DRUG TARGET DECONVOLUTION IN CANCER CELL LINES

Parisa Hariri

MSc Thesis
March 2020

DEPARTMENT OF MATHEMATICS AND STATISTICS

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

HARIRI, PARISA: Drug target deconvolution in cancer cell lines
MSc Thesis, 21 pages, 6 appendix pages
Mathematics
March 2020

The deconvolution problem to identify the critical protein targets behind drug sensitivity profiling is an important part of drug development. It helps us to understand the mechanism of action of anti-cancer drugs on the cell lines through protein targets in those cell lines. This problem can be formulated as a matrix deconvolution problem, with two matrices for the cell-based drug sensitivity profiling and drug-target interaction data, respectively. The model needs to be solved to identify the vulnerability of the cell lines to inhibition of critical targets.

We used drug sensitivity data for 265 anti-cancer compounds over 990 cell models taken from cancer patients and cultivated in the lab. Using the data on interaction of these drugs with the protein targets, we used a novel method called TDSBS (target deconvolution with semi-blind source separation) in order to determine the critical targets for each cell model. The critical protein targets determined using this method were found to be clinically relevant, as we could determine that the driver genes have higher TDSBS values compared to the non-driver genes in the cell models. In this thesis we demonstrate a general statistical model which can be used to identify the protein targets which are inhibited by anti-cancer drugs in drug/cell line sensitivity experiments.

Keywords: Target deconvolution, blind source separation, nonnegative matrix factorization.

Contents

1 Introduction

Cancer diseases of various types form one of the most significant reasons of death in most countries. According to the World Health Organisation (WHO), cancer caused an estimated 9.6 million deaths in 2018, corresponding to about 1 in 6 deaths globally. Therefore, cancer research is one of the key areas of medical and pharmaceutical research world-wide.

This thesis deals with analysing data made available online by the Sanger Institute, one of the leading institutes of genomic research. Our primary reference, a research paper by F. Ioris et al [5], describes the many steps of the data collection process and reports the results obtained from enormous data. In this pre-clinical study the authors reported the mapping of cancer-driven mutations in 11,289 tumors onto 1,001 human cancer cell lines and tested against 265 anti-cancer compounds. Clinical trials are usually expensive and laborious, therefore pre-clinical data such as the ones made available by the Sanger institute [21] are important because they could increase the likelihood of success in clinical trials.

It is a natural and important question to analyse the mechanisms of how drug treatments act on cancer cells. Deconvoluting the protein targets using the drug sensitivity profiles (drug-target deconvolution) is important for understanding the mode of action of those drugs which show potency on the cancer cells. This understanding is important for the drug development and repurposing applications. Various models have been suggested for this purpose. See Terstappen et al. for a review of a broad panel of experimental methods that can be applied to phenotype based deconvolution of targets [35].

Yadav et al. [43] developed a method, called drug sensitivity score (DSS). The DSS method integrates the dose-response relationships in high-throughput compound testing studies, see Figure 1. This method could be used to identify the cancer sensitive drugs on various cell models. Later on, Szwajda et al. [32] developed a method for target deconvolution which was both experimental and computational. The approach, called kinase inhibition sensitivity score (KISS), maps the sensitivity profiles of kinase inhibitor. Moreover, it uses the probability of kinase inhibitors being crucial for the survival of cancer cells to rank them.

Overall, several models have been developed for deconvolution of the cancer cells response to kinase inhibitors using computational approaches [11, 28, 37, 38].

In [42] a generalisation of the KISS method was developed for all the target types, called target addiction score (TAS). It was tested over 107 cell models and applied to primary leukemia patient cells.

Because of the current interest in so called precision medicine, it is important to look for other, perhaps better methods for target deconvolution. Motivated by these ideas, we will apply the TDSBS (target deconvolution with semi-blind source separation) method on data by the Sanger Institute and compare it to TAS [42]. The TDSBS method is based on the algorithm implemented by van Benthem and Keenan [2]. Blind source separation methods form a large class of methods mostly used for engineering purposes such as pattern recognitions, signal analysis, computer vision, and speech recognition [29]. The TDSBS method has not been formerly used for cancer related data analysis.

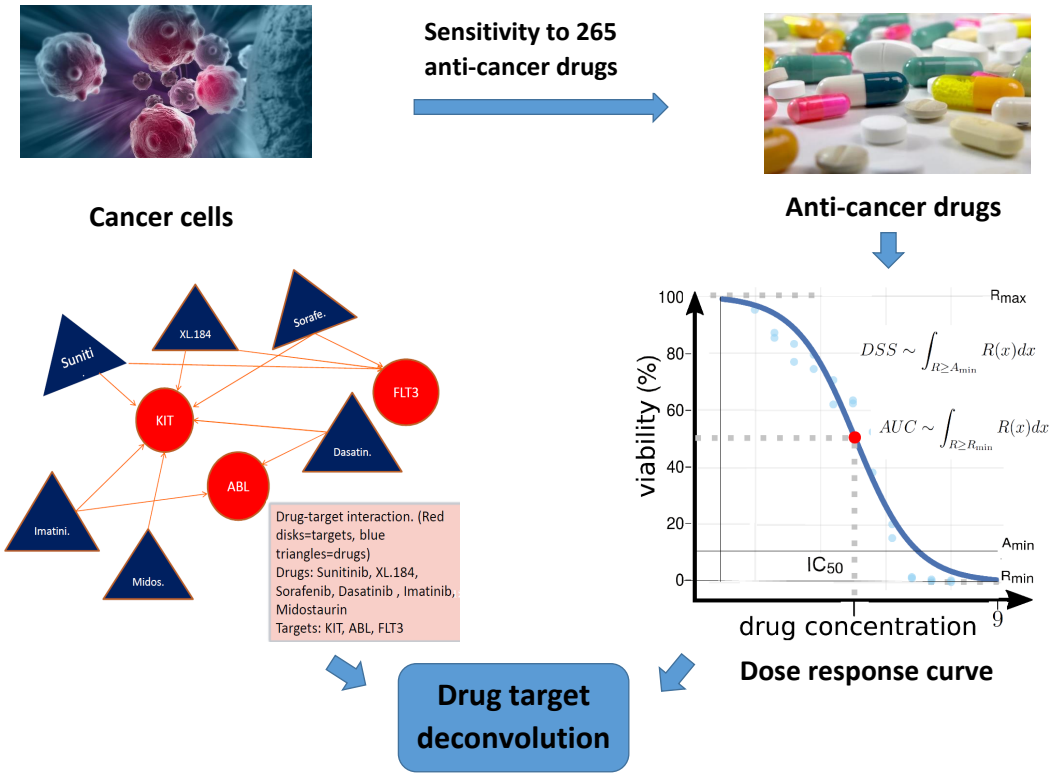

Figure 1: Schematic diagram for drug target deconvolution

The aforementioned data files studied in this thesis include the sensitivity profiling for 265 anti-cancer drugs over 990 cell lines, and drug target interaction data for those 265 drugs with 78 protein targets. The data files are incomplete in the sense that some data values are missing. In order to overcome the problem of missing data we will fill the missing values by values based on a statistical technique, a method called imputation.

Finally we compare the cell-line target interaction results obtained by this method with the corresponding results obtained by the TAS method. We also stratify the TDSBS values by driver or non-driver genes for the cell models. The main results are presented in Section 4.

1.1 Cancer biology

Cancers arise by changes in the genome that cause alteration in the function of cancer genes. Thereafter, cells start growing uncontrollably, do not die, get nutrients from blood to support their modified cell biology, and accumulating in body organs and form so called tumors, see [13]. Cancer cells can also invade distant organs by entering the bloodstream or the lymphatic networkin a process which is called metastasis, see [36].

