Identification of Cancer Drivers, Conserved Alteration Patterns and Evolutionary Trajectories in Tumors

by

Ermin Hodzic

M.Sc., Simon Fraser University, 2014
B.Sc., University of Sarajevo, 2012

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

in the Department of Computing Science Faculty of Applied Sciences

© Ermin Hodzic 2020
SIMON FRASER UNIVERSITY
Spring 2020

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.
Approval

Name: Ermin Hodzic
Degree: Doctor of Philosophy (Computing Science)
Title: Identification of Cancer Drivers, Conserved Alteration Patterns and Evolutionary Trajectories in Tumors

Examining Committee: Chair: Valentine Kabanets
Professor

Martin Ester
Senior Supervisor
Professor

Cenk Sahinalp
Supervisor
Senior Investigator, National Cancer Institute

Colin Collins
Supervisor
Professor, Department of Urologic Sciences, University of British Columbia

Faraz Hach
Supervisor
Assistant Professor, Department of Urologic Sciences, University of British Columbia

Kay C. Wiese
Internal Examiner
Associate Professor

Mona Singh
External Examiner
Professor, Princeton University

Date Defended: March 25, 2020
Abstract

Over the past decade, high-throughput sequencing efforts have provided an unprecedented opportunity to identify genomic alterations that can lead to changes in gene regulation, protein structure, and function. During tumor progression, cancer cells accumulate a multitude of genomic alterations, a small fraction of which provide tumor cells with selective advantage – known as “driver” alterations. However, most of them are inconsequential “passenger” alterations that are effectively neutral and greatly outnumber driver alterations. Moreover, due to a high amount of heterogeneity, alteration landscapes of tumors differ between different patients and different sites.

This thesis first presents HIT’nDRIVE, a method that integrates genomic and transcriptomic data to identify a set of patient-specific, sequence-altered potential driver genes, with sufficient collective influence over dysregulated transcripts through interactome. Applied to 2200 tumors, HIT’nDRIVE revealed many potentially clinically actionable driver genes and demonstrated its robustness in selecting cancer-implicated drivers. The results also show that small network modules seeded by HIT’nDRIVE-selected drivers significantly improve classification of cancer phenotypes and drug efficacy in pan-cancer cell lines compared to alternative methods and approaches.

Next, a method for detection of functionally meaningful and recurrent alteration patterns within gene interaction networks, cd-CAP, is presented. In a number of TCGA data sets, cd-CAP identified large subnetworks with identically conserved alteration patterns (across many tumor samples), that were significantly associated with patients’ clinical outcome.

As multi-region, time-series and single cell sequencing data become more widely available, computational methods have been developed with the goal of inferring the subclonal composition and evolutionary history of tumors from tumor biopsy sequencing data. Unfortunately, the phylogenetic trees reported for many tumor samples differ significantly from other tumors with similar characteristics. This thesis presents CONETT, the first computational method for detection of conserved trajectories of alteration events in tumor evolution. Applied to two multi-region sequencing data sets of 100 tumors each, CONETT confirms all findings of the original studies and identifies additional repeated trajectories.

**Keywords:** alterations; cancer drivers; pathways; tumor evolution; conserved trajectories
Acknowledgements

I would like to thank my supervisor, Dr. Cenk Sahinalp, for years of his guidance and leadership of the projects that I have been involved in. I would also like to thank Dr. Colin Collins, Dr. Martin Ester and Dr. Faraz Hach for their support and help throughout my studies.

I would like to thank the main collaborators on my projects, most notably Dr. Rau-nak Shrestha, who has been involved in all my projects, as well as Kaiyuan Zhu for his contribution to the cd-CAP project. I would also like to thank all members of the SFU Computational Biology lab, past and present, with whom I have shared thoughts, troubles and memories throughout these years.
# Table of Contents

Approval ii

Abstract iv

Acknowledgements v

Table of Contents vi

List of Tables ix

List of Figures x

List of Algorithms xiii

1 Introduction 1

1.1 Types of Somatic Alterations ................................. 2

1.2 Patterns of Somatic Alterations in Cancer .................... 4

1.3 Cellular Pathways in Cancer ................................. 6

2 Background on Detection of Cancer Driver Genes and Pathways 9

2.1 Driver SNVs and The Background Mutation Rate ................ 9

2.2 Modelling Driver Activity via Interaction Networks ........... 11

2.3 Modelling Topology-Derived Influence .......................... 14

2.3.1 Network Diffusion ........................................ 14

2.4 Cancer Pathway Detection via Patterns of Mutual Exclusivity ... 16

2.5 Cancer Pathway Detection via Patient Coverage ............... 18

3 Patient-Specific Multi-Driver Gene Prioritization for Precision Oncology 21

3.1 Methods .................................................. 23

3.1.1 Overview ............................................. 23

3.1.2 Gene Influence and (Multi-)Hitting Time ................... 24

3.1.3 Calculating Hitting Time on an Interaction Network ........ 26

3.1.4 Estimating Multi-Source Hitting Time via Single-Source Hitting Times 27

3.1.5 Random Walk Facility Location problem .................... 28
3.1.6 Driver Modules and Phenotype Classification ........................................... 31
3.1.7 Derivation of Expression-Outlier Genes ................................................. 31
3.1.8 Derivation of Expression-Outlier Gene Weights ...................................... 32
3.1.9 Association of Driver Modules With Patients’ Survival Outcome ............. 32
3.1.10 Statistical Significance of the Overlap With the CGC Database ............. 33
3.2 Data and Pre-Processing ............................................................................. 33
3.3 Results ....................................................................................................... 36
   3.3.1 HIT’nDRIVE-Predicted Candidate Driver Genes in Multi-Omics Cancer Datasets ........................................................................................................... 36
   3.3.2 Network Properties of Cancer Driver Genes .......................................... 39
   3.3.3 Sensitivity of Identification of Infrequent Drivers to Sample Size .......... 40
   3.3.4 Sensitivity of HIT’nDRIVE to the Underlying Network ....................... 43
   3.3.5 Randomized Alteration and Outlier Labels Testing ............................... 45
   3.3.6 HIT’nDRIVE’s Ability to Capture CGC genes ........................................ 46
   3.3.7 CGC Cancer Type-Specific Gene Enrichment ........................................ 46
   3.3.8 Phenotype Classification Using Dysregulated Modules Seeded With the Predicted Driver Genes ................................................................. 48
   3.3.9 Significance of Driver Modules for the Phenotype Classification .......... 50
   3.3.10 HIT’nDRIVE seeded driver genes accurately predict drug efficacy ....... 54
   3.3.11 Correlation of Predicted Driver Genes With Alteration Burden .......... 55
3.4 Discussion ................................................................................................. 57

4 Conserved Alteration Patterns in Cancer ......................................................... 60
   4.1 Contributions of cd-CAP ......................................................................... 61
   4.2 cd-CAP Formulation ............................................................................. 64
      4.2.1 Solving cd-CAP .............................................................................. 66
      4.2.2 Additional Constraints and Parameter Options ................................. 69
      4.2.3 Assessing the Statistical and Biological Significance of the Networks Identified by cd-CAP ................................................................. 70
   4.3 Results .................................................................................................... 71
      4.3.1 Datasets and data processing ............................................................. 71
      4.3.2 Maximal Colored Subnetworks Across Cancer Types ....................... 73
      4.3.3 Maximal Colorful Subnetworks Across Cancer Types ....................... 76
      4.3.4 Multiple-Subnetwork Analysis Across Cancer Types ....................... 82
      4.3.5 Empirical P-Value Estimates Confirm the Significance of cd-CAP Identified Networks ................................................................. 84
      4.3.6 Multi-colored Subnetwork Model Improves Survival Outcome Association ................................................................. 84
   4.4 Discussion .............................................................................................. 87
5 Identification of Conserved Evolutionary Trajectories in Tumors

5.1 Methods

5.1.1 ILP formulation for solving the Maximum Conserved Evolutionary Trajectory Tree problem

5.1.2 Not transitive input graphs

5.1.3 Optional Constraints and Parameters

5.1.4 Polynomial-Time Algorithm for the (Unconstrained) Maximum Depth Spanning Tree Problem

5.2 Assessment of the Statistical Significance of MPCSI Solutions

5.3 Data and Pre-Processing

5.3.1 Clear-cell renal cell carcinoma (ccRCC)

5.3.2 Non-small-cell lung cancer (NSCLC)

5.4 Conserved evolutionary trajectories in ccRCC

5.5 Conserved evolutionary trajectories in NSCLC

5.6 Discussion And Future Work

6 Summary

Bibliography
List of Tables

Table 4.1 Interaction networks used in cd-CAP analysis. . . . . . . . . . . . . . 72
Table 4.2 Properties of subnetworks identified in the multi-subnetwork mode . . 82
Table 4.3 P-value estimates for maximal colored subnetworks . . . . . . . . . . 85
Table 4.4 P-value estimates for maximal colorful subnetworks using STRING v10
  PPI subnetwork . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Acquisition of passenger and driver alterations in a single cell in cancer</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Types of single nucleotide and indel variants based on the effect that they have on the resulting protein product</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Recurrence frequency of altered genes in primary vs metastatic tumors</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>A schematic representation of the DriverNet approach</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>HotNet2 scheme</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>A set of mutually-exclusive genes in TCGA glioblastoma</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Overview of three settings for which BeWith was applied</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview of the HIT’nDRIVE algorithmic framework</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Summary of driver genes prioritized by HIT’nDRIVE</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>Network properties of driver genes</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Infrequent driver prioritization sensitivity to sample size</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>HIT’nDRIVE identified driver genes with respect to network perturbation in 100 select BRCA samples</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>HIT’nDRIVE identified driver genes with respect to underlying network used in 100 select BRCA samples</td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>HIT’nDRIVE identified CGC genes using randomized input data in 100 select BRCA samples</td>
<td>45</td>
</tr>
<tr>
<td>3.8</td>
<td>Likelihood of HIT’nDRIVE to capture CGC Genes</td>
<td>47</td>
</tr>
<tr>
<td>3.9</td>
<td>P-value Distribution of the likelihood of HIT’nDRIVE to pick CGC genes</td>
<td>48</td>
</tr>
<tr>
<td>3.10</td>
<td>Schematic Diagram of HIT’nDRIVE-unsupervised approach to prioritize driver-modules</td>
<td>49</td>
</tr>
<tr>
<td>3.11</td>
<td>Phenotype classification using driver-seeded modules</td>
<td>51</td>
</tr>
<tr>
<td>3.12</td>
<td>Comparison of HIT’nDRIVE+OptDis based modules against randomly selected modules</td>
<td>52</td>
</tr>
<tr>
<td>3.13</td>
<td>Comparison of HIT’nDRIVE+OptDis based modules against linear combination of all genes</td>
<td>52</td>
</tr>
<tr>
<td>3.14</td>
<td>Phenotype Classification using CGC Genes Seeded Modules</td>
<td>53</td>
</tr>
<tr>
<td>3.15</td>
<td>Unsupervised Clustering of BRCA subtypes in TCGA-BRCA cohort</td>
<td>53</td>
</tr>
</tbody>
</table>
Figure 3.16  Phenotype classification of BRCA subtypes using driver-seeded modules  

Figure 3.17  Drug efficacy predicted by HIT’nDRIVE seeded driver genes  

Figure 3.18  Correlation between the number of driver genes predicted by HIT’nDRIVE with mutation rate and copy-number burden  

Figure 4.1  Schematic Overview of cd-CAP Framework  

Figure 4.2  Comparison of cd-CAP’s framework to previous methods.  

Figure 4.3  Maximal conserved colored subnetworks using STRING v10 PPI network  

Figure 4.4  Sample conserved colored subnetworks using STRING v10 PPI network  

Figure 4.5  Maximal colorful subnetworks using STRING v10 PPI network  

Figure 4.6  Sample colorful subnetworks using STRING v10 PPI network  

Figure 4.7  Maximal conserved and colorful subnetworks using STRING v10.5 PPI network with high confidence and experimentally verified edges  

Figure 4.8  Maximal conserved and colorful subnetworks using HPRD PPI network  

Figure 4.9  Maximal conserved and colorful subnetworks using REACTOME 2015 PPI network  

Figure 4.10  Multiple Subnetwork Analysis  

Figure 4.11  Distribution of maximum subnetwork sizes identified by cd-CAP on permuted data for p-value calculation  

Figure 4.12  Comparison of cd-CAP to a single-color version on TCGA COAD and GBM datasets  

Figure 4.13  Comparison of cd-CAP to a single-color version on TCGA BRCA subtype datasets
Overview of CONETT’s framework. The method takes as its input a partial ancestor-descendant ordering of alteration events across a number of tumors and produces a tree of conserved evolutionary trajectories stemming from a common root event which maximizes the number of ancestor-descendant ordered pairs of nodes. **A) Input.** Tumor graphs represent partial ancestor-descendant order of alteration events in tumors. In this particular example, any pair of events that do not have an actual edge between them in a tumor should be thought to have an “anti-edge”; similarly any event not present in a tumor should be thought to have a “don’t care” edge to all events present in that tumor. **B) The Conserved Evolutionary Trajectory Tree.** In the most stringent setting CONETT computes the maximum total depth “consensus” phylogeny tree where the ordering of alteration events do not conflict with the partial orderings of alteration events in any of the tumors.

**Figure 5.2** The ILP formulation for inferring the conserved evolutionary trajectory tree

**Figure 5.3** Illustration of the importance of propagation of new paths throughout the subgraph during execution of the MPCSI-solving algorithm

**Figure 5.4** Recurrence frequency of altered genes and chromosome arms in TRACERx renal cohort

**Figure 5.5** Recurrence frequency of altered genes and chromosome arms in TRACERx lung cohort

**Figure 5.6** Conserved evolutionary trajectories identified through CONETT in renal cancer

**Figure 5.7** VHL-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC

**Figure 5.8** loss3p-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC

**Figure 5.9** Germline-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC

**Figure 5.10** PBRM1-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC

**Figure 5.11** loss14q-seeded tree of trajectories with $t = 11$ in TRACERx ccRCC

**Figure 5.12** gain5 and gain5q seeded trees of trajectories with $t = 10$ in TRACERx ccRCC

**Figure 5.13** Conserved evolutionary trajectories identified through CONETT in lung cancer

**Figure 5.14** Germline seeded tree of trajectories with $t = 5$ in TRACERx NSCLC

**Figure 5.15** TP53 seeded tree of trajectories with $t = 5$ in TRACERx NSCLC
# List of Algorithms

1. The ILP formulation for the WMSC problem ........................................ 30
2. Generating candidate subnetworks .................................................... 67
3. ILP formulation for solving MCSC ................................................... 68
4. Maximum Depth Spanning Tree ...................................................... 102
5. Maximum Depth Spanning Tree - Depth First Search ....................... 103
Chapter 1

Introduction

Over time, cells in the human body accumulate alterations in their DNA, which is continuously damaged by both internal and external factors. Normally, most of that DNA damage gets repaired through cellular DNA repair mechanisms, even though a small fraction of the damage might remain. In case of irreparable DNA damage, otherwise healthy cells then undergo the process of cell death, and new cells take their place. However, at times, a cell can acquire a sufficient set of alterations that give it enough of an advantage to proliferate unhindered, forming growths of new unneeded cells, which are called tumours. Tumours can be benign, in which case they are localised and do not spread outside of the location where they are formed. They can also be malignant, growing quickly and out of control, spreading by invading nearby tissues and causing major issues for the organism. Cancer is the collective term for a set of diseases characterised by formation and spread of malignant tumours. A common characteristic of cancer is the occurrence of metastasis, which is the spread of cancer cells from the primary tumour site to other places in the body, through blood or lymph system, forming secondary tumours of the same type in other organs and tissues.

Over the past decade, high-throughput sequencing efforts have revealed the importance of genomic alterations in the progression of cancer [125]. During the evolution of cancer in an organism, tumour cells accumulate numerous genomic alterations which affect genes in control of cell growth and differentiation through a cascade of abnormal cell reactions. These cancer-related genes can be placed into two major classes – oncogenes and tumour suppressors, which have opposite effects on cell proliferation. Oncogenes are genes with the potential to cause cancer, when they are altered or activated, by affecting growth of cancer cells, typically as a consequence of an increase in their own activity due to external factors or inherited alterations. Tumour suppressor genes (also called antioncogenes) are genes whose protein products help control cell growth by initiating apoptosis (programmed cell death in case of irreparable DNA damage) or blocking cell division. Their inactivation may lead to uncontrolled cell division and progression of tumour [151]. Development of
Figure 1.1: Acquisition of passenger and driver alterations in a single cell in cancer. Some alterations might be acquired while the cell is phenotypically normal (i.e. not a tumour cell), either through internal or external mutagen factors. As cancer starts developing, the cell starts acquiring further alterations, possibly due to damaged DNA repair mechanism. Some alterations do not have any positive effect for the cell growth, but certain alterations (drivers) will provide the cell with the necessary mechanisms for clonal expansion. Relapse after chemotherapy can be associated with resistance mutations that often predate the initiation of treatment. (Figure taken from [125])

malignant tumours typically occurs due to over-expression of oncogenes or under-expression or disabling of tumour suppressor genes.

Cancer is known to be mediated by somatically acquired alterations that accumulate in the genome over the lifetime of a human organism. Somatic alterations are alterations in the DNA which occur after conception and can be passed on through cell division. They do not affect germ cells (sperm and egg) and are not inherited, as opposed to germline alterations which are specifically occurring in germ cells and as such are incorporated into the DNA of every cell of the offspring. As a way to a better understanding of cancer and the underlying processes that lead to its development, a major goal is to analyse the alteration profiles of various tumours and extract useful patterns in the way that DNA of tumour cells is altered by somatic alterations. But, as will be described in the following subsections, there is a great amount of variability among the somatic alterations present in tumours between different patients, or even tumour samples within the same patient, which makes this task extremely difficult.

1.1 Types of Somatic Alterations

Somatic alterations in cancer genomes differ in the type of DNA (or cellular) damage that they inflict, as well as in ways that they can be detected to begin with. They generally fall in the following different classes. The first class are single nucleotide variations, which
Figure 1.2: **Types of single nucleotide and indel variants based on the effect that they have on the resulting protein product.**

A) A missense SNV changes the DNA sequence of a codon, causing it to code for another amino acid; B) A nonsense SNV changes a codon into the stop codon, causing the reading of DNA to stop prematurely, or removes an existing stop codon and causes translation to continue past the point at which it should stop; C) A silent SNV changes the DNA sequence of a codon without changing the resulting amino acid; D) An inframe indel inserts or removes nucleotides in total length that is a multiple of three, without changing the reading frame of the gene; E) A frameshift indel removes (or inserts) a number of nucleotides that is not a multiple of three, affecting any following stop codons and causing the reading frame to completely change; F) A splice site mutation happens at a boundary of an exon or an intron and can cause an exon not to be translated, or an intron to be translated, and as a result changes the amino acid sequence.

represent substitutions of one nucleotide with another at a specific location in the genome. Another class consists of insertions or deletions of small or large DNA segments into or out of (respectively) the genome. These changes are often given a common name of indels, and represent smaller-scale events. Splice-site alterations are alterations in the DNA sequence which are located at the boundaries of exons and introns (called splice sites), and whose effect results in inclusion of an intron or an exclusion of an exon from the messenger RNA. Lastly, there is the class of large-scale genomic rearrangements, where large segments of DNA (pieces of chromosome) are broken and cut out and then moved to another location or joined with another large broken segment; as well as copy number increases, where segments of genome can be repeated multiple times (increase from the two copies in normal diploid genome to multiple), and copy number decreases (loss of DNA). These events differ from indels in that they can occur at a much larger scale, affecting whole chromosomes.

Single nucleotide variations are further classified into nonsense mutations, which can cause loss of start or stop codons or their premature occurrence in the messenger RNA; missense mutations, which result in a codon that codes for a different amino acid than the original; and silent mutations, which do not significantly alter the phenotype. Among the
single nucleotide variations within the coding region (the part of DNA that codes for proteins), we differentiate between synonymous substitutions, which do not alter the resulting amino acid (which is possible because some amino acids are coded by multiple different codons), and nonsynonymous substitutions, which do alter the amino acid sequence.

Indels typically affect a larger number of nucleotides, and typically have a very high impact on the protein product if they affect the coding region of the DNA. Unless the length of the indel is a multiple of three, it will result in a frameshift mutation, which represents a change of the reading frame, impacting interpretation of all original codons following the indel. As such, frameshift mutations completely alter the result of translation and the resulting protein product.

Splice sites are located at the ends of introns (segments of DNA between exons), which are normally not passed on into messenger RNA (mRNA) from pre-mRNA, whereas exons are. An alteration in the splice site can cause the intron to be erroneously included in the mRNA, or cause the exon next to the intron to be lost. The alteration can be an insertion, deletion, frame shift, etc.

Large genomic rearrangements are alterations of large segments of chromosomes, possibly affecting whole chromosomes or their arms, ultimately causing structural changes to the chromosomes. Aside from deletions, duplications and insertions, they also include translocations, which cause pieces of chromosomes to break off and get attached to other chromosomes, sometimes trading places; and inversions, which completely reverse a segment inside the chromosome from one end to the other. Fusion genes can arise as a consequence of translocations, deletions and inversions.

1.2 Patterns of Somatic Alterations in Cancer

As previously mentioned, numerous somatic alterations accumulate over time within cancer cells, while the rest of the existing alterations are inherited from an ancestor of the cell. Even in normal cells, DNA is continuously damaged by both internal and external factors. While most of the damage in normal cells gets repaired, a small fraction of alterations may remain [125].

In cancer cells, the mutation rate is higher than in normal cells, leading to a very diverse alteration profile. However, not all alterations are significant for the development of cancer and some of them do not contribute to it at all. Therefore, a distinction is made between them based on the effect that they have with respect to progression of cancer. As a result, we divide all alterations into drivers and passengers. Passenger alterations are functionally inconsequential for the progression of cancer and arise as a consequence of driver alterations or they could be present in the ancestors of the cell. Driver alterations are those that confer crucial growth advantage to the cancer cells. The genes carrying these driver alterations are therefore known as cancer genes or driver genes.
Identifying driver alterations and their respective cancer genes is an interesting goal because of their potential to be used as therapeutic targets in treatment of cancer patients. With further development of targeted cancer therapies and improvement in genome analysis technology, genome-wide surveys of cancer are resulting in tools for diagnosis as well as discovery. They also represent interesting markers for diagnosis, as presence of a cancer driver that is linked to a specific subtype of cancer helps guide the patient towards a treatment with a more likely positive outcome. However, identification of driver alterations and the specific genes that they alter poses a significant challenge as they are greatly outnumbered by passenger alterations which do not contribute to progression of cancer, but they do contribute further towards cancer heterogeneity [125][150]. Tumours with damaged DNA repair mechanism can have thousands of point mutations and most solid tumours can have dozens of large genomic rearrangements [150]. Intuitively, one reason why cancer cells have a much higher rate of passenger mutations might be that they are more capable of surviving such large chromosomal alterations – for example by inactivating tumour suppressors and avoiding activation of cell death. Nevertheless, this “ability” just further complicates the analysis of the whole process, search for crucial driver alterations and identification of potential therapeutic targets in individual patients.

Tumour heterogeneity poses a very difficult challenge to the efforts to compose a complete list of genes responsible for driving cancer progression that would serve as targets for therapy, development of drugs or diagnosis. There are different kinds of heterogeneity in cancer. Intratumoural heterogeneity, which is the term for occurrence of different somatic mutations in different cells of the same tumour, increases the difficulty for computational
methods to extract useful information by providing background noise. While some of these unique somatic mutations are rarely significant in primary tumours, combined with metastasis they can result in several secondary tumours with very different genetic alteration profiles. On the other hand, the interpatient heterogeneity poses another challenge as no two patients with the same cancer type seem to have the same alteration profile. Not only are the altered genes often very different, but also the types of alterations in the genes that they do have altered in common are often very different [150]. Aside from issues with detection of drivers, the treatment of cancers is also complicated. There are differences in the way that therapy of oncogenes and tumour suppressors can be conducted, even if we assume that we are able to easily find drivers of cancer. For example, while in case of some oncogenes there might be drugs which can bind to their protein products and interfere with their activity, drugs generally cannot repair function of damaged genes, such as inactivated tumour suppressors.

1.3 Cellular Pathways in Cancer

Tumour heterogeneity and the problem of driver genes that are patient-specific make results of many of the methods that aim to detect driver genes have lesser impact, as what is discovered in one patient (or a small subset of patients) can rarely be translated into a useful pattern for other patients. However, that problem seems to stem from looking at cancer as a disease of genes, rather than as a disease of pathways. While different patients that have the same cancer type rarely have the same genes altered, their alterations tend to affect the same cellular pathways when viewed on a protein level. It is important to note that, contrary to what the term “path” means in graph theory of computer science, the term “pathway” in a biological setting can represent any set of proteins or genes and transmissions of signals among them. Therefore, the most accurate way to think about them would be as subgraphs of the signalling network of the cell.

Human cells are complex systems that are composed of many different and complex parts, whose behaviour is regulated by signals that are transmitted throughout the cell via signalling pathways. For example, gene expression, apoptosis and cell metabolism are all regulated. Regulation may originate from external factors, picked up by the receptors on the exterior of the cell, but it may also originate from within the cell. An example of a pathway that is very important in cancer is the p53 pathway, which activates upon DNA damage and acts as a tumour suppressor. In normal circumstances, the p53 protein is inactivated by MDM2 protein which negatively regulates its activity. DNA damage leads to breaking of the inhibition of p53 by MDM2, thereby activating p53. Once activated, p53 activates the downstream processes that induce cell cycle arrest, allowing the DNA to be repaired, or it leads to cell death in case that DNA damage is found to be irreparable. It is often the case in cancer cells that the functioning of the p53 pathway is disrupted, perhaps by the presence
of damage on the TP53 gene (whose protein product is the p53 protein) or overexpression of MDM2, which severely damages the cell’s ability of tumour suppression. This highlights the importance of research of pathways and the ways that genes are able to affect the functions of one another, as inactivation of a tumour suppressor typically activates growth-promoting downstream signal that was regulated and constrained by it.

Previous research indicates that many mutational events tend to target a limited set of biological pathways. For example, in TCGA glioblastoma dataset [101] there are alterations within the p53 pathway in up to 87% patients, even though the exact mutation mechanism varies from patient to patient. In the same dataset, it has been noticed that TP53 mutation and MDM2 copy number amplification are mutually exclusive (only one of them is altered) across almost all patients – in very few patient samples are both present [25]. Since we know that MDM2 protein inhibits the activity of p53’s protein level, and that disruption of normal function of either of the two genes (or their protein products) leads to dysregulation of the cell cycle arrest, DNA repair and apoptosis, it makes sense that tumour cells do not need both to be significantly altered in order to gain proliferative advantage. This explains the prevalence of their mutual exclusivity in tumour samples, as adding more alterations does not provide further selective advantage. Actually, it is possible that further increase of the alteration rate would actually hinder the tumour cell and put it at a disadvantage, as it could lead to, for example, cell death [25].

The above examples demonstrate that genes interact with one another. For example, MDM2’s protein product inhibits TP53’s protein product, and there is a negative feedback loop from TP53 that induces MDM2. Binding of MDM2 protein to p53 protein represents a protein-to-protein type of an interaction between two genes, which in itself is an indirect gene-to-gene interaction, since proteins are translated from RNA, which is transcribed from DNA. There are many types of indirect interactions between genes that happen on the protein level, an example being a protein-to-protein interaction called phosphorylation, where one protein phosphorylates another, thereby (de)activating it. In fact, the p53 tumour suppressor protein contains multiple phosphorylation sites and is heavily regulated [6], and has a large number of known direct interaction partners. Another example of an indirect gene-to-gene interaction is binding of transcription factor proteins to enhancer or promoter regions of DNA and up or down regulating transcription of the adjacent gene (protein-to-DNA type of interaction). This represents a gene-to-gene interaction where one gene is able to directly alter the expression level of another.

Being able to better understand how genes interact with one another and how signals are propagated throughout the signalling pathways of cells represents an important goal in trying to better understand cancers. For that purpose, numerous, publicly available, databases of known pathways and regulatory and functional interactions between genes and proteins have been created and improved upon over the recent years. Reactome [31][46] and KEGG [67][68][69] represent well-known databases of known cellular pathways that capture current
knowledge. Reactome also provides a list of protein and gene interactions that are extracted from known processes in the pathways within the database. On the other hand, there are databases focused only on maintaining a list of known protein (or gene) interactions, such as Human Protein Reference Database (HPRD) [110] and STRING [130], as opposed to collecting and providing full information about the structure and signal flow on a pathway level. STRING goes a step beyond by also trying to infer new and currently experimentally unconfirmed interactions between proteins by analyzing co-expression, protein homology and employing text mining. Since not all interactions are experimentally confirmed, all interactions are given probability-like scores that represent a level of likelihood that the interaction actually occurs. Since the process of tumour evolution differs in different tissue types, as different cells and tissues types differ in their response to oncogenic driver mutations [118], there are also databases that provide information about tissue-specific interactions for more precise analyses, such as TissueNet2 [9] and Integrated Interactions Database [76].
Chapter 2

Background on Detection of Cancer Driver Genes and Pathways

2.1 Driver SNVs and The Background Mutation Rate

The earliest computational methods for detection of cancer drivers were based on the assumption that driver mutations would represent a common pattern (shared by a significant portion of tumour samples) of point mutations among samples of the same tumour, and that they would occur at a rate that is higher than what could be expected from the available data. The notion of the expected mutation rate was given by what is called the background mutation rate. The way that background mutation rate is calculated differs based on the method, and has received significant upgrades in the literature over the past decade, but, in essence, it represents a mutation rate at which it is expected to see genes mutated without considering it to be a significant event. On the other hand, if a gene were found to be mutated at a rate that rises above the background mutation rate, then that would indicate a possible cancer driver. The underlying assumption is that passenger mutations are distributed more or less randomly, while driver mutations are expected to be clustered in a small subset of genes that are crucial for the development of cancer. Once the background mutation rate is calculated, it is compared to the mutation frequency of each gene, which is defined as the number of mutations per megabase. The biggest challenge to methods in this category lies in filtering out false positives, as simply looking at the frequency of the mutations is not enough to be certain that a gene is a cancer driver.

One of the early methods, by Greenman et al [52], assumes that mutations occur independently of one another, and gives a probabilistic model of the likelihood of observed counts and rates of silent, missense, nonsense and splice mutations based on a Poisson process. Then, using likelihood ratio tests and score statistics, they compare different hypotheses about selection pressure of different mutation types, in order to infer the most likely source of cancer progression based on their model. Their conclusion was that nonsense and splice mutations have a higher selection pressure in tumour samples.
In the following series of papers, where each represents work that improves upon the methodology of the previous, silent mutations are assumed to be passengers. Initially, the silent background mutation rate was calculated as the ratio of the number of base pairs in which a silent mutation is observed, and the total number of sequenced base pairs. Using that, the non-silent background mutation rate is calculated as a product of the silent background mutation rate and the average ratio of the number of potential non-silent mutations and the number of potential silent mutations (the two numbers being estimated by experiments). Youn and Simon [157] improved upon previous methods by considering additional patterns in the way that background mutation rate is estimated. The first of those was that not all mutations have the same impact. For example, nonsense mutations and frameshift indels have greater impact on protein function than missense mutations. In short, if mutation rate between two genes is similar, but mutations are more severe in one than the other, then it is more likely to be a driver. They assign a score to every non-silent mutation in the increasing order of severity of the effect it has on transcriptome: missense, inframe indel, mutation in splice sites, frameshift indel and nonsense. Their second contribution was to introduce sample-specific background mutation rate, as opposed to average rate across all samples. If a gene is mutated only in samples with high background mutation rate, then it is likely a passenger. If it is mutated only in samples with low background mutation rate, then it is more likely to be a driver, rather than random noise. Lastly, they consider that different aminoacids are encoded by different number of codons, leading to different sensitivity to mutations and their types.

