Contents

Chapter 1

Introduction

1.1 Drug Discovery

1.2 TFmir3

1.3 Connectivity Map

1.4 Data

1.5 Workflow

Chapter 2

Background

2.1 Related Work

2.2 Data Soruces

This section provides an overview of the RNA-Seq and drug data used in our work. It includes datasets for multiple cancer types, with matching cell lines in the CMap. In the subsequent subsections, we outline the primary sources of our data: section 2.2.1 highlights the Cancer Genome Atlas Program (TCGA), section 2.2.2 introduces the National Cancer Institute and section 2.2.3 describes the Therapeutic Target Database (TTD).

2.2.1 Cancer Genome Atlas Program

The Cancer Genome Atlas Program (TCGA), is a cancer genomics program, initiated in 2006 and completed in 2018, led by National Institute of Health (NCI) and National Human Genome Research Institute. Over the 12 years, TXGA molecular analyzed more 20,000 cancer and normal samples from 33 cancer types. The project has generated more than 2.5 petabytes of genomic, transcriptomic, epigenomic, and proteomic information, which is freely accessible to researchers. This resource was used to download the raw mRNA and miRNA read count data for multiple cancer types, with corresponding cell-line in the CMap database. We accessed these datasets using the R package TCGAbiolinks, and it can also be accessed through a web-interface as well https://portal.gdc.cancer.gov/.

2.2.2 National Cancer Institute

The National Cancer Institue (NCI), established in 1937 as a part of National Institute of Health (NIH), plays an important role in cancer research, focusing on prevention, diagnosis, and treatment. It provides comprehensive information regarding different types of cancer treatments like precision medicine, targeted therapy, chemotherapy, and immunotherapy. In this work, we utilized NCI as a primary source of FDA-approved drugs.

2.2.3 Therapeutic Target Database

The Therapeutic Target Database (TTD), is a free bioinformatics resource, developed through collaboration between the Innovative Drug Research and Bioinformatics (IDRB) team in China and the Bioinformatics and Drug Design (BIDD) in Singapore. It contains comprehensive information regarding drug targets, associated diseases, pathways, and therapeutic protein and nucleic acid targets documented in the literature. In this work, we used TTD as secondary resource for FDA-approved drugs, complementing the data obtained from the NCI.

2.3 Differential Expression Analysis

Differential gene expression analysis is done to identify and quantify changes in the gene expression levels between different experimental conditions such disease vs. healthy tissues. The input data is a matrix of normalized read counts, where each value corresponds to the number of reads mapped to a specific gene for a given sample. These counts are modeled using statistical distributions such as negative binomial, to capture variability that might be present in the data. The result of the analysis is a list of genes with associated p-values indicating statistical significance and log2 fold changes, which describe the magnitude and direction of expression changes. Fold changes is the ratio of expression level changes between conditions, is an important metric in the analysis. The values are log-transformed, with positive values indicating up-regulation and negative values indicating down-regulation. We performed differential analysis between normal and tumor samples, those that have corresponding cell-line in CMap.

In section 2.3.1, we first explain how the data is filtered and normalized. Then, in section 2.3.2, we describe how the count data is modeled. Lastly, 2.3.3 provides an overview of the R package DESeq2, which is used for differential expression analysis in this work.

2.3.1 Pre-processing and Normalization

Before performing differential expression analysis with DESeq2, it is essential to pre-process the raw count data to obtain accurate results properly. This step usually involves the removal of low-quality data and normalizing biases that could interfere with gene expression between conditions. The initial step in pre-processing involves raw filtering, which removes genes that have low counts across the samples. It is necessary to remove low-count genes as they would not provide any meaningful information in further downstream analysis. Typically, genes that have fewer than 10 counts across samples are discarded.

After raw filtering, normalization accounts for differences in sequencing depth and library size between samples. DESeq2 uses the median-of-ratios method to normalize count data. The median-of-ratios is like TMM; however, it is more robust. It calculates the size factors for each sample. For given a gene i in a sample j, its normalized count N_{ij} is calculated by dividing the raw count K_{ij} by the total count of the sample:

$$N_{ij} = \frac{K_{ij}}{\sum_{g=1}^{G} K_{gj}}$$

Subsequently the count is averaged across all samples in each condition c, and the median ratio of the average normalized counts is taken between the conditions. This gives us the size factor which is used to normalize the original raw counts for each gene in a sample.

$$\overline{N_{i,c}} = \frac{1}{n_c} \sum_{j \in c} N_{ij}$$

$$S_j = \text{Median}\left(\left(\prod_{k=1}^{n} K_{ik} \right)^{\frac{1}{n}} K_{ij} \right)$$

$$K_{ij}^{\text{norm}} = \frac{K_{ij}}{S_j}$$

In DESeq2, the size factors are automatically calculated using the *estimateSizeFactors()* function, which by default uses the median-of-ratios method for normalization.