Genetic materials consisting of chains of DNA control the normal behaviour of each cell. The DNA sequence mutations can affect the normal function of protein targets and cause cancer. Each cancer has a unique combination of mutations. Most

mutations that develop during the life time of an individuals are errors happening during multiplication of normal cells because of environmental risk factors and life habits. Only 10% of the genetic mutations are inherited, for more details see [30].

We now explain the terms we use for cancer biology in this thesis.

- **Cell lines.** Testing all the anti-cancer compounds on humans is not feasible in practice. Moreover, it is unethical to try new treatments without any evidence that it is the best possibility of curing a patient. Therefore, we need an easy-to-handle model to mirror the disease. Cell lines are one of the easiest models that provide an accurate mirror. A number of cells are taken from a tumour and subsequently grown in the laboratory [5]. The advantage of this method is that cells grow as long as they get nutrients while they reflect the properties of that cell in the tumour. The cell line mimic aspects of the disease biology [31].

- **The therapeutic window.** The goal of a therapeutic intervention is to determine which cells are diseased and kill them, while not influencing the normal function of the other cells. An ideal drug would be the one which kills all the bad cells and leaves the good ones unaffected. The therapeutic window refers to the range of concentration where a drug kills the target cells but does not impact the other cells [31].

- **Cytotoxic drugs.** Exposing the cancer cells to a toxic compound is the easiest way to kill them. This is likely to have a negative impact on the other cells. However, cancer cells grow more actively than the other cells, which makes them more susceptible for these types of treatments. Therefore, a therapeutic window exists for such treatments, for example in the case of chemotherapy. [31].

- **Targeted therapies.** The targeted therapies turn off the abnormally active proteins which contribute in the progression and spread of cancer. Targeted therapies block tumor cell proliferation, and therefore often called cytostatic.

  There are several differences between targeted therapies and cytotoxic drugs:

  - Targeted therapies block the specific targets which are associated with cancer, however cytotoxic drugs kill all fast-growing cells.

  Targeted drugs are the backbone of precision medicine, a medicine which uses the individual's unique genetic background to treat the disease. For more details see e.g. the article entitled 'Targeted Cancer Therapies', published by the National Cancer Institute [26].

- **Cancer driver genes.** The cancer-related genes have been classified as oncogenes and tumor suppressor genes [40]. When a proto-oncogene, which helps for the normal cell growth, mutates, it will become activated. Thereafter, the cell starts growing uncontrollably, and could lead to cancer. This gene called oncogene. Tumor suppressor genes are normal genes that control the division of cells. When tumor suppressor genes are not functioning, cells will grow

uncontrollably which could lead to cancer. The basic difference between tumor suppressors and oncogenes is the fact that tumor suppressors cause cancer when inactivated but the oncogenes lead to cancer when the proto-oncogenes are activated.

Mutations which contribute to development or progression of a cancer are called drivers, while mutations which a cancer may have caused and have no functional impact on the cell are called passenger mutations. Driver genes are usually activated oncogenes or tumor suppressor genes [31].

2 Drug sensitivity profiles and their preprocessing

The data on drug sensitivity and the interaction of the drugs and targets analysed in this thesis are available from the Genomics of Drug Sensitivity in Cancer (GDSC) database of the Sanger Institute [21]. GDSC is the largest public resource for drug sensitivity profiles on cell lines and their genomic and molecular characteristics. The article [5] reports in a detailed way how these data were collected and analysed in laboratory over a period of several decades. Below we explain the different steps of data preprocessing.

- Drug responses over cancer cell lines: The data samples were obtained from tumors in cancer patients and cultivated in laboratory. A total of 990 cell lines were exposed to 256 drug compounds and their responses were recorded generating 212,774 dose response curves. This high-throughput drug/cell line screening was performed by the Cancer Genome Project at the Wellcome Trust Sanger Institute (WTSI). The compounds used for screening include both cytotoxic (n=19), not included in the analysis, targetted therapies (n=242), and drugs which do not have specified functions (n=4).

  Of particular interest is the survival fraction of the cells when the drug dose is increased. This fraction is specific for a given drug and cell line pair. Naturally if the dose is 0, the survival rate equals 1 and it usually approaches 0 when the dose increases to some critical value. It is customary to consider the area under this survival probability curve above the x-axis as a measure of the effect of the treatment. This so called area-under-curve (AUC) value is between 0 and 1, see Figure 1. The AUC values allow us to draw conclusions about the effect of drug treatments on the cell lines. If the AUC value is small, it means that even small doses are enough to affect cell survival, which indicates a high potency of the drug on the particular cell line. We used the $1-$AUC values as the measurement in our analyses. Figure 1 presents a schematic diagram of the data as used in the analyses of this thesis.

  There are two preprocessing steps which must be made before analysing the data. The first step is to remove those drugs which have too heavy side effects or are toxic to cells. They are called cytotoxic drugs. We also remove drugs with undefined conditions meaning the drugs which are neither inhibiting nor cytotoxic. In the next step we must deal with the problem that the drug

cell line data are incomplete and about 19% of the data are missing. This problem of missing data entries can be treated with imputation. We will discuss imputation in subsequent sections. The dataset containing the drug responses in the cell lines will be referred to as a mixed matrix. These data can be represented by $X = [x_{ij}]_{m\times n}$, $m = 990$, $n = 242$, with the rows corresponding to the cell lines and the columns standing for the different drug compounds. Element x_{ij} is then the value of 1- AUC for cell line i and drug j.

- Drug target dataset: This dataset was originally a list of drugs with protein targets for each drug. Drugs were either targeted (n=242), cytotoxic (n=19) or without a defined impact. After removing the cytotoxic drugs, We mapped these data to a binary matrix $A = [a_{ij}]_{n\times k}$, $n = 242$, $k = 78$. The rows correspond to the drugs and the columns are the targets. In this matrix $a_{ij} = 1$ indicates that drug i is inhibiting the target j, and 0 otherwise. This matrix will be referred to as the mixing matrix. It is partially known because of the dual meaning of the zero entries. The interpretation of the zero entries for each pair of the drug and target is that there is either no interaction between them or the information about the interaction is missing.

3 Target addiction scoring (TAS)

The problem of drug target deconvolution for identifying key targets for cancer cells has been studied by several authors. In the paper by B. Yadav, et al [42], the authors implemented an experimental-computational approach which used polypharmacological effects of compounds in order to determine target addictions in cancer cell lines. The authors used a high-throughput genome profiling and drug screening data made available by Garnett et al. (2012) [7], including the AUC and IC_{50} (half-maximal inhibitory concentration) which are parameters for drug response, see Figure 1. Moreover, they used drug target interaction data to model the mode of action of drugs over targets. These data were taken from the Genomics of Drug Sensitivity in Cancer (GDSC) database of the Sanger Institute [21]. Using a collection secondary and downstream targets from several resources, the primary targets listed in this dataset were extended.

The computational method that Yadav et al. developed and used for the target deconvolution is called target addiction score (TAS) [42]. It was implemented in clinical and preclinical investigation over 107 cancer cell lines originated from various tissues, with the observed drug response to the panel of 138 anti-cancer drugs. This method estimates how sensitive is a cell to the inhibition of a particular protein target. Drug sensitivity score (DSS) was used as the primary drug response parameter (Yadav et al., 2014 [43]). The IC_{50} and AUC values were also tested as drug sensitivity metrics. The authors used distance-based congruence analysis to compare the target addiction score values with genetic signatures. It was found that when they used DSS metric in the TAS model it led to the best concordance. Using AUC values in the TAS model also showed slightly improved concordance. But the IC_{50} metric had an opposite trend in concordance.