A significant further effort in tackling the problem of false positives was done by Lawrence et al [79], who pointed out the problem resulting from the increase in sample size – the number of genes detected as drivers was rising rapidly with the number of samples, rather than being able to better filter out the passengers. As possible reasons, based on their data analysis, they name heterogeneity of patients within a given cancer type, in mutational spectrum and across the genome (intratumoral heterogeneity). Their method, MutSigCV, corrects for variation by accounting for patient-specific mutation frequency and spectrum, and gene-specific background mutation rates, incorporating expression level and replication time. It uses surrounding genes in covariate space for the adjustment of gene background mutation rate. They report an order-of-magnitude decrease in the number of significantly mutated genes reported. For example, they report decrease from 450 to 11 genes identified as drivers of TCGA lung squamous cell carcinoma [102], compared to their previous method MutSig which did not account for differences in patient-specific mutation rates, significantly reducing the false discovery rate.

The last step represents work done by Korthauer et al [75] who, in addition to measures employed by previously mentioned methods, notice the relevance of non-random spatial patterns of mutations [150]. For example, oncogenes appear to be recurrently mutated at few amino acid positions, while tumor suppressor genes tend to have an overabundance of
truncating mutations (frameshift indels, non-sense mutations or mutations at the normal stop codon). They present their method, MADGiC, which incorporates all of the above-mentioned features in a Bayesian framework.

One of larger drawbacks of these methods is that they focus only on point mutation data, ignoring other genomic alteration events (for example, copy number changes). All the methods also appear to be favouring longer genes, which intuitively comes from their ability to acquire more mutations by chance. Since the methods are restricted to identifying genes that are mutated at a high frequency, they are also very likely to miss rare personalized drivers which are affecting only a small subset of patients but may still be functionally relevant (a consequence of interpatient heterogeneity). Examples of such drivers are SPOP mutations and CHD1 deletions which are present in less than 20% of prostate cancer patients [8][50]. Whereas recurrent drivers are hypothesized to initiate carcinogenesis and are therefore present in the majority of tumor cells, rare drivers can arise during tumor evolution and be isolated to a smaller fraction of cells due to clonal expansion [51][39]. Another drawback of these methods is that they do not consider synergy or opposition among events – every mutation is observed as separately-functioning without conflict with or benefit by the presence of another. Ultimately, they completely ignore knowledge provided by known gene interactions and groupings of genes into functional pathways.

2.2 Modelling Driver Activity via Interaction Networks

Contrary to the previously described methods that largely ignore any information about functional interactions among genes, focusing entirely on background mutation rate of single nucleotide variants, there has been extensive work done on making use of information provided by various gene (or protein) interaction networks. The methods vary greatly in their approach, which ranges from modelling existing cell signalling networks in order to infer missing links, to linking possible causal events to observable changes that they might explain, to deriving patterns of significantly altered pathways that are rich in functional gene interactions.

A series of papers that started with [140] attempted to discover the most likely route of the signal flow throughout the interaction network towards a given target gene, identifying its most likely cause and the series of interactions that most likely lead to the observed effect in the target. These methods were developed outside of cancer research and attempt to go beyond just the identification of drivers, providing information on how the disruptions were passed through the network of molecular interactions. The intuition of this approach is that drivers leave a visible trace of their activity, typically in a form of a “footprint” on the expression signature. Therefore it should be possible to discover significant correlation between the two types of events. Tu et al [140] determine for a given target gene its gene expression variation cause by using probabilistic backwards-linking to identify a path leading
from the most likely causal gene to a transcription factor in neighbourhood of the given target gene in an interaction network. It uses a random walk-like method (although “random path” might be a more appropriate name since no node is visited more than once in this probabilistic exploration stage), where transition probabilities are based on the absolute values of Pearson correlation coefficient of the linking gene with the target gene (or a small $\varepsilon > 0$ probability in case of too small correlation coefficient), which stops once it reaches one of the potential causal genes. This procedure is repeated a large number of times and the candidate causal gene that has been reached by this random walk process the most is picked as the cause. The linear pathway leading back to the target is reconstructed by starting with the chosen causal gene and picking the neighbour with the highest visit-count until the target is reached.

eQED [129] builds upon the idea of Tu et al and represents an algorithmic improvement, attempting to fix issues caused by dead ends in many of the random paths explored which in turn affect the number of visits of candidate causal genes. It does so by exploiting the convergence of the number of visits in random walks to the amount of current flowing through the node if the same amount of current as the number of random walks is let through the electrical network of the same structure as the interaction network. Since the equivalence holds only in case of undirected edges, the linear program used to solve the problem is strictly speaking a heuristic due to treating transcription factor to DNA interactions as unidirectional.

Kim et al [73] extended the eQED electric circuit model by running a post-processing step on the set of identified causal genes. They assign weights to each causal gene based on the number of targets that they explain, and then use greedy approximation of weighted set cover to find the final causal genes. They also use an iterative heuristic to solve the electric circuit model by treating all edges as undirected, and then removing edges that were assigned current in the wrong direction in the solution until the number of such edges was sufficiently small. The authors applied their method to cancer, trying to detect copy number variations as potential causal events in TCGA GBM [101] that explain observed differential expression of target genes.

A slightly different approach is represented by PARADIGM [147], which uses factor graphs to model the information flow between entities in select NCI pathways using copy number and gene expression data from TCGA GBM [101]. In their factor graph model, entities are variables that represent gene DNA, mRNA, proteins and active proteins, and they can take values of ’activated’, ’nominal’, ’deactivated’. It builds a probabilistic model from an existing pathway and associates a factor with every entity and predicts the entity’s value based on the values of the parents via a majority vote. If the expected value corresponds to the observed value, factor is given a value close to 1, and a value close to 0 otherwise. Then, they use expectation maximization in order to learn the most likely values of hidden parameters (entities) given the observed (input) patient data.
Figure 2.1: A schematic representation of the **DriverNet approach**. Given the genomic aberration states for different patients and genes, gene expression data, and the influence graph, which captures biological pathway information, the bipartite graph shown on the right is constructed. Green nodes on the left partition of the bipartite graph correspond to aberrated genes and nodes on the right represent the outlying expression status for each patient where red indicates outlying patient-gene events from the gene expression matrix. The genes with the highest number of outlying expression events are nominated as putative drivers. (Figure taken from [10])

**DriverNet** [10] aims to link mutations and copy number aberrations to changes in the expression profile of a cohort of patients, modelling the links using a bipartite graph. In the bipartite graph, an altered gene is linked to an expression outlier if both events occur in the same patient, and there is a direct protein-protein interaction between the two genes (proteins) in a given interaction network. A set-cover approach is then used to identify the minimum number of altered genes that can explain changes in the expression profile, where an alteration is considered to cover an outlier if there is an edge between the two in the bipartite graph. The problem is solved using greedy approximation algorithm where the set (altered gene) that covers the most still-uncovered elements (expression outliers) is picked in each iteration. The order of selection of the altered genes is used as an ordering representing significance of each candidate driver gene. The novelty of DriverNet is that it aims to find the minimum number of potential drivers that can cover targets. It also does not pre-filter alterations based on background mutation rate, theoretically allowing discovery of rare drivers.
2.3 Modelling Topology-Derived Influence

Rather than simply relying on existence of a known interaction between two genes, or a path made of interactions in the network, which represents binary type of information, methods have been developed that attempt to measure amount of influence that a gene can have over other genes. The notion of influence of a gene over another is related to both the proximity of the nodes in the interaction network and the topology of the network itself. This approach follows the intuition that different genes have different likelihood of causing disruptions in functions of their interacting partners, and therefore should not be treated the same.

The simple idea of relying only on the proximity of nodes in the traditional sense (the shortest path in a graph) suffers from the presence of hubs. Hubs in interaction networks arise partly as a consequence of study bias, causing the network to contain a number of nodes that share an edge with a large fraction of nodes in the network. As an example, STRING-v10 network of high-confidence gene interactions (score greater than 0.9) has a diameter less than 10 – i.e. it is possible to reach every node from any other node with a shortest path of length less than 10. In an attempt to mitigate the dominance of hubs, alternative distance measure such as network diffusion and random walks have been used. The intuition of using, for example, the diffusion method is that while hubs receive more heat due to their high connectedness, they also lose more heat for the same reason, thereby balancing out the advantage that their high degree gives them during identification of interesting and influential genes in interaction networks.

2.3.1 Network Diffusion

Network diffusion process, as described in [112], is defined on a directed network and measures the amount of fluid (or alternatively, heat) that is diffused from selected source nodes to all other nodes in the network. Initially, nodes contain no fluid in them, and there is no flow happening. Then, fluid is constantly pumped into source nodes and diffuses throughout the network from node to node based on the edges and weights in the adjacency matrix. At each node, a $\gamma$ fraction of the fluid is lost and does not get passed through the edges (or in case of heat diffusion, a portion of heat is retained in the node). As this process reaches the equilibrium or steady state, no flow occurs anywhere in the network anymore and each node contains a certain amount of fluid (heat) which depends on the source nodes, network topology and the parameter $\gamma$.

Let $A_{n \times n}$ be the adjacency matrix of the network; $\vec{b} = (b_1, b_2, ..., b_n)$ a column vector where $b_i = 1$ if node $v_i$ is a source node, and $b_i = 0$ otherwise; and let $L = D - A + \gamma I$ (graph Laplacian shifted by $\gamma$), where $D$ is a diagonal degree matrix (each diagonal entry $D_{i,i}$ is equal to sum of entries in column $i$ of $A$, and non-diagonal entries are zeroes). Then the vector of the steady-state distribution of fluid remaining in the nodes of the network is
Figure 2.2: **HotNet2 scheme.** HotNet2 finds significantly mutated subnetworks using a diffusion process on a protein-protein interaction network. Each node (protein) is assigned a score (heat) according to the frequency and significance of SNVs or CNAs in the corresponding gene. Heat diffuses across the edges of a network. Subnetworks containing nodes that both send and receive a significant amount of heat (outlined) are reported. (Figure taken from [81])

given by $\vec{f}_b = L^{-1}\vec{b}$. Since $\vec{b}$ is a binary vector, its multiplication by $L^{-1}$ simply represents extraction and addition of certain columns from the matrix (those corresponding to non-zero dimensions of $\vec{b}$). Therefore, $L^{-1}$ contains the whole network topology information coupled with the parameter $\gamma$ representing the amount of fluid lost at each node, and is independent of the sources of the flow.

HotNet [144] uses the diffusion process on the interaction network to pre-compute influences of genes over one another, obtaining an influence matrix $F$ where $F_{i,j}$ represents the amount of fluid left at node $v_j$ at the steady state, given that the single source of the flow was node $v_i$. The interaction network is transformed into an influence graph by assigning weight to each edge equal to the minimum of the influence from one end of the edge to the other, and removing every edge whose weight is smaller than threshold $\delta$. HotNet employs an approximation algorithm to compute a connected subgraph of the influence graph of size $k$ which contains mutations from the maximum number of samples. The algorithm considers each node a seed, and proceeds to iteratively add nodes on paths chosen to maximize the ratio of the number of new samples covered by the nodes of the path over the number of new nodes added to the subnetwork. It stops once the subnetwork contains $k$ nodes. Resulting “optimal” subnetwork is chosen among the identified subnetworks for each seed. The method was used to identify subnetworks of the HPRD interaction network [110] enriched with genes mutated in TCGA glioblastoma and lung adenocarcinoma datasets [101].

Later developed HotNet 2 [81] represents an extension of the original method, used for analysis of TCGA Pan-Cancer data. The novelty of HotNet 2 lies in: (1) use of non-unit amount of heat placed in source node, introducing a heat vector that corresponds to mutation frequency of a gene; (2) identification of strongly connected components in the influence graph as the means of constructing significantly mutated subnetworks; (3)
automation of the selection of parameters corresponding to the amount of heat retained in each node during the diffusion process ($\gamma$) and the threshold value for the influence graph ($\delta$). $\gamma$ is chosen as the point at which further decrease in its value (from 1 to 0) would lead to decrease in the amount of heat present in the neighbors of the source. $\delta$ is chosen as the minimum threshold value such that the strongly connected components of the influence graph will have a given maximum size; chosen by testing on 100 permuted versions of the interaction network.

TieDIE [105] was used to implicate cancer genes in TCGA breast cancer data [100]. It identifies parsimonious subnetworks that link a set of sources (genes involved with somatic mutations) to a set of targets (genes involved in transcriptional effects) in interaction network, using “tied diffusion” process. The adjacency matrix has values $A_{i,j} = 1$ if node $i$ activates node $j$, and $A_{i,j} = -1$ if node $i$ deactivates or represses node $j$. Additionally, node scores are used that represent activity of genes involved with the disease being studied (calculated in a number of ways) and are given values between -1 and 1. The scores are then diffused throughout the network by running the network diffusion process twice – once on the adjacency matrix of the network with the given source set, and then on the transposed adjacency matrix with the target set used as the source. The final diffusion score for each intermediate node is chosen as the minimum of the two computed scores, after which thresholding is performed to remove all nodes with score below a certain value $\alpha$. The remaining connected structure represents a pathway leading from the source gene set to the target set, composed of genes with high diffusion score.

2.4 Cancer Pathway Detection via Patterns of Mutual Exclusivity

Noticing that certain genes exhibit patterns of mutual exclusivity in tumours [101][25], such that in most samples only one of the genes is altered, has given rise to a set of methods created to exploit the patterns of mutual exclusivity in order to constrain the search space. Since mutual exclusivity is defined on sets of genes and not individual genes, these methods naturally focus on identification of cancer pathways. Primary aim of these methods is to potentially discover mutual exclusivity patterns among implicated genes that were previously unknown.

MEMo [25] is a heuristic method for discovery of gene sets (which the authors name “driver networks”) whose gene members are frequently altered, exhibit patterns of mutational exclusivity and are likely to participate in the same biological pathway based on previous knowledge. It first constructs a binary event matrix indicating whether a gene is significantly mutated (determined by a statistical test based on background mutation rate) or affected by a copy number alteration, in each sample. Next, pairs of genes that are possibly involved in same pathways are identified by looking at whether they are di-
Figure 2.3: A set of mutually-exclusive genes in TCGA glioblastoma derived by Multi-Dendrix. Edges show known gene interactions among the genes. Orange cells in the mutation matrix mark co-occurrence, whereas blue ones mark mutual exclusivity. (Figure taken from [80])

rect interaction partners, or by looking at the number of common neighbours that they share in the interaction network. Then, a graph is built upon these pairs of genes. In this graph, maximal cliques are extracted (using a heuristic method) and any gene whose number of unique alterations is smaller than number of alterations shared with other genes in the clique is deemed non-informative and removed. Once the cliques no longer contain any non-informative genes, they are tested for significance of their mutual exclusivity by a permutation test – the binary event matrix has its non-zero values permuted (by selecting two gene-sample pairs and exchanging their row or column indices if they are not already present in the matrix), in a series of experiments, from which a p-value is extracted that measures the significance of the clique in non-permuted data. If a clique does not pass the test, its least-informative gene is removed and the procedure is repeated, so as not to restrict the attention only to maximal cliques and ignore their subsets.

Multi-Dendrix [80] simultaneously identifies multiple pathways whose genes exhibit patterns of mutational exclusivity, and does so without the use of information about gene interactions. The authors’ motivation for doing so is that using known gene interaction information would limit the search space based on still incomplete knowledge about interactions and pathways. Their integer linear programming formulation identifies a collection of gene sets with the maximum sum of weights, using parameters for the number of sets and the minimum and the maximum size of each set. Weight of a set is defined as the difference between the number of samples in which there is an alteration in one gene per set, and the number of patients with an alteration in more than one gene per set. To deal with the sensitivity of the resulting sets with respect to the choice of the three parameters in the integer linear program, the method is run with various parameter settings. Each gene pair
is assigned a weight based on fraction of runs (with unique parameter settings) in which they appear in same gene sets. By considering the gene pairs as edges of a graph and thresholding the weights, connected components are obtained from which final sets (modules) are extracted. They use the same permutation test as in MEMo to test the sensitivity of their method to random mutations.

More recently, methodology has shifted from using only patterns of mutual exclusivity to also using patterns of co-occurrence, in order to capture a more complete image.

SELECT [95] simultaneously considers co-occurrence and mutual-exclusivity of gene pairs affected by copy-number alterations and/or single nucleotide variations without the use of any prior biological knowledge (i.e. signalling interaction networks). It assigns a score to each pair of genes, based on significance of co-occurrence and mutual-exclusivity patterns in the cohort of tumours (using a permutation test). In order to discover higher order structures of evolutionary inter-dependency (in either the co-occurent or mutually exclusive sense), the significant pairs are then combined to form network modules by using seed-and-extend strategy around each functional event (considering “edges” represented by the scored pairs and greedily picking up to 10 nodes connected to the seed by the highest sum of edge scores), clustering the modules via hierarchical clustering and producing overlap modules from each cluster. The resulting modules are then again tested for statistical significance by 100 permutation tests of edge-scores.

2.5 Cancer Pathway Detection via Patient Coverage

The next set of methods has been developed with the aim to detect cancer pathways in form of connected subnetworks of an interaction network, such that they “cover” the maximum possible number of patients – meaning that they have at least one gene that is altered, in every patient. Contrary to the set of methods from the previous section, they do not focus entirely on patterns of mutual exclusivity, although some of them do incorporate such patterns as part of their framework.

nCOP [60] identifies a small subnetwork of a given interaction network that maximizes the number of patients which have at least one of the subnetwork’s genes altered (the importance of the small size of the subnetwork and the number of patients that are covered is balanced through a parameter in the objective function). They give an ILP formulation which solves the problem exactly, as well as a fast heuristic solution. nCOP was tested on TCGA pan-cancer data (24 different cancer types) with over 6000 tumor samples in total, identifying subnetworks enriched in CGC (cancer gene census) genes.

Similarly, Bomersbach et al. [13] give an ILP formulation that identifies a subnetwork of an interaction network of a user-given size such that it covers the maximum number of patients. They solve the formulation via branch and cut approach and apply it to TCGA pan cancer data, showing improvements in running time against other methods.
Figure 2.4: **Overview of the three settings with which BeWith operates.** (A) The goal of BeME-WithFun is to discover modules which have dense functional interactions within the modules while having mutually exclusive mutations with genes in other modules. (B) BeME-WithCo aims to identify modules which have co-occurring mutations within the modules while having mutual exclusivity between the modules. (C) BeCo-WithMEFun looks for modules of functional and mutual exclusivity relations inside a module with co-occurring mutations between modules. (Figure taken from [34])

In Module Cover [72], the aim is to identify a set of pathways (or modules) of genes, rather than just a single pathway. The genes within a pathway are chosen so that they are close in the interaction network and share similarities with respect to expression quantitative trait loci (eQTL). The objective is to select pathways so that each patient is covered by at least a user-defined number of them. They solve the problem using a greedy heuristic algorithm which adds genes to existing pathways or creates new pathways by balancing between the cost of expanding pathways and adding new ones.

MEMCover [71] uses a similar approach as in Module Cover, but incorporates patterns of mutual exclusivity rather than eQTL scores. For each pair of genes, a “within”, “between” and “across” tissue mutual exclusivity p-value is calculated via permutation testing. They use the “across” p-values as confidence scores for edges in a given interaction network, solving a similar objective function as in Module Cover. The method is applied to TCGA pan cancer data to identify pathways that exhibit patterns of mutual exclusivity and detect mutual exclusivity “hubs” that suggest possible drivers with significant growth advantage.

BeWith [34] combines information about co-occurrence and mutual exclusivity among pairs of genes (without the use of interaction networks) with functional interactions in a combinatorial optimisation framework, in order to compute sets of pathways that exhibit different relationship settings among themselves, as well as among the genes inside each pathway. They focused on three relationship settings: 1) genes belonging to different pathways exhibiting mutual exclusivity patterns, and genes within the same pathway being functional interaction partners; 2) genes belonging to different pathways exhibiting mutual exclusivity patterns, and genes within the same pathway exhibiting co-occurrence patterns; and 3) genes belonging to different pathways exhibiting co-occurrence patterns, and genes within the same pathway being mutually exclusive functional interaction partners. Figure
2.4 depicts these settings. The objective function for all three settings is:

$$Max \sum_{ij} \text{between}(i, j)z_{ij} + \sum_{ij} \sum_{k=1}^{K} \text{within}(i, j)x_{ijk}$$

where $z_{ij}$ indicates an edge between pathways, $x_{ij}$ indicates an edge within a pathway, $K$ represents the maximum number of pathways to detect, and the scores between and within depend on the particular setting. The three settings are then tested independently to compute different kinds of pathways.
Chapter 3

Patient-Specific Multi-Driver Gene Prioritization for Precision Oncology

Genomic and transcriptomic alterations are the major contributors of tumorigenesis and progression of cancer. Over the past decade, high-throughput sequencing efforts have provided an unprecedented opportunity to identify such genomic alterations that can lead to changes in gene regulation, protein structure, and function [125]. Genomic and transcriptomic data provide unique and complementary information about a particular tumor, but the translation of “big” molecular data into insightful and impactful patient outcomes is extraordinarily challenging [150]. During tumor progression, cancer cells accumulate a multitude of genomic alterations; however most are inconsequential “passenger” alterations that are effectively neutral. Nevertheless, a small fraction provide mission-critical “hallmark” functions and are known as “driver” alterations that modify transcriptional programs and therefore drive and sustain tumor progression [125, 150, 54]. Improving our knowledge on driver alterations, possibly through an integrative analysis of various omics data is critical to better understand cancer mechanisms and select appropriate therapies for specific cancer patients.

There are several computational methods for identifying cancer drivers. However, many of them rely on the recurrence frequency of single nucleotide variants (SNVs) with respect to the background mutation rate [53, 158, 79, 75]. As a result, these methods are restricted to identifying only highly recurrent mutations as driver events. Recent studies have implicated novel drivers that affect only a small subset of cancer patients. Notable examples include SPOP mutations and CHD1 deletions which are present in less than 20% of prostate cancer patients [8, 50]. Whereas recurrent drivers are hypothesized to initiate carcinogenesis and are therefore present in the majority of tumor cells, rare drivers can arise during tumor evolution and be isolated to a smaller fraction of cells due to clonal expansion [51, 39].
These rare driver genes may be functionally important but are likely to be missed by a frequency-based approach.

Perhaps the first computational method to consider large scale genomic alterations as driver events is CONEXIC [3], which correlates genes with highly recurrent copy number alterations (CNAs) with variation in gene expression profiles within a Bayesian network. Similarly, with no prior knowledge of pathways or protein interactions, MOCA correlates gene mutation information with expression profile changes in other genes [90]. Suo et al [128] also prioritize highly mutated genes that interact with large number of differentially expressed genes in a gene network. Another approach, (Multi) Dendrix [80] aims to simultaneously identify multiple driver pathways, assuming mutual exclusivity of mutated genes among patients, using either a Markov chain Monte Carlo algorithm or integer linear programming (ILP). XSEQ [38] uses probabilistic model to compute influence of mutated genes over expression profile changes in other genes by considering direct gene interactions. Suo et al [128] also prioritize highly mutated genes that interact with large number of differentially expressed genes in a gene network. Another approach, (Multi) Dendrix [80] aims to simultaneously identify multiple driver pathways, assuming mutual exclusivity of mutated genes among patients, using either a Markov chain Monte Carlo algorithm or integer linear programming (ILP). XSEQ [38] uses probabilistic model to compute influence of mutated genes over expression profile changes in other genes by considering direct gene interactions. Finally, MEMo [26], identifies sets of proximally-located genes from interaction networks, which are also recurrently altered and exhibit patterns of mutual exclusivity across the patient population.

Simultaneously with the above methods, several approaches were developed outside of cancer research to correlate the presence of causal genes with gene expression. For example, Tu et al [140] used a random walk approach on a molecular interaction network to associate causal genes and pathways. Similarly, ResponseNet [153, 78] relate genetic perturbations to transcriptomic response in yeast model thereby identifying sub-network of regulators mediating the interactions. ResponseNet formulates a minimum-cost flow optimization problem that aims to maximize the flow between the source and target while minimizing the cost of the connecting paths. Similarly, eQTL electrical diagrams (eQED) [129] integrates expression quantitative trait loci (eQTL) analysis with molecular interaction network using the circuit network model. To the best of our knowledge, NetQTL [73] is the first method to link CNAs to expression profile changes within an interaction network and connects specific “causal” aberrant genes with potential targets in the interaction network. EPoC [66] links CNAs to expression changes in an interaction network assuming steady-state perturbation effects. Similarly, PARADIGM [147], computes gene-specific inferences using factor graphs to integrate different genomic changes and infer pathways altered in a patient. MAX-DRIVER [23] uses maximum information flow to identify potential causal genes (CNAs) in an interaction network.

More recently, HotNet [144], was the first tool to use a network diffusion approach to compute a pairwise influence measure between the genes in the (protein interaction) network and identify sub-networks enriched for mutations in cancer. TieDIE [105] also uses the diffusion model to identify a collection of pathways and sub-networks that associate a fixed set of driver genes to expression profile changes in other genes. Briefly, the network diffusion approach aims to measure the influence of one node over another by calculating
the stationary proportion of a “flow” originating from the starting node that ends up in the destination node. Since this is based on the stationary distribution, the inferences that can be made by the diffusion model are time independent. In that sense, the diffusion approach is very similar to Rooted PageRank, the stationary probability of a random walk originating at a source node, being at a given destination node. Shi et al. [121] also prioritizes genes based on diffusion score matrix (derived from a tripartite graph of mutations, outliers and patients) rank aggregation. A final method, DriverNet [10], also aims to correlate genomic alterations with target genes’ expression profile changes, but only among direct interaction partners; the novel feature of DriverNet is that it aims to find the minimum number of potential drivers that can “cover” targets.

Among the above strategies, the ones based on mutual exclusivity still focus on frequent events. The others, based on “information flow” in gene/protein interaction networks, do not aim to discover cancer drivers, but rather are designed to identify dysregulated sub-networks or modules. In addition, the notion of influence they employ is based on stationary distribution of “information” originating at a particular gene/protein. As a result, none of the available methods aim to identify rare, patient-specific driver events, based on a time dependent notion of influence. Finally, none of the available techniques aim to simultaneously consider different types of genomic alterations as potential drivers.

To address the above challenges, this Chapter presents a novel combinatorial method, HIT’nDRIVE (a preliminary version of which was presented at the Research in Computational Molecular Biology (RECOMB) conference [123]). HIT’nDRIVE jointly analyzes genome and transcriptome data for identifying and prioritizing sequence-altered genes as potential cancer drivers. For that HIT’nDRIVE integrates patient-specific genomic alterations with the associated transcriptome profile, identifying driver genes that dysregulate large portion of each patient’s transcriptome. Drawing upon the domain knowledge of molecular interactions presented as a gene/protein interaction network, HIT’nDRIVE uses network topology to derive the influence of one (sequence-altered) gene over another (expression-outlier) gene and aims to identify the most parsimonious set of patient-specific driver genes that have sufficient “influence” over a large proportion of the expression-outliers.

3.1 Methods
3.1.1 Overview

The primary goal of HIT’nDRIVE is to link alterations at the genomic level to changes at the transcriptome level through a gene/protein interaction network. Intuitively, it aims to find the smallest set of altered genes that can explain most of the observed transcriptional changes in a given cancer patient cohort. In other words, HIT’nDRIVE aims to identify the minimum number of potential driver genes which can cause a user-defined proportion of the downstream expression effects observed. HIT’nDRIVE formulates this as a Random
Walk Facility Location (RWFL) problem, a combinatorial optimization problem introduced in this Chapter. Random Walk Facility Location Problem generalizes the classic Facility Location (FL) problem by changing the notion of distance it uses. As a reminder, in the classic Facility Location problem, the goal is to find a set of facilities (a subset of nodes) in a given graph, such that the maximum distance from each of the rest of the nodes towards their nearest selected facility is minimized. Given a gene interaction network, the Facility Location problem defines the distance between a potential driver gene and an outlier gene as the length of the shortest path of gene interactions between them. The Random Walk Facility Location problem, in contrast, uses “hitting time” (or “first passage time”) \[29, 84\], the expected length of a random walk between the two nodes, as their distance instead of the length of the shortest path. Under the use of hitting time, the FL problem completely changes nature: in the classical FL formulation the goal is to associate each outlier gene in the network with exactly one (the closest) driver gene. In the RWFL formulation, each outlier gene is associated with multiple driver genes through a collective distance to the outlier (which will no longer be the shortest pairwise distance), forming a many-to-many relation.

As per the standard facility location problem, RWFL is NP-hard, even to approximate. As a result, we reduce it to the weighted multi-set cover (WMSC) problem, for which we give an ILP formulation. Intuitively, in this new formulation, HIT’nDRIVE associates the genomic alterations with transcriptomic changes in the form of a bipartite graph with nodes on one partition representing the set of aberrant genes and nodes on the other partition representing the set of expression-altered genes, and each edge has an influence value equal to the inverse pairwise hitting time between the two nodes it connects (Figure 3.1A). The WMSC problem on this representation of data asks to find the smallest subset of aberrant genes (as potential drivers) whose total influence (sum of pairwise influence values) over a user defined fraction of expression-altered genes (for each patient) is sufficiently high. In order to quantitatively assess the genes identified by HIT’nDRIVE, we extended our lab’s previously developed algorithm, OptDis [35], for de novo identification of modules of small size inside the interaction network which are seeded by at least one predicted driver. The modules are chosen so that their discriminative power (for phenotype classification) is the greatest among connected sub-networks of similar size that contain the individual predicted driver genes (Figure 3.1B–C). We report the classification accuracy based on the identified driver-seeded modules as means of quantitative validation of our results (in the absence of ground truth). We also look at the genes that build the chosen modules (of high classification accuracy) in attempt to identify cancer-related pathways.

3.1.2 Gene Influence and (Multi-)Hitting Time

HIT’nDRIVE naturally integrates genome and transcriptome data from a number of tumor samples for identifying and prioritizing sequence-wise altered genes as potential drivers.
Figure 3.1: Overview of the HIT’nDRIVE algorithmic framework. (A) HIT’nDRIVE integrates sequence-wise changes in genome with expression changes in transcriptome obtained from patients’ tumor samples. The influence values derived from the protein interaction network indicate how likely a driver gene influences its downstream target genes in the network. (B) The predicted driver genes are used as seeds to discover modules of genes that discriminate between the sample phenotypes using OptDis. (C) Based on this the driver modules are ranked and thus prioritized.

It “links” sequence-wise altered genes to genes with expression changes through a gene or protein interaction network. For that, it aims to find the smallest set of sequence-wise altered genes that can “explain” most of the observed gene expression alterations in the cohort. In other words, HIT’nDRIVE identifies the minimum number of potential driver genes which can “cause” a user-defined proportion of the downstream expression effects observed.

In order to measure one gene’s ability to cause changes in the transcript of another, HIT’nDRIVE uses a particular “influence” value of a potential driver gene on other (possibly distant) genes based on the (gene or protein) interaction network in use. In order
to capture the uncertainty of interactions of genes with their neighbours, it considers a random walk process which propagates the effect of sequence alteration in one gene to the remainder of the genes through the network. Thus, the ability of one gene to influence a change in another is a function of both distance between them and the network topology. Specifically, HIT’nDRIVE models the notion of influence through the hitting time, which is the expected length (number of hops) of a random walk which starts at a given potential driver gene, and “hits” a given target gene the first time in a (protein or gene) interaction network (smaller hitting time corresponds to a greater influence). More specifically, for any two nodes \( u, v \in V \) of an undirected, connected graph \( G = (V, E) \), let the random variable \( \tau_{u,v} \) denote the number of hops in a random walk starting from \( u \) and visiting \( v \) for the first time. Then the hitting-time \( H_{u,v} \) is defined as \( H_{u,v} = E[\tau_{u,v}] \) [83].