2.3.2 Modeling of Count Data and Dispersion Estimation

To handle overdispersion, where variance exceeds the mean, DESeq2 employs a negative binomial distribution, similar to edgeR, to model count data. Overdispersion is commonly observed in RNA-Seq data due to various factors such as unaccounted technical noise or heterogeneity in the data. The negative binomial distribution can be expressed as follows:

$$K_{ij} \sim \text{NB}(S_j * q_{ij}, \alpha_i)$$

Where K_{ij} is read counts for a given gene i in each sample j that follows a negative binomial with gene's mean expression value and dispersion parameter. The mean expression of the gene is given by the product of sample's size factor S_j and relative abundance of gene in a sample q_{ij}. The dispersion term $alpha_i$ accounts for variability in gene expression of a given gene i beyond the Poisson assumption.

2.3.3 DESeq2

The R package DESeq2, available on Bioconductor, is an extensively used differential expression analysis tool specifically designed to analyze RNA-seq data, utilizing the generalized linear model (GLM) approach. The method models count data using the negative binomial distribution, effectively addressing overdispersion. It also incorporates empirical Bayes shrinkage for dispersion estimation. Normalization is achieved through the robust median-of-ratios method, to ensure consistent library size between samples.

The initial step in the analysis is the preparation of the count matrix using the *DESeqDataSetFromMatrix()* function if the input is a raw count matrix. The function creates a dataset object by incorporating raw count and associated metadata, such as sample conditions, which are essential for experimental design. Alternatively, if the data is available as a SummarizedExperiment object, the *DESeqDataSet()* function can be used. Once the data set object is created, we apply the *DESeq()* function to perform the core steps of the pipeline - normalization, dispersion estimation, and model fitting - in a single step.

Normalization of the data, through median-of-ratios, is implemented internally in the *DESeq()* function. However, this step can also be performed separately using the *estimateSizeFactors()* function mentioned in section 2.3.1. Then it will use the calculated size factor values for further analysis. Similarly, even though dispersion estimation is implemented within the DESeq() function, *estimateDispersions()* can be used to calculate the dispersion in the data. The differential expression analysis in DESeq2 uses GLM of the form:

$$K_{ij} \sim \mathrm{NB}(\ S_j * q_{ij}, \alpha_i)$$

$$\log_2(q_{ij}) = \mathbf{x}_j \cdot \boldsymbol{\beta}_i$$

The coefficient β_i represents the log2 fold changes in expression for each gene i across different conditions or samples.

Once the data is normalized and the model is fit, for hypothesis testing, DESeq2, by default uses the Wald Test to determine if there are statistically significant differences in expression between two conditions. This evaluates the null hypothesis $H_0 : \beta_i = 0$ against the alternative $H_a : \beta_i neq 0$. The test statistics for each gene is calculated as:

$$W = \frac{\hat{\beta}_i}{\mathrm{SE}(\hat{\beta}_i)}$$

Where $\hat{\beta}_i$ represents the calculated log fold 2 change, while SE is its standard error. A p-value is generated for each gene based on its test statistic. Since many genes are tested simultaneously, the p-values are adjusted for multiple comparisons; corrections such Benjamini-Hochberg procedure are applied to manage the risk of false discovery rate.

2.4 TFmiR3

2.4.1 Gene Regulatory Network Construction

2.4.2 Identification of Key Network Genes/Nodes

2.5 Connectivity Map

The Connectivity Map (CMap) is an online resource developed to generate hypotheses about the relationship between genes, diseases, and therapeutics. This resource represents all biological states - whether physiological, pathological, or induced by genetic or chemical perturbations – through genomic signatures. It is a comprehensive database of over one million gene expression profiles developed by treating different cell lines with small molecules or perturbagens under controlled conditions. These signatures are generated using the L1000 assay, a high-throughput technology that measures changes in gene expression when treated with a certain perturbagen. To complement its database, CMap provides analytical tools like QueryApp, which compares user-provided expression signatures against the reference profiles in the database. It can be accessed freely at https://clue.io/ and can also be accessed through an API.