TAS values are calculated by taking the average of the observed drug responses

over all the n_t compounds that inhibit the target protein t:

$$TAS_t = \sum_{i=1}^{n_t} \frac{DR_i}{n_t}, \quad DR = \text{Drug response}\,.$$

In this method only the observed values for drug response contribute to the calculation of the TAS value for each target and therefore there is no need to use imputation methods to replace missing values. The implementation of the TAS model is made available online. We had access also to the drug-response (DSS) and drug target interaction data used in this paper, kindly provided by Dr Bhagwan Yadav, Institute for Molecular Medicine Finland (FIMM).

4 Blind and semi-blind source separation

In many signal processing problems only measurements of the mixed signals are known and the problem is how to build suitable projections to find the unmixed signals of interest. Blind Source Separation (BSS) belongs to a class of computational data analysis techniques to estimate the source components from their mixtures. It is called blind because we do not use any other information besides the mixtures [8]. See Figure 2 for the procedure of BSS.

Because of its simple mathematical form, the BSS method has been applied widely for speech signal processing [1], communication systems [4], and processing of biomedical signals [14], [17]. For other application of BSS see [20].

The BSS is a statistical model to decompose the observed multivariate data. The data could be either a linear or nonlinear mixture of unknown variables with unknown mixing coefficients. In practice however, BSS methods make some assumptions about either the sources or the mixing system, or both, in order to gain traction on the problem. Furthermore, when we have partial knowledge about the mixing process we could perform so called semi-blind source separation (SBSS). Incorporating partial information about the mixing process into the model gives more accurate solution and hence it could be easily interpreted for the applications [15]. When we know all of the sources, or the complete mixing matrix, we denote this as supervised source separation (SSS) and note that separating the sources is relatively trivial in this case.

In our deconvolution problem $A * S = X$. If X is known but only partial information from A is known, this partial information could be very useful for improving the separation method. In this kind of problem one could use a semi-blind source separation method (SBSS). SBSS takes the partial information from the mixing matrix A into account in order to solve the deconvolution problem and estimate S. In the application of this thesis, partial information in the mixing matrix refers to the drug target interaction. The mixing matrix is a binary matrix. For each drug and target pair if the value in the matrix is 1 it means that there is interaction between that specific drug and target, but if the value is zero it has dual meanings, it could either mean that there is no interaction between them, or the information about the interaction between them is missing.

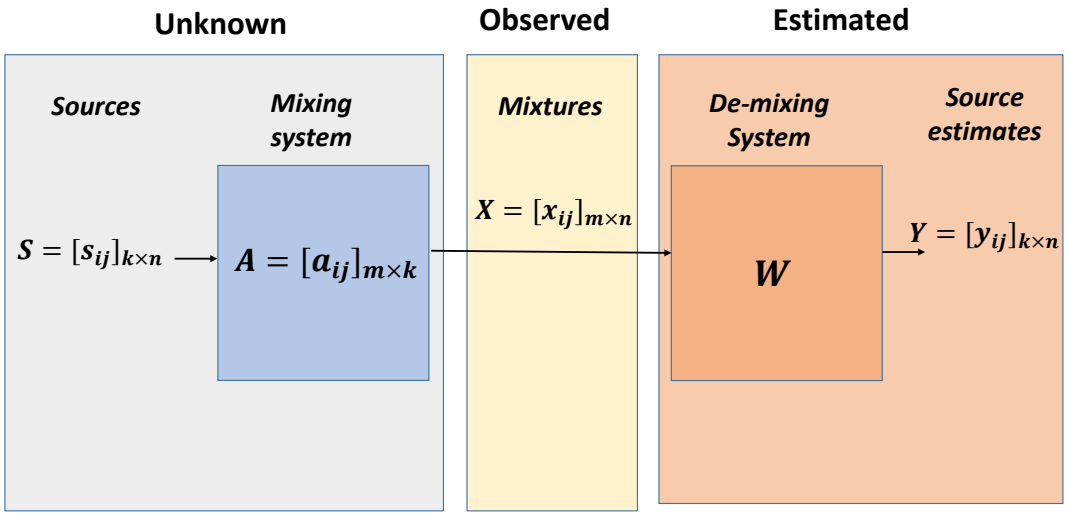

Figure 2: Blind Source Separation Model. A group of unknown sources are mixed together to produce a set of observed mixture signals. A source separation algorithm (and its associated demixing operation) estimates the sources.

4.1 Nonnegative Matrix Factorization (NMF)

Depending on the nature and the assumptions of the problem, there are several ways to use a source separation method for deconvolution problem. Some examples are nonnegative matrix factorization (NMF) [22], [24], independent component analysis (ICA) [34], principal componenet analysis (PCA) [16], singular value decomposition (SVD), etc. In our problem the matrices have positive entries and therefore NMF is a suitable method. NMF was first introduced in 1994 by Paatero and Tapper [27], and popularised in an article by Lee and Seung [24] in 1999. It has become popular because of its ability to automatically extract sparse and easily interpretable factors. NMF is an approach for matrix decomposition that decomposes a nonnegative matrix into two low rank nonnegative matrices. It has successfully found applications in biological data mining, see [22].

If we have a nonnegative matrix $X \in \mathbb{R}^{m\times n}$, the standard NMF decomposes X into two non-negative factors $A \in \mathbb{R}^{m\times k}$ and $S \in \mathbb{R}^{k\times n}$, such that $X \approx AS$. The value of k is chosen according to the rule $k < nm/(n+m)$. It is usually a tricky task to choose the factorization rank k. One way to choose it is to try various values of k and choose the one which performs best for our application. We used NMF for two purposes. First, to impute the missing values. The choice of factorization rank k for this purpose is explained in Section 4.1. Second, we used NMF for target deconvolution. In that case the factorization rank was the same as the number of targets $k = 78$. The methods to solve the NMF problem are iterative. One way to solve it is the Alternating Least Square algorithm. The first step in this algorithm is to determine an objective function. The objective function is a criterion for the

goodness of approximation. In this algorithm the NMF problem can be formulated as the following optimization problem:

$$\min_{A,S} ||X - AS||_F^2 = \min_{A,S} \sum_{i=1,..,m,j=1,..,n} (X - AS)_{ij}^2, \quad \text{subject to} \quad S \geq 0\,, A \geq 0\,. \quad (1)$$

In our application, A stands for the drug target interaction matrix (mixing process), X stands for the matrix of AUC values (mixed matrix), and S is the interaction between cell lines and targets and is our source matrix. Here, $||.||_F^2$ denotes squared Frobenius norm and the conditions $S \geq 0$, $A \geq 0$ amount to S and A being both entry-wise nonnegative. The Frobenius norm assumes the error E present in the matrix $X = AS + E$ to be normally distributed. The aim of this approach is to minimize the error E and get the best estimate for the source matrix. Minimizing the Gaussian error could be done by maximising the log-likelihood function [22]. The k rows of S are viewed as new bases, whereas the k columns of A are regarded as the coefficients for which the original samples are representable as linear combinations of the bases. We consider S as a low dimensional representation of X because we have $k < m$.

Most of the algorithms designed to solve (1) are using the method which keeps one of the factors A or S fixed and optimizes over the other factor. The reason for keeping one of the factors fixed is that the subproblem in one factor is convex; see below for a justification. It is then a nonnegative least squares (NNLS) problem. For example if we keep S fixed, we will need to solve $\min_{A\geq 0} ||X - AS||_F^2$. We have

$$||X - AS||_F^2 = tr((X - AS)(X - AS)^T) = \sum_{i=1}^{p} A_i(SS^T)A_i^T - 2A_i(SX_i^T) + ||X_i||_F^2\,,$$

where A_i and X_i are the column vectors of A and X respectively. This optimization problem is indeed convex for fixed X, S since $A_i \mapsto (X_i - A_iS)^2$ is convex. Therefore this problem can be decomposed into p independent NNLS problems in k variables. Alternating Least Squares is appealing in several senses. At each iteration, it is minimizing a convex function, meaning that there is a unique local and global minimum, and it is easy to implement, since there are many least-squares routines publicly available, see [9]. Another popular algorithm is called multiplicative update by Lee and Seung [24].