In order to capture synthetic lethality like scenarios, HIT’nDRIVE considers multiple sequence-wise altered genes as potential drivers and thus generalises the notion of hitting-time to multiple sources. Given a set of multiple genes, we measure their collective influence over a given target gene through multi(source)-hitting time, the expectation of the smallest number of hops in one of the random walk processes, simultaneously starting at each one of the potential driver genes and ending at a given expression-wise altered gene for the first time. More specifically, let \( U \subseteq V \) be a subset of nodes of \( G \) and \( v \in (V - \{U\}) \) be a single node. We thus define the multi(source)-hitting time \( H_{U,v} \) as \( H_{U,v} = E[min_{u \in U} \tau_{u,v}] \).

### 3.1.3 Calculating Hitting Time on an Interaction Network

To calculate exact values of pair-wise hitting times, \( H_{u,v} \), for all pairs of nodes in the network, we use the matrix inversion method as explained by Tetali et al. [134]. Here we will briefly describe the method. For proofs please refer to [134].

Let \( P \) denote the transition probability matrix (of size \( n \times n \)) of the interaction network with \( n \) nodes, and \( H \) its hitting-time matrix. \( P_{i,j} \) represents the probability that random walk picks node \( u_j \) as its next step from node \( u_i \), and \( H_{i,j} \) represents the hitting-time \( H_{u_i,u_j} \). We assume that \( P_{i,i} = 0 \), for all \( i \in \{1, ..., n\} \), which forces the random walk to move from current node to one of its neighbours in every step. Lastly, let \( \pi \) be the stationary distribution of the network, where \( \pi_i \) represents the proportion of time that an infinite-length random walk in the network spends visiting node \( u_i \).

Given \( P, H, \) and \( \pi \), we define \( (n - 1) \times (n - 1) \) matrices \( \tilde{P} \) and \( \tilde{H} \) as follows: \( \tilde{P}_{i,j} = \pi_i, \tilde{P}_{i,j} = -\pi_i P_{i,j} \) and \( \tilde{H}_{i,i} = H_{i,n} + H_{n,i}, \tilde{H}_{i,j} = H_{i,n} + H_{n,j} - H_{i,j} \), for all \( i, j \in \{1, ..., n - 1\} \) such that \( i \neq j \). We show how to calculate hitting-times based on the following claim:

**Claim 1.** Given \( P, H, \tilde{P} \) and \( \tilde{H} \) as defined above, \( \tilde{P} \tilde{H} = I_{n-1} \).

For proofs please refer to [134].

Note that \( \tilde{P} \) can be computed directly from the transition probability matrix \( P \) (following its definition) and we obtain \( \tilde{H} \) by inverting \( \tilde{P} \). Using the definition of \( \tilde{H} \) and proof of
Claim 1 (see Theorem 2.2 in [134]), we obtain following formulae:

\[ H_{i,n} = \sum_k N_i^k \]

\[ H_{n,i} = \bar{H}_{i,i} - H_{i,n} \]

\[ H_{i,j} = H_{i,n} + H_{n,j} - \bar{H}_{i,j} \]

Standard \( O(n^3) \) matrix inversion method based on Gaussian elimination finishes this pre-processing step of calculating the exact hitting-times in a few hours for the interaction networks we analyzed (of around \( 10000 \) nodes).

### 3.1.4 Estimating Multi-Source Hitting Time via Single-Source Hitting Times

Calculating exact multi-hitting time from a set of nodes \( U \) to a given target \( v \) is a hard problem even when the set \( U \) is fixed, and only gets more complicated in an optimisation scenario where we are looking to find an “optimal” set \( U \). Therefore, HIT’nDRIVE estimates the multi-hitting time \( H_{U,v} \) between a set of nodes \( U = \{u_1, u_2, \ldots, u_k\} \) and a single node \( v \), as a function of independent hitting times \( H_{u,v} \) for all \( u \in U \). We use the following linear estimate:

\[ H_{U,v} \approx \frac{1}{\sum_{i=1}^k \frac{1}{H_{u_i,v}}} \]

Since we do not have a strong bound for this estimate’s accuracy, we attempted to measure the quality of the estimate by comparing it to experimentally-computed multi-source hitting time values of a randomly chosen set \( U \) via simulations. Obviously, computing the exact multi-hitting time over all possible sets of facilities in the network is computationally not feasible. Therefore, we performed 1000 iterations of the following experiment on the STRING v10 interaction network:

- Choose 10 nodes of the network uniformly at random, to be a set of facilities \( U \).
- For each node \( v \notin U \), estimate multi-source hitting time by performing 5000 random walk simulations from every \( u \in U \) and in each simulation, measure the minimum time required for the first one of them to reach \( v \). Take the average of the observations and call it the “exact” multi-hitting time \( MHT_{U,v} \).
- Compute the relative error of the estimate \( H_{U,v} \) (estimated based on pair-wise hitting times) compared to \( MHT_{U,v} \).
- Take the average relative error over all nodes \( v \notin U \).

With 5000 random walk simulations, we hoped to get accurate enough estimate to be able to compute the relative error accurately without it taking too long. Over the 1000 iterations,
we observed average error rate of 5.96%, with average error over non-facility nodes per iteration ranging from 3.03% to 8.06%. This suggests that our estimate (which has the practical benefit of being linear and thus useful in a linear programming setting) is quite accurate in practice.

Furthermore, we would like to point out that the most extreme cases of our approximation being inaccurate are where one of the $u \in U$ is an immediate neighbour of $v$ of degree one, so that $H_{u,v} = 1$ and the true multi-hitting time of the entire set $U$ should be 1. It is easy to see that if $v$ has $k$ such neighbours then the estimate will be approximately $\frac{1}{k}$, when the true multi-hitting time is 1. We have analysed the candidate driver-outlier pairs in our model (in the datasets that we analyzed with HIT’nDRIVE) that have hitting time 1 and obtained the following stats: (a) TCGA-BRCA – 11 pairs with hitting time 1, where maximum number of candidate drivers connected to a single outlier with hitting time 1 is 1; (b) TCGA-GBM – 0 pairs with hitting time 1; (c) TCGA-OV – 12 pairs with hitting time 1, where maximum number of candidate drivers connected to a single outlier with hitting time 1 is 1; (d) TCGA-PRAD – 1 pair with hitting time 1. Since average non-zero hitting time in the STRING v10 hitting time matrix is 129322, and selected number of drivers in our results is in order of tens (meaning that in most cases the estimate of $H_{U,v}$ will be inverse of the sum of number 1 and inverses of much larger numbers), the estimate in these extreme cases will be close to 1 which represents the exact solution. Combined with the the result of the previously-explained randomized test that estimated average error to be 5.96%, it serves as further evidence that our approximation is quite accurate in practice.

### 3.1.5 Random Walk Facility Location problem

HIT’nDRIVE links alterations at the genomic level to changes at the transcriptome level through a gene/protein interaction network. More specifically, HIT’nDRIVE identifies the minimum number of potential driver genes which can cause a user-defined proportion of the outlier downstream expression effects observed. We formulate this as a “Random Walk Facility Location” (RWFL) problem, a new combinatorial optimization problem which generalizes the classical “Facility Location” (FL) problem by the use of a novel distance measure. Given a network, FL problem defines the distance between a potential driver gene and an outlier gene as the length of the shortest path between them. The RWFL problem, in contrast, uses “multi hitting time”, the expected length of the shortest random walk originating from any of the sequence altered genes, and ending at an outlier. Specifically, the Random Walk Facility Location problem asks to find the smallest set of sequence-altered genes from which one can reach outliers in the network within a user defined multi-hitting time. Formally, let $\mathcal{X}$ be a set of potential driver genes and $\mathcal{Y}$ be a set of expression altered (outlier) genes. Then, for a user defined $k$ in a single patient, the RWFL problem asks to find
$k$ potential driver genes as solution to the following optimisation problem:

$$\arg \min_{X \subseteq \mathcal{X}, |X| = k} \max_{y \in \mathcal{Y}} H_{X,y}$$

where $H_{X,y}$ denotes the multi-hitting time from the gene set $X$ to the gene $y$. Alternatively, for a user defined $\Delta$, one might seek to find the minimum-size subset of potential driver genes from which multi-hitting time towards each outlier is at most $\Delta$:

$$\arg \min_{X \subseteq \mathcal{X}} |X| \quad \text{s.t.} \quad \forall y \in \mathcal{Y} : H_{X,y} \leq \Delta$$

Since RWFL is a computationally hard problem, and cannot be solved in a reasonable amount of time in its original formulation, we reduce the RWFL problem to the **Weighted Multi-Set Cover** (WMSC) problem and give an ILP formulation to solve it. The reduction is achieved through the estimate of the multi-hitting time, described and discussed in the previous section, as a function of independent hitting times of the drivers to an outlier, which provides an upper bound on the multi-hitting time. As a reminder, the estimate of the multi-hitting time $H_{U,v}$ from a set of genes $U = \{u_1, u_2, \ldots, u_k\}$ to a given target gene $v$ is:

$$H_{U,v} \approx \frac{1}{\sum_{i=1}^{k} \frac{1}{H_{u_i,v}}}$$

Given the estimate, constraining $H_{U,v}$ by $\Delta$ is equal to constraining $\frac{1}{H_{u_i,v}}$ by $\frac{1}{\Delta}$:

$$\frac{1}{\sum_{i=1}^{k} \frac{1}{H_{u_i,v}}} \leq \Delta \iff \sum_{i=1}^{k} \frac{1}{H_{u_i,v}} \geq \frac{1}{\Delta}$$

Thus, the constraint becomes linearized and solvable through an integer linear program (ILP).

In this new form, HIT’nDRIVE associates the genomic alterations with transcriptomic changes in the form of a bipartite graph $G_{bip}(\mathcal{X}, \mathcal{Y}, \mathcal{E})$ where $\mathcal{X}$ is the set of aberrant genes, $\mathcal{Y}$ is the set of patient-specific expression-altered genes, and $\mathcal{E}$ is the set of edges (Figure 3.1A). If gene $x_i$ is mutated in a patient $p$, we set edges between $x_i$ and all of the expression altered genes in the same patient $(y_j, p)$, where the edges are weighted by the inverse pairwise hitting times $w_{ij} := H_{x_i,y_j}^{-1}$. The WMSC problem on this representation of data asks to find the smallest subset of $\mathcal{X}$ (as potential drivers) whose total influence (sum of pairwise influence values) over a user defined fraction of expression-altered genes (for each patient) is sufficiently high:

$$\arg \min_{X \subseteq \mathcal{X}} |X| \quad \text{s.t.} \quad \forall y \in \mathcal{Y} : \sum_{x \in X} w_{x,y} \geq \gamma_y \quad \text{for some } Y \subseteq \mathcal{Y}, |Y| \geq \alpha |\mathcal{Y}|$$
The resulting WMSC problem can then be formulated as the ILP shown in Algorithm 1, which is efficiently solvable by ILP solvers such as Gurobi or IBM CPLEX for all data sets we considered.

The above ILP formulation for the WMSC problem introduces binary variables $x_i, y_j, e_{ij}$, respectively, for each potential driver, expression-alteration event, and edge in the bipartite graph. The objective of the ILP is to minimize the number of drivers (i.e. the sum of $x_i$ values) subject to four constraints. Constraint (1) ensures that a selected driver contributes to the coverage of each of the expression alteration events it is connected to (in each patient, if multiple patients are available). Constraint (2) ensures that selected (patient-specific) driver genes contribute enough to cover at least a ($\gamma$) fraction of the sum of all incoming edge weights to each expression alteration event. This constraint corresponds to setting an upper bound on our estimate on the inverse of multi-hitting time of the selected (patient specific) drivers on an expression alteration event. Constraint (3) ensures that, for each patient, the top $\beta$ fraction of the expression altered genes with highest weights ($\lambda_j$) are always covered. Finally, constraint (4) ensures that the selected driver genes collectively cover at least an $\alpha$ fraction of the set of expression alteration events. The outlier weights ($\lambda_j$) represent importance of an outlier (greater value indicating greater importance), and play a role both in the significance of covering the outlier (constraint (3)) as well as scaling the maximum multi-hitting time distance (constraint (2)). In our experiments, we set $\lambda_j$ to a function of the z-score of outlier $y_j$ (see Section 3.1.8).

As indicated above, our ILP formulation for WMSC problem can be generalized to multiple patients with the objective of minimizing the total number of driver genes across all patients, subject to the constraint that a user-defined proportion of outlier genes in each of the patients are covered by the subset of driver genes present in that patient. The resulting ILP model has $|\mathcal{X}| + |\mathcal{Y}|$ binary variables (variables $e_{i,j}$ can be omitted and replaced by $x_i$ in constraint (2)) and $O(|\mathcal{Y}|)$ constraints.
3.1.6 Driver Modules and Phenotype Classification

In order to quantitatively assess the genes identified by HIT’nDRIVE, we extended our lab’s previously developed algorithm, OptDis [35], for de novo identification of modules of small size inside the interaction network which contain (i.e. are seeded by) at least one predicted driver gene. The modules are chosen so that their discriminative power (for phenotype classification) is the greatest among connected sub-networks of similar size that contain the individual predicted driver genes. In general, OptDis performs supervised dimensionality reduction on the set of connected sub-networks. It projects the high dimensional space of all connected sub-networks to a user-specified lower dimensional space of sub-networks such that, in the new space, the samples belonging to the same class are closer and the samples from different class are more distant to each other (i.e. minimize in-class distance and maximize out-class distance) with respect to a normalized distance measure (typically $L_1$). Then we use module features (average expression of genes in the module) for phenotype classification. Using such module features, we hope that the classifier in use does not overfit on rare driver genes and is able to generalize the signal coming from rare drivers to new patients. We report the classification accuracy based on the identified driver-seeded modules as means of quantitative validation of our results (in the absence of ground truth). We also look at the genes that build the chosen modules (of high classification accuracy) in attempt to identify cancer-related pathways. Additionally, we use the classification accuracy of driver modules for evaluation of significance of our prioritised drivers compared with drivers selected via different approaches.

3.1.7 Derivation of Expression-Outlier Genes

We used generalized extreme studentized deviate (GESD) test [117] to obtain the outlier genes. Unlike Grubbs test and the Tietjen-Moore test, GESD test only requires that an upper bound for the suspected number of outliers be specified. Given the upper bound, $r$, the GESD test essentially performs $r$ separate tests: a test for one outlier, a test for two outliers, and so on up to $r$ outliers.

Hypothesis: The GESD test is defined for the hypothesis:

- $H_0$: There are no outliers in the data set
- $H_a$: There are up to $r$ outliers in the data set

Test statistic: Compute

$$ R_i = \frac{\max_i |x_i - \mu|}{\sigma} $$

with $\mu$ and $\sigma$ denoting the sample mean and sample standard deviation, respectively. Remove the observation that maximizes $|x_i - \mu|$ and then recompute the above statistic with $n - 1$ observations. Repeat this process until $r$ observations have been removed. This results in the $r$ test statistics $R_1, R_2, ..., R_r$. 31
Critical region: Corresponding to the \( r \) test statistics, compute the following \( r \) critical values

\[
\lambda_i = \frac{(n - i)t_{n-i-1,p}}{\sqrt{(n - i - 1 + t^2_{n-i-1,p})(n - i + 1)}}
\]

where \( i = 1, 2, ..., r \), and \( t_{p,v} \) is the \( 100p \) percentage point from the \( t \) distribution with \( v \) degrees of freedom.

\[
p = 1 - \frac{\alpha}{2(n - i + 1)}
\]

here, \( \alpha \) denotes the significance level.

The number of outliers is determined by finding the largest \( i \) such that \( R_i > \lambda_i \).

3.1.8 Derivation of Expression-Outlier Gene Weights

Outlier-gene weights were calculated as follows. Let \( i \) denote genes, \( j \) denote patients and \( x_{ij} \) denote the gene-expression value of gene \( i \) in patient \( j \). We then calculated the absolute value of z-score \( (z_{ij}) \).

\[
z_{ij} = \frac{|x_{ij} - \mu_i|}{\sigma_i}
\]

where \( \mu_i \) and \( \sigma_i \) respectively denotes mean and standard deviation of expression value of gene \( i \). Next we performed Student’s t-test in the gene-expression values of normal and tumor phenotypes, where \( \psi_i = -\log(p_{\text{value}_\text{ttest}}) \). Finally, we calculate the outlier weight \( \omega_{ij} \) as

\[
\omega_{ij} = \frac{\psi_i z_{ij}}{\sum_i \psi_i z_{ij}}
\]

3.1.9 Association of Driver Modules With Patients’ Survival Outcome

To test for association of driver modules with patients’ survival outcome, we developed a risk-score based on multi-gene (component genes of the module) expression. The risk-score \( (S) \) defined as a weighted sum of the normalized gene-expression values of the component genes in the module weighted by their estimated univariate Cox proportional-hazard regression coefficients [11] as given in the equation below.

\[
S = \sum_i^k \beta_i x_{ij}
\]

Here \( i \) and \( j \) represents a gene and a patient respectively, \( \beta_i \) is the coefficient of cox regression for gene \( i \), \( x_{ij} \) is the normalized gene-expression of gene \( i \) in patient \( j \), and \( k \) is the number of component genes in a gene-module. The normalized gene-expression values were fitted against overall survival time with living status as the censored event using univariate Cox proportional-hazard regression (Exact method).
Based on the risk-score values, patients were stratified into two groups: low-risk group (patients with $S < 33$ percentile of $S$), and high-risk group (patients with $S > 66$ percentile of $S$). Patients that fall in between (i.e. patients with $S >= 33$ percentile of $S$ and <= 66 percentile of $S$) were discarded from the further analysis as these patients fall into intermediate-risk group and are bound to introduce noise while performing log-rank test.

Both Cox regression coefficients of each gene and risk-score cutoff values for each module were estimated from TCGA-BRCA cohort (training dataset), later these values were applied to METABRIC cohorts (test dataset). To assess whether the risk-score assignment to high/low categories was valid, a log-rank test was performed for each module in both training and test datasets. Finally, to identify the significant list of driver-modules that were robust enough to predict patients’ survival, we calculated log-rank test p-value, hazard-ratio (HR) (Wald test) and concordance-index (c-index) (Wald test).

3.1.10 Statistical Significance of the Overlap With the CGC Database

Suppose, for a cohort of cancer patients, that we predict $n_{total}$ number of driver genes using HIT’nDRIVE, out of which $n_{cgc}$ of driver genes are present in the Cancer Gene Census (CGC) database (of known cancer driver genes). Let, $x$ be the total number of sequence altered genes (i.e. all potential driver genes) and let $y$ of these $x$ sequence altered genes be in CGC. This means that the probability that a randomly selected gene out of these sequence altered genes happens to be a CGC gene is $\frac{y}{x}$. Thus, the probability (p-value) that at least $n_{cgc}$ out of $n_{total}$ identified driver genes are in the CGC database is:

$$pvalue = \sum_{i=n_{cgc}}^{n_{total}} \binom{n_{total}}{i} \left( \frac{y}{x} \right)^i \left( 1 - \frac{y}{x} \right)^{n_{total}-i}$$

Next we consider driver genes in each patient. We also calculated the p-value for HIT’nDRIVE picking at least $p$ CGC drivers out of $p'$, and picking at most $q$ non-CGC drivers out of $q'$ as follows:

$$pvalue = \sum_{x=p'}^{x=p'+q'} \binom{p+q}{x} \left( \frac{p}{p+q} \right)^x \left( \frac{q}{p+q} \right)^{p'+q'-x}$$

3.2 Data and Pre-Processing

We used publically available datasets of four major cancer-types: glioblastoma multiforme (GBM), Ovarian serous cystadenocarcinoma (OV), breast adenocarcinoma (BRCA) and prostate adenocarcinoma (PRAD) from The Cancer Genome Atlas (TCGA) project. All data were obtained from TCGA data-portal in May 2014 which were mapped to GRCh37 genome build. Although TCGA has recently made available all data re-aligned to the newer
GRCh38 genome build, to ensure compatibility, all TCGA data we have used in this study has been mapped to GRCh37.

**Somatic mutation.**
Calls (level 2 data) from all available platforms/centres were merged. Only missense, nonsense and splice-site mutations were marked as somatic-mutation alteration events.

**Copy number aberrations (CNAs).**
For GBM and OV, Agilent Human Genome CGH Microarray 244A (level 1) data files were used and for PRAD and BRCA, Affymetrix Genome-Wide Human SNP Array 6.0 (level 3) data files were used to generate the copy number profiles. These Agilent FE format sample files were loaded into BioDiscovery Nexus Copy Number software v7.0, where quality was assessed and data was visualized and analyzed. All samples were mapped to the most recent genome build (hg 19, NCBI build 37) via Agilent probe identifiers and annotation (downloaded from Agilent’s website) based on the 1M SurePrint G3 Human CGH Microarray 1x1M design platform. BioDiscovery’s FASST2 segmentation algorithm, a Hidden Markov Model based approach, was used to make copy number calls. The FASST2 algorithm, unlike other common HMM methods for copy number estimation, does not aim to estimate the copy number state at each probe but uses many states to cover more possibilities, such as mosaic events. These state values are then used to make calls based on a log-ratio threshold. The significance threshold for segmentation was set at \( 5 \times 10^{-6} \) also requiring a minimum of 3 probes per segment and a maximum probe spacing of 1000 between adjacent probes before breaking a segment. The log ratio thresholds for single copy gain and single copy loss were set at 0.2 and -0.23, respectively. The log ratio thresholds for two or more copy gain and homozygous loss were set at 1.14 and -1.1 respectively. Upon loading of raw data files, signal intensities are normalized via division by mean. All samples are corrected for GC wave content using a systematic correction algorithm. Only the high confidence copy number aberrations i.e. high copy number gain or homozygous deletions were marked as copy-number aberrant events. Finally, genes that harbour either a somatic-mutation aberrant event or a copy-number aberrant event were taken to be the final list of aberrant genes at the genomic level.

**Gene expression.**
We used microarray based gene-expression (Affymetrix HT Human Genome U133 Array Plate Set) (level-1) for GBM and OV data sets. In case of BRCA and PRAD data sets, RNA-seq derived gene-expression were used (level-3). Gene expression profiles of normal and tumor phenotype were used as sample groups.
Gene fusions.
Transcript fusions prediction calls for GBM, OV, BRCA and PRAD were obtained from TCGA Fusion gene Data Portal (http://www.tumorfusions.org) [154]. The fusion partner genes were tagged for gene-fusion alteration.

Genomics of drug sensitivity in cancer.
Somatic mutation, copy-number alterations and gene-expression, and drug screening data of cancer cell lines were downloaded from Genomics of Drug Sensitivity in Cancer (GDSC) [61] website http://www.cancerrxgene.org/downloads. Data downloaded on August 2016.

Interaction networks.
We used STRING version 10 [131] protein-interaction network which contains high confidence functional protein-protein interactions (PPI). Self-loops and interactions with missing HGNC symbols were discarded and interaction scores were divided by 1000 to obtain percentage-like reliability score. Only high confidence interactions with combined score of 0.9 or greater were selected. As a result we obtained a network of 10971 nodes with 214298 interactions.

In the case of prostate cancer, we integrated STRING-10 protein-protein interaction network with protein-DNA interaction network derived from Chip-seq experiments for transcription factors highly relevant to prostate cancer - REST, FOXA1, AR, EZH2 [120] and ERG [115] resulting in a new combined network of 13517 nodes and 220190 interactions.

To simulate HIT’nDRIVE using different underlying network we used two additional interaction networks: Human Protein Reference Database - Protein-Protein Interaction Database (HPRD-PPI) network (version 9.0) [70] and REACTOME pathway database (version 2015) [46].

Pathway enrichment analysis.
The selected set of genes were tested for enrichment against gene sets of pathways present in Molecular Signature Database (MSigDB) v5.0 [126]. A hypergeometric test based gene set enrichment analysis was used for this purpose (see Software and Code Availability for details). A cut-off threshold of false discovery rate (FDR) ≤ 0.01 was used to obtain the significantly enriched pathways. Same procedure, as above, is used to assign biological function to the gene-modules.

Derivation of outlier-genes.
We used generalized extreme studentized deviate (GESD) test [117] to obtain the outlier genes (see Section 3.1.7 for details).

Validation datasets.
For the validation of driver-modules via phenotype classification, we used the following

3.3 Results

3.3.1 HIT’nDRIVE-Predicted Candidate Driver Genes in Multi-Omics Cancer Datasets

We applied HIT’nDRIVE to prioritize driver genes in four major cancer types - glioblastoma multiforme (GBM) [135], ovarian serous cystadenocarcinoma (OV) [136], breast adenocarcinoma (BRCA) [137] and prostate adenocarcinoma (PRAD) [138]; obtained from the The Cancer Genome Atlas (TCGA) data portal. Only samples with matched genomic alterations (SNVs and/or CNAs and/or gene fusions) and transcriptomic changes (outlier genes from gene-expression profile) were used in our study. We used the fusion prediction calls as reported in the TCGA Fusion gene Data Portal [154].

In GBM, we obtained 48 unique candidate driver genes altered at varying frequencies across 258 GBM patients. *EGFR* (36%), *TP53* (29.5%), *PTEN* (28%) and *CHEK2* (26%) were the most frequently altered driver genes in GBM; followed by *CDKN2A* (16%), *RB1* (13%) and *SEC61G* (12%). Previous efforts in GBM genome characterization identified amplification in *EGFR*, *PDGFRA*, mutations in *CHEK2*, *TP53*, *PTEN*, *RB1*, *NF1* and deletions in *CDKN2A* to be associated with GBM [103, 135, 148]. HIT’nDRIVE prioritized all of the above alterations. Alterations in *EGFR* are characteristic of the classical GBM subtype, *NF1* with mesenchymal subtype, *PDGFRA* and *IDH1* with pro-neural subtype of GBM [148]. Fifteen out of 48 driver genes predicted by HIT’nDRIVE (pvalue = 8 · 10^{-4}), were present in Cancer Gene Census (CGC) database [49], which contains genes for which mutations have been causally implicated in cancer (Figure 3.2A). *GSTT1* (deleted in 21 patients), a key player in drug metabolism, was neither found in CGC nor in Catalogue of Somatic Mutations in Cancer (COSMIC) [48] databases. Twelve GBM driver genes were found to be actionable targets. Actionable genes were extracted from TARGET database [143], which contains genes directly linked to a clinical action. In addition to the above list, 6 other driver genes were druggable (Figure 3.2B). We extracted the list of druggable genes from Drug-Gene Interaction Database (DGIdb) [55]. Interestingly, around 85% of the patients in GBM cohort harbour at least one actionable driver gene and further 5% of patients have druggable targets (Figure 3.2C). HIT’nDRIVE also identified 12 infrequent driver genes, which we define as genes altered in at most 2% of the cases. Among the infrequent genes, *SACS* is known to be associated with neurological functions, *NLRP3* is involved in apoptosis, and *TIAM2* is involved in invasion and metastasis.
Figure 3.2: **Summary of driver genes prioritized by HIT’nDRIVE.** Top: The distribution of the number of driver genes identified by HIT’nDRIVE in individual cancer cohorts. The leftmost panel shows the distribution of driver genes in a combined cohort of GBM, OV, PRAD and BRCA patients. (A) Distribution of predicted driver genes in cancer genes databases. CGC database contains genes for which mutations have been causally implicated in cancer. Genes curated in CGC database represents likely drivers of cancer. COSMIC is a comprehensive database of somatic mutations that have been reported in different cancers. However, every gene present in COSMIC database may not represent drivers of cancer. (B) Distribution of driver genes in druggable genes databases. Actionable genes in cancer therapy were derived from TARGET database. List of druggable genes were extracted from DGI database. (A-B) The numbers in the panel represent the number of genes in respective categories. (C) Distribution of patient druggability. Patient druggability was accessed using information in TARGET and DGI databases. The numbers in the panel represent the number of patients in respective categories.
The 526 OV patients harboured a total of 85 unique driver alterations. TP53 mutations were prevalent in more than half (58%) of the patients in the cohort. Consistent with the previous findings, we found OV patients to be driven by genomic copy-number changes rather than recurrent point mutations [28, 104]. Recurrent somatic CNAs were observed in GSTT1 (32.3%), WWOX (28.1%), FAM49B (15.0%), UGT2B17 (14.6%), CCNE1 (13.1%), SLC39A4 (13.1%) and MYC (12.5%). Mutations in TP53, BRCA1/2 and loss of RB1, NF1 and CCNE1 were previously associated with OV [136, 104]. HIT’nDRIVE revealed 18 CGC driver genes (pvalue = 2 \cdot 10^{-5}) (Figure 3.2A), among which 13 genes were actionable targets and other 12 genes were at least druggable (Figure 3.2B). More than 75% of OV patients harboured at least one actionable targets and additional 6% of patients have druggable target (Figure 3.2C). GSTT1 (altered in 170 patients), in OV, is involved in estrogen and drug metabolism. It was neither found in CGC nor in COSMIC databases. We identified 13 infrequent genes, among which MAPK1 is known to play an important role in oncogenic pathways in cancer.

HIT’nDRIVE identified 40 driver genes across 333 PRAD patients. Copy number loss of SPECCIL (23.7%), STEAP1B (13%), WWOX (10%) and amplification of NSD1 (16.2%), SIRPB1 (16.2%) were the most recurrent events in PRAD patients. We also found recurrent somatic mutation in MUC4 (11%), SPOP (10.5%) and TP53 (10%). The most common alterations in PRAD genomes are fusion of androgen-regulated promoters with ERG and other members of ETS family of transcription factors mainly, TMPRSS2-ERG fusions [139]. Since we relied on the gene fusion predictions obtained from TCGA Fusion gene Data Portal [154] which analyzed only 178 (out of 333) patients, we observed ERG gene fusion in only 5.7% cases. The more recent TCGA publication [138] reported ERG fusions in almost half of the patients in the cohort. Moreover, the tools used for gene fusion detection, in the two studies, were different as a result of which we observed much smaller number of ERG fusions than reported previously. SPOP, TP53, FOXA1 and PTEN are the most frequently mutated genes which have been previously associated with prostate cancer [8]. PRAD patients harboured 12 driver genes present in CGC database (pvalue = 9 \cdot 10^{-4}) (Figure 3.2A), out of which 8 driver genes were actionable (Figure 3.2B). Approximately a quarter of PRAD patients could benefit with actionable targeted therapy (Figure 3.2C). Moreover an additional 14% of patients harboured druggable genes which warrants deeper investigation of drug repurposing opportunities. NBPF1 (mutated in 17 patients), is a known tumor suppressor gene known to have neural function and also involved in cell-cycle arrest, was neither found in CGC nor in COSMIC databases. We identified 11 infrequent genes in PRAD, among which IDH1 mutant patients were recently identified as a distinct molecular-subtype of PRAD [138], NKX3-1 is required for normal prostate tissue development and CDKN1B was previously associated with PRAD.

In BRCA, HIT’nDRIVE identified 107 driver genes across 1090 patients. Somatic mutation of PIK3CA (30.5%) and TP53 (30.2%) were the most recurrent events in BRCA.
This was followed by somatic mutation of CHD1 (11.2%), GATA3 (10.5%), MUC16 (6.9%), MAP3K1 (6.9%), and CNA amplification of NSD1 (8.7%) and MED1 (6.9%). BRCA patients harboured 16 genes present in CGC database (pvalue = 9.310^{-3}) (Figure 3.2A), among which 10 genes were actionable targets (Figure 3.2B). More than 60% of BRCA patients could benefit with the actionable targeted therapy. Furthermore, additional 11% of BRCA patients harboured at least one of the 19 potentially druggable genes (Figure 3.2C). ACACA (altered in 36 patients mostly from HER2 subtype), involved in fatty-acid metabolism, was neither found in CGC nor in COSMIC databases. We identified 46 infrequent driver genes among which BRCA2 and GNAS have been previously linked to BRCA.

Although the driver events per tumor sample greatly varied, the median number of driver genes among the the 2207 tumor samples in all four cancer types was three (Figure 3.2). Twenty-three percent of 2207 tumor samples harboured just a single driver gene. Remaining 77% of tumor samples harboured two or more driver events which may either indicate the existence of multiple sub-clonal populations within the tumor, or the presence of collaboration among multiple sequence altered genes in an oncogenic pathway.