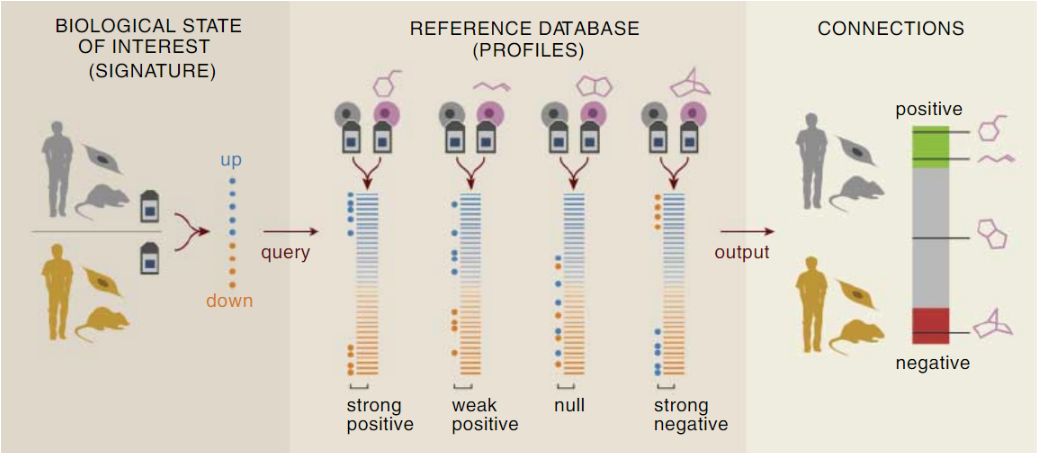

Figure 2.1: CMap

In our analysis, we used the Query App, figure 2.1, which allows users to compare a biological state of interest – characterized by a list of differentially expressed genes (DEGs) and their direction of expression change – with its reference profile, identifying compounds that have the ability to reverse the disease-specific gene expression. The compounds are ranked according to their "connectivity scores" ranging from 100 to -100, which indicates how closely it matches or oppose the user's query. A positive score suggests similarity, that is the compound produces effects similar to the input query, while a negative score indicates an opposing effect, suggesting therapeutic potential. Compounds with scores 95 or -95 are considered as highly relevant for further investigation. These scores provide a foundation for generating therapeutic hypotheses for a biological state of interest.

The associated data and L1000 assay are further explained in subsection 2.5.1 and the calculation of the connectivity score of the compounds is explained in subsection 2.5.2.

2.5.1 Associated Data and L1000 Assay

The CMap gene expression data is organized into a matrix format, where each column corresponds to an experiment involving a perturbagen treatment applied to a specific cell type while each row represents gene expression level across the experiment. Each value in the matrix represents a z-score, which reflects the expression change of a specific gene compared to its median expression across all other wells in the experiment. A column of that matrix corresponds to a gene signature, which is essentially the gene expression generated by applying a specific perturbagen to a particular cell line. Not all experiments in the CMap database hold the same relevance depending on the specific research goal. For instance, studies that focus on understanding the molecular mechanisms, it's more convenient to investigate compounds that have detailed annotations regarding their mechanism of action and targets. To support this, CMap has subdivided its data into the Touchstone and Discover datasets. We utilized the Touchstone dataset in this work and is also the primary resource used in the CMap Query App. This dataset is well annotated, which, as of the latest version, contains 81,979 perturbagens, out of which 33,609 are small molecule compounds measured in nine core human cell lines. This results in more than one million signatures. In contrast, the Discover dataset consists of over 15,000 unannotated small molecules perturbagens and can be used to discover new compounds with therapeutic potential and generate hypotheses.