4.2 Missing value imputation using NMF

In the AUC dataset, about 19% of values are missing. This could be either because they were not screened or did not pass quality control and were hence not released. The missing data has negative impact on the performance of the data analysis. One method which is used frequently for processing missing values is imputation. There are many different ways to estimate the missing values in a dataset. The first step is to identify the missing patterns in the dataset. In our AUC dataset the missing values are missing completely at random (MCAR), because the missing values were present simply because the drug responses were not screened for reasons not related to any drug or cell line features. For the pattern of missing values in the AUC values see Figure 3.

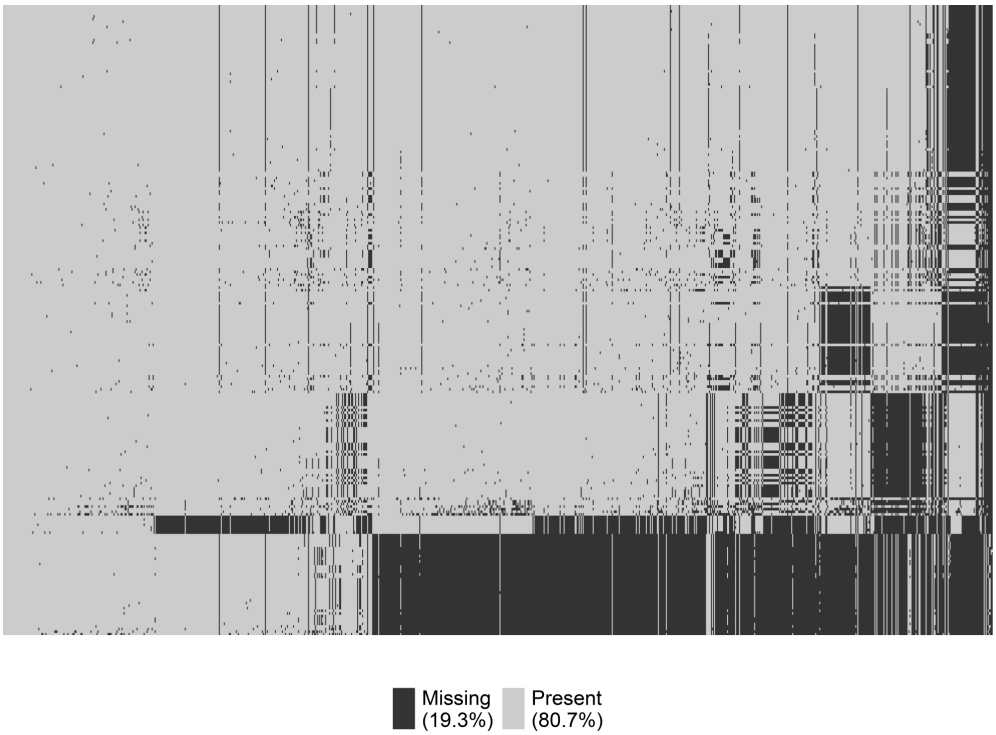

Figure 3: The missing pattern for the AUC values, rows correspond to 242 drugs and columns correspond to 990 cell lines

We describe three approaches for processing the data with missing values. The first one is to choose a subset of the data with no missing values, however, we may lose some information as the missing AUC values may be diverse and play an important role in the following analysis. The second method replaces the missing values with simple numerical methods of the AUC values, such as their mean or mode or imputed by zero. Despite this method requiring a lot of computations, it could introduce a rather large error for the analysis. This approach would also result in lots of same values which is not realistic. The third method involves imputing the missing values using their estimated values based on the observed entries in the dataset. Recent studies indicate that this method has a better imputation accuracy [12].

We use a novel method for imputing missing values using Nonnegative Matrix Factorization (NMF) [23]. Compared to the other imputation methods, the advantage of this method is that it uses all the observed entries for imputing a single missing value, and therefore it captures complex dependency among the observed entries.

First we need to determine the factorization rank k for which we will apply the NMF method. In order to determine the factorization rank k we deleted some of the entries randomly from the AUC values and then imputed by NMF with different choices of k. These imputed entries were next compared to their observed values, and the k that gave the smallest error was our choice. For our problem $k = 4$ was the best choice for the factorization rank which gave the smallest error using normalized

root mean squared error measure (NRMSE), see Figure 4. NRMSE is a measure to evaluate the similarity between the original data X and imputed data X'. The NRMSE value is calculated by taking the root mean squared difference between the original X and estimated values X' of the missing entries, divided by the root mean squared original values in these entries:

$$NRMSE = \sqrt{\frac{\text{mean}((X - X')^2)}{\text{mean}(X^2)}}$$

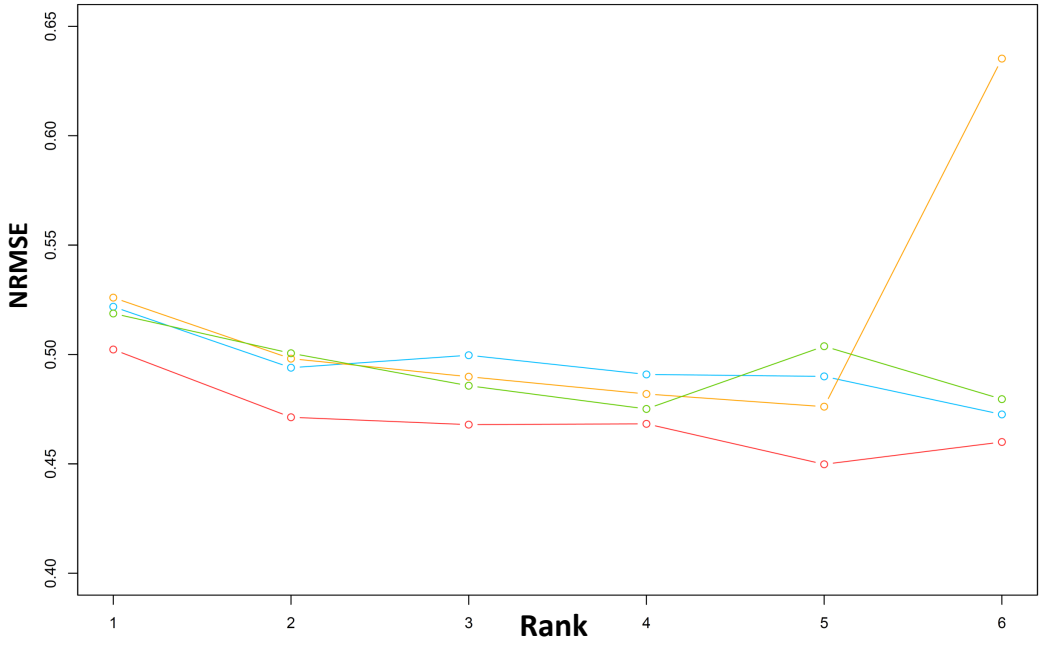

Figure 4: Determining the optimal rank k in NMF using imputation, four replications are indicated by different colors. In the modified dataset, 20% of data is replaced by missing values.

Algorithm for imputation:

- Step 1. We denote the matrix with missing values by X. We define a weight matrix w with the same dimensions as X. All entries of this weight matrix w are equal to one, except those corresponding to missing values in X which are set to be zero.