### 3.3.2 Network Properties of Cancer Driver Genes

Cancer driver genes are known to occupy critical positions in the interactome. To check whether HIT’nDRIVE predicted driver genes also occupy similar positions in the interaction network, we used the node degree as a “local measure”, and node betweenness (the number of shortest paths between node pairs that pass through the node) as a “global measure” of centrality. The driver genes predicted by HIT’nDRIVE include a number of well-known high-degree hubs – TP53, EGFR, RB1, MYC, PIK3CA, ERG, CHD1, that are “central” in the interactome with high degree and high betweenness (Figure 3.3A). Although there was very weak correlation between the number of edges (i.e. degree centrality) of a node and the number of samples/patients in which it is identified as a driver, each hub gene was typically altered in a large fraction of patients. Because of their centrality perturbations, hub genes are likely to dysregulate several other genes and the associated signalling pathways. HIT’nDRIVE also identified low-degree genes (IDH1, MTAP, NF1, NRG1, NSD1) that reside in the periphery of the interaction network. In particular, in prostate cancer, there seems to be an inverse correlation between the degree and how often the gene is picked as a driver. Most of these low-degree genes are altered in a small fraction of patients, indicating that HIT’nDRIVE, does not primarily return hubs that are altered in a large number of patients but is capable of identifying rare driver genes without trivial topological biases.

As discussed in previous sections, we used hitting time to compute the influence of a node in an interaction network. The influence from a source to a target node depends on the topological position of the target node in the network. We observed that the nodes occupying central positions in the network, i.e. with high betweenness centrality, tend to receive more influence than the nodes in the periphery of the network (Pearson correlation coefficient (R)
This is because the distance between any source node and a central target node (i.e. a hub) is usually very short, implying a low hitting time, and thus high influence of the source node on the target. We also observed negative correlation between a node’s total incoming influence and the median outgoing influence ($R = -0.54$) (Figure 3.3C). Although, the central hub nodes (for example - UBC, TP53) are good receivers of influence, when individual influences are considered, they do not contribute a lot.

Next we examined the influential driver genes that are selected as responsible for driving cancer. For this, we computed the total outgoing influence from each altered gene (which has been chosen as a driver), defined as the weighted sum of all influence values from the source to all outlier genes it is connected to (targets), weighted by the corresponding outlier weights. First we investigated driver genes with high influence values within each cancer type. We observed that on average the total influence of driver genes was higher than that of other altered genes in all cancer types (Figure 3.3D). EGFR, PTEN, CHEK2, TP53 and CDKN2A were the most influential driver genes in GBM which together exerted 38.5% of the total influence on the GBM patient cohort. In OV, TP53, GSTT1 and MYC together exerted 20% of the total influence. Similarly, in PRAD cohort, SPOP, MUC4 and TP53 were the most influential genes exerting 23.7% of the total influence. PIK3CA, TP53 and CHD1 were the most influential genes exerting 23% of the total influence on the BRCA patient cohort. Moreover, the gene influence was positively correlated to its alteration frequency (Figure 3.3E).

3.3.3 Sensitivity of Identification of Infrequent Drivers to Sample Size

In order to assess whether the identification of infrequent driver genes (those present in at most 2% of samples) is simply due to the large sample size in our experiments, we have performed the following experiment in order to measure the likelihood of HIT’nDRIVE selecting infrequent drivers even at a lower sample size. Using 1000 TCGA-BRCA tumors as the “original set”, different subsets of tumor samples with varying sample-size were chosen such that the sub-sample tumor population has similar alteration-frequency distribution to that of the original 1000 tumor samples. Our objective was to sub-sample TCGA-BRCA tumors with different sample-sizes such that the frequency distributions of mutations in the selected sub-samples are similar to that of original 1000 BRCA tumors (Figure 3.4).

For this we first estimated the alteration-frequency distribution of the original 1000 tumor samples and calculated the mean ($\mu_{\text{target}}$) and standard-deviation ($\sigma_{\text{target}}$) of the distribution. Our aim here is to find the sub-set of samples (with defined sample-size) such that the mean ($\mu_{\text{obs}}$) and standard-deviation ($\sigma_{\text{obs}}$) of the sub-sampled population is very close to $\mu_{\text{target}}$ and $\sigma_{\text{target}}$ respectively. This can be represented as the following goal, provided that we give equal penalty to both the mean and standard-deviation difference:

$$\text{MINIMIZE} \ (\text{Score} = |\mu_{\text{obs}} - \mu_{\text{target}}| + |\sigma_{\text{obs}} - \sigma_{\text{target}}|)$$
Figure 3.3: Network properties of driver genes. (A) The centrality of the predicted drivers in STRING v10 network. The size of the circles is proportional to the alteration frequency of the driver gene. The color scale represents the total influence of the driver gene on the expression outliers. (B) Correlation between influence and centrality. Each dot represents a target node receiving certain amount of influence from all source nodes in the network. A Lowess regression line is represented in blue. (C) Correlation between incoming and outgoing influence of a node. Each dot represents a node in the network and the color scale represents its betweenness centrality. A linear regression line is represented in blue. (D) Boxplot of the total influence of driver genes predicted by HIT’nDRIVE on the expression outliers compared to that of other altered genes (genes not predicted as drivers). (E) Correlation between gene influence and its alteration frequency in the respective patient cohort. (F) Relative influence of driver genes in each patient in GBM cohort with mutation in ABCB1. (G) Relative influence of driver genes in each patient in PRAD cohort with mutation in BRAF. (All gene influence values have been multiplied by 10^5 before log transformation.)
Figure 3.4: **Infrequent driver prioritization sensitivity to sample size.** (A) TCGA-BRCA cohort with 1000 tumors were selected for randomization experiment (labelled here as “original”). Tumor samples of different sample size were sub-sampled such that the sample alteration frequency in the sub-sample population is very similar to that of the original 1000 tumors. HIT’nDRIVE simulation was performed in all of the above sub-sampled tumor populations. The least frequent driver gene was identified for each sub-sampled population. (B-C) Number of samples (and percentage of sub-sampled tumor population) in which the least frequent driver gene is present.

We took a heuristic approach to solve the problem – we randomly sub-sampled, with a user defined sample-size, from the original set of tumor samples and calculated the above score. For a given target sample size, this step was repeated 10,000 times (i.e. 10,000 different combination of samples of defined sample-size). Then the sub-sample set with least score was chosen for further HIT’nDRIVE analysis.

HIT’nDRIVE analysis was performed on the chosen subsets of tumor samples independently to identify driver genes and then compared the frequency of driver genes detected. HIT’nDRIVE detected driver genes that were prevalent in just a single patient tumor with a sample size of 50, 25 or 15 tumors representing 6.5%, 4%, and 2% of the tumor population,
Figure 3.5: HIT’nDRIVE identified driver genes with respect to network perturbation in 100 select BRCA samples. The edges of the STRING v10 network were perturbed to different extent (between 1-10%), preserving the degree of the nodes in the network. HIT’nDRIVE was run for each of the perturbed networks. Proportion of common driver genes between the unperturbed network and each of the perturbed network were calculated and is plotted on the figure.

respectively. Even when the sample-size was increased to 700-900 samples, HIT’nDRIVE was able to detect driver genes prevalent in just 4 patients representing < 0.5% of the tumor population. This demonstrates that HIT’nDRIVE prioritizes infrequent driver genes independent of the sample-size.

3.3.4 Sensitivity of HIT’nDRIVE to the Underlying Network

In our analysis, we used the STRING v10 interaction network of high confidence edges. However, different interaction networks can differ significantly and thus impact the output of our method. Therefore, we tested the sensitivity of HIT’nDRIVE to changes in the interaction network with two tests on a randomly chosen subset of 100 TCGA-BRCA samples (following the same distribution-preserving scheme as explained in Section 3.3.3).

The first test was a perturbation test, where the edges of the STRING v10 network were perturbed to different extent (between 1 – 10%), while making sure to preserve the degree of the nodes in the network. HIT’nDRIVE analysis was then performed for each perturbed network with varying values of the \( \gamma \) parameter, and the proportion of common driver genes between the unperturbed network and each of the perturbed networks was calculated (Figure 3.5). We observed that even though the edges of the network were perturbed, the list of driver genes did not change to a great extent (i.e. the overlap of driver genes was very high) as compared to the non-perturbed network, even when the edges of the network were perturbed by up to 10%. The plot shows that there is a 70 – 80% overlap for values of the \( \gamma \) parameter that were used in our experiments (\( \gamma = 0.8 \)). This demonstrates that HIT’nDRIVE output is fairly robust with respect to small network perturbations.
Figure 3.6: HIT’nDRIVE identified driver genes with respect to underlying network used in 100 select BRCA samples. (A) Venn Diagram showing the overlap of nodes in the three different networks used - STRING v10 (only high-confident interactions), HPRD v9.0 and REACTOME v2015. (B) Comparison between the number of nodes in the network. (C) Comparison between the number of edges in the network. (D) Comparison between the number of driver genes detected using different networks. (E) Proportion of common driver genes between the networks (STRING-REACTOME and HPRD-REACTOME) as compared to driver genes detected using REACTOME network. A subset of 100 BRCA samples from TCGA were used for the simulation.

The second test aimed to measure the difference in HIT’nDRIVE output when different interaction networks were used (rather than just a perturbation of the edges). We evaluated the robustness of HIT’nDRIVE on three networks, STRING v10, HPRD 9 and REACTOME 2015. Only 34% of the vertices in STRING, HPRD and the REACTOME networks are shared in all three networks; in terms of edges, an even smaller proportion is shared. Not surprisingly, the more nodes the network has, the more driver genes HIT’nDRIVE predicts; this is consistently observed across various parameter settings. What is noteworthy is that the percentage overlap between the driver genes predicted on the three networks is quite robust, i.e., the percentage of driver genes shared between all three networks is preserved across various parameter settings and significantly higher than the relative overlap of genes of the interaction networks. This overlap is above 60% between the REACTOME and any of the other two networks, across various values of γ. In fact, the driver genes predicted on the STRING network are almost a superset of those predicted on REACTOME.
Figure 3.7: HIT’nDRIVE identified CGC genes using randomized input data in 100 select BRCA samples. Driver genes predicted by HIT’nDRIVE in non-randomized data compared with the driver genes predicted using randomized (i.e. by gene label swapping for 100 iterations). (A) Randomized altered genes and (B) Randomized outlier genes. The Cancer Gene Census (CGC) genes present in the predicted driver geneset is displayed in the plot. A subset of 100 BRCA samples from TCGA were used for the simulation (See Section 3.3.3 for details).

3.3.5 Randomized Alteration and Outlier Labels Testing

In order to assess the tendency of HIT’nDRIVE to make false positive predictions, we performed tests in which we permuted alteration and outlier labels and measured the number of CGC genes in the output of HIT’nDRIVE on the permuted data. We randomly selected a subset of 100 TCGA-BRCA samples on which the analysis was performed. In
100 experiments, we permuted alteration (outlier) labels (thereby preserving the recurrence frequency of each altered gene and outlier) in the chosen set of 100 TCGA-BRCA samples and ran HIT’nDRIVE on the permuted data. Likewise, the non-randomized results were obtained from the same 100 samples, just without permuting the original labels of their alterations and outliers. The average number of CGC genes in the output of HIT’nDRIVE in the permuted data was plotted against the non-randomized data on Figure 3.7, both for randomized alterations and randomized outliers. The figure shows that there is a stark contrast between the two sets of driver predictions (i.e. drivers obtained via non-randomized vs randomized data) with respect to their overlap with the Cancer Gene Census (CGC) data set - conserved through different values of the $\gamma$ parameter. Driver genes predicted in the non-randomized alteration (or non-randomized outliers) data not only (i) included a higher number of CGC genes (i.e. more number of true driver genes) as compared to that in driver genes predicted from randomized alterations (or randomized outliers) data, but also (ii) the number of CGC driver genes predicted through the use of non-randomized data increased quickly with increasing $\gamma$ parameter, whereas it stays roughly the same when randomized data was used. Note that while performing randomization, the original gene labels (sequence-wise altered genes or expression-outlier genes) were randomly replaced by new ones while preserving their recurrence frequency distributions.

### 3.3.6 HIT’nDRIVE’s Ability to Capture CGC genes

To assess HIT’nDRIVE’s ability to capture true driver genes, we perform a significance analysis in which we assume that the cancer-type specific genes listed in CGC database are the true driver genes i.e. the ground truth. As described earlier, we predicted potential driver genes in patients from four major cancer types using HIT’nDRIVE. In this analysis, for every analyzed patient, we compared the amount of CGC true driver genes present in the input (i.e. all sequence-wise altered gene) and the output (i.e. subset of the input sequence-wise altered genes that are predicted as potential driver genes) data for HIT’nDRIVE, to find how many of them are captured by our method.

Figure 3.8 summarizes the results of this analysis. As can be seen, the likelihood of a sequence-wise altered CGC gene to be selected by HIT’nDRIVE is much higher than that of a non-CGC gene. Next, for each patient, we calculated the likelihood of HIT’nDRIVE to capture CGC genes (as described in Section 3.1.10). We found that majority of the samples analyzed have a very significant p-value (i.e. $< 0.01$) (Figure 3.9). This analysis demonstrates that HIT’nDRIVE is able to capture cancer driver genes, to a larger extent, in the patient samples analyzed.

### 3.3.7 CGC Cancer Type-Specific Gene Enrichment

In order to further compare HIT’nDRIVE with the other similar method, DriverNet, we looked into the list of prioritized driver genes by both HIT’nDRIVE and DriverNet and
Figure 3.8: Sequence-wise altered Cancer Gene Census (CGC) genes prioritized by HIT’nDRIVE v.s. that of non-CGC genes, for each patient sample, across four cancer types. Only CGC genes specific to the particular cancer type are considered here. **Green:** Cancer specific sequence-wise altered CGC genes prioritized by HIT’nDRIVE; **Red:** Cancer specific sequence-wise altered CGC genes NOT-prioritized by HIT’nDRIVE; **Orange:** Sequence-wise altered non-CGC genes prioritized by HIT’nDRIVE; **Purple:** Sequence-wise altered non-CGC genes NOT prioritized by HIT’nDRIVE. The right panel depicts absolute numbers and the left panel depicts relative proportions. As can be seen the likelihood of a sequence-wise altered CGC gene to be prioritized by HIT’nDRIVE is much higher than that of a non-CGC gene.

their overlap with the known CGC genes (Figure 3.11B). DriverNet selects a much larger number of driver genes, as compared to HIT’nDRIVE, to cover most outlier genes (across all four cancer types) due to its model considering only direct interactions in the network. In particular, in OV and BRCA, the number of HIT’nDRIVE identified driver genes is an order of magnitude smaller than that of DriverNet. Although in GBM and PRAD datasets, the number of driver genes identified by DriverNet is somewhat lower and comparable to that identified by HIT’nDRIVE (primarily because most outliers were filtered out due to sharing no interaction edge with candidate altered genes), HIT’nDRIVE-identified driver genes
cover a significantly larger number of outliers. More importantly, even though HIT’nDRIVE identifies a smaller number of driver genes, a larger fraction of these driver genes can be found in CGC database – in comparison to the DriverNet identified driver genes (Figure 3.11B, right side). In fact, even a larger fraction of CGC genes specific to the relevant cancer type can be found among HIT’nDRIVE identified driver genes. Specifically, HIT’nDRIVE predicted four glioblastoma specific CGC genes (\textit{IDH1}, \textit{PDGFRA}, \textit{PIK3CA} and \textit{PIK3R1}) in TCGA-GBM dataset. Among them, \textit{IDH1}, \textit{PDGFRA} and \textit{PIK3CA} were not identified by DriverNet. Similarly, four ovarian cancer specific CGC genes (\textit{BRCA1}, \textit{BRCA2}, \textit{CCNE1} and \textit{MAPK1}) were predicted in TCGA-OV dataset. \textit{CCNE1} was not identified by DriverNet. Five prostate cancer specific CGC genes (\textit{BRAF}, \textit{ERG}, \textit{FOXA1}, \textit{PTEN} and \textit{SPOP}) were predicted in TCGA-PRAD dataset. \textit{BRAF} and \textit{SPOP} were not identified by DriverNet. And seven breast cancer specific CGC genes (\textit{BRCA2}, \textit{CCND1}, \textit{CDH1}, \textit{GATA3}, \textit{MAP3K1}, \textit{PIK3CA} and \textit{TP53}) were predicted in TCGA-BRCA dataset. Among them, \textit{CDH1} and \textit{MAP3K1} were not identified by DriverNet.

### 3.3.8 Phenotype Classification Using Dysregulated Modules Seeded With the Predicted Driver Genes

Evaluating computational methods for predicting cancer driver genes is challenging in the absence of the ground truth (i.e. follow-up biological experiments). Therefore, we focused on testing whether our predictions provide insight into the cancer phenotype and improve classification accuracy on an independent cancer dataset (not the one that drivers were obtained from). To test association of the driver genes identified by HIT’nDRIVE with the cancer phenotype, as explained in Section 3.1.6, we used the driver gene seeded gene-modules – a set of functionally related genes (e.g. in a signaling pathway) from the protein interaction network, as features for classifying the cancer phenotype. Using OptDis (here referred to as HIT’nDRIVE-OptDis), we identified small connected sub-networks that include (i.e. are
Driver-outlier interaction pairs are identified from the bipartite graph. We perform a hypergeometric test to identify significant driver-outlier interaction pairs across the patient cohort (p-value < 0.001). Each driver-module is seeded with one HIT’nDRIVE identified driver gene and includes outlier genes with significant driver-outlier interaction pairs.

Seeded by) predicted driver genes in a greedy fashion. More specifically, we prioritized sub-networks (of at most seven genes) iteratively so that in each iteration we identified the sub-networks that maximally discriminate sample phenotypes in a gene-expression matrix, among the sub-networks that share very few genes (at most 20%) with the sub-networks already prioritized.

Furthermore, we have also developed an unsupervised method for module identification (here referred to as HIT’nDRIVE-unsupervised), i.e. one that does not depend on any phenotype information. This unsupervised method seeds each module with one HIT’nDRIVE-identified driver gene, and includes outlier genes that it has influence over and co-occurs with significantly across patients (Figure 3.10). For this, we perform a hypergeometric test to identify significant driver-outlier interaction (i.e. mutual presence) pairs across the patient cohort (p-value < 10^{-3}).

Additionally, we compare HIT’nDRIVE-OptDis and HIT’nDRIVE-unsupervised to another network-based driver genes prioritization method – DriverNet [10]. DriverNet itself does not aim to identify modules that we can use to compare against HIT’nDRIVE-OptDis or HIT’nDRIVE-unsupervised modules; rather, DriverNet identifies driver genes in an iterative fashion, where in each iteration, DriverNet picks the driver genes which “covers” the maximum number of uncovered outliers. We use this driver and the outlier genes it covers as the “next” DriverNet module.

We first used the set of prioritized sub-networks, i.e. the driver modules, to perform binary sample classification: tumor vs normal. For this, we used gene-expression data for each of the four cancer types (GBM, OV, PRAD and BRCA) from TCGA as discovery datasets to calculate the mean gene expression value for each sub-network/driver module, for each patient. On these sub-networks, we used the k-nearest neighbour (KNN) classifier (with k = 1), to perform classification on both the expression values from TCGA, as well as

Figure 3.10: Schematic Diagram of HIT’nDRIVE-unsupervised approach to prioritize driver-modules. Driver-outlier interaction pairs are identified from the bipartite graph. We perform a hypergeometric test to identify significant driver-outlier interaction pairs across the patient cohort (p-value < 0.001). Each driver-module is seeded with one HIT’nDRIVE identified driver gene and includes outlier genes with significant driver-outlier interaction pairs.
additional validation gene-expression datasets. The additional validation datasets were used in order to assess the capability of the modules identified on TCGA cohort in classifying other cohorts.

For every dataset analyzed, the maximum classification accuracy achieved by HIT’nDRIVE modules (either HIT’nDRIVE-unsupervised or HIT’nDRIVE-OptDis), for any number of modules considered, was higher than that achieved by DriverNet modules (Figure 3.11A). Moreover, in most datasets, HIT’nDRIVE methods achieve maximum or near-maximum accuracy using a smaller fraction of modules. All three methods achieved perfect or near perfect classification accuracy in TCGA-GBM, TCGA-OV and TCGA-BRCA datasets, except for TCGA-PRAD dataset (where the maximum classification accuracy achieved was 90% by HIT’nDRIVE-Unsupervised, 95% by HIT’nDRIVE-OptDis and 86% by DriverNet). Overall, the driver modules (identified in the TCGA cohort) were able to distinguish the tumor phenotype from normal very well in validation datasets (on other, non-TCGA, cohorts) supporting the relevance of the identified driver genes to the cancer phenotype.

3.3.9 Significance of Driver Modules for the Phenotype Classification

Our rationale for using driver-module identified by OptDis as a feature of classifying phenotypes (eg. tumor vs normal) is that the observable effects of true driver alterations on their immediate vicinity (in an interaction network) should be sufficient to discriminate normal samples from tumour samples. To demonstrate that HIT’nDRIVE-OptDis-selected driver modules provide better classification accuracy than known gene sets, as well as randomly selected ones, we have compared the classification accuracy of modules selected by OptDis and seeded by HIT’nDRIVE-predicted drivers against the whole set of genes, randomly generated modules, OptDis modules seeded by CGC genes and an unsupervised clustering approach.

We compared the classification accuracy of randomly formed modules in the TCGA-PRAD dataset (which is the single most challenging dataset), against modules identified using HIT’nDRIVE-OptDis. To generate the random modules, 60 random sets of genes, of random size (up to 7 genes), were selected in 20000 experiments. Phenotype classification was then performed using KNN (k = 1) classifier. HIT’nDRIVE-OptDis modules demonstrated superior classification accuracy compared to randomly selected modules (Figure 3.12).

Next, we computed cross validation performance of whole set of differentially expressed genes and compared it to the accuracy of the modules on the TCGA-BRCA subtype expression datasets (Figure 3.13). There were 4657 genes on which the classifier was trained (using R caret LGOCV train method), achieving following accuracy: BASAL 93.41%, HER2 88.16%, LUMA 66.08% and LUMB 77.02%. The fact that the classifier using only a few modules composed of genes in the immediate vicinity of HIT’nDRIVE identified driver genes performs better than a classifier (albeit unsupervised) with access to the entire set of genes
Figure 3.11: **Phenotype classification using driver-seeded modules.** (A) Phenotype (tumor vs normal) classification accuracy in gene-expression datasets of different cancer-types using three different methods – HIT’nDRIVE-unsupervised (left panel), HIT’nDRIVE-OptDis (middle panel) and DriverNet (right panel). (B) Comparison of HIT’nDRIVE with DriverNet.

provides a strong evidence that HIT’nDRIVE identified genes are likely to be true driver genes.

We have also performed a comparison of classification accuracy against modules seeded by known drivers from the Cancer Gene Census (CGC) and selected by OptDis. As can be seen, HIT’nDRIVE identified driver seeded modules provide higher classification accuracy in the TCGA PRAD dataset (Figure 3.14A), potentially due to novel drivers identified by HIT’nDRIVE, since the PRAD cohort included non-trivial number of patients with no previously known drivers. In contrast, breast cancer is possibly the best studied cancer type with respect to known drivers. Thus it is not surprising that Basal, Her2 and Luminal-B subtypes show negligible differentiation between HIT’nDRIVE predictions and CGC based predictions (Figure 3.14B). This is due to big overlap between HIT’nDRIVE discovered
Figure 3.12: **Comparison of HIT’nDRIVE+OptDis based modules against randomly selected modules.** Phenotype Classification accuracy of HIT’nDRIVE driver seeded module identified by OptDis in TCGA-PRAD data against classification accuracy using randomly selected modules.

Figure 3.13: **Comparison of HIT’nDRIVE+OptDis based modules against linear combination of all genes.** Comparison of phenotype classification accuracy achieved by HIT’nDRIVE-OptDis with that achieved by the best possible nearest neighbor classifier that uses a linear combination of all differentially expressed genes (part of R’s ‘caret’ package). As can be seen, in all subtypes but especially for the Luminal-A subtype HIT’nDRIVE-OptDis provides a much higher classification accuracy.

modules and CGC modules (e.g. in BASAL, top 4 HIT’nDRIVE modules almost perfectly match the top 4 CGC modules - which, again, is not surprising since BRCA is a very well studied cancer with respect to drivers). However, HIT’nDRIVE shows some advantage in Luminal-A: HIT’nDRIVE outperforms the CGC genes - from 43rd module onward. This
Figure 3.14: Phenotype Classification using CGC Genes Seeded Modules. Phenotype Classification accuracy of HIT’nDRIVE driver seeded module vs Cancer Gene Census (CGC) genes seeded modules. (A) TCGA-PRAD gene-expression dataset with Tumor and Normal samples. (B) Subtype classification accuracy of HIT’nDRIVE identified driver seeded modules vs CGC BRCA driver seeded modules on the TCGA-BRCA cohort with respect to four subtypes of breast cancer (Basal, Her2, Luminal-A and Luminal-B).

Figure 3.15: Unsupervised Clustering of BRCA subtypes in TCGA-BRCA cohort. Unsupervised classification of BRCA subtypes based on the gene-expression profiles. The dendrogram was generated using Euclidean distance and Ward’s minimum variance method (via hclust, R’s hierarchical clustering function). As can be seen, unsupervised clustering can not identify BRCA subtypes well. In particular, LUMINAL-A and LUMINAL-B subtypes are well mixed in the dendogram.

may be due to HIT’nDRIVE predicted driver genes (seeds) such as DMD, ROCK1, AGAP1, SHANK2 which are not part of CGC; possibly playing important roles in some patients.

The classification of breast cancer subtypes (in TCGA samples) based on their gene-expression profiles using an unsupervised classification approach is shown in the Figure 3.15. The results of such an approach seem at best mixed. Based on the dendrogram shown
Figure 3.16: Phenotype classification of BRCA subtypes using driver-seeded modules. Phenotype (tumor vs normal) classification accuracy in gene-expression datasets of different breast cancer subtypes using HIT’nDRIVE-OptDis.

In Figure 3.15, the BASAL subtype is very easy to classify since it forms a distinct cluster. However, for the remaining subtypes there are no distinguishable clusters, and such subtypes are hence difficult to classify in this manner. In contrast, HIT’nDRIVE+OptDis modules are able to achieve classification accuracy of roughly 80% and above even on external validation datasets (Figure 3.16). For these reasons, unsupervised classification is not a suitable approach for the problem and HIT’nDRIVE+OptDis classification approach using driver modules is demonstrably superior over it.

3.3.10 HIT’nDRIVE seeded driver genes accurately predict drug efficacy

We obtained somatic mutation, copy number aberration and gene expression data of pan-cancer cell lines from Genomics of Drug Sensitivity in Cancer (GDSC) project [61]. We used HIT’nDRIVE to identify driver genes of individual cancer cell lines. Following up on the premise by [61] that potential driver genes (i.e. cancer genes, which include the CGC genes) alone could predict drug efficacy fairly well, the predicted driver genes were used as seeds in the network (STRING v10) to identify sub-networks that discriminate between the drug-response phenotypes – i.e. sensitive vs resistant cell lines. As available in GDSC, 265 different drug treatments were tested on each cell line provided. We present results for 25 cancer types (the remaining 5 cancer types for which only a very limited number of cell lines are available are statistically insignificant and thus have not been used).

Perhaps our most interesting result is that, for many drugs, the top HIT’nDRIVE predicted driver module for a specific cancer type (more specifically, OptDis modules seeded by HIT’nDRIVE-identified driver genes, prioritized with respect to drug efficacy) not only includes the drug target but also the associated (downstream) signalling pathway. As importantly, we measured the accuracy of drug-response phenotype classification using HIT’nDRIVE+OptDis for each drug-treatment in different cancer types (Figure 3.17). In most cancer types, HIT’nDRIVE-OptDis correctly predicted the response to more than 25% of the drugs in 95% of the patients or more. Specifically, Stomach adenocarcinoma (STAD) and Chronic Myelogenous Leukemia (LCML) are the cancer types with highest fraction of drugs predicted with an accuracy of 95% or more, whereas Liver hepatocellular carcinoma.
Figure 3.17: Drug efficacy predicted by HIT’nDRIVE seeded driver genes. Accuracy of drug-response phenotype classification for all 265 drugs used in GDSC study across 25 cancer types (the remaining 5 cancer types for which only a very limited number of cell lines have been made available are statistically insignificant and thus have not been used). The classification accuracy for each drug on each cancer type is measured based on the collective use of at most 10 best discriminating modules, i.e. the accuracy is maximized across the range of 1 to 10 (best discriminating) modules. Note that many of the drugs were not tested on all cancer types; in fact for the vast majority of cancer types only a handful of drugs were tested.

3.3.11 Correlation of Predicted Driver Genes With Alteration Burden

To obtain the mutation rate, we calculated the somatic mutation frequency per Mb (considering mutations in protein-coding genes only). We obtained copy-number burden values (i.e. percentage of somatic copy-number genome changed) using BioDiscovery Nexus Copy Number software. Figure 3.18A summarizes the correlation between Mutation rate and copy-number burden. As reported in many recent studies, samples in OV, PRAD and LIHC) and GBM are the cancer types with the lowest fraction of drugs predicted with the same accuracy.
BRCA had a high copy-number burden. In case of GBM, majority of samples had more or less equal mutation and copy-number burden. A large number of COAD samples were hypermutated and few other samples had high copy-number burden. Figure 3.18B shows the correlation of number of HIT’nDRIVE predicted driver genes with Mutation rate. Except for a number of hypermutated samples in COAD and few highly mutated samples in BRCA, the number of driver genes predicted by HIT’nDRIVE was not correlated with the somatic mutation rate of the respective sample.

In case of COAD, a large number of driver genes were identified in hypermutated samples (30 driver genes per sample in average) as compared to non-hypermutated samples (10 driver genes per sample in average). Finally, Figure 3.18C shows the correlation of number
of HIT’nDRIVE predicted driver genes with copy-number burden. Here too we observed the number of HIT’nDRIVE predicted driver genes were largely independent of the somatic copy number burden in the genome. Therefore, except for the hypermutated cases, the number of HIT’nDRIVE predicted driver genes is independent of both mutation rate and copy-number burden.

3.4 Discussion

In recent years, there has been an unprecedented increase in the multi-dimensional high-throughput data profiling (especially genome and transcriptome) of cancer patients. This has revealed extensive mutational heterogeneity observed in the cancer (sub)types, yielding a long-tailed distribution of mutated genes across the patients, implying the existence of many rare/private driver genes. Thus, there is a great need for computational methods to mine these massive datasets and prioritize clinically actionable driver events to aid treatment modalities using precision oncology.

Here, we have presented a network-based combinatorial method, HIT’nDRIVE, which models the collective effects of sequence altered genes on expression altered genes. HIT’nDRIVE aims to solve the “random-walk facility location” (RWFL) problem on a gene/protein interaction network – which differs from the standard facility location problem by its use of “hitting time”, the expected minimum number of hops in a random-walk originating from any sequence altered gene (i.e. a potential driver) to reach an expression altered gene, as a distance measure. We introduced the notion of “multi-hitting time” and presented efficient and accurate methods to estimate it based on single-source hitting time in large-scale networks. HIT’nDRIVE reduces RWFL (with multi-hitting time as the distance) to a weighted multi-set cover problem, which it formulates and solves as an ILP.

As a measure of influence, hitting time – the expected length of a random walk between two nodes, or its general version, the multi-hitting time, is quite different from the diffusion-based measures or Rooted PageRank, which are based on asymptotic distributions. We argue that hitting time is a better measure for our purposes as it is: (i) parameter free (diffusion model introduces at least one additional parameter - the proportion of incoming flow “consumed” at a node in each time step), (ii) it is time dependent (while the diffusion model and PageRank measures the stationary behavior) and (iii) it is more robust with respect to small perturbations in the network [58].

