'')} - \frac{D(x'')}{D(x')}) && \text{(A.23)}\\
&= (-N(x'') + \pi' D(x''))(\frac{1}{D(x')} - \frac{1}{D(x'')}) && \text{(A.24)}\\
&= (-F(\pi'') + (\pi' - \pi'')D(x''))(\frac{1}{D(x')} - \frac{1}{D(x'')}) && \text{(A.25)}
\end{aligned}
$$

■

Lemma 9 (Yue et al. [YGGY13] Appendix, Lemma 12)
Let $x' \in \operatorname{argmax}\{N(x) - \pi' D(x)\}$ *and* $x'' \in \operatorname{argmax}\{N(x) - \pi'' D(x) : x \in \mathscr{F}\}$ *with* $F(\pi^*) = N(x^*) - \pi^* D(x^*) = 0$, *then* $\pi^* - f(x') \leq (\pi^* - \pi')(1 - \frac{D(x^*)}{D(x')})$.

Proof.

$$
\begin{aligned}
\pi^* - f(x') &= f(x^*) - f(x') && \text{(A.26)}\\
&\leq (-F(\pi^*) + (\pi' - \pi^*)D(x^*))(\frac{1}{D(x')} - \frac{1}{D(x^*)}) && \text{(A.27)}\\
&= (\pi' - \pi^*)(\frac{D(x^*)}{D(x')} - 1) && \text{(A.28)}\\
&= (\pi^* - \pi')(1 - \frac{D(x^*)}{D(x')}) && \text{(A.29)}
\end{aligned}
$$

where the inequality follows from Lemma 11. ■

Proposition 2 can now be demonstrated as follows:

Proof of proposition 2. Let $F(\pi^*) = 0$, i.e. $\pi^* = \max\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$. For $i \in \mathbb{N}$, let $\pi_{i+1} = \frac{N(x_i)}{D(x_i)} = f(x_i)$ where $x_i \in \operatorname{argmax}\{N(x) - \pi_i D(x) : x \in \mathscr{F}\}$ it follows with Lemma 9:

$$\frac{\pi^* - \pi_{i+1}}{\pi^* - \pi_i} = \frac{\pi^* - f(x_i)}{\pi^* - \pi_i} \leq 1 - \frac{D(x^*)}{D(x_i)}$$

Since $\pi_i \leq \pi^* = \max\{\frac{N(x)}{D(x)} : x \in \mathscr{F}\}$ it follows with Lemma 5 $\frac{D(x^*)}{D(x_i)} \leq 1$ and since $\frac{D(x^*)}{D(x_i)} > 0$ one obtains

$$0 \leq \frac{\pi^* - \pi_{i+1}}{\pi^* - \pi_i} < 1$$

for all $i \in \mathbb{N}$. The latter inequality demonstrates superlinear convergence. ■

Abbreviations

AP-1 Activator protein 1. 71

BCL2 B-cell lymphoma 2. 79

BIRC5 Baculoviral IAP Repeat Containing 5. 69, 70

CASP9 aspase recruitment domain-containing protein 9. 78

CD19 B-lymphocyte antigen CD19. 78

CDC25C Cell Division Cycle 25 Homolog C / M-phase inducer phosphatase 1. 70

CDK1 Cyclin-dependent Kinase 1. 70

CHK1 Checkpoint Kinase 1. 78

CTNNB1 β-catenin. 69–71

CXCL1 Growth-regulated alphaprotein. 78

CYP Cytochrome P450. 71

FOS AP-1 transcription factor subunit / Fos proto-oncogene. 71–73

FZD10 Frizzled 10. 69

GPC3 Glypican-3. 67, 69

HRAS GTPase Hras. 78

JAK3 Januskinase 3. 79

JUN AP-1 transcription factor subunit / Jun proto-oncogene. 71–73

LCP2 Lymphocyte cytosolic protein 2. 79

LEF1 Lymphoid Enhancer-Binding Factor 1. 69

MAPK Mitogen Activated Kinase-like Protein. 70, 78

PI3KCD Phosphatidylinositol 4,5-bisphosphate 3-kinase catalytic subunit delta. 78