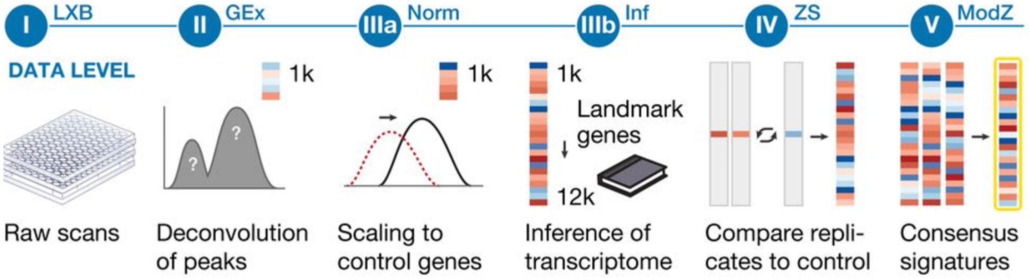

Figure 2.2: L1000 Assay Method

The CMap data, gene expression profiles, is generated by the L1000 assay, a high-throughput technology that measures the abundance of 978 'landmark' genes in human cells. The assay then infers the expression of the remaining 11,350 genes, by comparing the landmark genes to the RNA-Seq profiles from the GTEx program, which contain expression profiles from more than 100 tissues. It is a cost-effective and efficient technology designed to capture gene expression changes in response to perturbagens. The figure 2.2 explains how the assay generates the gene signature. It is a five-step data processing pipeline for the generation of CMap gene expression profiles. It starts with raw intensity measurement, which is then deconvoluted to gene expression levels in the next step. In Step 3, these values are normalized using control genes. In the last two steps, the expression of additional genes is also inferred in this step. This data is then converted to z-score profiles to quantify differential expressions, and the replicates are collapsed into a consensus signature.

2.5.2 Connectivity Scores

Chapter 3

Materials and Methods

3.1 Dataset and Preproccessing

The harmonized Transcriptomic Profiling data covering gene and miRNA expression quantification for multiple cancer entities was obtained. The datasets were sourced from TCGA and included data from breast, lung, prostate, colon, kidney and liver cancer. The data was retrieved using the R package TCGAbiolinks, which provides access to raw HTSeq and BCGSC read counts. The following table summarizes the cancer types included in the study, their respective TCGA project code, and the number of samples used for analysis:

Column 1	Column 2	Column 3	Column 4
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3	Row 1, Col 4
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3	
Row 3, Col 1	Row 3, Col 2	Row 3, Col 3	

Table 3.1: TCGA

(Mention skin cancer, and also mirnas table alongwith the number of genes table after filtering) To ensure that non-coding and irrelevant RNAs are not included the gene expression metrics were filtered to include only the protein coding genes. This was achieved by extracting gene information from the SummarizedExperiment object and subsequently sub-setting the count matrix based on the gene type attribute. To reduce noise by removing uninformative RNAs, features with total counts less than 10 across all samples were removed. By doing this filtering step, we made sure that the focus was on genes that had sufficient read coverage for reliable statistical analysis. DESeq2 internally normalizes the data using the estimateSizeFactors function, which takes into account the differences in sequencing depth and library sizes. The function uses a median-of-ratios method to account for variability in the data while also preserving the biological differences.

3.2 Differential Expression Analysis

In previous studies, Mark Bauer used two methods, edgeR and limma, while Johanna Strauß only used edgeR for differential expression analysis. We used DESeq2 for differential expression analysis due to its ability to effectively handle datasets with varying library sizes and sequencing depths. DESeq2's generalized linear model (GLM) used negative binomial distribution which makes the modelling of RNA-Seq data more robust, specifically when the sample size is large and complex experimental design. Furthermore, DESeq2 has a built-in normalization and filtering methods, making it ideal for our multi-cancer analysis. Differential expression analysis was performed between normal and tumor samples, this information was retrieved from the clinical metadata of the summarizedExperiment object. The DESeq object was created using the un-normalized counts matrix along with its associated clinical metadata. DESeq function was used to estimate size factors, dispersion, and genralized linear model through the integration of the estimateSizeFactors, estimateDispersion and nbinomWaldTest functions within a single workflow. For assessing the statistically significant gene expression differences between the conditions, Wald Test was used. The final p-values were adjusted for multiple testing using the Benjamin-Hochberg method, controlling the false discovery rate. We then identified differentially expressed genes based on the criteria if their adjusted p-value is below 0.01 and the absolute log2 fold change greater than 1. This ensures the selection of biologically significant genes for each cancer type.

3.3 Detection of Key Network Genes

Key genes associated with breast, lung, prostate, colon, kidney, liver, and skin cancer were identified using the latest version of TFmiR. We used this tool to construct co-regulatory networks integrating mRNA and miRNA data derived from the DESeq2 differential expression analysis. The network was build using all the up and down regulated genes that follows the criteria from the respective datasets. A p-value threshold of 0.05, which is the default value, was applied to ensure statistical significance. Only experimentally validated interactions were included as evidence for gene regulation. For the datasets used in the study, we set the related tissues and diseases according to the respective cancer type under investigation. For example, for breast cancer, the related tissue was set to breast with the disease specified as "Breast Neoplasm". Correspondingly, the appropriate tissue and disease were applied for the other cancer types in the study. Protein-Protein/Gene-Gene interactions with a cutoff of 0.8, default value, were also considered for network construction. The identification of key disease genes were carried out using multiple approaches: Hub genes, Minimum Dominating Set, Vertex Sorting, and Breadth-First Search. Hub genes and MDS were used in previous works to identify key genes, while we also included vertex sorting and BFS, which provides additional layers of network analysis. These methods provide a more comprehensive identification of influential genes that regulate the molecular mechanism of the disease.