In this thesis Chapter, we have demonstrated that (1) HIT’nDRIVE increases our ability to identify potential genomic driver alterations. (2) HIT’nDRIVE prioritizes clinically actionable driver genes many of which happen to be private drivers. This implies that it is possible to replicate the lengthy and costly experimental approaches for detecting driver genes in common tumor types by HIT’nDRIVE – in-silico, strongly supporting the biological relevance of HIT’nDRIVE’s algorithmic framework. The fact that a high portion
of HIT’nDRIVE prioritized drivers in well studied cancer types overlap with known driver genes increases our confidence in the calls made by HIT’nDRIVE in rarer tumor types for which driver genes are mostly unknown. In fact, the initial results of the Pan Cancer Analysis of Whole Genomes (PCAWG) project reveal that more than 20% of tumors do not have a single (genomically altered) driver gene from CGC. (3) HIT’nDRIVE prioritizes driver genes present in both the centre and periphery of an interaction network. (4) Our analysis revealed that driver genes have higher collective influence on the transcriptome than other altered genes. Some of these driver genes are central and naturally have high influence, however there are also many non-central driver genes with high influence over other genes in the network. (5) HIT’nDRIVE is especially suitable for identifying such non-central driver genes or infrequent/private drivers. (6) We demonstrated that it is also possible to perform accurate phenotype prediction for tumor samples by only using HIT’nDRIVE implied driver genes and their “network modules of influence” (small sub-networks involving each driver gene where the aggregate expression profile correlates well with the cancer phenotype) as features, providing additional evidence that these genes may be driving the cancer phenotype. The network modules we identified may provide new insights into the biological mechanisms underlying tumor progression. (7) HIT’nDRIVE can capture subtype specific driver genes and such driver seeded modules can indeed differentiate between different subtypes of a cancer. (8) We have demonstrated that subtype specific driver modules are also associated with patients’ survival outcome providing additional evidence that these driver genes have clinical significance. (9) We also demonstrated that HIT’nDRIVE seeded driver genes (more specifically, OptDis modules seeded by HIT’nDRIVE identified driver genes, prioritized with respect to drug efficacy) enable excellent classification accuracy with respect to drug efficacy.

We also note that targeted therapeutics are being extensively used in clinical trials but the drug response rate is very poor (only ~5% of patients in clinical trials have good response to targeted therapeutics) [111]. This is most likely because even if a cancer patient harbours an alteration for which targeted therapeutics are available, we do not know if that alteration is responsible for driving the tumor [12]. HIT’nDRIVE could potentially play a key role by prioritizing potential driver alterations from a vast pool of passenger alterations. In our study, we have used drug efficacy data from pan-cancer cell lines in order to demonstrate that the potential genomic drivers (more precisely driver gene seeded modules) of the cell-lines can be used as features to predict drug-efficacy. Following similar procedure in clinical trials, we believe that the application of HIT’nDRIVE to predict drug efficacy would likely improve the drug response rate.

Protein-protein interaction (PPI) networks representing physical interactions now include thousands of proteins and over a million (undirected) interactions between them. Regulatory networks, on the other hand, represent gene/protein regulation occurring at multiple levels of biological systems through directed links. Since available regulatory net-
works are very limited in size and scope, our study focuses on PPI networks. However, HIT’nDRIVE can easily be applied to regulatory networks as they grow in size and scope. In addition, the use of multi-hitting time as a distance measure between two or more driver genes and a target gene enables HIT’nDRIVE to capture synthetic rescue like scenarios; this is ideally suited for undirected PPI networks, but in principle can be extended to regulatory networks in the future.

HIT’nDRIVE is a driver gene prioritization tool that is flexible enough to incorporate different types of -omics data. Both principles under RWFL and HIT’nDRIVE can be utilized to identify the causal genes in different complex disease facing analogous problems to cancer. Finally, we believe that applications of RWFL problem may extend beyond its application to driver gene identification – to influence analysis in social networks, disease networks and others.
Chapter 4

Conserved Alteration Patterns in Cancer

Recent large scale tumor sequencing projects such as PCAWG (Pan Cancer Analysis of Whole Genomes) have revealed multitude of somatic genomic, transcriptomic, proteomic and epigenomic alterations across cancer types. However, only a select few of these alterations provide proliferative advantage to the tumor and hence are called “driver” alterations [150]. Identifying driver alterations and their respective cancer genes, and distinguishing between them and functionally inconsequential random “passenger” alterations, is critical due to their potential to be used as therapeutic targets in treatment of cancer patients. While there are a large number of genes that have been found to be drivers in specific types of cancer, due to cancer heterogeneity it is very difficult to extract regular patterns of their activity and interdependence with other genes (potentially other drivers) in tumor samples.

Cancers are often driven by alterations to multiple genes [122, 7]. Whereas genomic alterations are likely consequences of endogenous or exogenous mutagen exposures [5], their evolutionary selection depends on the functional role of the affected genes [150] and their synergistic combinations. For example, TMPRSS2-ERG gene fusion is an early driver event in almost half of prostate cancer cases, and it often co-exists with copy-number loss of PTEN and NKX3-1 [21, 74, 77]. Another example is the concomitant deletion of four cancer genes - BAP1, SETD2, PBRM1, and SMARCC1 in chromosome locus 3p21, identified as a driver event in clear cell renal cell carcinoma (ccRCC) [22], uveal melanoma [116], and mesotheliomas [156]. These genes are involved in chromatin remodeling process, and their loss further impairs DNA damage repair pathway in tumors [116].

Alterations in two or more genes might be evolutionary co-selected because alteration in one gene might enhance the deleterious effect of the others [16]. Such co-selected genes are often active in a functionally significant subnetwork (i.e. module or pathway) within the human gene/protein interaction network and aberrations in such subnetworks are common to particular cancer types as demonstrated by recent sequencing efforts (e.g. PCAWG) [17]. For instance, TMPRSS2 interacts with ERG and PTEN (see the example above) in
STRING v.10 protein-protein interaction network; in fact all three genes co-operate to mod-
ulate NOTCH signaling pathway in \textit{TMPRSS2-ERG} positive prostate cancer progression
\cite{77}. As a result, it is desirable to identify subsets of functionally interacting genes which
are commonly (genomically or transcriptomically) altered in specific tumor types.

Recently, a number of computational methods have been developed to identify recurrent
genomic (as well as transcriptomic) alteration patterns across tumor samples. Some of these
methods have been designed to identify multiple gene alterations simultaneously based on
their co-occurrence or mutual exclusivity relationships in a tumor cohort, either with \cite{71}
or without \cite{95, 34} reference to a molecular interaction network. Other methods have been
developed to identify subnetworks within a molecular interaction network with specific
characteristics, e.g. the subnetwork of a fixed size with the highest total “weight” \cite{42, 86} or
the subnetwork seeded by a particular node that can be derived through a diffusion process
\cite{145, 81}; naturally these methods do not capture recurrent alteration patterns across a
cohort. A direction particularly relevant to our method is motivated by a number of related
works \cite{145, 72, 142, 4}, and explored by Bomersbach \textit{et al}. \cite{13}, which suggests to find a
subnetwork of a given size \( k \) with the goal of maximizing \( h \), the number of samples for which
at least one gene of the subnetwork is in an altered state. (A similar formulation where the
goal is to maximize a weighted difference of \( h \) and \( k \), for varying size \( k \), can be found
in \cite{60}.) Although the above combinatorial problems are typically NP-hard, they became
manageable through the use of state of the art integer linear programming (ILP) solvers
or greedy heuristics, or by the use of complex preprocessing procedures which significantly
reduce the problem size.

Complementary to the ideas proposed above, there are also several approaches to identify
mutually exclusive (rather than jointly altered) sets of genes and pathways \cite{94, 27, 146}. These approaches utilize the mutational heterogeneity prevalent in cancer genomes, and
are driven by the observation that mutations acting on same pathway are often mutually
exclusive across tumor samples. Although, from a methodological point of view, these ap-
proaches are very interesting, they are not trivially extendable to the problem of identifying
co-occurring alteration patterns (involving more than two genes) conserved across many
samples.

\section{4.1 Contributions of cd-CAP}

In this chapter of the thesis, a novel computational method is presented – cd-CAP (combinatorial
detection of Conserved Alteration Patterns), for detection of subnetworks of an interaction
network, each with an alteration pattern conserved across a large subset of a tumor sample
cohort. The framework of cd-CAP allows each gene to be labeled (or “colored”) with one or
more distinct alteration types (e.g. somatic mutation, copy number alteration, or aberrant
expression) with the goal of identifying one or more subnetworks, each with a specific alter-
Figure 4.1: **Schematic Overview of cd-CAP framework.** Multi-omics alteration profiles of a cohort of tumor samples are identified using appropriate bioinformatics tools. The alteration information is combined with gene-level information in the form of a sample-gene alteration matrix. Each alteration type is assigned a distinct color. Using a (signaling) interaction network, cd-CAP identifies subnetworks with conserved alteration patterns. (labeling) pattern, that is shared across many samples (Figure 4.1). As such, cd-CAP solves a novel problem that has not been tackled in the literature. In fact, the very notion of *conserved subnetworks* used by cd-CAP is novel: in [13, 60] the subnetworks of interest are composed of nodes such that in each patient at least one is altered (one way or another). In contrast, cd-CAP insists that each node is altered in each patient, and each node preserves its alteration type in each patient. Additionally, unlike [60] which employ heuristics to solve a highly restrictive problem and thus cannot guarantee optimality, cd-CAP uses a very efficient exhaustive search method (a variant of the a-priori algorithm, originally designed for association rule mining [2]) to quickly solve a very general problem optimally. Figure 4.2 summarizes the above-mentioned comparison between cd-CAP and the work of [13] and [60].

cd-CAP offers two basic modes: the “single-subnetwork” mode identifies the largest subnetwork altered the same way in at least \( t \) samples by solving the *maximum conserved subnetwork identification* problem optimally; the “multi-subnetwork” mode identifies \( l \) subnetworks of size (at most) \( k \) (\( k \) and \( l \) are user defined parameters) that collectively cover the maximum number of nodes in all samples by solving the *maximum conserved subnetwork cover* problem via ILP. In both modes, cd-CAP runs in two steps. The first step computes a set of all “candidate” subnetworks (each with a distinct alteration pattern) with at most \( k \) nodes, and which are shared by at least \( t \) samples. However, the two modes differ in the sec-
ond step: the first returns a single largest subnetwork, and the second returns \( l \) subnetworks collectively covering the maximum number of nodes from the set of candidate subnetworks.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Problem</th>
<th>1 color</th>
<th>1 out of multiple colors</th>
<th>multiple colors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heuristics</td>
<td></td>
<td>[13, 60]</td>
<td>-</td>
<td>cd-CAP (not reported)</td>
</tr>
<tr>
<td>ILP</td>
<td></td>
<td>[13, 60]</td>
<td>cd-CAP (not reported)</td>
<td>cd-CAP (not reported)</td>
</tr>
<tr>
<td>a-priori</td>
<td></td>
<td>cd-CAP</td>
<td>cd-CAP</td>
<td>cd-CAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-priori + ILP</td>
<td></td>
<td>cd-CAP</td>
<td>cd-CAP</td>
<td>cd-CAP</td>
</tr>
</tbody>
</table>

Figure 4.2: Comparison of cd-CAP’s framework to previous methods.

Additionally cd-CAP provides the user the ability to add or relax some constraints on the subnetworks it identifies. Specifically, the user can ask cd-CAP to (i) return “colorful” subnetworks (i.e. subnetworks of nodes with at least two distinct colors), or (ii) allow up to a \( \delta \) fraction of nodes in the subnetwork to have no alteration (as a result, not colored) in some of the samples that share the subnetwork.

cd-CAP has been applied – with both single and multi-subnetwork mode, with the basic setting (which only requires that each node has the same alteration type across the samples), as well as each of the possible additional options above, i.e., (i), (ii) – to The Cancer Genome Atlas (TCGA) breast adenocarcinoma (BRCA), colorectal adenocarcinoma (COAD), and glioblastoma multiforme (GBM) datasets. On these datasets, which collectively include more than 1000 tumor samples, cd-CAP identified several connected subnetworks of interest, each exhibiting specific gene alteration pattern across a large subset of samples.

In particular, cd-CAP results with the basic setting demonstrated that many of the largest highly conserved subnetworks within a tumor type solely consist of genes that have been subject to copy number gain, typically located on the same chromosomal arm, and are thus likely a result of a single large scale amplification. One of these subnetworks cd-CAP observed (in about one third of the COAD samples [18]) include 9 genes in chromosomal arm 20q, which corresponds to a known amplification recurrent in colorectal tumors. Another copy-number gain subnetwork cd-CAP observed in breast cancer samples corresponds to a recurrent large scale amplification in chromosome 1 [33]. It is interesting to note that cd-CAP was able to re-discover these events without specific training.

Several additional subnetworks identified by cd-CAP solely consist of genes that are aberrantly expressed. Further analysis with option (ii) in the multi-subnetwork mode of cd-CAP revealed subnetworks that capture signaling pathways and processes critical for oncogenesis in a large fraction of tumors. We have also observed that the subnetworks identified through all different options of cd-CAP are associated with patients’ survival outcome and can hence be clinically important.
In order to assess the statistical significance of subnetworks discovered by cd-CAP - in the single-subnetwork mode, we introduce for the first time a model in which likely interdependent events, in particular amplification or deletion of all genes in a single chromosome arm, are considered as a single event. Conventional models of gene amplification either consider each gene amplification independently [24] (this is the model we implicitly assume in our combinatorial optimization formulations, giving a lower bound on the true p-value), or assumes each amplification can involve more than one gene (forming a subsequent sequence of genes) but with the added assumption that the original gene structure is not altered and the duplications occur in some orthogonal “dimension” [119, 44, 159]. Both models have their assumptions that do not hold in reality but are motivated by computational constraints: inferring evolutionary history of a genome with arbitrary duplications (that convert one string to another, longer string, by copying arbitrary substrings to arbitrary destinations) is an NP-hard problem (and is difficult to solve even approximately) [30, 99]. By considering all copy number gain or loss events in the same chromosomal arm as a single event, we are, for the first time, able to compute an estimate that provides an empirical upper bound to the statistical significance (p-value) of the subnetworks discovered. Note that this is not a true upper bound since a duplication event may involve both arms of a chromosome - but that would be very very rare. Through this upper bound, together with the lower bound above, we can sandwich the true p-value and thus the significance of our discovery.

4.2 cd-CAP Formulation

Consider an undirected and node-colored graph $G = (V, E)$, representing the human gene or protein interaction network with $n = |V|$ nodes, where nodes $v_j \in V$ represent genes and edges $e = (v_i, v_j) \in E$ represent interactions among the genes/proteins. Given $m = |P|$ different samples/patients $P_i$ in a cohort, every given $P_i$ has a specific coloring of $G$, namely $G_i = (V, E, C_i)$, where each node $v_{i,j}$ (corresponding to node $v_j \in V$) is colored with one or more possible colors to form the color set $C_{i,j}$ (i.e. $C_i$ maps $v_{i,j}$ to a possibly empty subset of colors $C_{i,j}$). Each color represents a distinct type of alteration harbored by a gene/protein: specific alteration types we consider are somatic mutation (single nucleotide alteration or short indel), copy number gain, copy number loss or significant alteration in expression (this set of alterations can be trivially expanded to include genic structural alteration - micro-inversion or duplication, gene fusion, alternative splicing, methylation alteration, non-coding sequence alteration) observed in a gene or its protein product. Note that $C_{i,j} = \emptyset$ implies none of the alteration types we consider are observed at $v_{i,j}$. Also note that given a node $v_j$, its occurrences $v_{i,j}$ and $v_{i',j}$, in respective samples $P_i$ and $P'_i$, have at least one matching color if $C_{i,j} \cap C_{i',j} \neq \emptyset$. 
Now, consider a connected subnetwork $S = (V_S, E_S)$ of the interaction network $G$, where each node $v_j \in V_T$ is assigned a single color $c_j$. Such a colored subnetwork is said to be shared by a collection of colored sample networks $\{G_i : i \in I\}$ if the color $c_j$ assigned to each vertex $v_j$ is in the color set $C_{i,j}$ of each $v_i,j$ ($i \in I$), i.e. $c_j \in \bigcap_{i \in I} C_{i,j}$ for each $v_j \in V_S$. Note that $v_i,j$ is said to be covered by a colored subnetwork $S$ if that colored subnetwork is shared by $G_i$. Intuitively, a colored subnetwork $S$ represents a conserved pattern or a network motif. The main goal of cd-CAP is to identify conserved patterns (i.e. identically colored) of connected subnetworks across a subset of colored sample networks $G_i$.

In this thesis, two formulations of cd-CAP are explored – one for detecting the single largest colored subnetwork $S$ shared by at least a given number of samples, and one for simultaneously detecting multiple colored subnetworks $\{S_i\}$ that can cover the maximum amount of nodes (thereby trying to avoid mutual overlap among them). cd-CAP solves these two problems via the Maximum Conserved Subnetwork Identification (MCSI) and Maximum Conserved Subnetwork Coverage (MCSC) problem formulations, as described below.

Maximum Conserved Subnetwork Identification problem (MCSI). The first formulation that is presented in this thesis (MCSI) is used by cd-CAP in the “single subnetwork mode” in order to detect the single largest network motif that is common to a large number of all the colored sample graphs. More formally, given a network $G$, and a set of its copies $G_i$ that are node-colored in a sample-specific manner (as described in the previous section), Maximum Conserved Subnetwork Identification problem (MCSI) aims to identify the largest connected subnetwork $S$ of the network $G$, with colors assigned to its nodes, that occurs in at least $t$ (a user specified number) samples $P$, such that each node in $S$ has the exact assigned color in each sample $P_i \in P$.

Note that this formulation is orthogonal to that used in [13] and [60], where the goal is to maximize the number of samples that share a fixed size subnetwork. Unlike these formulations, MCSI admits a generalization of the a-priori algorithm, which we use to solve it efficiently.

Maximum Conserved Subnetwork Coverage (MCSC) problem. The second formulation (MCSC) is used by cd-CAP in the “multi subnetwork mode” to simultaneously detect multiple subnetworks that each represent common motifs and collectively cover the maximum amount of colored nodes across all the colored sample graphs. A parameter-free variant of the multi-subnetwork formulation would aim to cover all nodes $v_i,j$, in all input sample networks $G_i$, with the smallest number of subnetworks $S_i$ that are each shared by at least one sample network. We refer to this combinatorial optimization problem as Minimum Subgraph Cover Problem for (Node) Colored Interaction Networks (MSC-NCI). While this formulation is parameter-free, in a realistic multi-omics cancer dataset, the number of genes far exceeds the number of samples represented. Under such conditions, the solution to the
MSC-NCI problem will primarily include subnetworks that are large connected components that are shared by only one sample network. To account for this situation, we introduce the following parameters/constraints akin to those for the MCSI formulation: (1) we require that the nodes in each subnetwork have their assigned color shared by at least \( t \) samples (in the remainder of the discussion, \( t \) is referred to as depth of a subnetwork); and (2) we require that each subnetwork returned contains at most \( k \) nodes. Note that this variant of the problem is infeasible for certain cohorts (consider a particular node which has a unique color for a particular sample; clearly requirement (1) cannot be satisfied if \( t > 1 \)). Even if there is a feasible solution, the requirement that each subnetwork in \( T \) is of size at most \( k \) makes the problem NP-hard (the reduction is from the problem of determining whether \( G \) can be exactly partitioned into connected subnetworks, each with \( k \) nodes [40]).

Therefore, cd-CAP solves a slightly more constrained variant of this problem in the multi-subnetwork mode. More formally, given a network \( G \), and a set of its copies \( G_i \) that are node-colored in a sample-specific manner, Maximum Conserved Subnetwork Coverage problem (MCSC) aims to identify at most \( l \) subnetworks, of size at most \( k \) and whose color scheme is conserved across at least \( t \) sample graphs \( G_i \), so as to maximize the total number of nodes \( v_{i,j} \) in all the sample graphs \( G_i \) that are covered by the chosen subnetworks.

### 4.2.1 Solving cd-CAP

In this section, the algorithmic framework of cd-CAP is explained in detail, which consists of two steps for both its single and multi-subnetwork modes. The key insight that is used as the basis of the algorithm is that in all instances of interest, only a limited number of genes are colored in comparison to the total number of nodes \( m|V| \). This allows usage of an exhaustive search method that is designed for association rule mining [2] to build a list of all “candidate subnetworks” exactly and efficiently (e.g. in comparison to the ILP or heuristic solutions in [13, 60]). Note that the used exhaustive search method is an extension of the a-priori algorithm with the difference that cd-CAP requires the candidate subnetworks to maintain connectivity as they grow. In the first step, cd-CAP first computes the candidate subnetworks (each with a distinct alteration pattern) with at most \( k \) nodes, and which are shared by at least \( t \) samples in both modes. In the next step, in the single-subnetwork mode, cd-CAP simply returns the largest subnetwork among the constructed candidate subnetworks, while in the multi-subnetwork mode it solves the maximum coverage problem (MCSC) on the set of candidate subnetworks via an ILP formulation (which will be given later).

**First step of cd-CAP: Generating candidate subnetworks.** We generate the complete list of candidate subnetworks with minimum depth \( t \) by the use of *anti-monotone* property [82]: if any subnetwork \( S \) has depth \( < t \), then the depth of all of its supergraphs \( S' \supset S \) must also be \( < t \) (since adding additional nodes can only further restrict the set
of samples that support the subnetwork). This makes it possible to grow the set \( S \) of valid subnetworks comprehensively but without repetition (described as “optimal order of enumeration” in [92]) through the following breadth-first network growth strategy.

1. For every node \( v_j \) and each of its colors \( c_\ell \in \cup_i C_{i,j} \) that the node has in at least \( t \) graphs \( G_i \), we create a candidate subnetwork of size 1 (i.e. with single node) containing the node with color \( c_\ell \). All samples in which the node is colored \( c_\ell \) trivially share this subnetwork.

2. We inductively consider all candidate subnetworks of size \( s = \{1, 2, \ldots, k-1\} \) with the goal of growing them to subnetworks of size \( s + 1 \) as follows. For a given subnetwork \( S \) of size \( s \), consider each neighboring node \( u \) outside of \( S \), i.e. \( u \in N(S) \). For each possible color \( c'_\ell \) of \( u \), we create a new candidate subnetwork of size \( s + 1 \) by extending \( S \) with \( u \) - colored with color \( c'_\ell \). We maintain this subnetwork for the next inductive step only if the number of samples sharing this new subnetwork is at least \( t \); otherwise, we discard it.
Algorithm 3: ILP formulation for solving MCSC

Maximize \( \sum_{v_{i,j} \in U} Y[i,j] \)

s.t. (1) \( \sum_{S_p \in SC_{i,j}} X[p] \geq Y[i,j] \) (\( \forall v_{i,j} \in U \))

(2) \( \sum_{S_i \in S} X[i] \leq l \)

The procedure is repeated until none of the subnetworks of size \( s + 1 \) cover at least \( t \) samples (typically in the single-subnetwork mode), or until subnetworks of the maximum given size \( k \) have been constructed (typically in the multi-subnetwork mode). Once the procedure terminates, the single-subnetwork mode simply returns all subnetworks constructed in the final iteration (of maximum size). The multi-subnetwork mode requires additional processing as will be described below. Note, however, that during the extension of \( S \) above, if the new node \( u \) does not reduce the number of samples sharing it then \( S \) becomes redundant and is not considered as one of candidate subnetworks in the ILP formulation in the multi-subnetwork mode. Algorithm 2 shows the pseudocode for the described procedure.

It is worth noting that for the sake of simplicity, pseudocode of Algorithm 2 omits duplicity check which is present in the actual implementation of cd-CAP, which ensures that we do not add more than one instance of a subnetwork in line 17 (since two subnetworks \( S_a \) and \( S_b \) which differ in only a single node can be extended into the same subnetwork of larger size by picking the unique node of the other subnetwork as the neighbour). cd-CAP does so through the use of hash tables of bitmasks that represent subnetwork nodes and their assigned colors.

In the worst case, the number of generated subnetworks is \( O(|C|^k) \), where \(|C|\) is the number of different colors that nodes can have. The worst-case running time of Algorithm 2 is thus \( O(|C|^k|E(G)|) \), as nearly every edge of the graph might be explored when extending a subnetwork. However, in practice, both the coloring of nodes in each patient and the underlying interaction network are very sparse. Thus, for any interesting values of \( t \) (at least 15% of the size of the cohort), the algorithm completes in at most a few minutes.

Second step of cd-CAP: Solving MCSC for multi-subnetwork mode. Given the universe \( U = \{ v_{i,j} | C_{i,j} \neq \emptyset, i = 1, \cdots, m; j = 1, \cdots, |V(G)| \} \), containing all the colored nodes in all the sample graphs \( G_i \), and the collection of all subnetworks \( S = \{ S_i | S_i \) is shared by \( \geq t \) samples & contains \( \leq k \) nodes \( \} \) as constructed by algorithm 2, our goal
is to identify up to \( l \) subnetworks from the set \( \mathcal{S} \) which collectively contain the maximum possible number of elements of the universe \( \mathcal{U} \).

After the list of all candidate subnetworks \( \mathcal{S} \) is constructed (as described in the previous step), we represent the MCSC problem with the integer linear program shown in algorithm 3 and solve it using Gurobi or IBM ILOG CPLEX, by solving the maximum coverage problem where the sets are the conserved subnetworks and the elements are the colored nodes. A binary variable \( Y[i,j] \) corresponds to whether colored node \( v_{i,j} \) was covered by at least one chosen subnetwork, and binary variable \( X[i] \) corresponds to whether colored candidate subnetwork \( S_i \) was one of the chosen. Similarly, \( \mathcal{SC}_{i,j} \) represents the set of all subnetworks of \( S \) which contain node \( v_{i,j} \) properly colored in them. Constraint (1) ensures that a node is considered covered only if at least one subnetwork that contains it is picked. The objective is to pick at most \( l \) subnetworks (constraint (2)) such that the number of colored nodes that are covered by those subnetworks across all the different graphs \( G_i \) is the maximum possible.

### 4.2.2 Additional Constraints and Parameter Options

In addition to the exactly-conserved colored subnetworks obtained through the general MCSI or MCSC formulation as described above, cd-CAP offers the user to add or relax constraints through new parameters, in both single and multi-subnetwork mode.

**“Colorful” Conserved Subnetworks.** In some of the datasets that were analyzed, certain variant types (i.e. colors) were dominant in the input to an extent that all subnetworks identified by our method had all nodes colored identically. By insisting that the identified subnetworks are *colorful*, it is possible to, e.g., capture conserved genomic alterations and their impact on their interaction partners (form of expression alterations). For this purpose we introduce the notion of a *colorful subnetwork*, \( \mathcal{S} \), as a subnetwork that has at least two distinct colors represented in the coloring of its nodes, i.e. \( c_l, c_h \in \bigcup_{v_j \in \mathcal{S}} c_j \quad (c_l \neq c_h) \).

In order to identify colorful subnetworks instead of arbitrary subnetworks, we update the first step of cd-CAP so that it specifically keeps track of colorful subnetworks (rather than all subnetworks) in each iteration; this is because any colorful network must contain a connected colorful subnetwork.

**Subnetworks Conserved Within Error Rate \( \delta \).** In order to reduce the sensitivity of cd-CAP to noise (that emerges during the assignment of variant types to genes - due to limited precision of sequence or statistical analysis methods) in the input data, we provide the user the option to allow *errors* in identifying conserved subnetworks. For that, cd-CAP provides the user the option to specify an error rate \( \delta \) that represents the fraction of nodes in a subnetwork \( \mathcal{S} \) that can have no assigned color in any sample that shares \( \mathcal{S} \). We implemented this by updating the first step of cd-CAP so that it expands the set of
samples that share each candidate subnetwork $S$ to every sample in which $S$ occurs with $\leq \delta|S|$ color omissions.

4.2.3 Assessing the Statistical and Biological Significance of the Networks Identified by cd-CAP

It is possible to assess the statistical significance of the subnetworks identified by cd-CAP by applying the conventional permutation test [71, 13, 146] on the color assignments of nodes - under the assumption that each gene is altered independently: let $C_{i,j}$ represent the set of colors assigned to a node $v_{i,j}$ and let $C_i = \{(v_{i,j}, C_{i,j})\}$, represent the entire set of color assignments to nodes $v_{i,j}$ in network $G_i$. We can obtain a random permutation of the color assignment $C'_i$, by independently shuffling each color $c \in \bigcup_j C_{i,j}$ across the nodes of $G_i$, which results in an assignment of a new color set $C''_{i,j}$ to each node $v_{i,j}$, under the constraint that the total number of nodes with each color $c$ is preserved. For a subnetwork $S = (V_S, E_S)$ of size $k$ covering $t$ samples returned by cd-CAP in the single-subnetwork mode, we can carry out a permutation test as follows. First we generate a permuted color assignment (as described above) for each sample. Then we run cd-CAP in the single-subnetwork mode (possibly with the option (i) or (ii) as described in the previous section) and identify the largest subnetwork which covers at least $t$ samples. We repeat this sufficiently many (by default 1000) times to compute $P_{1,S}$, the number of times we end up with a subnetwork of size at least $k$ in $t$ or more samples, normalized by the number of attempts. We can use $P_{1,S}$ as an empirical p-value for subnetwork $S$ of size $k$.

$P_{1,S}$ forms an empirical lower bound for the p-value of $T$ rather than an accurate estimate since it ignores the inter-dependencies among gene alteration events (i.e. node colors). In particular, whole chromosome or chromosome arm level copy number amplifications/deletions are commonly observed in cancer - such events should be reflected in the permutation test we employ. To address this issue, we apply the following procedure to compute $P_{2,S}$ as an empirical upper-bound for the p-value of $S$, under the assumption that copy number alterations take place in whole chromosome arms. For a given color $E$, corresponding to either copy number gain or loss events, let $N_{i,E}$ denote the number of nodes with color $E$ in $G_i$. For each chromosomal arm $A$, consider the set of nodes $V_{i,A}$ that have been assigned at least one color in $G_i$. Now we can reassign colors to vertices such that (1) colors $E$ corresponding to copy number gain or loss are assigned to all genes in a chromosome arm simultaneously; specifically the set of nodes $V_{i,A}$ in a chromosome arm $A$ are all assigned the same color $E$ independently with probability $\frac{N_{i,E}}{\sum_j |C_{i,j}|}$ (which guarantees that the expected number of nodes with color $E$ in $G_i$ is preserved); (2) the remaining colors (not related to copy number gain or loss) are assigned randomly to those nodes without a color assignment thus far (as described in the computation for $P_{1,S}$). This process provides a new randomly permuted color assignment $C''_i$ which we use to obtain an empirical upper bound on the p-value of a subnetwork $S$ discovered by cd-CAP. For that we perform this process simultaneously in
all $G_i$ and check whether the largest subnetwork shared by at least $t$ samples exceeds the size of a subnetwork $S$ (identified on the input dataset by cd-CAP). We repeat this process sufficiently many times and record the number of times the largest subnetwork obtained indeed exceeds the size of $S$; that value normalized by the number of times the process is executed is the value $P_{2,S}$, the empirical upper bound on the p-value of $S$. The true p-value of $S$ must be in the range $[P_{1,S}, P_{2,S}]$ (provided that chromosome arms form the largest units of alteration).

Pathway enrichment analysis  We tested the set of genes in the subnetworks obtained by cd-CAP for enrichment against gene sets corresponding to pathways present in the Molecular Signature Database (MSigDB) v6.0 [126]. A hypergeometric test based gene set enrichment analysis [126] was used for this purpose. A false discovery rate (FDR) $\leq 0.01$ was used as a threshold for identifying significantly enriched pathways.