PI3K Phosphatidylinositol-4,5-bisphosphate 3-kinase. 70

PLK1 Polo-like Kinase 1. 70

PTK2 Protein Tyrosine Kinase. 70, 71

RAC2 Ras-related C3 botulinum toxin substrate 2. 79

S. cerevisiae *Saccharomyces cerevisiae* (budding yeast). vii, 65, 108

SOX2 SRY-box 2. 69, 70

SPP1 Secreted Phosphoprotein 1. 70, 71

SRC SRC proto-oncogene/Non-receptor tyrosine kinase. 70, 71, 78

SRY Sex-Determining Region Y. 69

SYK Spleen Tyrosine Kinase. 74, 75, 77–79, 86, 87, 108

TERT Telomere Reverse Transcriptase. 69

VAV Guanine nucleotide exchange factor. 79

VEGGC vascular endothelial growth factor-C. 79

WNT3A Wnt Family member 3a. 69

WNT Wingless-related integration site. 67–71, 107

1C one-carbon. 65, 85

ABI Applied Bioinformatics Group. xi

ANN Artificial Neural Network. 82, 84

API Application Programming Interface. ix, xiv, 8, 89–106, 108

AUC Area under the curve. 81

BMI Body Mass Index. 80

BSD Berkeley Software Distribution. 58, 101

CNCF Cloud Native Computing Foundation. 90

CRUD Create-Read-Update-Delete. 98, 100, 104

DMWCSP Directed Maximum Weight Connected Subgraph Problem. 17, 18, 22

ECM Extracellular Matrix. 78

FLP Fractional linear program/programming. 43

FMILP Fractional mixed-integer linear program/programming. 40–43, 47

FPR False Positive Rate. 62, 63

GB Gigabyte. 62

GHz Gigahertz. 62

GraphML Graph Markup Language. 92, 97, 100, 103

GSE Gene Set Enrichment. 2–4

GSEA Gene set enrichment analysis. 3, 10, 79, 80, 82

HCC Hepatocellular Carcinoma. 67, 69–71, 78

HPC High-Performance Computing. 64

HTTP Hyper-text transfer protocol. 90, 91, 97, 99, 100

ID Identifier (of an API resource). 90, 91, 93, 95, 103

IMPRS International Max Planck Research School. xi

JSON JavaScript Object Notation. 99

JWT JSON Web Token. 99, 100

KEGG Kyoto Encyclopedia of Genes and Genomes. 47, 59, 66, 69, 70, 80, 85, 86, 91

LIHC Liver Hepatocellular Carcinoma. xiv, xv, 66–70, 72–74, 77–80, 86, 149–154, 157

LP Linear program/programming. 51, 52, 56

MAWCSP Maximum Average Weight Connected Subgraph Problem. xiii, 26, 28

MILP Mixed-integer linear program/programming. xiv, 40, 44, 46–50, 56, 57, 62, 64

MWCSP Maximum Weight Connected Subgraph Problem. 17–19, 21, 31, 52

ORA Over-representation analysis. 3, 11, 14

OSI Open Source Initiative. 58

PCSF Prize-collecting Steiner forest. 22, 25

PCST Prize-collecting Steiner tree. 18, 19, 21, 25

RAM Random Access Memory. 62

RBF Radial Basis Function. 82, 84

REST Representational State Transfer. xiv, 89, 90, 92, 94, 96, 98, 100, 102, 104, 106

RMAWCSP Rooted Maximum Average Weight Connected Subgraph Problem. 26

RMWCSP Rooted Maximum Weight Connected Subgraph Problem. 18, 23

RNA Ribonucleic Acid. xv, 66, 67, 70, 71, 93, 149, 152

ROC Receiver Operating Characteristic. 81

SE Size Efficiency. 63

SIF Simple Interaction Format. 96

SPIA Signaling Pathway Impact Analysis. 13, 14

SSD Solid-State Disk. 62

ssGSEA Single Sample Gene Set Enrichment Analysis. 80

SVC Support Vector Classifier. 82, 84

TCGA The Cancer Genome Atlas. vii, ix, xiv, xv, 47, 65–75, 77–81, 83, 86, 93, 102, 107, 149–154, 157