3.4 Potential Therapeutic Compounds Identification

For the identification of small- molecule compounds that reverse the disease associated gene expression signatures, the Connectivity Map Query App was utilized. The analysis was performed using the gene signatures derived from DESeq2 and the key network genes identified using TFmiR3 for all cancer types under study. We only considered cell-line specific compounds that matched the relevant cancer type, making sure that the selected compounds were directly relevant to the cancer types under study, thereby increasing the possibility of identifying potential therapeutic compounds with targeted effects. The result consists of compounds ranked based on their connectivity scores, with lower scores indicating a strong inverse relationship with the gene signature. For the analysis, we focused on the top 10, 20, 50, 100, 150, and 200 compounds. Relevant compounds with connectivity score $\tau \leq -95$, were also considered.

3.4.1 Query Design

- Top 100-300 TFmiR3 Hub genes selected by $log_2(FC)$, up- and down-regulated genes of equal size
- Top 100-300 TFmiR3 VS genes selected by $log_2(FC)$, up- and down-regulated genes of equal size
- Top 100-300 TFmiR3 BFS genes selected by $log_2(FC)$, up- and down-regulated genes of equal size

Some of the queries for gene sets selected via MDS could not be formulated. So, the following queries for TFmiR3 MDS genes were created:

- Top 100 LUAD TFmiR3 MDS genes selected by $log_2(FC)$, up- and down- regulated genes of equal size
- Top 100-150 LUSC TFmiR3 MDS genes selected by $log_2(FC)$, up- and down- regulated genes of equal size
- Top 100-150 BRCA TFmiR3 MDS genes selected by $log_2(FC)$, up- and down- regulated genes of equal size
- Top 100-150 COAD TFmiR3 MDS genes selected by $log_2(FC)$, up- and down- regulated genes of equal size
- Full set LUAD TFmiR3 MDS genes
- Full set LUSC TFmiR3 MDS genes
- Full set BRCA TFmiR3 MDS genes

- Full set COAD TFmiR3 MDS genes
- Full set PRAD TFmiR3 MDS genes
- Full set LIHC TFmiR3 MDS genes

Most of the queries with gene sets of variable size could not be created. For some of the cancer types the there were not enough down regulated genes. For LUAD, queries for DEGs selected by $|\log_2(\text{FC})|$ could not be created as the lists were very unbalanced. For LUSC, queries for DEGs and TFmiR3 Hub genes could not be formulated. No queries for DEGs, TFmiR3 Hub, and TFmir3 BFS could be created. The following queries for variable gene set sizes were created:

- 50-100 BRCA TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down-regulated genes of variable size
- 100 LIHC TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down- regulated genes of variable size
- 200 LIHC TFmiR3 VS selected by $|\log_2(\text{FC})|$, up- and down- regulated genes of variable size
- 100 PRAD TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down- regulated genes of variable size
- 100 COAD TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down- regulated genes of variable size
- 100-200 LUSC TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down-regulated genes of variable size
- 100-200 LUSC TFmiR3 VS selected by $|\log_2(\text{FC})|$, up- and down- regulated genes of variable size
- 100-150 LUAD TFmiR3 MDS selected by $|\log_2(\text{FC})|$, up- and down-regulated genes of variable size
- 100-200 LUAD TFmiR3 Hub selected by $|\log_2(\text{FC})|$, up- and down-regulated genes of variable size

- 100-200 LUAD TFmiR3 BFS selected by $|\log_2(\mathrm{FC})|$, up- and down-regulated genes of variable size

- 100-200 LUAD TFmiR3 VS selected by $|\log_2(\mathrm{FC})|$, up- and down-regulated genes of variable size