Association between cd-CAP identified sub-networks and patients’ survival outcome  In order to assess the association between each cd-CAP identified subnetwork $S$ with patients’ survival outcome, we used a risk-score based on the (weighted) aggregate expression of all genes in the subnetwork $S$. The risk-score ($R$) of a patient is defined as the sum of the normalized gene-expression values in the subnetwork, each weighted by the estimated univariate Cox proportional-hazard regression coefficient [11], i.e., $R = \sum_j \beta_j x_{ij}$. Here $j$ and $i$ represent a gene and a patient respectively, $\beta_j$ is the coefficient of Cox regression for gene $j$, $x_{ij}$ is the normalized gene-expression of gene $j$ in patient $i$, and $k$ is the number of genes in the subnetwork. The normalized gene-expression values were fitted against overall survival time with living status as the censored event using univariate Cox proportional-hazard regression (exact method). Based on the risk-score values, patients were stratified into two groups: low-risk group (patients with $R < \text{mean of } R$), and high-risk group (patients with $R \geq \text{mean of } R$). Note that only those patients that are covered by the subnetwork are considered for the analysis above. In fact, with respect to survival outcomes, the set of patients covered by a subnetwork identified by cd-CAP would not necessarily differ from those that are not, since the latter set is likely to be highly heterogeneous with respect to cancer subtypes.

4.3 Results

4.3.1 Datasets and data processing

We obtained somatic mutation, copy number aberration and RNA-seq based gene-expression data from three distinct cancer types - glioblastoma multiforme (GBM) [135], breast adenocarcinoma (BRCA) [137], and colon adenocarcinoma (COAD) [18] from The Cancer Genome Atlas (TCGA) datasets. The TCGA datasets were downloaded in February 2017 from the
### Table 4.1: Interaction networks used in cd-CAP analysis

<table>
<thead>
<tr>
<th>Network</th>
<th># nodes</th>
<th># edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPRD v9</td>
<td>9297</td>
<td>73564</td>
</tr>
<tr>
<td>REACTOME 2015</td>
<td>7507</td>
<td>346108</td>
</tr>
<tr>
<td>STRING v10 HiConf</td>
<td>10971</td>
<td>428596</td>
</tr>
<tr>
<td>STRING v10.5 HiConf Experimental</td>
<td>6933</td>
<td>63128</td>
</tr>
</tbody>
</table>

Overview of the types of interaction networks used and their sizes, reflected in the number of nodes and directed edges.

National Cancer Institute-Genomic Data Commons NCI-GDC (https://portal.gdc.cancer.gov/).

In addition, we distinguish four commonly observed molecular subtypes (i.e. Luminal A, Luminal B, Triple-negative/basal-like and HER2-enriched) from the BRCA cohort. For each sample, we obtained the list of genes which harbor somatic mutations, copy number aberrations, or are expression outliers as per below.

**Somatic Mutations.** All non-silent variant calls that were identified by at least one tool among MUSE, MuTect2, SomaticSniper and VarScan2 were considered.

**Copy Number Aberrations.** CNA segmented data from NCI-GDC were further processed using Nexus Copy Number Discovery Edition Version 9.0 (BioDiscovery, Inc., El Segundo, CA) to identify aberrant regions in the genome. We restricted our analysis to the most confident CNA calls selecting only those genes with high copy gain or homozygous copy loss.

**Expression outliers.** We used HTSeq-FPKM-UQ normalized RNA-seq expression data to which we applied the generalized extreme studentized deviate (GESD) test [117]. In particular, we used GESD test to compare the transcriptome profile of each tumor sample (one at a time) with that from a number of available normal samples. For each gene, if the tumor sample was identified as the most extremely deviated sample (using critical value $\alpha = 0.1$), the corresponding gene was marked as an expression-outlier for that tumor sample. This procedure was repeated for every tumor sample. Finally, comparing the tumor expression profile of these outlier genes to the normal samples, their up or down regulation expression patterns were determined.

**Interaction networks** We used the following human protein-interaction networks in the identification of the most significant subnetworks specific to the cancer types mentioned above. (1) STRING version 10 [131] protein-interaction network which contains high confidence functional protein-protein interactions (PPI). Self-loops and interactions with missing HGNC symbols were discarded and interaction scores were normalized (divided by 1000).
to obtain a reliability score in the range $[0, 1]$. Only high confidence interactions with combined score of 0.9 or greater were selected. (2) STRING network with high confidence and experimentally verified edges. Combined score was still required to be at least 0.9, but experimental score had to be greater than 0. (3) Human Protein Reference Database (HPRD) version 9 [70]. (4) REACTOME version 2015 [46]. Figure 4.1 shows the sizes of the mentioned networks.

4.3.2 Maximal Colored Subnetworks Across Cancer Types

We used cd-CAP to solve the maximum conserved subnetwork identification (MCSI) problem exactly on each of the protein-interaction networks we considered on all cancer types - for every feasible value of network depth. As can be easily observed, the depth and the size of the identified subnetwork are inversely related. We say that a network depth value is feasible if (i) the depth is at least 10% of the cohort size, (ii) the maximum network size for that depth is at least 3, (iii) the number of “candidate” subnetworks are at most 2 millions per iteration when running cd-CAP for that depth.

The number of maximal solutions of cd-CAP as a function of feasible network depth for each cancer type (COAD, GBM, BRCA Luminal A, and BRCA Luminal B) is shown in Figure 4.3A-D on STRING v10 PPI network with high confidence edges (see Figures 4.7–4.9 for the results on alternative PPI networks). In general, for a fixed network size, the number of distinct networks of that size decreases as the network depth increases. One can observe that the ends of “valleys” (the point right before a sharp rise) in the colored plots in Figure 4.3A-D correspond to the largest depth that can be obtained for a given subnetwork size.

In the remainder of the text we focus only on the single colored subnetwork of each given size that has the maximum possible depth (corresponding to the ends of “valleys” in the plots). (If for a given subnetwork size and the corresponding maximal depth, cd-CAP returns more than 1 subnetwork, they are ignored.) Many of the subnetworks we focused on, especially those with large depth, only consisted of expression outlier genes (typically all upregulated or all downregulated) (Figure 4.4A-D) - across all four cancer types. In Luminal A data set for example, cd-CAP identified a subnetwork of eight downregulated genes with a network depth 90 (Figure 4.4A) - consisting of genes EGFR, PRKCA, SPRY2, and NRG2, known to be involved in EGFR/ERBB2/ERBB4 signaling pathways (Figure 4.4B). EGFR is an important driver gene involved in progression of breast tumors to advanced forms [133] and its altered expression is observed in a number of breast cancer cases [33]. The subnetwork also included MET, another well-known oncogene [96], and is enriched for members of the Ras signaling pathway, which is also known for its role in oncogenesis and mediating cancer phenotypes such as over-proliferation [47].

cd-CAP additionally identified some (uni-colored) copy-number gain networks, typically with lower depth: a prominent example is in the COAD dataset with depth 163 (out of 463
Figure 4.3: Maximal conserved colored subnetworks using STRING v10 PPI network. (A-D) Number of maximal solutions and the size of the conserved colored subnetwork obtained using the MCSI formulation, as a function of network depth $t$, in each of four cancer types analyzed, on STRING v10 (with high confidence edges) PPI network. The horizontal axis denotes the depth (number of patients) of the network. For the blue plot, the vertical axis denotes the maximum possible network size (in terms of the number of nodes) and thus it is strictly non-increasing by definition. For the plots with different colors, the vertical axis denotes the number of distinct networks with network size equal to that indicated by the blue plot. As can be seen, the red plots depict networks where all nodes have a copy number gain, the yellow plots depict networks where all nodes are expression outliers and purple plots depict colorful networks (with at least two distinct colors). A total of 41 subnetworks across all cancer types (10 COAD, 4 GBM, 11 Luminal A and 16 Luminal B) correspond to the end of “valleys” in the color plots - and were further analyzed.

patients in the cohort) (Figure 4.4D). This network forms the core of larger (maximal) subnetworks cd-CAP identifies for lower depth values; it corresponds to a copy number gain of the chromosomal arm 20q - a well known copy number aberration pattern highly specific to colorectal adenocarcinoma tumors [18]. Another subnetwork cd-CAP identified in 15% of the 422 BRCA Luminal-A samples corresponds to a copy number gain on chromosome 1 which is again a known aberration associated with breast cancer [33].

A complete list of subnetworks of focus (from STRING v10 with high confidence edges), across all cancer data sets, is provided in the Table 4.3. For each of these subnetworks, and for each patient covered by a particular subnetwork, we calculated a risk-score defined as a linear combination of the normalized gene-expression values of the genes in the subnetwork weighted by their estimated univariate Cox proportional-hazard regression coefficients (see Methods section for details). Based on the risk-score values, the patients covered by the subnetwork were stratified into two risk groups (high risk and low risk group).

The expression outlier subnetwork we mentioned above for the Luminal A dataset was the most significant among all subnetworks identified in this dataset (Figure 4.4C). The
Figure 4.4: **Sample conserved colored subnetworks using STRING v10 PPI network.** (A-D) Two of the most interesting subnetworks identified using STRING 10 PPI network - both of which are uni-colored. The number in parenthesis next to each node represents the univariate Cox proportional-hazard regression coefficient estimated for each gene, used as its weight in the risk-score calculation to stratify patients into two distinct risk groups. (See Methods section for details).

(A-C) One of the 11 maximal colored subnetworks identified in BRCA Luminal A dataset: it consists solely of downregulated expression outlier genes and has depth 90 (patients). (A) The colored subnetwork (with 8 nodes) topology. (B) Pathways dysregulated by alterations harboured by the genes in the subnetwork - these genes are involved in EGFR, ERBB2, and FGFR signaling pathways. (C) Kaplan-Meier plot showing the significant association of the subnetwork with patients’ clinical outcome. Patients “covered” by the subnetwork were stratified into two groups, namely High Risk (8 patients) vs Low Risk (82 patients), based on their gene expression levels. (See Methods for details.)

(D-F) One of the 10 maximal colored subnetworks identified in COAD dataset - it consists solely of copy number amplified genes and has a depth of 163 (patients). Genes in this subnetwork belong to the same chromosomal locus 20q13. (D) The colored subnetwork (with 9 nodes) topology. (E) Pathways dysregulated by the alterations harboured by the genes in the subnetwork - these genes are involved in signal transduction and apoptotic process. (F) Kaplan-Meier plot showing the significant association of the subnetwork with patients’ clinical outcome (73 High Risk vs 83 Low Risk patients).
patients in the high-risk group have poor overall survival outcome suggesting clinical importance of the identified subnetwork by cd-CAP.

Another copy-number gain subnetwork shared among 163 patients in the COAD dataset (Figure 4.4D) was comprised of genes from chromosome locus 20q13 - likely indicating a single chromosomal amplification event. Intriguingly, these genes form a linear structure in the protein interaction network. Among them is a group of functionally related genes consisting of transcription factors and their regulators (genes \textit{CEBPB, NCOA’s, UBE2’s}), which are known to be involved in the intracellular receptor signaling pathway (Figure 4.4E). \textit{CEBPB} and \textit{UBE2’s} are also involved in the regulation of cell cycle [65]. At the other end of the linear subnetwork, there are \textit{MMP9} and \textit{SDC4}, established mediators of cancer invasion and apoptosis [93, 20]. We also confirmed that these genes are highly predictive of the patients’ survival outcome (Figure 4.4F). All these results seem to support that cd-CAP identified subnetworks are functionally important with potential clinical relevance.

### 4.3.3 Maximal Colorful Subnetworks Across Cancer Types

We used cd-CAP to solve the maximum conserved colorful subnetwork identification problem in each of the four protein interaction networks and each cancer type that we considered (see Section “Additional Constraints and Parameter Options” for details). Again, cd-CAP was run with every feasible value (as defined above) of network depth. The number of maximal solutions of cd-CAP as a function of network depth for each cancer type (COAD, GBM, BRCA Luminal A, and BRCA Luminal B) is shown in Figure 4.5A-D on STRING v10 PPI network with high confidence edges (see Figures 4.7–4.9 for the results on alternative PPI networks). Note that we pay special attention to subnetworks with at least one sequence altered gene (i.e. a gene that is somatically mutated or copy number altered) since the sequence alteration(s) may explain expression-level changes in the remaining genes of the subnetwork (Figure 4.6A provides such an example).

One such COAD subnetwork is composed of several overexpressed genes and one copy number gain gene - covering 108 patients (Figure 4.6A). This subnetwork is mainly enriched for genes involved in ribosome biogenesis (Figure 4.6B). Cancer has been long known to have an increased demand on ribosome biogenesis [97], and increased ribosome generation has been reported to contribute to cancer development [106]. The biological relevance of this subnetwork is also supported by survival analysis, which shows a strong differentiation between the high-risk and low-risk groups - see Figure 4.6C.

Another subnetwork we observed in 58 BRCA Luminal A samples consists of four copy number gain genes, an overexpressed gene, and two underexpressed genes, including \textit{EGFR} (Figure 4.6D). All copy-number gain genes and the overexpressed gene are located in chromosome 1q, commonly reported in breast cancer [33]. The subnetwork involves an interesting combination of the down-regulation of the cancer gene \textit{EGFR} and the amplification of a group of genes involved in T-cell receptor signaling (\textit{PTPRC}, \textit{CD247}, and \textit{ARPC5}; see Fig-
Figure 4.5: **Maximal colorful subnetworks using STRING v10 PPI network.** (A-D) Number of maximal solutions and the size of the conserved colorful subnetwork obtained using the MCSI formulation, as a function of network depth $t$, in each of cancer types analyzed on the STRING v10 (high confidence edges) PPI network. The horizontal axis denotes the depth (number of patients) of the network. For the blue plot, the vertical axis denotes the maximum possible network size (in terms of the number of nodes) and thus it is strictly non-increasing by definition. For the plots with different colors, the vertical axis denotes the number of distinct networks with network size equal to that indicated by the blue plot. As can be seen, the purple plots depict colorful subnetworks and the green plots depict networks that include one to two nodes which are not expression outliers. A similar analysis was performed on the STRING v10 (experimentally validated edges), REACTOME and HPRD PPI networks. A total of 104 colorful subnetworks corresponding to the end of “valleys” of the plots were identified across the 4 cancer types in all the above PPI networks.

Figure 4.6E). Thus we may surmise that the covered population of patients potentially have relatively low cancer proliferation index with higher anti-tumor immune response, which can be highly relevant indicators with respect to clinical outcome. Indeed, this subnetwork is significantly associated with patients’ survival (Figure 4.6F).
Figure 4.6: Sample colorful subnetworks using STRING v10 PPI network. Two of the most interesting color subnetworks identified using STRING 10 PPI network are provided here. The number in parenthesis next to each node represents the univariate Cox proportional-hazard regression coefficient estimated for that gene, used as its weight in the risk-score calculation to stratify the patients into two distinct risk groups. (See Methods section for details).

(A-C) One of the maximal colorful subnetworks identified in the COAD dataset, consisting of at most 2 non-expression outlier (for this case copynumber gain) genes, with depth 108 (patients). (A) The colored subnetwork (with 9 nodes) topology - obtained from STRING v10 (with experimentally validated edges) PPI network. (B) Pathways dysregulated by alterations harboured by the genes in the subnetwork - these genes are involved in Ribosome biogenesis and RNA processing. (C) Kaplan-Meier plot showing the significant association of the subnetwork, with patients’ clinical outcome (59 High Risk vs 47 Low Risk patients).

(D-F) One of the maximal colorful subnetworks identified in the Luminal A dataset with no color restrictions, with depth of 58 (patients). (D) The colored subnetwork (with 8 nodes) topology - obtained in the REACTOME PPI network. (E) Pathways dysregulated by the alterations harboured by the genes in the subnetwork. (F) Kaplan-Meier plot showing the significant association of the subnetwork with patients’ clinical outcome (30 High Risk vs 30 Low Risk patients).
Figure 4.7: Maximal conserved and colorful subnetworks using STRING v10.5 PPI network with high confidence and experimentally verified edges. A comprehensive view of the size of maximum subnetworks (identified through the MCSI formulation, both without and with the colorful constraint), and the number of such maximum subnetworks obtained from STRING 10 PPI network with experimentally verified edges, as a function of input network depth $t_i$ in each of the four TCGA datasets. The horizontal axis denotes the depth (number of patients) of the network. For the blue plot, the vertical axis (right) denotes the maximum possible network size (in terms of the number of nodes) and thus it is strictly non-increasing by definition. For the plots with different colors, the vertical axis (left) denotes the number of distinct networks with network size equal to the indicated by the blue plot. As can be seen, the red plots depict networks where all nodes have a copy number gain, the yellow plots depict networks where all nodes are expression outliers and purple plots depict colorful networks (with at least two distinct colors).
Figure 4.8: **Maximal conserved and colorful subnetworks using HPRD PPI network.** A comprehensive view of the size of maximum subnetworks (identified through the MCSI formulation, both without and with the colorful constraint), and the number of such maximum subnetworks obtained from HPRD PPI network, as a function of input network depth $t$, in each of the four TCGA datasets. The horizontal axis denotes the depth (number of patients) of the network. For the blue plot, the vertical axis (right) denotes the maximum possible network size (in terms of the number of nodes) and thus it is strictly non-increasing by definition. For the plots with different colors, the vertical axis (left) denotes the number of distinct networks with network size equal to the indicated by the blue plot. As can be seen, the red plots depict networks where all nodes have a copy number gain, the yellow plots depict networks where all nodes are expression outliers and purple plots depict colorful networks (with at least two distinct colors).
Figure 4.9: Maximal conserved and colorful subnetworks using REACTOME 2015 PPI network. A comprehensive view of the size of maximum subnetworks (identified through the MCSI formulation, both without and with the colorful constraint), and the number of such maximum subnetworks obtained from REACTOME PPI network, as a function of input network depth $t$, in each of the four TCGA datasets. The horizontal axis denotes the depth (number of patients) of the network. For the blue plot, the vertical axis (right) denotes the maximum possible network size (in terms of the number of nodes) and thus it is strictly non-increasing by definition. For the plots with different colors, the vertical axis (left) denotes the number of distinct networks with network size equal to the indicated by the blue plot. As can be seen, the red plots depict networks where all nodes have a copy number gain, the yellow plots depict networks where all nodes are expression outliers and purple plots depict colorful networks (with at least two distinct colors).
Table 4.2: Properties of subnetworks identified in the multi-subnetwork mode.
5 subnetworks identified by cd-CAP in multi-subnetwork mode for each cancer type: respective columns below depict the subnetwork size, depth, and the number of nodes in the subnetwork with copy number amplification (AMP), expression increase (EXP-UP) or decrease (EXP-DOWN).

4.3.4 Multiple-Subnetwork Analysis Across Cancer Types

We next sought to detect up to 5 subnetworks per cancer type that collectively cover maximum possible number of colored nodes by solving the MCSC problem on STRING v10.5 network (with experimentally validated edges). The subnetwork extension error rate was set to 20%, and we restricted the search space to subnetworks which do not consist only of expression outlier nodes, in order to obtain what we believe to be more biologically interesting results. The network depth $t$ was chosen for each dataset in a way that made it possible to construct all candidate subnetworks of maximum possible size while keeping the total number of candidate subnetworks below $2 \times 10^6$, making the problem solvable in reasonable amount of time. We set $t$ to 69 (15% of the patients), 62 (10% of the patients), and 110 (10% of the patients) respectively for COAD, GBM, and BRCA datasets. Table 4.2 shows the sizes of the selected subnetworks, per sample depth, and the coloring of the nodes in the resulting subnetworks.

We note that the subnetworks identified in the GBM dataset had the lowest depth (10-15% of the samples). COAD and BRCA datasets on the other hand have much larger depth (respectively 30-48% and 15-32% of the samples). Smaller subnetworks of the GBM dataset solely consist of copy number gain genes on chromosome 7q, a known amplification in GBM [15]. The two large subnetworks each contain a single gene with copy number gain ($SEC61G$ and $EGFR$, respectively) accompanied by several of overexpressed genes. BRCA
Figure 4.10: Multiple Subnetwork Analysis. Two of the largest subnetworks identified across the COAD, GBM and BRCA data sets (5 networks were identified per cancer type) through the MCSC formulation of cd-CAP on STRING v10.5 (with experimentally validated edges) PPI network. The number in parenthesis next to each node represents the univariate Cox proportional-hazard regression coefficient estimated for that gene, used as its weight in the risk-score calculation to stratify the patients into two distinct risk groups. (See Methods section for details).

(A-C) The largest of the 5 COAD subnetworks with a network depth of 149 (patients). (A) The subnetwork topology (with 15 nodes). (B) Pathways dysregulated by alterations harboured by the genes in the subnetwork. (C) Kaplan-Meier plot showing the significant association of the subnetwork with patients’ clinical outcome (69 High Risk vs 78 Low Risk patients).

(D-F) The largest of the 5 BRCA subnetworks with a network depth of 313 (patients). (D) The subnetwork topology (with 15 nodes). (E) Pathways dysregulated by the alterations harboured by the genes in the subnetwork. (F) Kaplan-Meier plot showing the significant association of the subnetwork with patients’ clinical outcome (33 High Risk vs 278 Low Risk patients).
dataset exhibits a similar pattern: each of the four large subnetworks contain a single copy number gain gene from chromosome 8q, \(\textit{NSMCE2}\) in one and \(\textit{MYC}\) in the remaining three subnetworks). Subnetworks detected in COAD dataset were much more colorful and recurrently conserved in a larger fraction of samples than those in the other datasets. All genes with copy number gain are located in chromosome 20q.

We identified a subnetwork with 15 nodes (11 genes with copy number gain, 1 overexpressed and 3 underexpressed genes) in 149 COAD patients (Figure 4.10A). All 11 copy number gain genes belong to chromosome 20q. \(\textit{IL6R}, \textit{PLCG1}, \textit{PTPN1}, \text{and \textit{HCK}}\) are involved in cytokine/interferon signaling to activate immune cells to counter proliferating tumor cells [124] (Figure 4.10B). \(\textit{UBE2I}, \textit{AURKA}, \text{and \textit{MAPRE1}}\) are involved in cell cycle processes. This subnetwork was found to be associated with patients’ survival outcome (Figure 4.10C).

We identified another subnetwork with 15 nodes (14 overexpressed and 1 copy number gain genes) in 313 breast cancer patients (Figure 4.10D). Genes in this subnetwork are involved in cell cycle processes (Figure 4.10E). In particular the cell cycle checkpoint processes were dysregulated - which is known to drive tumor initiation processes [149]. The subnetwork was found to be associated with patients’ survival outcome (Figure 4.10F) suggesting its potential clinical relevance.

4.3.5 Empirical P-Value Estimates Confirm the Significance of cd-CAP Identified Networks

To evaluate the significance of cd-CAP’s findings, we performed the permutation test described in Section 4.2.3, 1000 times on each cancer type for each possible setting of subnetwork constraints. Tables 4.3–4.4 and Figure 4.11 demonstrate the distribution of the empirical p-value upper bound estimates with STRING 10 (high confidence edges) PPI network, while the lower bound results look similar to what is presented in the figure and thus are omitted. In the permutation tests all cd-CAP identified subnetworks (without additional constraints) of size 2-5 were composed solely of expression altered genes; in contrast there are several larger CNV rich subnetworks observed in the TCGA COAD data set and others, further confirming the significance of our findings. Colorful subnetworks presented in Figure 4.5 are even less likely to occur at random.

4.3.6 Multi-colored Subnetwork Model Improves Survival Outcome Association

cd-CAP identified several multi-colored subnetworks in our analysis. In order to demonstrate the value of using multiple colors in cd-CAP’s framework (one for each alteration or expression outlier type), rather than just a single color indicating the presence of an alteration (or a lack of color in case that there is no alteration on a gene nor is it an expression outlier), we have run cd-CAP in both the standard \textit{multi-color} as well as a \textit{single-color}
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Depth</th>
<th>Size</th>
<th>Max Sim. Size</th>
<th>Minmax Sim. Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COAD</td>
<td>t = 283</td>
<td>4</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 234</td>
<td>6</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 224</td>
<td>7</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 163</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 162</td>
<td>12</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 159</td>
<td>15</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 156</td>
<td>18</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 153</td>
<td>22</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 149</td>
<td>27</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>COAD</td>
<td>t = 147</td>
<td>30</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>GBM</td>
<td>t = 167</td>
<td>11</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>GBM</td>
<td>t = 165</td>
<td>20</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>GBM</td>
<td>t = 164</td>
<td>27</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>GBM</td>
<td>t = 163</td>
<td>30</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 301</td>
<td>3</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 259</td>
<td>4</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 178</td>
<td>5</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 120</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 103</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 90</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 60</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 57</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 56</td>
<td>15</td>
<td>5</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 51</td>
<td>28</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 50</td>
<td>33</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 49</td>
<td>40</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumA</td>
<td>t = 48</td>
<td>46</td>
<td>6</td>
<td>4</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 170</td>
<td>3</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 159</td>
<td>4</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 143</td>
<td>5</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 136</td>
<td>6</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 131</td>
<td>7</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 124</td>
<td>8</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 115</td>
<td>9</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 109</td>
<td>10</td>
<td>≤ 2</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 104</td>
<td>11</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 99</td>
<td>12</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 94</td>
<td>13</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 90</td>
<td>14</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 83</td>
<td>16</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 80</td>
<td>17</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 77</td>
<td>18</td>
<td>3</td>
<td>≤ 2</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>LumB</td>
<td>t = 70</td>
<td>20</td>
<td>4</td>
<td>3</td>
<td>&lt; 0.001</td>
</tr>
</tbody>
</table>

Table 4.3: P-value estimates for maximal colored subnetworks. Empirical p-value estimates (including the maximum and minimum largest subnetwork sizes obtained in 1000 permutation tests) for the 41 maximum-sized subnetworks identified by cd-CAP shown in Figure 4.3. Compared with the subnetworks observed in real mutation profiles, those identified by cd-CAP in permutation tests (with identical t values) were much smaller, implying a p-value of < 0.001 for each of the colored subnetworks presented in Figure 4.3.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Depth</th>
<th>Size</th>
<th>Max Sim. Size</th>
<th>Minmax Sim. Size</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COAD</td>
<td>( t = 177 )</td>
<td>3</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 172 )</td>
<td>3</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 171 )</td>
<td>5</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 167 )</td>
<td>6</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 160 )</td>
<td>10</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 158 )</td>
<td>13</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 156 )</td>
<td>16</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 153 )</td>
<td>19</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>COAD</td>
<td>( t = 146 )</td>
<td>28</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>GBM</td>
<td>( t = 75 )</td>
<td>9</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>GBM</td>
<td>( t = 74 )</td>
<td>53</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 79 )</td>
<td>3</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 74 )</td>
<td>4</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 68 )</td>
<td>5</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 62 )</td>
<td>6</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 60 )</td>
<td>7</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 55 )</td>
<td>12</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumA</td>
<td>( t = 54 )</td>
<td>16</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 91 )</td>
<td>3</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 87 )</td>
<td>4</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 85 )</td>
<td>5</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 76 )</td>
<td>8</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 74 )</td>
<td>9</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 72 )</td>
<td>10</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 68 )</td>
<td>12</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 65 )</td>
<td>14</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 62 )</td>
<td>15</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
<tr>
<td>LumB</td>
<td>( t = 60 )</td>
<td>16</td>
<td>( \leq 2 )</td>
<td>( \leq 2 )</td>
<td>(&lt; 0.001 )</td>
</tr>
</tbody>
</table>

Table 4.4: **P-value estimates for maximal colorful subnetworks using STRING v10 PPI subnetwork.** Empirical p-value estimates (including the maximum and minimum largest subnetwork sizes obtained in 1000 permutation tests) for the 28 maximum-sized colorful subnetworks identified by cd-CAP using the STRING v10 high confidence edges PPI network (out of a total of 104 subnetworks using all 4 different PPI networks), shown in Figure 4.5. Compared with the subnetworks observed in real mutation profiles, those identified by cd-CAP in permutation tests (with identical \( t \) values) were much smaller, implying a p-value of \(< 0.001 \) for each of the colored subnetworks presented in Figure 4.5A-D.
Figure 4.11: Distribution of maximum subnetwork sizes identified by cd-CAP on permuted data for p-value calculation. A typical network size distribution for the maximum size subnetworks identified by cd-CAP in 1000 permutation tests with selected range of depth $t$ (notice that we do not differentiate the cases where cd-CAP return a subnetwork of size 0, 1 or 2). Compared with the subnetworks observed in real mutation profiles, those identified by cd-CAP in permutation tests (with similar or identical $t$ values) were much smaller.

mode (i.e. a limited version of cd-CAP that does not differentiate alteration types), and compared the association of the resulting subnetworks with patient survival. The subnetwork extension error rate was set to 20%, and in case of multi-color mode, we restricted the search space to subnetworks which did not only consist of expression outliers; thereby ensuring that subnetworks are indeed colorful. For each dataset (COAD, GBM, Basal, LumA and LumB), the value of the minimum required network depth ($t$) was set to the minimum amount for which we could obtain a solution of maximum size within a reasonable amount of time (up to 30 minutes) in the single-color mode. In multi-color mode, we used matching value of $t$, for each dataset. The specific values were 88 for COAD, 160 for GBM, 95 for Basal, 42 for LumA and 70 for LumB.

The benefits of cd-CAP’s ability to identify multi-colored subnetworks are demonstrated in Figures 4.12–4.13. The figure shows that, especially in COAD and GBM, the survival outcomes of samples that include the cd-CAP identified subnetworks differ significantly from those samples that do not include such subnetworks. In the BRCA Basal, LumA and LumB subtype data sets, the multi-color formulation results in subnetworks which are better associated with patient survival in all datasets except LumB (where the subnetworks identified in both modes consisted of exactly the same genes).

4.4 Discussion

In this Chapter, we introduced a novel combinatorial framework and an associated tool named cd-CAP which can identify (one or more) subnetworks of an interaction network where genes exhibit conserved alteration patterns across many tumor samples. Compared with the state-of-the-art methods (e.g.[4, 60]), cd-CAP differentiates alteration types asso-
Figure 4.12: Comparison of cd-CAP to a single-color version on TCGA COAD and GBM datasets. Kaplan-Meier plot showing the association of the subnetworks with patients’ clinical outcome (High Risk vs Low Risk patients), on TCGA COAD and GBM datasets and STRING v10.5 network. The left side shows single-color formulation plots, and the right column using the multi-color formulation.

cd-CAP provides the user with two major options. (a) In single-subnetwork mode, it computes the largest colored subnetwork that appears in at least $t$ samples. This option exhibits significant speed advantage over available ILP-based approaches; its a-priori based algorithmic formulation allows flexible integration of special constraints (on maximal subnetworks) – not only simplifying complicated ILP constraints, but also further reducing the number of candidate subnetworks in iteration steps (a good example for this is the “colorful conserved subnetworks” as introduced in Section “Additional Constraints and Parameter Options”). However, the identified subnetworks are required to be conserved, i.e., each node only admits one alteration type among the samples sharing it (although we have relaxed constraints that allow samples to have a few nodes without any alterations, i.e. colors). In the future, we may be able to extend the definition of a network to include nodes with color mismatches (for example, according to the definition in [142]) or [4] with a modification.
to cd-CAP’s candidate subnetwork generation algorithm. (b) In multi-subnetwork mode, it solves the maximum conserved subnetwork cover (MCSC) problem to cover the maximum number of nodes in all samples with at most \( l \) colored subnetworks (\( l \) is user defined) via ILP. In the future we aim to refine the MCSC formulation with reduced number of parameters and hope to develop exact or approximate solutions.