TPR True Positive Rate. 62, 63

w.l.o.g. without loss of generality. 40

w.r.t with respect to. 75, 80, 81

Supporting Figures

Global upregulated RNA-Seq subgraphs (TCGA-LIHC)

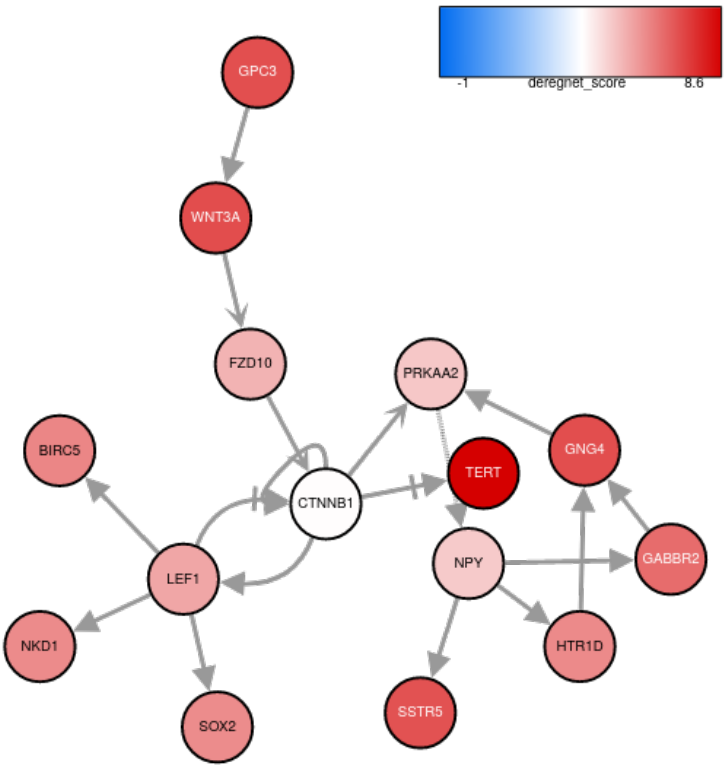


Figure A.1: Optimal upregulated global subgraph for TCGA-LIHC

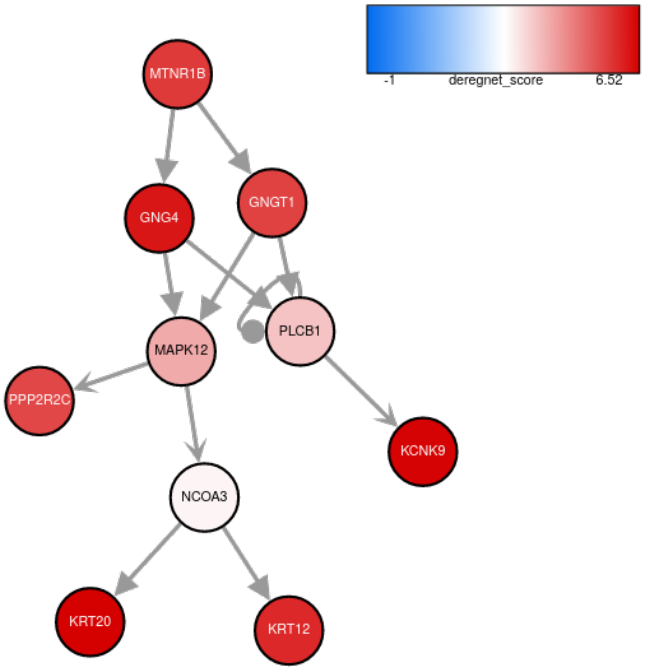


Figure A.2: 1^{st} **suboptimal upregulated global subgraph for TCGA-LIHC**