- Step 2. Decompose $X = A * S$, using LS-NMF, by the weight function as defined in Step 1.

- Step 3. Form $A * S$ again to obtain X this time without missing values.

- Step 4. The result can be used to impute the missing values.

Before the actual imputation, in order to assess the accuracy of this imputation algorithm, we first tested it on the AUC dataset. The data were transformed into a complete matrix by removing all cell lines containing at least one missing value. We generated missing values completely at random (MCAR) in the dataset. Different missing value rates (q=$5-95\%$) were used. Missing values were generated 20 times for each missing value rate q, see Figure 5.

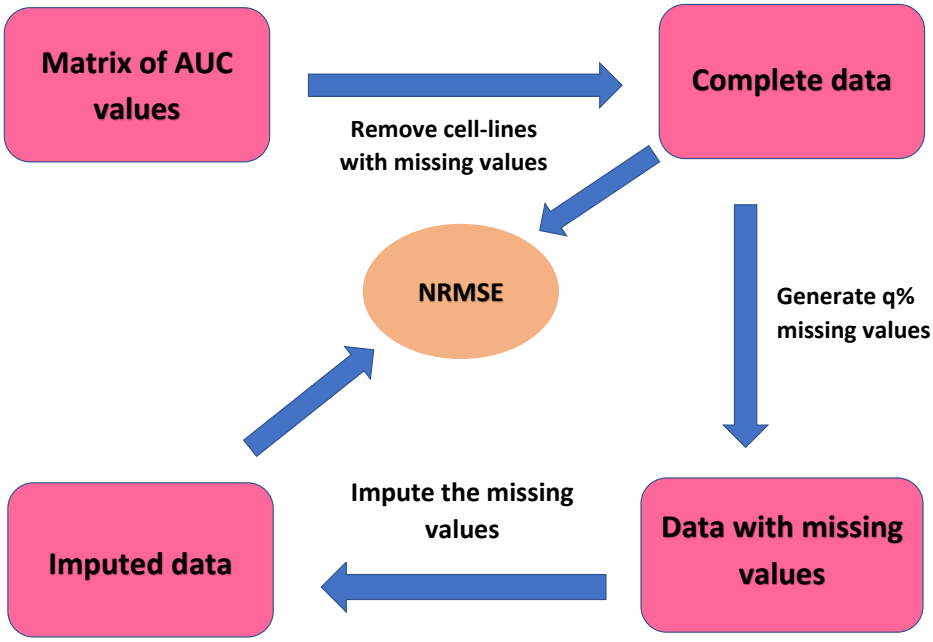

Figure 5: The testing procedure was repeated for a range of missing value rates ($q = 5 - 95\%$)

We evaluated the accuracy of the imputation method using the normalized root mean squared error measure (NRMSE). If we impute the missing values by zero, the NRMSE value will be equal to one, which will give a convenient reference error level for our imputation method. As we see in Figure 6, the accuracy of imputation decreases when the missing value rate is increasing.

4.3 Target deconvolution by semi-blind source separation (TDSBS)

In our deconvolution problem $A * S = X$, $A = [a_{ij}]_{242\times 78}$ is the drug/target matrix, $X = [x_{ij}]_{242\times 990}$ is the drug/cell line matrix, and $S = [s_{ij}]_{78\times 990}$ is the target/cell line matrix which will show the critical targets for each cell line. The matrix A will act as the mixing process in our source separation problem. This binary matrix is known only partially, therefore, in order to solve this problem we can use a semi-blind source separation method. The matrix X is a continuous drug response matrix containing the $1-$AUC values. About 19% of these data were missing and imputed using the method discussed in Section 4.2. Given A and X we aim to estimate S.

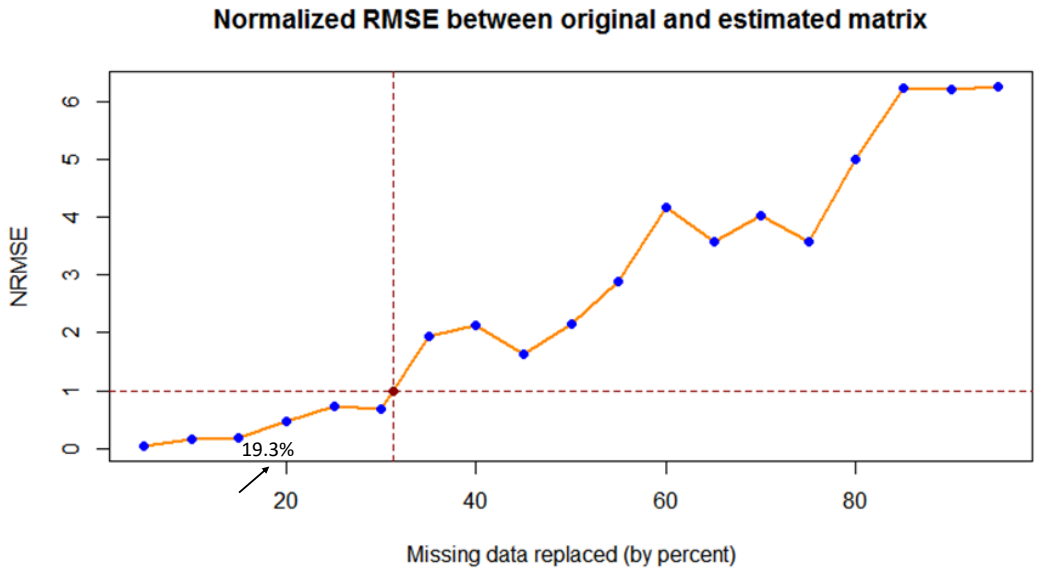

Figure 6: Ability of the NMF method to reproduce the original AUC value measurements. This plot shows the mean NRMSE values over 20 replicates of missing dataset. Dotted lines indicate the percentage at which NRMSE value is 1

We used the Fast Combinatorial Nonnegative Least-Squares model to solve this problem. The method was introduced in [2] where the authors presented a new NNLS solution algorithm for the constrained least squares problem. For large observation vectors, their algorithm reduced the computational burden of the NNLS problems. For the details of this algorithm see Figure 7.

Kim et al (2007) [18] introduced a novel formulation of sparse NMF using the algorithm implemented by van Benthem and Keenan, 2004 [2]. They showed that using the alternating non-negativity-constrained least squares method this new formulation could lead to a convergent sparse NMF algorithm. Gaujoux [6] developed this algorthim to be fitted in the R package 'NMF' as function 'fcnnls'. We used 'fcnnls' which solves (1) for the drug/cell line matrix X and the drug/target matrix A of dimension 242×990 and 242×78 respectively. It estimates the target/cell line matrix $S = [s_{ij}]_{78\times 990}$ using the algorithm shown in Figure 7.

4.4 Comparison of TDSBS and TAS

We applied the TDSBS method on our dataset, in order to identify critical protein targets in the 990 cancer cell lines by 242 targeted anti-cancer drugs. This approach uses the observed drug response profile (1-AUC values) which is a continuous matrix, and the interaction of those drugs with the protein targets which is a binary matrix. The TDSBS method estimates the target/cell line matrix, which is a continuous matrix. The estimated matrix gives a ranking for protein targets by their functional importance in the given cancer cell line.