3.5 Qualitative Validation Based on Approved Drugs

For validation of the results from the CMap analysis, we opted to perform qualitative validation using the FDA-approved cancer drugs. A reference set of these drugs was created by integrating data from two sources: NCI and TTD, explained in sections 2.2.2 and 2.2.3. The FDA-approved for each cancer type included in our work were retrieved from these two primary sources. The number of drugs obtained from each source are summarized in table 3.2. For the identification of identical compounds, we used PubChem compound identifier (CID) as the unique identifier that distinguishes compounds. The list of compounds was mapped to their respective CIDs using the python package PubChempy. This approach facilitated automated retrieval and mapping of compound names to CIDs. There were cases where a synonym could not be mapped or was mapped to multiple CIDs. In case of being mapped to multiple CIDs, all the returned CIDs were aggregated to represent the compound. And for compounds with no mapped CID, they were excluded from further analysis. Similarly, CMap compounds were mapped to their CIDs using the same approach. To ensure non-redundancy, compounds whose CID sets were subsets of others were removed, as they were considered identical in this context. The reference set of approved drugs was further filtered to retain only those compounds available in the CMap database.

To validate the CMap results, the overlap coefficient, Szymkiewicz-Simpson coefficient, was calculated for the top 10-200 connectivity score-ranked compounds and those with score $\tau \leq -95$, relative to the reference set of approved drugs. A similar approach was followed in previous studies. The overlap coefficient is defined as:

Column 1	Column 2	Column 3	Column 4
Row 1, Col 1	Row 1, Col 2	Row 1, Col 3	Row 1, Col 4
Row 2, Col 1	Row 2, Col 2	Row 2, Col 3	
Row 3, Col 1	Row 3, Col 2	Row 3, Col 3	

Table 3.2: Drugs

$$\text{overlap}(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

where X and Y are two sets. The formula ensures that if one set is a subset of the other, the overlap value becomes 1, making the results more intuitive. To evaluate significance, the overlap was compared to a random distribution generated by selecting random subsets of compounds from the CMap and calculating overlap with the reference set in 1000 random iterations. The significance of the overlap results was assessed using one-sample Wilcoxon signed-rank test with the null hypothesis $H_0 : m \geq m_0$, while the alternative hypothesis is $H_a : m < m_0$. Here, m_0 is the overlap result of a given query and m is the median of respective random runs. A p-value threshold of 0.05 was used to determine significance, where value below this lead to the rejection of the null hypothesis.

3.6 Characterization of Compounds and GO Terms

After performing qualitative validation, we identified what kind of drugs the CMap retrieves particularly those that overlapped with the reference set. CMap BigQuery package in python was used to access the CMap metadata regarding the compounds. Using the compound names, we were able to extract the drug target and mechanism of action for that compound. Additionally, functional roles of the drug targets were explored through Gene Ontology annotation using the python package $MyGeneInfo$. The biological process (BP) and molecular functions (MF) provided further insight into the biological mechanisms of action associated with each drug target. By including these GO terms, we were able to add biological context to the analysis helping to better understand the molecular pathways associated with the identified drugs.

3.7 Workflow Implementation

The Cancer Genome Atlas Data (TCGA) was accessed using the R package TCGAbiolinks. This package provides essential functions for downloading, processing, and analysing cancer data. For each cancer type, used in our work, we retrieved raw mRNA expression and miRNA count matrices. The data was filtered to include only protein coding genes based on the gene type information present in the row annotation. Each matrix was saved as a *SummarizedExperiment* object in a RDS file.

We wrote a python script that calls the CMap API and also performs the validation procedure explained in section 3.5. The input to the script consists of TFmiR3 output file, two files for up and down regulated genes, cancer name, and method name. The script finds the key genes from the TFmir3 network_properties.yml and node_properties.tsv files, based on the given method name, convert the gene symbols to Entrez IDs, as the API only works with BING Enterz IDs other identifiers will not work. Afterwards, the script processes these files to create GMT files. For the specific cancer type and method, the script submits a POST request to the CMap API, along with the API key, necessary parameters, and the GMT files. Upon submission, the job ID is saved in a .csv file in case of later retrieval. The script periodically checks if the analysis is completed by querying the server every 30-seconds. It usually takes approximately 30 minutes for CMap to analyse one query. Once analysis is completed, using the job ID, the script sends a request to the CMap API to download the results, which is a compressed TAR file consisting of GTCX matrix files. The file ps_pert_cell.gctx stores compounds and the associated cell-line specific connectivity score. This file is used by the script for validation purposes with the list of FDA-approved drugs of the given cancer type.

Chapter 4

Results