Subnetworks identified by cd-CAP in COAD, GBM and BRCA datasets from TCGA are typically enriched with genes harboring gene-expression alterations or copy-number gain. Notably, we observed that genes in subnetworks with copy-number amplification are universally located in the same chromosomal locus. Many of these genes have known interactions and are functionally similar, demonstrating the ability of cd-CAP in capturing functionally active subnetworks, conserved across a large number of tumor samples. These subnetworks seem to overlap with pathways critical for oncogenesis. In the datasets analyzed, we observed cell cycle, apoptosis, RNA processing, and immune system processes that are known to be dysregulated in a large fraction of tumors. cd-CAP also captured subnetworks relevant to EGFR/ERBB2 signaling pathways, which have distinct expression patterns in specific subtypes of breast cancer [33, 108]. Survival analysis of cd-CAP identified subnetworks also highlighted their potential for clinical relevance. In the future, it may be possible to use tissue-specific interaction data (such as [76] or [9]) to capture subnetworks with gene interactions that are more relevant to a specific cancer and tissue type.
Figure 4.13: **Comparison of cd-CAP to a single-color version on TCGA BRCA subtype datasets.** Kaplan-Meier plot showing the association of the subnetworks with patients' clinical outcome (High Risk vs Low Risk patients), on TCGA COAD and GBM datasets and STRING v10.5 network. The left side shows single-color formulation plots, and the right column using the multi-color formulation.
Chapter 5

Identification of Conserved Evolutionary Trajectories in Tumors

Cancer is a disease initiated by somatic genomic alterations. Accumulation of such alterations drives the progression of a cancer to an advanced form. As the tumor grows, new subpopulations of the tumor cells emerge with distinct genomic alteration profiles. After subsequent rounds of selection and expansion, these subpopulations give rise to substantial intra-tumor heterogeneity, which is arguably one of the main challenges in cancer management and treatment.

As multi-region, time-series and single cell sequencing data become more widely available, it is becoming clear that certain tumors share evolutionary characteristics with others. With new computational methods to identify recurrent cancer progression patterns from multi-dimensional tumor sequencing data, it may become possible to predict the likely course of evolution and perhaps an effective treatment strategy for certain cancer types [59, 85, 37]. E.g., novel approaches for targeted immuotherapy [32] could be developed into combinatorial strategies that simultaneously target multiple subclonal populations if a comprehensive characterization of recurrent subclonal expansion patterns could be established. Unfortunately, elevated levels of intra-tumor heterogeneity, with intricate subclonal mutational landscapes dominated by inconsequential passenger alterations, may make it difficult to discriminate predictable tumor progression signals from noise introduced by these statistically insignificant subpopulations.

In the last few years, several computational methods have been developed with the goal of inferring the subclonal composition and evolutionary history of tumors from tumor biopsy sequencing data [43, 36, 109, 87, 45, 62, 160, 41, 88, 89]. Phylogenetic trees for representing tumor evolution processes reported by such methods provide (at least one of) the likely relative temporal order of genomic alteration events. Unfortunately, the trees they report for many tumor samples differ significantly from other tumors with similar characteristics. In
contrast, some recent studies have been able to recapitulate tumor evolution more accurately through long-term clinical followup [152, 1, 141]. These studies were able to find evidence of recurrent progression patterns involving two or more co-occurring genomic alterations. Since, until recently, no computational tool was purposefully developed to identify recurrent patterns of tumor evolution across multiple tumors, these studies typically relied on manual inspection of all possible sequences of alteration events, starting from the root of the tumor phylogeny (representing germline), all the way to leaf nodes [141]. Recently, methods have been developed that cluster tumor phylogenetic trees into groups of closely related trees with respect to edge similarity/distance measures [91, 19, ?. In particular, the REVOLVER method [19] employs a maximum-likelihood learning strategy to construct a joint hidden tree model for all tumors in a cohort, which is then used to infer an individual tree depicting the temporal order of clonal alterations in each individual tumor (this particular aspect of the method is later improved upon by [?]). This is followed by a hierarchical clustering of the trees for the purpose of detecting edges shared within each cluster. We note here that REVOLVER neither constructs likely evolutionary trajectories (which may be done manually to a degree [19] via the use of shared edges) nor does it look into whether the trajectories are conserved in a significant portion of tumors.

In parallel to the effort summarized above, the problem of finding common sets of genomic alterations in cross-sectional tumor data has been investigated by several computational approaches. These approaches typically employ molecular interaction networks to identify connected subnetworks of genes that have been subjected to somatic alterations, that are shared across tumor samples in the sense that at least a fixed number (e.g. one) of the genes in the subnetwork is altered in each tumor sample [145, 81, 72, 142, 4, 13, 60]. Recently, cd-CAP [57] took a stricter approach, requiring preservation of every alteration in the subnetwork in at least a user defined number of tumor samples in a cohort, introducing the notion of a conserved alteration pattern. Through cd-CAP it has become possible to detect sizable networks of genes, altered in a similar way across tumor samples. However none of these approaches consider the temporal order of alterations to make inference about evolutionary processes common in certain tumor types.

In this thesis, we present a novel computational method, CONETT (CONserved Evolutionary Trajectories in Tumors), for combinatorial detection of recurrent tumor evolution trajectories. CONETT considers as input a partial temporal ordering of alteration events for each tumor in a given set of tumors; for any tumor the input is given in the form of a directed graph where each alteration event is represented by a distinct node. Given a pair of alteration events $a$ and $b$, one of the following must be true for each tumor: (i) a directed edge from $a$ to $b$ only (alternatively from $b$ to $a$ only) indicates that $a$ is an ancestor of $b$, (ii) a directed edge from $a$ to $b$ as well as a directed edge from $b$ to $a$ (or a bidirectional $a-b$ edge), indicates that the two events are known to have an ancestor-descendant relationship but their specific ordering is unknown, (iii) an “anti-edge” between $a$ and $b$ indicates that
the two events belong to two distinct lineages, or (iv) a “don’t care” edge between $a$ and $b$ indicates that either $a$ or $b$ or both are not observed in that tumor or no information is available with respect to their ordering. Given such input, and a “root” (e.g. driver) alteration event $s$, CONETT constructs a “consensus” phylogeny tree on a large subset of tumors whose topology “captures” the ancestor-descendant relationship for the largest possible number of event pairs in these subset of tumors. A path from the root of that phylogeny to any other event is said to form an “evolutionary trajectory” of alteration events, “conserved” in this subset of tumors.

In its most stringent setup, CONETT constructs a consensus tree (rooted at the germline “event” $g$) of all tumors in the cohort with maximum total node “depth” (distance from the root), under the constraint that if event $a$ is an ancestor of event $b$ in the consensus tree, then in each individual tumor in which $b$ has been observed, there must exist a directed edge from $a$ to $b$. As a result, not all alteration events can be a part of the consensus tree. Thus CONETT inferred consensus tree maximizes the number of ancestor-descendant orderings between pairs of alteration events that do not conflict with those orderings observed in any individual tumor graph. In a more general setup, CONETT can relax this constraint for a small fraction of tumor graphs. It also allows the user to specify a node $s \neq g$ as the “driver” event.

We have applied CONETT to the TRACERx clear-cell renal cell carcinoma (ccRCC) data set [141] involving 100 tumors, as well as TRACERx non-small-cell lung cancer (NSCLC) data set [63] involving 99 tumors, and identified a number of significantly conserved evolutionary trajectories which involve sequence-altered genes and copy number alteration events – not reported in the original studies.

5.1 Methods

Let $\mathcal{P}$ be a set of tumors for which it is possible to infer (possibly through the use of time series, multi-region, single cell or single molecule sequencing data) a partial ancestral ordering of alteration events, labelled by a gene or chromosome and the type of alteration affecting it (e.g. somatically altered single gene or copy number altered entire chromosomal arm) \(^2\). Let $\mathcal{A}$ be the set of all alteration events that are observed in at least one tumor $p \in \mathcal{P}$. For a given tumor $p \in \mathcal{P}$, we define a tumor graph $G_p$ as a graph that has a node

\[^1\]We require that the input graph for each tumor should be transitive, i.e. if there are directed edges from $a$ to $b$ and from $b$ to $c$ then there must exist a directed edge from $a$ to $c$.

\[^2\]Note that CONETT allows the user to distinguish distinct types of somatic alterations such as single nucleotide alterations, short insertions, deletions, inversions and duplications within a gene, as well as copy number gains and losses within a chromosomal arm. However, the studies that published the data sets that CONETT was applied to in this paper, did not associate significant biological differences with these distinctions, and thus, for the sake of simplicity and the ability to compare our results, we similarly largely avoided employing such distinctions.
Figure 5.1: **Overview of CONETT’s framework.** The method takes as its input a partial ancestor-descendant ordering of alteration events across a number of tumors and produces a tree of conserved evolutionary trajectories stemming from a common root event which maximizes the number of ancestor-descendant ordered pairs of nodes. **A) Input.** Tumor graphs represent partial ancestor-descendant order of alteration events in tumors. In this particular example, any pair of events that do not have an actual edge between them in a tumor should be thought to have an “anti-edge”; similarly any event not present in a tumor should be thought to have a “don’t care” edge to all events present in that tumor. **B) The Conserved Evolutionary Trajectory Tree.** In the most stringent setting CONETT computes the maximum total depth “consensus” phylogeny tree where the ordering of alteration events do not conflict with the partial orderings of alteration events in any of the tumors.
for each alteration event in $A$ (note that $G_p$ includes a node for each event in $A$, but not every node in $G_p$ corresponds to an event observed in $p$). For each pair of distinct nodes $a, b \in V(G_p)$ exactly one of the following must be true: the set of edges $E(G_p)$ contains (i) either a directed edge from $a$ to $b$ only or from $b$ to $a$ only, indicating that $a$ is an ancestor of $b$ (or alternatively, that $b$ is an ancestor of $a$); (ii) a directed edge from $a$ to $b$ as well as a directed edge from $b$ to $a$, indicating that the two events are known to have an ancestor-descendant relationship but the specific ordering is unknown; (iii) an “anti-edge” between $a$ and $b$, indicating that the two events belong to two distinct lineages; or (iv) a “don’t care” edge between $a$ and $b$, indicating that alteration events corresponding to either one or both of the nodes are not observed in that tumor or no information is available with respect to their ordering. We require that each tumor graph should be transitive with respect to its directed edges, i.e. if there are directed edges from $a$ to $b$ and from $b$ to $c$ then there must exist a directed edge from $a$ to $c$. Thus, edges of a tumor graph $G_p$ capture the complete available information about partial ancestor-descendant ordering of alteration events in tumor $p$.

We say that an ordered set of nodes $e = (v_1, \ldots, v_k)$ represents an evolutionary trajectory that is conserved in a tumor graph $G_p(p \in P)$ if for each pair of nodes $v_i, v_{i+1}$ in $e$ there is a directed edge from $v_i$ to $v_{i+1}$. In its most stringent setting, CONETT builds a tree $T$ that includes ancestor-descendant relationships observed in all tumors in $P$ - i.e. the path from the root to any given node $v$ in $T$ is conserved in all tumor graphs $G_p$ for which the alteration event corresponding to $v$ is observed in tumor $p$. Among such trees $T$, CONETT’s objective is to compute the tree $T_P$ that maximizes the total path length from the root to every other node $v$; we call this tree, the “consensus” tree for $P$, and each path in $T_P$ from the root to any other node a “conserved evolutionary trajectory”. Intuitively, the consensus tree maximizes the number of orderings between alteration event pairs. We note here that we represent the “germline” as a special (pseudo) alteration event $g$ from which each $G_p$ has a directed edge to every other node $v$. In this stringent setting, the root of $T$ is specified to be $g$, and thus $T$ will include each event in $A$ as a node - sometimes as a singleton leaf, with $g$ as its parent. Singleton leaves can later be filtered out as they reveal no new information.

In a more general setting (which we use for all our experimental results), CONETT offers the ability to identify longer evolutionary trajectories that are conserved only in a subset of tumor graphs in $P$, by relaxing the constraint that each trajectory should be conserved in every tumor graph. Additionally, this setting allows tumor graphs to be non-transitive (for reasons explained later in this section). Finally, this general setting allows the user to specify as the root, any alteration/node $s \in A$, to be identified as the “driver” alteration event. For every other event $v$, let $\mathcal{T}(s)_v$, denote the set of all tumor graphs $G_p$, where there is a directed path from $s$ to $v$. We then formulate the following problem.

3This is because the $T_P$ maximizes the total path length.
Maximum Conserved Evolutionary Trajectory Tree problem (MCETT): Construct a tree $T_{P,s}$ rooted at $s$ with the maximum total path length from $s$ to every other node $v$, such that every path $(s, u_1, ..., u_k, v)$ in $T_{P,s}$ is conserved in at least $\max(1, \gamma |T(s)_v|)$ tumor graphs in $T(s)_v$, for a user defined $\gamma \in (0, 1]$.

Note that in order to achieve maximum total path length (i.e. node “depth”), the solution to the MCETT problem necessarily includes all nodes $v$ for which $T(s)_v \neq \emptyset$. CONETT solves MCETT problem through an ILP formulation as described in Section 5.1.1. However if $s$ is set to be the germline node and $\gamma = 1$, the solution to the MCETT problem is the consensus tree we defined above for the most stringent setting of CONETT and for this case we describe a simple polynomial time algorithm below.

A Polynomial Time Algorithm for MCETT problem for $\gamma = 1$. Let $s$ denote the root alteration event (not necessarily representing the germline). For any alteration/node $v \in A$, let $E^*_v$ denote the set of edges $(u, v)$, conserved in every tumor graph in $T(s)_v$ and let $E^* = \cup_v E^*_v$. The solution to the MCETT problem for this setting is then the spanning tree of the nodes in $E^*$ in which the sum of distances between root $s$ and every other node $v$ is maximum possible. It is possible to compute this spanning tree through a simple depth first search strategy as shown in Supplementary Methods.

For the more general setting of MCETT where $\gamma < 1$, the maximum depth spanning tree strategy above could be used as an efficient heuristic that works reasonably accurately for higher values of $\gamma$. However it is possible to solve MCETT exactly for any value of $\gamma$ through an integer linear programming formulation as described below in Section 5.1.1. In this formulation, we do not differentiate (i) node pairs with a single directed edge from (ii) those with edges in both directions. We also do not differentiate (iii) node pairs with an anti-edge from (iv) those with a don’t care edge. We show how to differentially “incentivise” the first two edge types towards positive contributions, and “penalize” the next two edge types towards negative contributions within the objective of this ILP formulation later in the paper.

5.1.1 ILP formulation for solving the Maximum Conserved Evolutionary Trajectory Tree problem

Given a set of tumor graphs $\{G_p\}_{p \in P}$, a root alteration $s$ and a value of the parameter $\gamma \in (0, 1]$, the goal is to construct a tree $T$ as the solution to the Maximum Conserved Evolutionary Trajectory Tree problem. First, for each alteration event $v \in A$, CONETT finds the maximum-size subset $T(s)_v$ of all tumor graphs in which there is a directed path from $s$ to $v$ in each $G_p \in T(s)_v$. Then CONETT forms an evolutionarily conserved alteration set $S = \{v \in A \mid T(s)_v \neq \emptyset\}$, which is the set of all nodes $v$ for which $T(s)_v$ is non-empty and thus the node must be included in the solution tree.
Maximize: $\sum_{v \in S} D_v$

s.t.

1. $s : D_s = 0$
2. $\forall (v \in S) : X_{v,s} = 0$
3. $\forall (u, v) : PD_{u,v} \leq D_u$
4. $PD_{u,v} \leq X_{u,v} \cdot |S|$
5. $\forall (v \neq s) : D_v = 1 + \sum_u PD_{u,v}$
6. $\forall (v \neq s) : \sum_u X_{u,v} = 1$
7. $s : \sum_u f(e_{s,u}) \leq |S| - 1$
8. $\forall (v \neq s) : \sum_u f(e_{u,v}) = 1 + \sum_u f(e_{v,u})$
9. $\forall (u, v) : |S| \cdot X_{u,v} \geq f(e_{u,v})$
10. $X_{u,v} \leq f(e_{u,v})$
11. $\forall (G_p \in T(s)_s) : PTH^p_s = 1$
12. $\forall (G_p \notin T(s)_s) : PTH^p_s = 0$
13. $\forall (u, v, G_p) : PTHA^p_{u,v} \leq PTH^p_u$
14. $PTH\bar{A}^p_{u,v} \leq X_{u,v}$
15. $PTH\bar{A}^p_{u,v} \geq PTH^p_u + X_{u,v} - 1$
16. $\forall (v, G_p) : PTH^p_s = \sum_{(u,v)} \alpha_{u,v} \cdot PTHA^p_{u,v}$
17. $\forall (v \neq s) : \sum_{\rho} PTH^p_s \geq \max(1, \gamma \cdot |T(s)_v|)$

Figure 5.2: The ILP formulation for inferring the conserved evolutionary trajectory tree

Given an evolutionarily conserved alteration set $S$, CONETT’s ILP formulation in Figure 5.2 specifies the tree $T_{P,s}$ built upon the node set $S$, with $s$ as its root, that represents the maximum node depth tree in which each path represents a conserved evolutionary trajectory with respect to the value of $\gamma$. In the ILP formulation, binary variable $X_{u,v}$, defined for each pair of nodes $u, v \in S$, is set to 1 if $(u, v)$ is a tree edge of $T_{P,s}$, and set to 0 otherwise. For each tumor graph $G_p$ and node $v \in S$, variable $PTH^p_v$ is set to 1 if the exact path in $T_{P,s}$ leading from $s$ to $v$ is conserved is tumor graph $G_p$. Lastly, for each node $v \in S$, the variable $D_v$ represents the distance between $s$ and $v$ in $T_{P,s}$. The objective of the ILP is to maximize the sum of distances from the root to all nodes in $S$.

Now we describe each of the constraints in the ILP formulation. First, the root node $s$ is set to have distance 0 by constraint (1), and it cannot be a child of any other node due to constraint (2). Every other node $v$ has distance that is equal to the distance of its parent node $u$ increased by 1. The edge to node $v$ from its parent $u$ is determined by auxiliary variables $PD_{u,v}$: Due to the constraint (4), $PD_{u,v}$ is set to 0 if $u$ is not the parent of $v$; The maximization in the objective function, combined with the constraints (3) and (5) ensures that $PD_{u,v}$ is set to $D_u$ if $u$ is the parent of $v$ and that distance of $v$ is equal to the distance of its parent $u$ increased by 1.

To ensure that $T_{P,s}$ is a tree, we require that each node can have only one parent; this is ensured by constraint (6). We enforce that the nodes of $T_{P,s}$ form a connected component
of size $|S|$ by considering a fictitious network flow originating at root node $s$ of $|S| - 1$ units – by constraint (7). For each directed edge $(u, v)$, the value of flow along that edge is represented by variables $f(e_{u,v})$.

The flow loses 1 unit at each node - by constraint (8). If there is a positive amount of flow through the edge $(u, v)$, then it is a tree edge in $T_{P,s}$ – by constraint (9). If there is no flow through the edge $(u, v)$ then it is not in $T_{P,s}$ – by constraint (10).

The set of constraints (11–17) ensures that the evolutionary trajectories represented by paths in $T_{P,s}$ are conserved. Constraint (17) requires each trajectory (or path) from the root $s$ to a node $v$ in the tree to be conserved in at least $\gamma|T(s)_v|$ tumor graphs. Similar to the use of auxiliary variables $PD_{u,v}$ above, we introduce auxiliary variables $PTHA^p_{u,v}$, set to 1 if and only if there is a path from $s$ to $u$ in tumor graph $G_p$, and $(u, v)$ is a tree edge of $T_{P,s}$ – by constraints (13), (14) and (15). A tree path from $s$ to $v$ exists in a tumor graph $G_p$ if $G_p$ both includes the path from $s$ to $u$ as well as the tree edge from $u$ to $v$ - by constraint (16). Here, the indicator constant $\alpha_{u,v}$ takes the value 1 if the edge $(u, v) \in G_p$, and 0 otherwise. By default, the root node $s$ has a path in tumor graph $G_p$ if it is observed in the tumor $p$ – by constraints (11) and (12).

Note that only $X_{u,v}$ variables are required to be binary. The decision variables $PTH^p_v$ can be continuous, with the addition of constraints that they can not be larger than 1. This reduces the complexity of the model and the running time and space required to solve it. The resulting mixed integer linear programming (MILP) model then consists of $|S|^2$ binary variables and $O(|S|^2|P|)$ constraints.

### 5.1.2 Not transitive input graphs

Section 5.1.1 describes finding the evolutionarily conserved alteration set $S = \{ v \in A \mid T(s)_v \neq \emptyset \}$ in a prior step to building the maximum conserved evolutionary trajectory tree from a given root $s$. The transitivity of directed edges in the input tumor graphs makes this easy – we simply add to $S$ all nodes $v$ for which there exists a directed edge $(s, v)$ in at least one of the tumor graphs.

However, in some applications the user may wish to break the transitivity in a tumor graph $G_p$ so that while it includes directed edges $(u, w)$ and $(w, v)$, it does not have $(u, v)$. This could be desirable if there is strong belief/confidence that the intermediate node/alteration $w$ was necessary for the emergence of alteration $v$ in $G_p$. In such cases, the user may ensure that an evolutionary trajectory involving alterations $u$ and $v$ does not “skip” a high-confidence intermediate alteration $w$ by excluding the edge $(u, v)$. Note that this may lead to some nodes being excluded from the resulting consensus tree: e.g. given $u$ as the root, suppose that half of the tumor graphs in the cohort include the path $u, w, v$ and the other half include the directed edge $(u, v)$ but not the directed edge $(w, v)$. If $\gamma > 0.5$ then $v$ will not be included in the consensus tree.
Furthermore, the relaxation of the transitivity property makes the problem of finding the evolutionarily conserved alteration set non-trivial; this is due to the fact that in any $G_p$, not all nodes on the directed path from root $s$ to another node $v$ will necessarily be the children of $s$. We show how to address this issue by a preprocessing step described here. Let $S^*$ be the largest set of nodes such that for each node $v \in S^*$, there are at least $\gamma |\mathcal{T}(s)_v|$ graphs which include a path from $s$ to $v$ consisting only of nodes in $S^*$. If a node $v$ is in the resulting consensus tree, the path from $s$ to $v$ must consist only of nodes in $S^*$ (since $S^*$ is defined as the largest such set). This implies that a node that is not in $S^*$ cannot be in the consensus tree, implying that computing the set $S^*$ and pruning out all nodes not included in it can reduce the solution space. Given a node $s$, we formulate the maximum path-conserved subgraph identification problem whose solution is the set $S^*$ and describe a polynomial-time algorithm to compute it. Note that on an input set of directed graphs $\{G_i\} (i \in \mathcal{I})$, we use the notation $\mathcal{G} = (\cup_{i \in \mathcal{I}} V(G_i), \cup_{i \in \mathcal{I}} E(G_i))$ to denote the smallest supergraph of all graphs $G_i (i \in \mathcal{I})$ (so that each $G_i$ is a subgraph of $\mathcal{G}$).

**Maximum Path-Conserved Subgraph Identification problem (MPCSI):** Given $\{G_i\} (i \in \mathcal{I})$, their smallest supergraph $\mathcal{G}$, a root node $s \in \mathcal{G}$, and a number $t_v$ for each $v \in \mathcal{G}$, find the largest set of nodes $S^*$ of $\mathcal{G}$ such that for each node $v \in S^*$ there are at least $t_v$ graphs $G_i$ which include a path from $s$ to $v$ consisting only of nodes in $S^*$.

To solve MPCI problem for a given root $s$, CONETT uses an iterative algorithm which starts with $S^* = \{s\}$. In each iteration, the algorithm considers adding to $S^*$ a new node $v$ to which there is a directed edge from any node in $S^*$ in at least $t_v$ graphs. Node $v$ is only added to $S^*$ if there are at least $t_v$ graphs that include a path from $s$ to $v$ consisting only of nodes already in $S^*$. For finding out whether $v$ satisfies this condition, CONETT constructs a data structure $P(v)$ for each node $v \in \mathcal{G}$, which maintains the set of graphs that have a directed edge from any node in $S^*$ to $v$ in the form of a bitmap. Naturally $P(s)$ is initially equal to the set of all graphs where $s$ is present. For every other node $v$, $P(v)$ is initially equal to the set of tumor graphs that include edge $(s, v)$. The algorithm terminates if there is no $v \in (\mathcal{G} - S)$ such that $|P_v| \geq t_v$.

Otherwise it identifies any node $u \in (\mathcal{G} - S^*)$ such that $|P_u| \geq t_v$ and adds it to $S^*$. Then it checks for each node $v \in S^*$ for which $(u, v) \in E(\mathcal{G})$, whether the addition of $u$ to $S^*$ adds any new graphs to $P_v$. If this is the case, the algorithm inserts the new graphs to $P_v$ and recursively propagates the new graphs to any neighbour $w \in S^*$ of $v$ that does not already have them in $P_w$. Figure 5.3 illustrates the necessity of the recursive path propagation within $S^*$. Finally, the algorithm updates $P_z$ for every node $z \in (\mathcal{G} - S^*)$ for which there is an edge $(v, z) \in E(\mathcal{G})(v \in S^*)$.

The correctness of the algorithm follows from the invariant that at each iteration, the algorithm considers paths from $s$ to nodes that are either in $S^*$ or are neighbours of nodes
Figure 5.3: **Illustration of the importance of propagation of new paths throughout the subgraph during execution of the MPCSI-solving algorithm.** Boxes next to each node represent tumors (total of 5), and there is a dot in a box if the corresponding tumor contains the alteration in the node. 

A) Let $t = 3$. If we add nodes to $S$ in order $S \rightarrow A \rightarrow B \rightarrow C$, then $D$ cannot be added unless $C$ updates paths of $B$. Since the graph has a topological ordering, adding nodes in topological order would avoid the need for updating subgraph nodes’ paths, e.g. $S \rightarrow A \rightarrow C \rightarrow B \rightarrow D$.  

B) This graph contains multiple cycles and does not have a topological ordering of its nodes. Each newly added node has to update paths of every other subgraph node before they are pushed to outside neighbours of the subgraph.

in $S^*$, thereby ensuring the existence of at least $t_v$ paths (each in a separate graph). We also note that for any given $v \in S^*$, the total number of times that $P_v$ is updated by the algorithm can not be more than the number of graphs. Furthermore the total number of times the algorithm can visit $v$ is no more than the number of edges in $G$. As a result, the running time of the algorithm is $O(|V(G)| \cdot (|P| + |E(G)|))$.

### 5.1.3 Optional Constraints and Parameters

CONETT allows the user to add additional constraints to the ILP solution to the MCETT problem described in Section 5.1.1, via a new set of options/parameters as follows: 

1. CONETT allows the user to restrict the set of nodes which are included into the consensus tree to more recurrent alteration events that “follow” the root event in a specified minimum number of tumor graphs $t$. This is achieved by removing all nodes $v$ from the evolutionarily conserved alteration set $S$ for which $|T(s)_v| < t$, prior to building the tree.

2. CONETT also allows the user to add constraints on the ancestor-descendant ordering of nodes in conserved evolutionary trajectories of the resulting tree $T_{P,s}$. Specifically, it
calculates a “confidence score” of \( u \) being an ancestor of \( v \) based on evidence provided by all tumor graphs in which root \( s \) is an ancestor of \( u \) and \( v \) that do not have “don’t care” edges. Given a user-defined value \( \delta \), representing the threshold for confidence scores between each pair of nodes in the evolutionarily conserved alteration set \( S \), CONETT adds constraints to the ILP formulation that require each ancestor-descendant pair of nodes in the tree to have a confidence score at least \( \delta \).

CONETT’s confidence score for node \( u \) being an ancestor of node \( v \), penalizes those tumor graphs that include (i) a directed edge \((v,u)\) without a directed edge \((u,v)\) and (ii) an anti-edge between \( u \) and \( v \), while considering tumor graphs with a don’t care edge between \( u \) and \( v \) neutral. Specifically, let \( T(s)_{u,v} = T(s)_{u} \cap T(s)_{v} \) be the subset of all tumor graphs in which there exists a directed path from \( s \) to both \( u \) and \( v \) (note that no tumor graph in this set has a “don’t care” edge between \( u \) and \( v \)). Let \( P_{u,v} \subseteq T(s)_{u,v} \) be the subset of tumor graphs that contain a directed edge \((u,v)\) (the presence of directed edge \((v,u)\) does not make a difference). Also let \( R_{u,v} \subseteq T(s)_{u,v} \) be the set of tumor graphs that include the directed edge \((v,u)\) but not the edge \((u,v)\). Similarly, let \( Q_{u,v} \subseteq T(s)_{u,v} \) be the subset of tumor graphs which include an anti-edge between \( u \) and \( v \). Note that \( T(s)_{u,v} = P_{u,v} \cup R_{u,v} \cup Q_{u,v} \). Then the confidence score of \( u \) being an ancestor of \( v \) in the subset of tumor graphs \( T(s)_{u} \cup T(s)_{v} \) is set to be \( W_{u,v} = \frac{|P_{u,v}| - |R_{u,v}|}{|T(s)_{u,v}|} \). As can be seen below, this (as well as alternative) confidence score(s) penalizing (or incentivizing) specific edge types is easily incorporated in CONETT’s ILP formulation.

\[
\begin{align*}
\forall (u,v,w) & : PA_{u,v,w} \leq A_{u,w} \\
PA_{u,v,w} & \leq A_{w,v} \\
PA_{u,v,w} & \geq A_{u,w} + A_{w,v} - 1 \\
\forall (u,v) & : A_{u,v} \geq X_{u,v} \\
|S| \cdot A_{u,v} & \geq \sum_{w \in S} PA_{u,v,w} \\
A_{u,v} & \leq X_{u,v} + \sum_{w \in S} PA_{u,v,w} \\
\forall (u,v) \text{ s.t. } W_{u,v} < \delta : X_{u,v} = 0 \\
A_{u,v} & = 0
\end{align*}
\]

For each pair of nodes \( u \) and \( v \) in \( S \), we add a variable \( A_{u,v} \), which is set to 1 if \( u \) is an ancestor of \( v \) in \( T_{P,s} \), and 0 otherwise. Constraints (24) and (25) ensure that all ancestor-descendant pairs of nodes in \( T_{P,s} \) have a high confidence score. We introduce auxiliary variables \( PA_{u,v,w} \), set to 1 if \( u \) is an ancestor of \( w \) and \( w \) is an ancestor of \( v \) and set to 0

\(^4\)In the case that transitivity of directed edges in the tumor graphs is broken, then \( P_{u,v} \) represents the subset of tumor graphs in which there is a directed path from \( u \) to \( v \) (they may lie on a cycle). Then \( R_{u,v} \) represents the subset of tumor graphs in which there is a directed path from \( v \) to \( u \) but no directed path from \( u \) to \( v \) is present. Since, in general, anti-edges do not need to be explicitly specified, so long as don’t care edges are specified, we may consider \( Q_{u,v} \) to represent the subset of tumor graphs in which there is no directed path between either \( u \) to \( v \) or \( v \) to \( u \).
otherwise—by constraints (18), (19) and (20). A parent of a node is its ancestor by default, by constraint (21). Constraints (22) and (23) impose transitivity on ancestor-descendant relationships: \( u \) is an ancestor of \( v \) if and only if \( u \) is an ancestor of \( w \) and \( w \) is an ancestor of \( v \), for some intermediate node \( w \).

5.1.4 Polynomial-Time Algorithm for the (Unconstrained) Maximum Depth Spanning Tree Problem

As opposed to the constrained problem of constructing a spanning tree with the maximum total node depth, which we solve through an ILP formulation in Section 5.1.1, the Maximum Depth Spanning Tree Problem does not impose constraints (11)–(25). In its unconstrained form, the problem formulation is as follows:

**Maximum depth spanning tree (MDST):** Given an unweighted, directed graph \( S \) and a root node \( s \in S \), find a spanning tree with the maximum sum of node depths.

We solve the MDST problem via an incremental algorithm which starts with the node \( s \) and then expands the tree, one node at a time. We use a modified version of depth-first search (DFS) to add new nodes that do not already belong to the tree, and then check whether depth of any nodes that are already in the tree can be increased via the new node (Algorithms 4 and 5).