We also used the TAS method on our data using the R package implemented

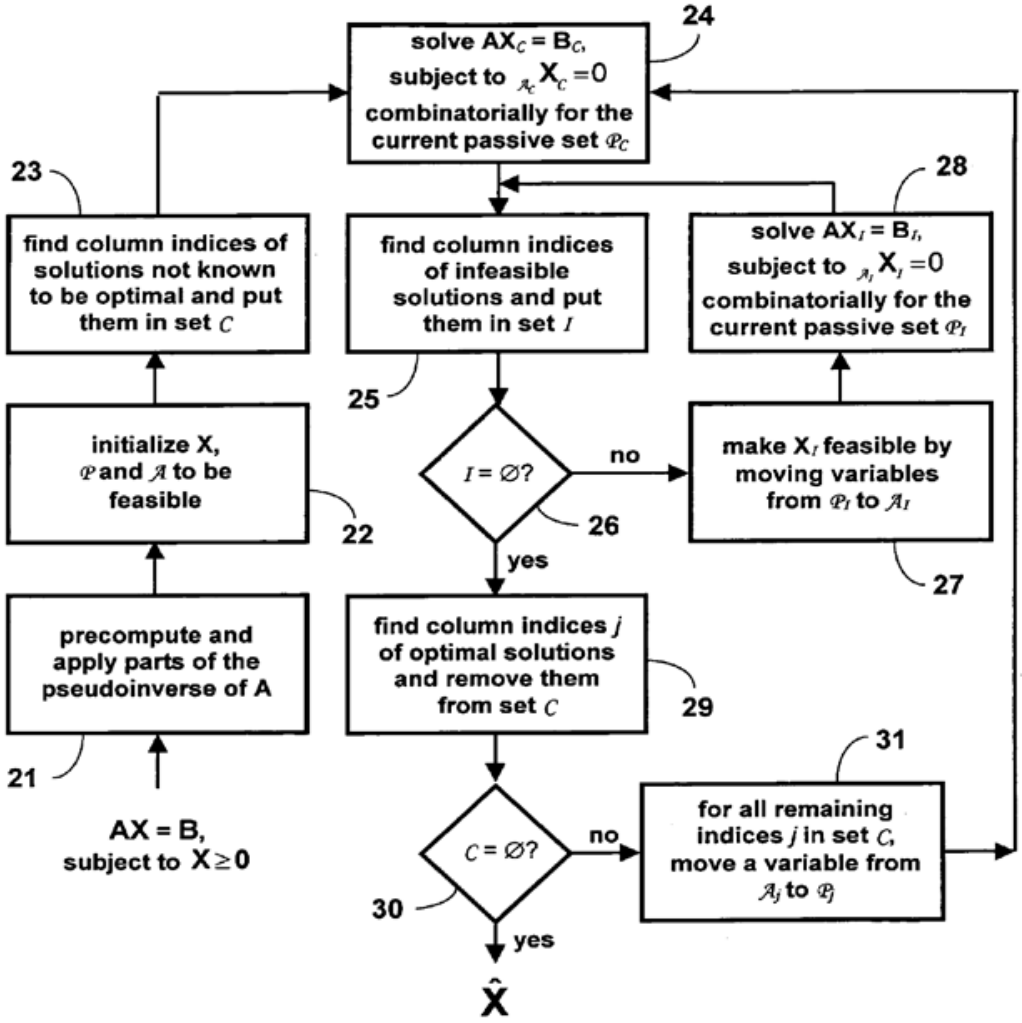

Figure 7: The fast combinatorial algorithm for the solution of least squares problem $AS = X$, subject to the non-negativity constraint $S \geq 0$. [3]
(In our application $X = S$ and $B = X$).

by Yadav. et.al [42]. The correlation plots over the cell lines with no missing AUC values, e.g. 'ALL-PO', show that the there is positive correlation between TDSBS and TAS values. As the TAS values increase, TDSBS values also tend to increase. However, it is not a perfect relationship. If we look at a specific TAS value, say 0.4, we see that there is a range of TDSBS values associated with it. We conclude that some protein targets with low TDSBS values have higher TAS values. However, the general tendency that TDSBS and TAS values increase together is unquestionably present, see Figure 8.

When we choose the cell lines with some missing AUC values for the correlation plot, e.g. 'MEL-HO', the correlation is still positive, but for some protein targets, TDSBS method gives a larger value than TAS, see Figure 9. Looking specifically

those targets we observe cancer driver genes based on mutation data. For example looking at the MEL-HO cell line with 66% of missing drug responses, the driver gene 'BRAF' was determined by the TDSBS method to be critical for this cell line, but its value by TAS method was zero. We expect the driver genes to have high TDSBS/TAS values for each cell line. The high TAS/TDSBS values for each pair of cell line and target indicates that specific cell line is vulnerable to inhibition of the target. The reason for this is that, we are including $1-$AUC values in the model, and high values indicate that specific drug has shown high potency over the cell line.

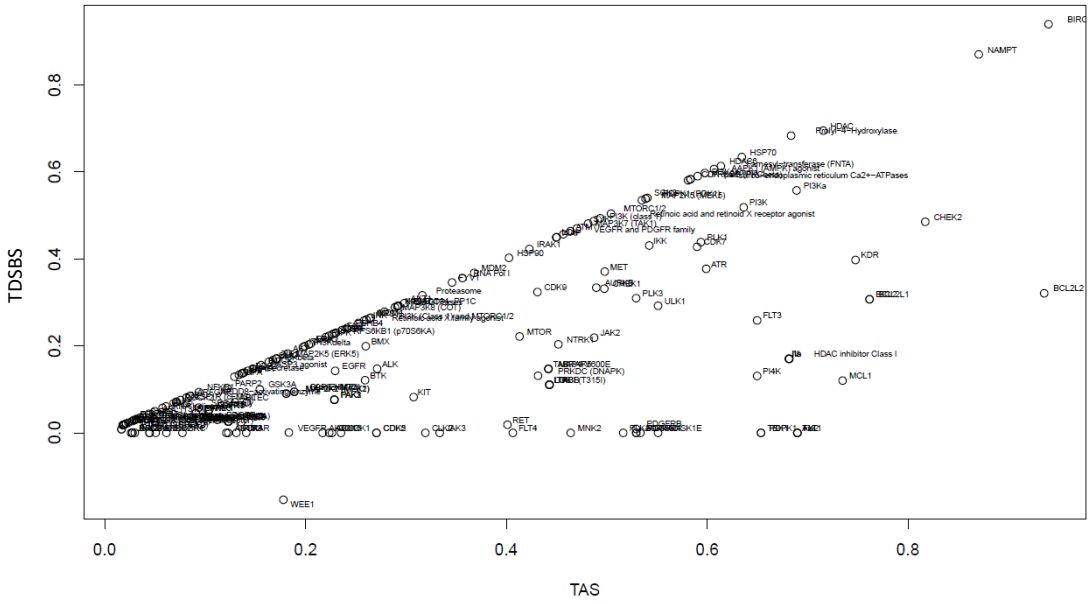

Figure 8: Correlation between TDSBS and TAS values for the cell line 'ALL-PO'. There are no missing AUC values for this cell line.

4.5 Validation of TDSBS

In the GDSC project of Sanger database, there are list of driver and non-driver genes for each cell line. We have used this information to validate our results, about critical targets for each cell line.