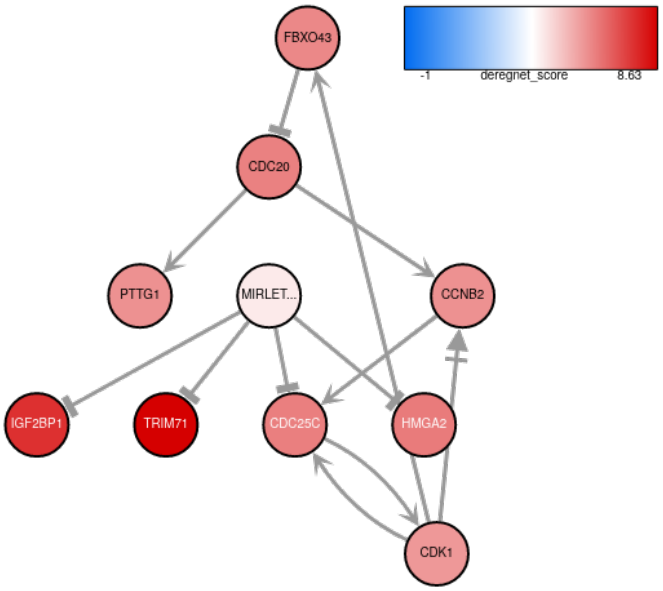


Figure A.3: 2^{nd} **suboptimal upregulated global subgraph for TCGA-LIHC**

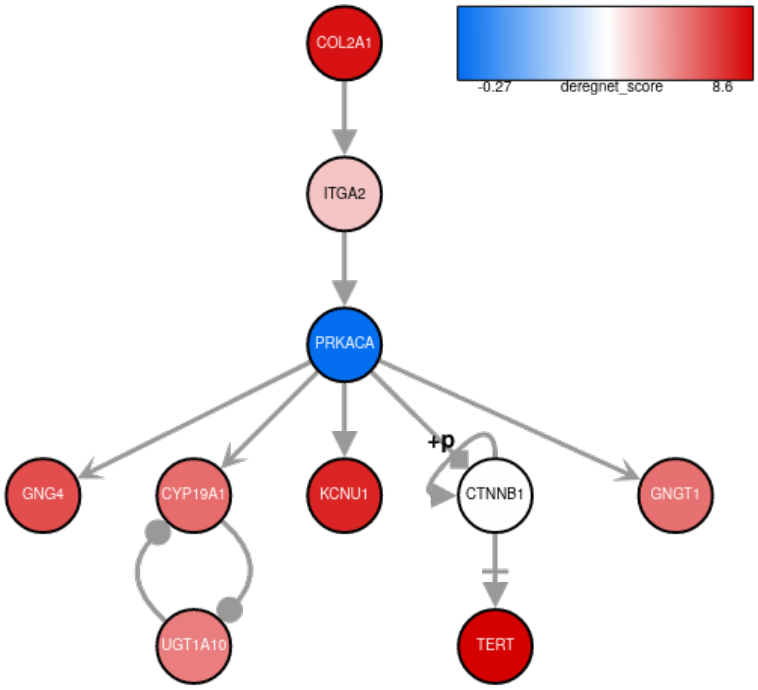


Figure A.4: 3^{rd} **suboptimal upregulated global subgraph for TCGA-LIHC**

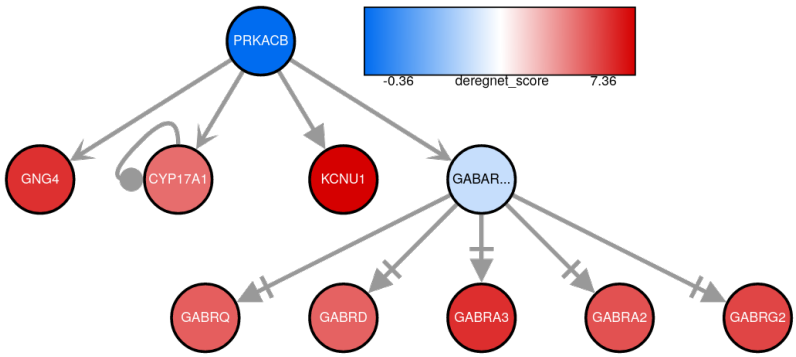


Figure A.5: 4^{th} **suboptimal upregulated global subgraph for TCGA-LIHC**

Global downregulated RNA-Seq subgraphs (TCGA-LIHC)