<table>
<thead>
<tr>
<th>Algorithm 4: Maximum Depth Spanning Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Unweighted graph ( S ), root node ( s )</td>
</tr>
<tr>
<td><strong>Output:</strong> Parent array ( P ), depth array ( D )</td>
</tr>
<tr>
<td><strong>Result:</strong> Constructs a spanning tree which maximizes the depth of each node</td>
</tr>
<tr>
<td>1 for ( i = 1, 2, \ldots,</td>
</tr>
<tr>
<td>2 ( P_i = -1 )</td>
</tr>
<tr>
<td>3 ( D_i = -1 )</td>
</tr>
<tr>
<td>4 ( A_i = false ); \hspace{1cm} // Flag for node being in the recursion stack</td>
</tr>
<tr>
<td>5 end</td>
</tr>
<tr>
<td>6 ( P_s = s )</td>
</tr>
<tr>
<td>7 ( D_s = 0 )</td>
</tr>
<tr>
<td>8 MDST-DFS(( s, P, D, A ))</td>
</tr>
<tr>
<td>9 return ( P, D )</td>
</tr>
</tbody>
</table>

The correctness and the running time bounds are provided by the following two claims.

**Claim 2.** Algorithm 4 maximizes the total node depth.

**Proof.** Let \( u \) be a newly added node at the current iteration of the DFS recursion. First, we show that in the current iteration node \( u \) has no better choice for a parent in the tree than the one that added it. If it did, then there would be an edge from another node in
**Algorithm 5: MDST-DFS**(\(u, S, P, D, A\))

**Input:** Unweighted graph \(S\), node \(u\), parent array \(P\), depth array \(D\), ancestor flag array \(A\)

**Result:** Constructs a spanning tree rooted in \(u\) which maximizes the total depth of nodes

1. \(A_u = true\); /* Ancestor of any new node that it adds */
2. \(U = \emptyset\)
3. for \((u,v) \in E(S)\) do
   4. if \((P_v > -1)\) and \((A_v == false)\) and \((D_v < D_u + 1)\) then
   5. \(U = U \cup \{v\}\)
   6. end
4. end
5. sort nodes in \(U\) by their depth, in increasing order
6. for \(i = 1, 2, ..., |U|\) do
   7. if \((D_{U_i} < D_u + 1)\) then
   8. updateDepthsInSubtree(\(U_i, S, P, D\)) ; /* Simple BFS */
   9. end
10. end
11. /* Next, discover new nodes */
12. for \((u,v) \in E(S)\) do
13. if \((P_v == -1)\) then
14. \(D_v = D_u + 1\)
15. \(P_v = u\)
16. \(A_v = true\)
17. MDST-DFS(\(s, P, D, A\))
18. end
19. \(A_u = false\); /* No longer ancestor */
20. return

the tree towards \(u\), and by nature of DFS, it would have already been added to the tree before. Hence this must be the first time that we have encountered node \(u\) and the current placement is the only possible one.

Next, we show that edge choices that the algorithm makes are optimal by proving the correctness of the update procedure. The proof follows via the DFS tree created by the order in which nodes are visited. Let \((u,v)\) be an edge considered by the algorithm. Note that forward and back edges can be safely ignored. If \((u,v)\) is a a forward edge, then \(v\) has already been processed by the recursion and \(D_v \geq D_u + 1\), so no change is necessary. If \((u,v)\) is a back edge, then \(v\) is an ancestor of \(u\), and including this edge would create a cycle. In order to break the cycle, some node \(w\) on path \(v \rightarrow \ldots \rightarrow u\) would have to be unlinked from its current parent and become child of an ancestor of \(v\) in order to keep the graph connected. Since any ancestor of \(v\) has depth that is smaller than depth of \(v\) by at

103
least 1, this edge swap at best keeps the depth of all nodes in the cycle the same as before. Let \((u, v)\) be a cross edge. If \(D_v \geq D_u + 1\), then the depth of node \(v\) and its descendants would not increase by changing its parent to \(u\). If \(D_v < D_u + 1\), then by unlinking \(v\) from its current parent and placing it as a child of \(u\), algorithm 5 would increase depths of \(v\) and all its descendants. However, since there might be an ancestor \(z\) of \(v\) that \(u\) also has an edge towards, we do not want to select \((u, v)\) as a tree edge, but \((u, z)\) instead. Since this is true for every such ancestor, it follows that updating nodes in topologically sorted order results in optimal depth and eliminates the need to worry about existence of such ancestors.

From the previous two paragraphs, it follows that at every iteration of the algorithm, the extension step correctly places node \(u\) in the tree, and correctly updates all remaining nodes’ parent edges. Therefore, at the last iteration, we will have an optimal tree. □

Claim 3. Running time of Algorithm 4 on a connected graph \(G = (V, E)\) is \(O(|V||E|)\).

Proof. The main DFS skeleton of the code traverses the graph in a depth-first manner, adding each node that is already not in the tree in total time \(O(|E| + f(n) \cdot |V|)\), where \(f(n)\) is the time needed to update the tree after each newly added node. In the worst case, the new node can cause depth update for each node in the tree. Each update needs to be pushed from the updated node down to its whole subtree. In order to avoid updating same nodes multiple times, we topologically sort the nodes and push updates starting with nodes that are closest to the root and doing the same for every subsequent node whose depth isn’t already updated by one of the previous nodes. Topological sort is carried out in time \(O(|V|)\) via counting sort, since we already have depth values of the nodes and they are integers smaller than \(|V|\). Since each node gets visited and updated at most once via normal BFS, total time that the update step takes is \(O(|V| + |E|)\). Since our graph is always connected, i.e. \(|E| = \Omega(|V|)\), the total running time of the algorithm is \(O(|V||E|)\). □

5.2 Assessment of the Statistical Significance of MPCSI Solutions

In order to assess the statistical significance of an evolutionarily conserved alteration set \(S\) computed by CONETT (or a solution to the MPCSI problem for graphs that are not transitive), we perform a variation of the standard permutation test on the input data. Let \(N_{a,p}\) represent the number of nodes (genes and chromosome arms) altered by alteration type \(a\) in tumor graph \(G_p\), and let \(F_{v,a}\) represent the number of tumors in which node \(v\) occurs altered by \(a\). As mentioned earlier, CONETT can be set to distinguish alteration types on individual genes or chromosomal arms. We have not treated distinct alteration types differentially in the algorithms described above; however, in case certain genes/chromosomal arms included in \(S\) are not strongly associated with a specific
by a in at least one tumor graph in the original input. The nodes are randomly drawn from a distribution based on their recurrence frequency, i.e. each node \( v \) is drawn with the probability \( \frac{F_{v,a}}{\sum_{u \in V(G)} F_{u,a}} \). That way, the average recurrence frequency of each node and alteration type in the permuted data is equal to the original recurrence frequency, preserving hidden inter-dependencies among somatic mutations and copy number alterations that were the cause of the observed recurrence frequencies and evolutionarily conserved alteration sets. Additionally, consistent with observations on the TRACERx ccRCC data set we use in this thesis, special care was devoted to ensuring that a copy number event affecting a whole chromosome does not appear together with a copy number event altering either of its arms within the same tumor. Lastly, the randomly chosen altered nodes replace the original nodes of the input tumor graph \( G_p \) after being subject to a random permutation.

The above process is repeated 1000 times. In order to estimate the statistical significance of a set \( S \) of size \( k \), obtained with root alteration event \( s \) that occurs altered in \( F_s \) tumors, we run the method on each generated permutation of the input data using the same parameters, and record the number of times that the resulting set has a size of \( k \) or greater when computed with a root that occurs altered in \( F_s \) or more tumors. That number, divided by the total number of experiments (1000 for our case), gives us the empirical p-value estimate for \( S \).

### 5.3 Data and Pre-Processing

#### 5.3.1 Clear-cell renal cell carcinoma (ccRCC)

We obtained ccRCC data from the TRACERx renal study [141], which gathered multi-region sequencing data from 100 tumors. The study collected non-silent mutation data in about 110 genes that are deemed to be high-confidence ccRCC driver genes, obtained through a sequencing panel from which single nucleotide variants (SNVs), dinucleotide variants (DNVs) and small insertion and deletions (INDELs) were inferred. The study also detected somatic copy number alterations (SCNAs), which were reported only on the level of chromosome arms or whole chromosome – provided at least 50% of the chromosomal arm had a copy number alteration. Then, in each tumor, the study clustered all the alterations and reconstructed a phylogenetic tree of clonal hierarchy.

For every somatically altered gene and copy number altered chromosomal arm that is found in a particular tumor’s phylogenetic tree, we create a node labeled by the name of the gene (or chromosome) and the type of alteration that affects it. Since, in this dataset, SNVs, DNVs and small insertion and deletion events result in inactivation of the genes that they affect, we treated them all as a single “inactivating” alteration type. Then, we build
Figure 5.4: **Recurrence frequency of altered genes and chromosome arms in TRACERx renal cohort.** The plot shows recurrence frequency of 137 inactivating sequence alteration (blue), 52 copy number loss (red) and 53 copy number gain (yellow) events. Events are sorted by their $y$ value along the $x$ axis for clarity. The $y$ axis is in log scale.

A directed graph on this set of nodes for a particular tumor, such that a directed edge is drawn from node $u$ to node $v$ if the clonal node in the phylogenetic tree which carries the alteration affecting node $u$ is an ancestor of the clonal node in the phylogenetic tree which carries the alteration affecting node $v$. In the case of alteration events $A$ and $B$ that belong to the same clone, we draw parallel directed edges, in opposite directions, between the nodes that are associated with $A$ and $B$. In 5 tumor trees, there was immediate branching right after the germline node. To ensure that there are no cross links in our tumor graphs (which can happen if a gene is present in both branches), leading to false precedence relationship between ancestors of the cross link source and descendants of the cross link destination, we split such trees at the root and treat them as two separate tumor phylogenetic trees.

As a result, we obtained tumor graphs from 105 “tumors” (after splitting the 5 trees with branching at germline node), containing 242 different nodes in total, representing driver genes affected by inactivating sequence alterations, and copy number gains and losses on chromosomes – both on whole chromosome and arm level. Figure 5.4 shows recurrence frequencies of the 242 events; the $y$ axis shows the recurrence frequency (in log scale), and events are sorted along the $x$ axis based on their $y$ values for clarity purposes. The figure reveals the sparsity of the data – more than a third of genes that carry sequence alterations each occur in only a single tumor, and more than a half occur in at most 2. Among more recurrently altered genes and chromosome arms (those that are found altered by the same...
alteration type in at least 10 tumors), there are 19 (35%) copy number gain-altered, 26 (50%) copy number loss-altered and 12 (9%) sequence-altered nodes.

5.3.2 Non-small-cell lung cancer (NSCLC)

We obtained data from 99 NSCLC patients from [19]. Originating from whole-exome sequencing of multiple spatially separated regions from the TRACERx lung cancer study [63], [19] provide cancer cell fraction (CCF) values, as well as clustering information, for single nucleotide variations and focal copy number alterations in 79 putative driver genes of NSCLC (we note that there was no case of a gene being affected by alterations of different types in any patient). Using CITUP [87], we inferred phylogenetic trees with the clustering information provided by [19], and for each patient we selected the tree with the minimum error score (which was always unique). We then construct tumor graphs from the phylogenetic trees similarly to the procedure used to generate tumor graphs in the ccRCC data (Section 5.3.1).

**Figure 5.5:** Recurrence frequency of altered genes and chromosome arms in TRACERx lung cohort. The plot shows recurrence frequency of 73 inactivating sequence alteration (blue) and 14 gene amplification (red) events. Events are sorted by their $y$ value along the $x$ axis for clarity. The $y$ axis is in log scale.

Contrary to the TRACERx ccRCC data, in which the clustering of alterations contained chromosome-level copy number alterations, the clustering of alterations did not contain copy number alterations on a chromosome level in this dataset. Since in [19], the analysis
was performed on the gene level, ignoring the specific alteration type (or having just one universal alteration type), we took a similar approach to make our results comparable by merging SNVs and gene deletions into a single “inactivating” alteration type. However, we kept gene amplifications as separate alteration type since the downstream effect significantly differs from SNVs and deletions.

Figure 5.5 shows recurrence frequencies of the 87 events; the y axis shows the recurrence frequency (in log scale), and events are sorted along the x axis based on their y values for clarity purposes.

### 5.4 Conserved evolutionary trajectories in ccRCC

We first used CONETT to identify statistically significant and highly-conserved alteration sets that consisted of highly-recurrent alteration events following the root event, and associated trajectories in the TRACERx renal study (see Section 5.3.1), by considering as a root node each gene and chromosomal arm that are found to be altered in at least 10 tumors. For each root, in this experiment we set the optional parameter \( t \) in the MPCSI formulation to a high fraction of the seed’s recurrence frequency (\( \geq 48\% \)). We empirically assessed the statistical significance of these alteration sets using methodology described in Section 5.2. From the evolutionarily conserved alteration sets, we constructed conserved evolutionary trajectory trees, setting the \( \gamma \) parameter in the MCETT formulation to \( \frac{1}{3} \) and the ancestor-descendant order confidence score threshold of \( \delta = 0.85 \) (see Section 5.1.3). Below we report on the 10 most significant evolutionarily conserved alteration sets.

Four of these sets contain 2 or 3 nodes, involving genes and copy number altered chromosome arms that are co-clonal (Figure 5.6E): these include sequence alterations on driver gene VHL and copy number loss on chromosome 3p, which are known to be founder alterations in ccRCC. Conserved evolutionary trajectories seeded either by sequence alterations on VHL or copy number loss on 3p (Figure 5.6A-B) both include sequence alterations in PBRM1 and copy number loss on chromosome 14q as highly-conserved follow-up events (p-value \( < 0.001 \)), with sequence alteration in PBRM1 typically occurring before copy number loss on 14q. Copy number gain on chromosome 5 (occurring in about a quarter of all tumors) is accompanied by co-clonal sequence alteration in VHL, and is followed by copy number loss on chromosome 14q and copy number gain on chromosome 7, each observed in about half of the tumors (p-value \( < 0.008 \), Figure 5.6C). The remaining three conserved trajectories are all seeded by the copy number loss on chromosome 6, occurring in 19 tumors (Figure 5.6D shows the largest trajectory tree). It is followed by losses on 14q and 9 at a high conservation rate of 70% (p-value \( < 0.001 \)), gains on chromosome 7 at 55% (p-value \( < 0.010 \)) and sequence alterations of VHL and loss on chromosome 4 at 50% (p-value \( < 0.001 \)). Loss on chromosome 4 follows loss of chromosome 9 in more than a third of the tumors.
Figure 5.6: Conserved evolutionary trajectories identified through CONETT in renal cancer. The numerical value in each node is the total number of tumors in which the root event is an ancestor of the node. Next to each edge, the number of tumors that the exact evolutionary trajectory ending with that edge is conserved in is displayed. Each undirected edge (orange) marks a pair of co-clonal alterations that cannot be temporally ordered. **A, B, C, D)** Conserved evolutionary trajectory trees constructed for highly-conserved alteration sets. **E)** Statistically significant highly-conserved, co-clonal sets, from which it is not possible to reconstruct evolutionary trajectories with a known temporal order of nodes. **F)** Select trajectories from conserved evolutionary trajectory trees seeded by the germline node, \( VHL \) and loss on chromosome 3p.
Next, we looked into trees of trajectories seeded by the most recurrent events in the data set. As sequence alterations in VHL (found in 77% tumors) and copy number loss on chromosome 3p (found in 89% tumors) play a key role in initiation of ccRCC, we examined the two evolutionarily conserved alteration sets and conserved evolutionary trajectories that are seeded with either of these events and contain nodes that are altered in at least 10 tumors each ($t = 10$, both of the sets have p-value $< 0.001$). We also examined conserved evolutionary trajectories that start with the germline node, with same value of $t$, as it is present in the whole data set. The resulting trees of trajectories are shown on Figures 5.7, 5.8, and 5.9; while Figure 5.6F shows a number of select trajectories from those trees.

Consistent with the trees reported in the original TRACERx renal study, the trajectories obtained by CONETT include a series of sequence alteration events $VHL \rightarrow PBRM1 \rightarrow SETD2$ observed in 14 tumors, as well as $VHL \rightarrow BAP1$ observed in 15 tumors. Additionally, CONETT also detected two conserved evolutionary trajectories terminating with a sequence alteration in gene KDM5C that collectively cover 80% of tumors where it is present: $VHL \& \text{loss}_3p \rightarrow \text{gain}_5q \rightarrow KDM5C$ in 8 tumors and $VHL \rightarrow PBRM1 \rightarrow KDM5C$ in 9 tumors; these trajectories were not reported in the original study. Note that KDM5C is a chromatin modifier, similar to BAP1; its dysregulation alters the activity of many other genes [56]. Additionally, $VHL \rightarrow PBRM1 \rightarrow TSC1$ is found in 9 tumors, representing 75% tumors where $TSC1$ is present. CONETT also identified a number of less frequently altered genes that do not display high confidence temporal ordering. These alteration events in genes such as $MTOR$ (in 12/17 tumors), $OBSCN$ (in 7/12 tumors), and $MUC16$ (in 10/16 tumors) all seem to branch out from an alteration in $VHL$.

Inactivations of both $PBRM1$ and $BAP1$ are known to rarely co-occur in ccRCC [107, 141], and trees of conserved evolutionary trajectories on Figures 5.7, 5.8 and 5.9 always place them into different branches. However, most of the time when they do co-occur, $PBRM1$ alteration precedes $BAP1$ alteration. We thus ran CONETT using $PBRM1$ alteration as the seed event, to find out whether $BAP1$ sequence alteration appears as a conserved follow-up event, using $t = 10$ (p-value $< 0.001$). The resulting tree of conserved evolutionary trajectories is shown on Figure 5.10. Alteration on $BAP1$ was not identified as a downstream event, strengthening the hypothesis that their co-occurrence is most likely due to upstream events such as loss on 3p and inactivation of $VHL$.

Loss on chromosome 14q is the second most recurrent large-scale copy number loss event (after loss on 3p), occurring in 50 tumors. However, contrary to loss on 3p which regularly appears as the earliest alteration event in tumors, loss on 14q is always placed later, after both $VHL$ and loss on 3p. It is often preceded by inactivation of $PBRM1$, although not in every single tumor (Figure 5.10 shows that loss on 14q follows inactivation of $PBRM1$ in 22 tumors). There is also strong evidence of precedence of gains on chromosomes 5 and 5q (which are themselves mutually exclusive) to loss on chromosome 14q – trajectories seeded by loss on 14q do not detect gains on chromosome 5 or 5q as downstream events (Figure
5.11), but trajectories seeded by gains on 5 and 5q do detect loss on 14q as a downstream event (Figure 5.12). Additionally, Figure 5.12A shows presence of mutual exclusivity of losses on chromosome 3p and 14q in tumors that contain gain on chromosome 5, whereas that relationship is not noticed in tumors that contain gain on 5q (Figure 5.12B). This pattern of mutual exclusivity was not reported in the original ccRCC TRACERx study.
Figure 5.7: VHL-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC.
Figure 5.8: loss3p-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC.
Figure 5.9: Germline-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC.
Figure 5.10: PBRM1-seeded tree of trajectories with $t = 10$ in TRACERx ccRCC.

Figure 5.11: loss14q-seeded tree of trajectories with $t = 11$ in TRACERx ccRCC.
5.5 Conserved evolutionary trajectories in NSCLC

We used CONETT to identify conserved evolutionary trajectories in the TRACERx non-small-cell lung cancer study (see Section 5.3.2). This data was analyzed by REVOLVER [19], allowing us to compare the CONETT-identified trajectories with the edges in clusters identified by their method. Since REVOLVER uses “collective” information from the whole set of tumors to infer individual tumor phylogenetic trees, in order to make our results more comparable we set the value for the optional parameter $\delta = 1$ – ensuring that the temporal order of ancestor-descendant pairs of nodes in our trees is conflict-free with not just those tumors in which the specific trajectory that they lie on is conserved, but also the wider set of tumors in which the root is ancestor of both the nodes (please see Section 5.1.3 for exact details). We used the same value for the parameter $\gamma = \frac{1}{3}$ as in the analysis of the ccRCC dataset.

In [19], the authors highlighted evolutionary trajectories constructed from edges in cluster C5 (Figure 5.13B). Using CONETT, with the SNV-altered gene $CDKN2A$ as the seed, we were able to recover the full topological order of the genes in cluster C5 (Figure 5.13A), as well as additional ancestor-descendant relationship between amplification on $TERT$ and SNVs on $FAT1$ and $NOTCH1$. The CONETT-identified tree captures more occurrences of tree edges $CDKN2A \rightarrow TP53$ and $TP53 \rightarrow TERT$, as well as edges from $TP53$ towards $NOTCH1$ and $COL5A2$, which were tree edges in cluster C5 but are just forward edges in CONETT (dashed gray lines); due to CONETT trajectories capturing additional ancestor-descendant relationship between amplification of $TERT$ and SNVs on sibling nodes $FAT1$ and $NOTCH1$.

Additionally, we used CONETT to construct trees of trajectories with known early drivers (as well as the germline node) as seeds (Figures 5.14, 5.15). Figure 5.13C shows select conserved trajectories which are supported by both trees. The trajectories capture $TP53 \rightarrow SOX2 \& PIK3CA$ amplifications $\rightarrow NFE2L2$ (in 4/9 tumors), as well as $TP53 \rightarrow KEAP1$ (in 6/12 tumors) – inactivations of $NFE2L2$ and $KEAP1$ may be linked to increased chemoresistance [64]. All $TP53$-seeded evolutionarily conserved alteration sets were found to be statistically significant (p-value < 0.001). Interestingly, trajectories seeded by SNV and amplification of $EGFR$ did not pick up any node other than $TP53$, which they are often found in the same cluster with (Figure 5.13D). Even when we merge amplifications and SNVs of $EGFR$, $RB1$ and $PIK3CA$ just barely emerge as downstream altered genes. $KRAS$-seeded trajectories capture $MGA$ and $KMT2D$, consistent with the germline-seeded tree (Figure 5.14).
Figure 5.12: gain5 and gain5q seeded trees of trajectories with $t = 10$ in TRACERx ccRCC.

Figure 5.13: Conserved evolutionary trajectories identified through CONETT in lung cancer. The numerical value in each node is the total number of tumors in which the root event is an ancestor of the node. Next to each edge, the number of tumors that the exact evolutionary trajectory ending with that edge is conserved in is displayed. A) CONETT-constructed tree of conserved evolutionary trajectories rooted by an SNV on the gene CDKN2A ($t = 3$). It captures all the genes shown in the likely tree of cluster C5, selected by REVOLVER (shown in B). It captures more occurrences of the genes and the edges and it shows additional ordering of the alterations. B) A likely evolutionary tree drawn from the edges in cluster C5, identified by REVOLVER. Edge labels show the number of occurrences of the edge within the cluster. C) Select trajectories identified by CONETT which have a TP53 SNV as the clonal event. D) Trajectories rooted by recurrent NSCLC drivers KRAS and EGFR. Rightmost tree has EGFR amplifications and SNVs merged together.
Figure 5.14: Germline seeded tree of trajectories with $t = 5$ in TRACERx NSCLC.
5.6 Discussion And Future Work

In this chapter of the thesis, CONETT, a novel computational method for combinatorial detection of conserved evolutionary trajectories for tumors was presented. CONETT is the first such method that goes beyond just identifying clusters of common edges and directly produces a consensus phylogeny which aims to maximize the number of temporally-ordered pairs of alteration events. Given a root alteration event, we show how to construct a conserved evolutionary trajectory tree consisting of the maximum set of events that can be found on conserved evolutionary trajectories from the root via an ILP formulation. We applied our method to recently published datasets of 100 ccRCC tumors and 99 NSCLC tumors, and identified conserved evolutionary trajectories of sequence-altered genes and copy number altered chromosomal arms. CONETT identified all conserved evolutionary trajectories involving sequence altered genes reported as significant by the original studies; it also identified several additional conserved trajectories which were not reported earlier.

As per the original studies of TRACERx renal and lung cancer data, we have not differentiated sequence-alteration types and have treated all sequence-altered genes identically; this allowed us to compare CONETT results with those of the original studies. However CONETT’s framework is sufficiently general to offer the user the ability to differentiate somatic alteration types in the trajectories it identifies so as to perform in depth analysis of larger and richer data sets to be published in the near future.

We acknowledge that the consensus tree that CONETT’s ILP formulation currently constructs represents a single tree topology in which each node occurs only once. That forces the method to select only a single trajectory from a given seed towards each node, even though there might be multiple different trajectories that each occur in a significant fraction of tumors (as evidenced by the two different trajectories towards KDM5C discovered by CONETT and shown on Figures 5.7 and 5.9, together covering 80% of tumors in which the gene is sequence-altered). In future, it would be useful to devise a way to simultaneously identify different conserved trajectories in tumor evolution towards the same node.

As larger and more precise data sets emerge, as well as depending on the motivation for and the specific application of the analysis, it might become desirable to use an alternative version of tumor graphs which do not have the property of transitivity of the directed edges. This could be desirable if there is strong belief that an ancestor event \( u \) was necessary for the emergence of another follow-up event \( v \). In such cases, removing transitive edges towards \( v \) would ensure that detected evolutionary trajectories involving \( v \) would not be able to “skip” the high-confidence intermediate alteration event \( u \). Our method can incorporate such data into its framework as is; we also give a polynomial time algorithm to detect a set of events that is a superset of the nodes in the resulting tree – allowing to prune the search space and reduce the running time and space of the ILP model.
Additionally, we introduced an optional scoring scheme which measures the confidence of ancestry for a pair of nodes based on the evidence given by the temporal order of the two events in the input tumor data. Currently, our method uses a common threshold for all possible ancestor-descendant pairs of nodes, imposing a uniform level of stringency. It is worth exploring how to impose a level of stringency specific to each individual pair of nodes on evidence available for their partial ordering in the future.
Figure 5.15: TP53 seeded tree of trajectories with $t = 5$ in TRACERx NSCLC.
Chapter 6

Summary

Over time, cells in the human body accumulate alterations in their DNA, which is continuously damaged by both internal and external factors. This occurs even in healthy cells, but most of that DNA damage gets repaired through cellular DNA repair mechanisms. However, sometimes a small fraction of the alterations might remain and accumulate over time. In case of irreparable DNA damage in otherwise healthy cells, the process of cell death is initiated and new cells take their place. However, at times, a cell can acquire a sufficient set of alterations that give it enough of an advantage to proliferate unhindered, turning it into a cancer cell. Such cells form malignant growths, tumors, with a serious negative impact on the health of the patient in which this occurs.

Over the past decade, high-throughput sequencing efforts have provided an unprecedented opportunity to identify genomic alterations in tumors. However, the available data shows that there is a great amount of heterogeneity in alteration profiles of tumors, even those that belong to the same cancer type, or even different tumor sites within the same patient. This heterogeneity is at least partly explained by viewing cancer as a disease of cellular pathways, rather than a disease of individual genes. In order to gain selective advantage that it needs to divide uncontrollably, a cancer cell may only need one (or a few) of its important cellular pathways to be perturbed in any sufficient manner, rather than always have the same genes altered. Thus, in different tumors, the exact manner in which the cell arrives at this end goal may differ.

The rate at which alterations are acquired is higher in tumor cells than in healthy cells, typically as a consequence of disabled DNA repair mechanisms. However, not all alterations are important for the progression of tumor. With respect to the significance of an alteration for the tumor progression, we divide all alterations into two groups: driver alterations and passenger alterations. Driver alterations are those whose presence gives the cell the selective advantage that it needs to divide unhindered. Passenger alterations are largely inconsequential and typically arise as a consequence of the driver alterations. Similarly, we call genes that harbor driver (passenger) alterations in tumors driver genes (passenger genes). Once identified, the driver alterations and genes can serve as therapeutic
targets, or they can be used in diagnosis as potential markers of cancer subtypes; allowing clinicians to guide the patient towards a treatment with a more likely positive outcome. However, it is very difficult to determine all drivers of cancer. Due to tumor heterogeneity, it is very difficult to identify common patterns in alteration profiles; driver alterations are greatly outnumbered by passenger alterations; and certain drivers may be subtype specific or playing a crucial role at a later stage in tumor progression (perhaps during and after treatment), thus being present in only a small fraction of tumors. This task is made somewhat easier through the use of multi-region, single cell or time series sequencing data, as that allows construction of phylogenetic trees of likely tumor evolution, which makes the identification of early drivers more straightforward. However, such data is still not widely available and also suffers from heterogeneity both regarding the alterations that are found in different tumors, as well as regarding the exact way in which the tumor evolves (phylogenetic tree structure).

This thesis presents three computational methods that address areas not covered by the existing tools in the analysis of tumors. The first method, HIT’nDRIVE, aims to identify a subset of sequence-altered genes that is most likely to have influenced observed changes in the expression profile of expression-outlier genes in a cancer patient cohort. A novel distance measure is introduced, the multi-hitting time, which represents a notion of a collective distance of an outlier gene from a subset of sequence-altered genes in a gene interaction network. Through multi-hitting time, this thesis defines the Random Walk Facility Location problem and solves it through a reduction to the Weighted Multi-Set Cover problem. Applied to 2200 tumors (from TCGA), HIT’nDRIVE revealed many potentially clinically actionable driver genes and demonstrated its robustness in selecting cancer-implicated genes. HIT’nDRIVE was able to identify, as potential drivers, genes that have been found altered in only a small fraction of the analyzed cohort, demonstrating its potential to identify subtype-specific drivers. In an attempt to enhance means of quantitative validation of potential drivers, this thesis explores the ability of cancer drivers to improve phenotype classification based on gene expression data, with encouraging results. In a select part of that analysis, the highly-successful ability of likely drivers to correctly classify drug-resistant and drug-responsive samples in cancer cell lines has been demonstrated.

The second method, cd-CAP, has been developed with the aim to detect regular patterns in alteration profiles of tumors. A notion of a conserved alteration pattern is introduced as a recurring subnetwork of functionally-interacting genes such that the exact alteration type affecting a gene in the subnetwork is conserved across a large fraction of tumors in the analyzed cohort. An association rule mining like approach has been given in order to identify large conserved alteration patterns, as well as a maximum-coverage formulation in order to simultaneously identify multiple conserved alteration patterns that cover as many alterations as possible. cd-CAP was used to identify highly-recurrent conserved alteration patterns in three major cancer types (from TCGA), in form of functionally-interacting
sequence-altered genes, or in the form of likely downstream footprint of driver activity through functionally-interacting expression-outlier genes. Many of the identified conserved alteration patterns have a high recurrence and show significant association with the patient survival outcome.

The third method, CONETT, has been developed for use in the analysis of tumor evolution data, and is the first method to offer automated computational detection of conserved evolutionary trajectories in tumor evolution. CONETT constructs consensus phylogenies that maximize the number of temporally-ordered pairs of alteration events and whose structure contains conserved evolutionary trajectories – in form of paths from the root of the tree to a node. CONETT has been applied to two published multi-region targeted sequencing datasets from the TRACERx project (with data sequenced from renal and lung cancer patients), of about 100 tumors each. The analysis of the data with CONETT was able to capture all results presented by the original publications, and it also revealed several conserved evolutionary trajectories that were not reported in the original studies.

The analysis reported in this thesis has demonstrated the ability of the presented methods to find rare personalized cancer drivers and identify recurrent patterns of alteration events in cross-sectional sequencing data and repeated trajectories in tumor evolution. With all three tools being open-source and available to the public, it is the authors’ hope that these tools will be used to augment existing tumor analysis frameworks and lead to new insights into regularities in the way cancer forms and evolves.
Bibliography


[21] Brett S Carver, Jennifer Tran, Anuradha Gopalan, Zhenbang Chen, Safa Shaikh, Arkaitz Carracedo, Andrea Alimonti, Caterina Nardella, Shohreh Varmeh, Peter T.


[50] Catherine S. Grasso, Yi-Mi Wu, Dan R. Robinson, Xuhong Cao, Saravana M. Dhanasekaran, Amjad P. Khan, Michael J. Quist, Xiaojun Jing, Robert J. Lonigro, J. Chad Brenner, Irfan A. Asangani, Bushra Ateeq, Sang Y. Chun, Javed Siddiqui, Lee Sam, Matt Anstett, Rohit Mehra, John R. Prensner, Nallasivam Palanisamy,


[79] Michael S. Lawrence, Petar Stojanov, Paz Polak, Gregory V. Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L. Carter, Chip Stewart, Craig H. Mermel, Steven A.