To understand whether the TDSBS values could determine cancer driver genes, we compared the driver and non-driver genes by their TDSBS values. In order to assess this comparison we selected a subset of 80 cell lines from the target/cell line matrix estimated by the TDSBS method. Next we extracted the list of all the genes mutated in those 80 cell lines from the "Catalogue of Somatic Mutations in Cancer (COSMIC) cell lines project" `http://cancer.sanger.ac.uk/cell_lines`. We next created a binary matrix for each cell line and its targets. In the matrix ones represent the driver genes and zeros represent the non-driver genes. Using these two matrices we stratified the TDSBS values by driver and non-driver genes. The main conclusion visualised in Figure 10 is that the TDSBS values for the driver genes

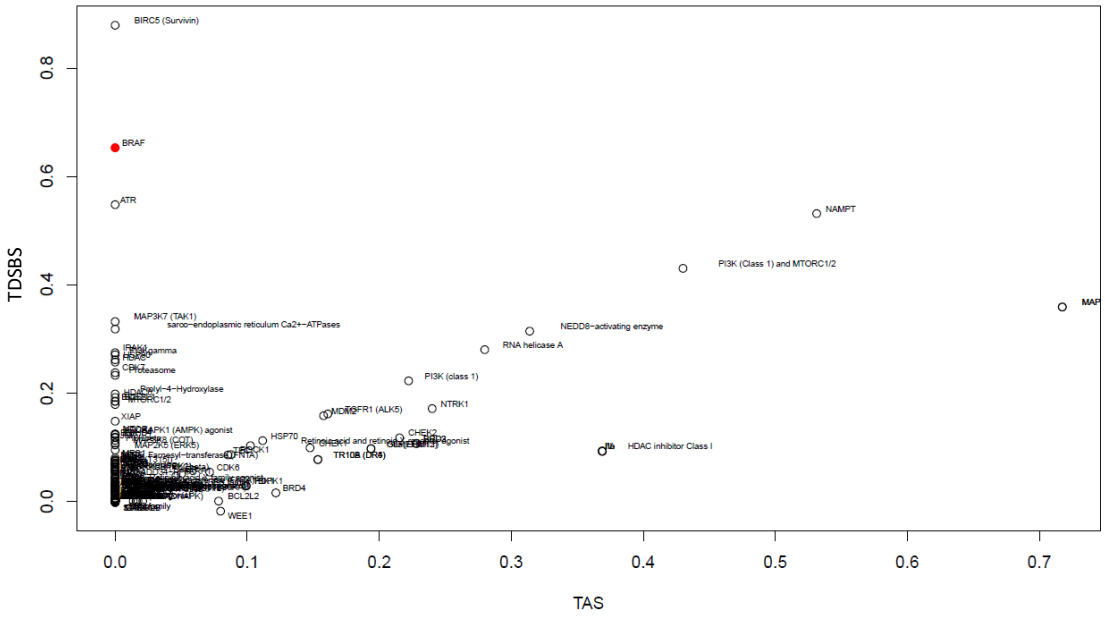

Figure 9: Correlation between TDSBS and TAS values for the cell line 'MEL-HO'. 66% of the AUC values are missing for this cell line. The red colored target 'BRAF' is a driver gene for which the TDSBS value is higher than TAS value.

over 80 cell lines are higher than for non-driver genes. Although the difference is not statistically significant, there is a clear pattern to support this conclusion.

We also performed the same analysis for the TAS values of the cell lines and targets, see Figure 11. This figure shows the difference of the TAS values between two categories of driver and non-driver genes. We used student t-test to assess the group differences. Based on the p-values which we obtained the difference was not statistically significant, but the patterns show that the driver genes have higher TDSBS/TAS values compared to non-driver genes.

5 Concluding remarks

The TDSBS model applied here is a computational target deconvolution method which provides a novel approach to the target deconvolution problem. This was used to identify critical targets for different cancer types or primary cell models. This thesis also applied a novel method for imputing missing values, which has a high precision for data where missing values are missing completely at random and if the proportion of missing values is less than 20%.

The computational target deconvolution method could help in understanding the mechanism of action of anti-cancer drugs and therefore be useful for drug development processes and repurposing applications. The targets which were found to be critical for the cell lines were concordant only in part with the information about the driver genes extracted for each cell model. This suggests that the computational methods for target deconvolution provide complimentary information

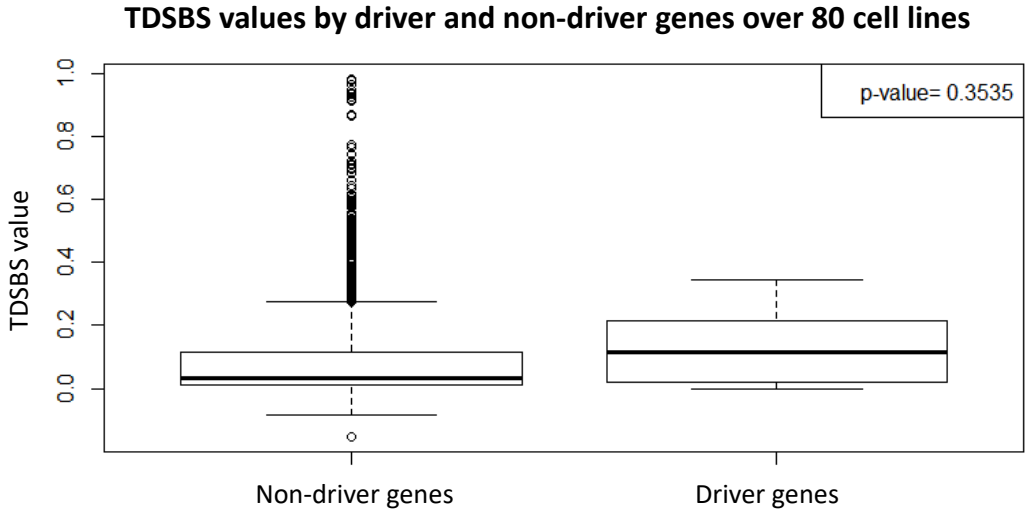

Figure 10: Stratification of target/cell line values by driver and non-driver genes (TDSBS)

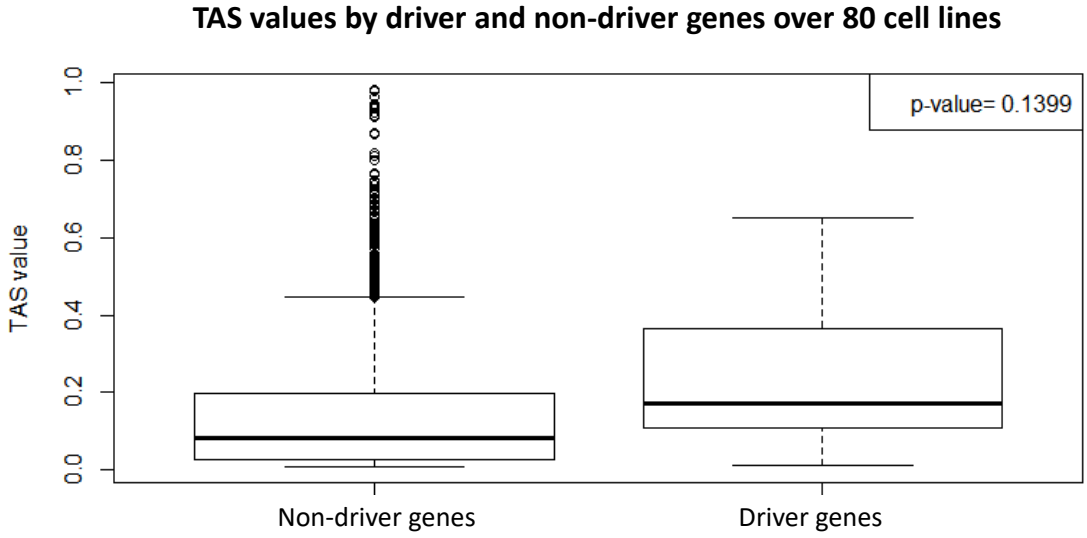

Figure 11: Stratification of target/cell line values by driver and non-driver genes (TAS)

compared with the genomic-only-based target deconvolution approaches (see [7]). However, additional genomic information could be used to validate the results of the computational methods.