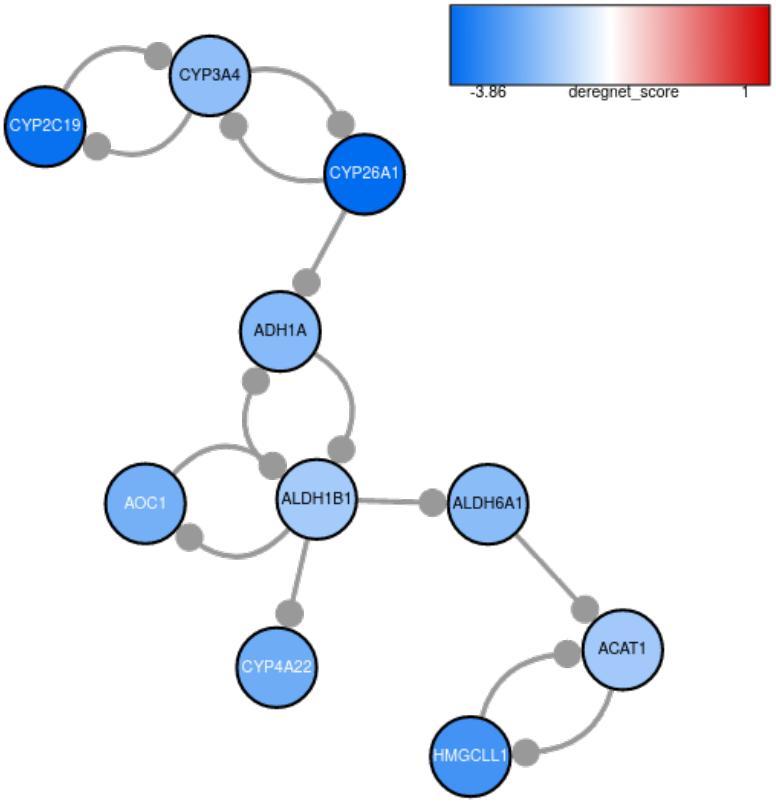


Figure A.6: Optimal downregulated global subgraph for TCGA-LIHC

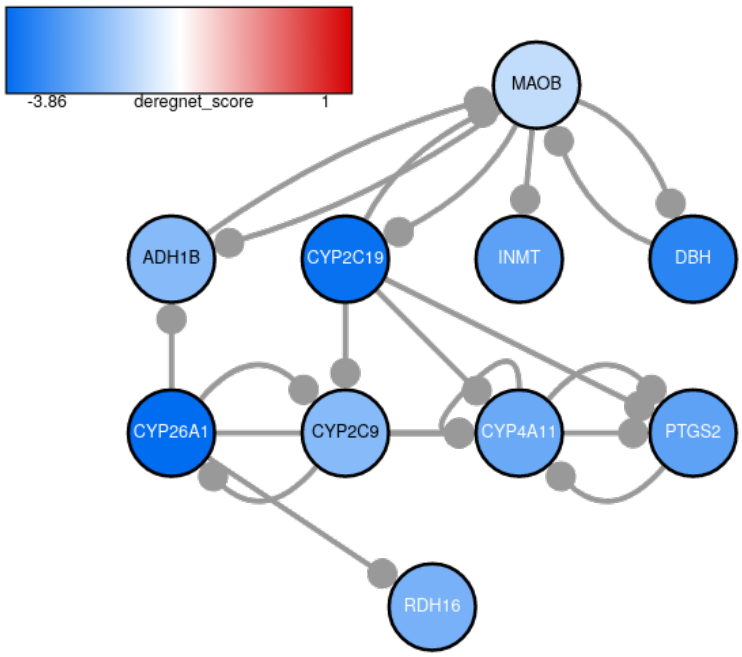


Figure A.7: 1^{st} **suboptimal downregulated global subgraph for TCGA-LIHC**

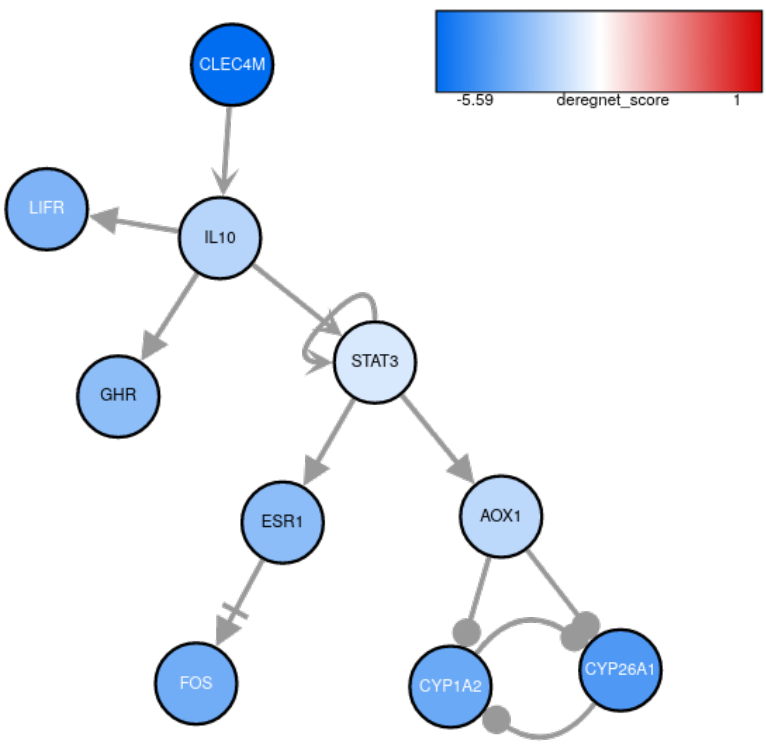


Figure A.8: 2^{nd} **suboptimal downregulated global subgraph for TCGA-LIHC**

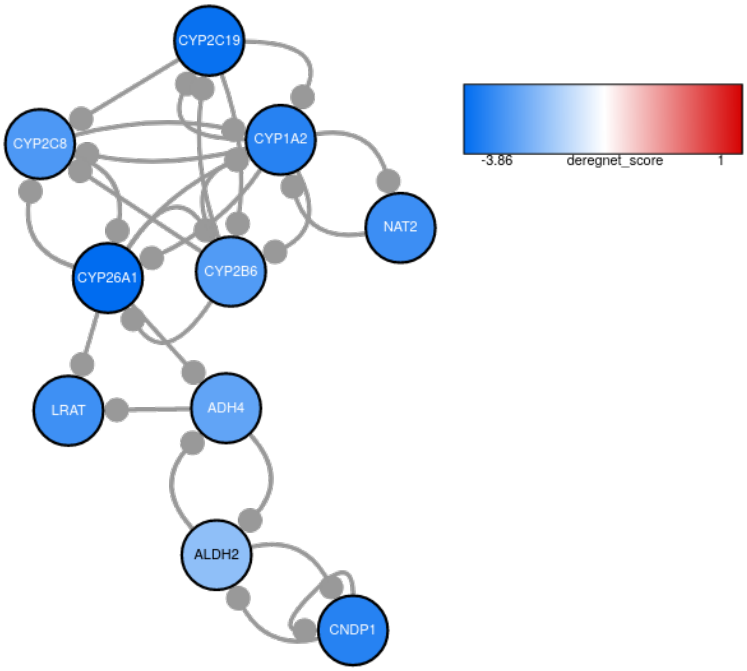


Figure A.9: 3^{rd} **suboptimal downregulated global subgraph for TCGA-LIHC**

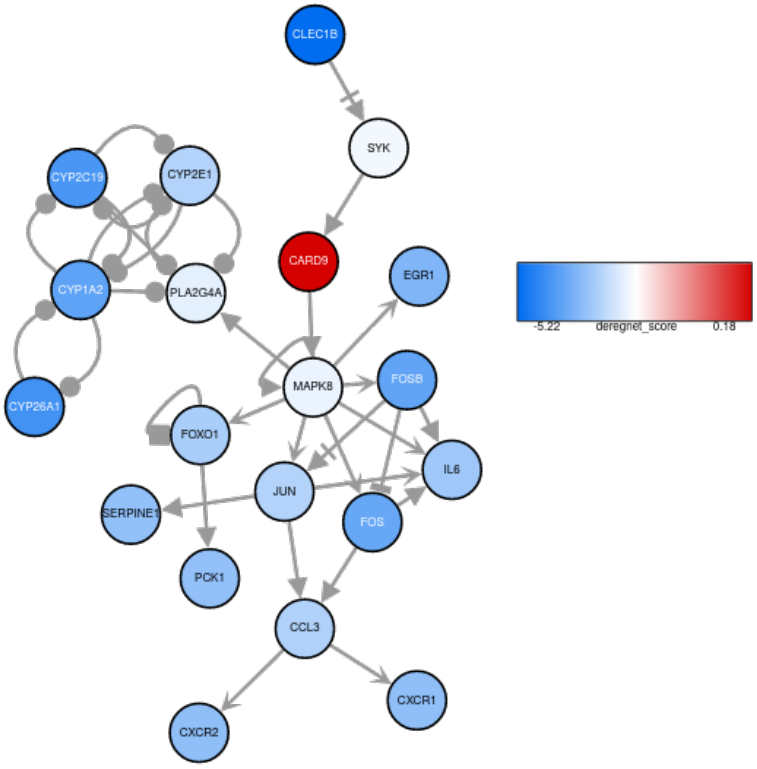


Figure A.10: 4^{th} **suboptimal upregulated global subgraph for TCGA-LIHC**

Contributions

Jennifer Ewald added the legend in the network figure 4.11 and improved the layout. Ivana Winkler arranged the network figure 4.10. Ivana Winkler and Alfred Nordheim double-checked the biological interpretations of the deregulated subgraphs and gave valuable hints for improvements. Mirjam Figaschewski implemented the tool BioGraphVisArt [bio] which was used to generate the supporting figures.

Publications

The main methodological work on DeRegNet and the application to the TCGA-LIHC dataset are described in the following paper [WWF+] and available as a preprint [WWF+21]:

S. Winkler, I. Winkler, M. Figaschewski, T. Tiede, A. Nordheim, and O. Kohlbacher. *DeRegNet - An approach to de-novo identification of maximally deregulated subnetworks based on multi-omics data.* Submitted.

Further papers [WTD+], [ZWS+19b][1], [WBW+20], [CAC+19]:

I. Winkler, A. Thavamani, G.S. Ducker, **S. Winkler**, K. Matic, D. Weichenhan,M. Graeve, S. Pietschmann, R. Geffers, S. Czemmel, M. Codrea, S. Nahnsen,O. Kohlbacher, R. Weiskirchen, C. Plass, B. Macek, J.D. Rabinowitz, and A. Nordheim. *Metabolic Reprogramming of Serine Synthesis, Mitochondrial One-Carbon Metabolism, and Methionine Cycle Activity in Hepatocellular Carcinoma.* In Preparation.

L .Zhang, **S. Winkler**, F. P. Schlottmann, O. Kohlbacher, J. E. Elias, J. M. Skotheim, and J. C. Ewald. *Multiple layers of phospho-regulation coordinate metabolism and the cell cycle in budding yeast.* Frontiers in Cell and Developmental Biology, 7:338, 2019

I. Winkler, C. Bitter, **S. Winkler**, D. Weichenhan, A. Thavamani, J. G. Hengstler, E. Borkham-Kamphorst, O. Kohlbacher, C. Plass, R. Geffers, R. Weiskirchen, and A. Nord-

[1] [ZWS+19a]

heim. *Identification of pparγ-modulated miRNA hubs that target the fibrotic tumor microenvironment.* Proceedings of the National Academy of Sciences, 117(1):454–463, 2020

S. Choobdar, M. E. Ahsen, J. Crawford, M. Tomasoni, T. Fang, D. Lamparter,J. Lin, B. Hescott, X. Hu, J. Mercer, T. Natoli, R. Narayan, A. Subramanian, J. D.Zhang, G. Stolovitzky, Z. Kutalik, K. Lage, D. K. Slonim, J. Saez-Rodriguez, L. J.Cowen, S. Bergmann, D. Marbach, F. Aicheler, N. Amoroso, A. Arenas, K. Azhage-san, A. Baker, M. Banf, S. Batzoglou, A. Baudot, R. Bellotti, S. Bergmann,K. A. Boroevich, C. Brun, S. Cai, M. Caldera, A. Calderone, G. Cesareni,W. Chen, C. Chichester, S. Choobdar, L. Cowen, J. Crawford, H. Cui, P. Dao,M. De Domenico, A. Dhroso, G. Didier, M. Divine, A. Del Sol, T. Fang, X. Feng,J. C. Flores-Canales, S. Fortunato, A. Gitter, A. Gorska, Y. Guan, A. Guénoche,S. Gómez, H. Hamza, A. Hartmann, S. He, A. Heijs, J. Heinrich, B. Hescott,X. Hu, Y. Hu, X. Huang, V. K. Hughitt, M. Jeon, L. Jeub, N. T. Johnson, K. Joo,I. Joung, S. Jung, S. G. Kalko, P. J. Kamola, J. Kang, B. Kaveelerdpotjana, M. Kim,Y. A. Kim, O. Kohlbacher, D. Korkin, K. Krzysztof, K. Kunji, Z. Kutalik, K. Lage,D. Lamparter, S. Lang-Brown, T. D. Le, J. Lee, S. Lee, J. Lee, D. Li, J. Li, J. Lin,L. Liu, A. Loizou, Z. Luo, A. Lysenko, T. Ma, R. Mall, D. Marbach, T. Mattia, M. Medvedovic, J. Menche, J. Mercer, E. Micarelli, A. Monaco, F. Müller,R. Narayan, O. Narykov, T. Natoli, T. Norman, S. Park, L. Perfetto, D. Perrin,S. Pirro, T. M. Przytycka, X. Qian, K. Raman, D. Ramazzotti, E. Ramsahai,B. Ravindran, P. Rennert, J. Saez-Rodriguez, C. Schärfe, R. Sharan, N. Shi,W. Shin, H. Shu, H. Sinha, D. K. Slonim, L. Spinelli, S. Srinivasan, A. Subra-manian, C. Suver, D. Szklarczyk, S. Tangaro, S. Thiagarajan, L. Tichit, T. Tiede,B. Tripathi, A. Tsherniak, T. Tsunoda, D. Türei, E. Ullah, G. Vahedi, A. Valdeo-livas, J. Vivek, C. von Mering, A. Waagmeester, B. Wang, Y. Wang, B. A. Weir,S. White, **S. Winkler**, K. Xu, T. Xu, C. Yan, L. Yang, K. Yu, X. Yu, G. Zaffaroni,M. Zaslavskiy, T. Zeng, J. D. Zhang, L. Zhang, W. Zhang, L. Zhang, X. Zhang,J. Zhang, X. Zhou, J. Zhou, H. Zhu, J. Zhu, and G. Zuccon. *Assessment of network module identification across complex diseases.* Nat Methods, 16(9):843–852, 09 2019