The imputation method which we used depends on the pattern of missingness. In our data the missing values were missing completely at random (MCAR). One could study the precision of this imputation method for different patterns of missingness and compare with the other conventional missing value imputation methods. Looking at Figures 11 and 10 we observe several outliers for non-driver genes. This could be because the outlier genes were not identifiable with the genome-only-based data, but they were critical. One could also use the genome-wide gene expression data from the Sanger institute and study whether those target genes that have high TDSBS score but are classified as non-drivers have higher gene expression in the same cell line where it was found important by the TDSBS analysis, compared to the other genes classified as non-drivers across all the other cell lines.

We also remark that the drug screening at the Sanger institute is an ongoing project, and drug response data have been developing further since we extracted the data for the analysis of thesis. Therefore, it would be worthwhile to analyse the latest data from the institute and compare with the results of this thesis.

In order to predict the effect of anti-cancer drugs, wide range of mathematical methods have been developed. Tang. J et al., developed a method for target inhibition inference which uses maximisation and minimisation averaging (TIMMA) [33]. They used the TIMMA method in order to identify the selective target combinations for specific cancer cells using large scale drug response profiles and drug target interaction. It would be interesting to compare our method with their target deconvolution approach.

In this thesis we used AUC values as drug response metric in target deconvolution. However one could also use other drug response metrics e.g. IC_{50} or DSS values and compare the results. It would be interesting to see how our results as presented in Figure 10 would change if we only used previously reported somatic mutations or verified mutations.

Acknowledgements. First I would like to thank my advisor, professor Tero Aittokallio, for suggesting a very interesting topic for my thesis and walking me patiently through the essential terms in cancer biology and his helpful feedback on the results of this thesis. I am also grateful to professor Kari Auranen, for reviewing my thesis and his helpful comments. I also want to mention that he was willing to check my thesis while he was busy with modeling the COVID-19 virus during the outbreak. I thank graduate students of the department of statistics, Markus Matilainen and Joni Virta, for answering my questions on R programming. I also thank my PhD thesis advisor in mathematics, professor Matti Vuorinen, for being patient and supportive when I did the thesis work during the last year of my PhD studies in mathematics. Finally, My special thanks go to my husband, Dr Joni Teräväinen, for reviewing my thesis and his useful feedback, also for his full support and encouragement during all stages of my thesis work.

References

[1] F. R. Bach and M. I. Jordan: Blind one-microphone speech separation: A spectral learning approach, in Advances in Neural Information Processing Systems 17, L.

K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA, USA: MIT Press, 2004, pp. 65–72.

[2] M.H. Van Benthem and M.R. Keenan: Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems, Journal of Chemometrics ,Volume 18, Issue 10, October 2004, pp. 441–450.

[3] M.H. Van Benthem and M.R. Keenan: Fast combinatorial algorithm for the solution of linearly constrained least squares problems, U.S. Patent 7451173B1, issued November 11, 2008.

[4] P. Comon and E. Moreau: "Improved contrast dedicated to blind separation in communications," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), vol. 5. Apr. 1997, pp. 3453–3456.

[5] F. Iorio, et al (20 authors): A Landscape of Pharmacogenomic Interactions in Cancer. Cell. 2016 Jul 28;166(3):740-54.

[6] R. Gaujoux, C. Seoighe: A flexible R package for nonnegative matrix factorization. BMC Bioinformatics 11, 367 (2010). https://doi.org/10.1186/1471-2105-11-367

[7] M. Garnett, E. Edelman, S. Heidorn, et al.: Systematic identification of genomic markers of drug sensitivity in cancer cells. Nature 483, 570–575 (2012). https://doi.org/10.1038/nature11005

[8] K. Gilbert.. A framework for multiple algorithm source separation. 2019, PhD thesis, doi:10.13140/RG.2.2.33999.53924.

[9] N. Gillis. The Why and How of Nonnegative Matrix Factorization. Regularization, Optimization, Kernels, and Support Vector Machines, 2014, arXiv:1401.5226v2

[10] TS. Gujral, L. Peshkin and MW. Kirschner: (2014) Exploiting polypharmacology for drug target deconvolution. Proc Natl Acad Sci USA 111, 5048– 5053.

[11] LA. Mathews Griner, R. Guha, P. Shinn, RM. Young, JM. Keller, D. Liu, IS. Goldlust, A. Yasgar, C. McKnight , MB. Boxer, et al.: (2014) High-throughput combinatorial screening identifies drugs that cooperate with ibrutinib to kill activated B-cell-like diffuse large B-cell lymphoma cells. Proc Natl Acad Sci USA 111, 2349–2354.

[12] A. Gruzdž, A. Ihnatowicz, D. Ślkezak: Gene expression clustering: Dealing with the missing values. Intelligent Information Processing and Web Mining. Edited by: Klopotek, M.A. 2005, Springer, Gdansk, Poland, 521-530.

[13] Hanahan, Douglas and Robert A Weinberg: (2000). The Hallmarks of Cancer. Cell 100.1, pp. 57–70.

[14] C. W. Hesse and C. J. James: On semi-blind source separation using spatial constraints with applications in EEG analysis, IEEE Trans. Biomed. Eng., vol. 53, no. 12, pp. 2525–2534, Dec. 2006.

[15] W. Hwang, K. Lu and J. Ho: Constrained Null Space Component Analysis for semiblind Source Separation Problem, in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 2, pp. 377-391, Feb. 2018. doi: 10.1109/TNNLS.2016.2628400

[16] Jolliffe Ian T. and Cadima Jorge: Principal component analysis: a review and recent developments, Philos Trans A Math Phys Eng Sci. 2016 Apr 13;374(2065):20150202. doi: 10.1098/rsta.2015.0202.

[17] T.-P. Jung et al.: Removing electroencephalographic artifacts by blind source separation, Psychophysiology. 2000 Mar;37(2):163-78.

[18] H. Kim and H. Park: (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics (Oxford, England) pp. 1495-502. ISSN 1460-2059, http://dx.doi.org/10.1093/bioinformatics/btm134, http://www.ncbi.nlm.nih.gov/pubmed/17483501.

[19] H. Li, C. Zhao, F. Shao, et al.: A hybrid imputation approach for microarray missing value estimation. BMC Genomics. 2015;16 Suppl 9(Suppl 9):S1. doi:10.1186/1471-2164-16-S9-S1

[20] C.-T. Lin, S.-F. Tsai, and L.-W. Ko: EEG-based learning system for online motion sickness level estimation in a dynamic vehicle environment, IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 10, pp. 1689–1700, Oct. 2013.

[21] Database, Genomics of Drug Sensitivity in Cancer. http://www.cancerrxgene.org/. (GDSC)

[22] Y. Li and A. Ngom: The non-negative matrix factorization toolbox for biological data mining. Source Code for Biology and Medicine 2013 8:10, doi:10.1186/1751-0473-8-10.

[23] Lee DD, Seung S: Learning the parts of objects by non-negative matrix factorization. Nature 1999, 401:788–791.

[24] D. D. Lee and H. S. Seung: Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems 13, 556–562, 2001, MIT Press

[25] LA. Loeb, , CF. Springgate, and N. Battula: (1974). Errors in DNA replication as a basis of malignant changes. Cancer Res. 1974 Sep;34(9):2311-21

[26] National Cancer Institute, https://www.cancer.gov/.

[27] P. Paatero, U. Tapper: Positive matrix factorization. A non-negative factor model with optimal utilization of error estimates of data values", Environmetrics, vol. 5, no. 2, pp. 111-126, 1994.

[28] R. Pal and N. Berlow: (2012) A kinase inhibition map approach for tumor sensitivity prediction and combination therapy design for targeted drugs. Pac Symp Biocomput 2012, 351–362.

[29] D.T. Pham: Handbook of Blind Source Separation, 2010.

[30] R. Sengupta and K. Honey: AACR Cancer Progress Report 2019: Transforming Lives Through Innovative Cancer Science, Clin Cancer Res September 15 2019 (25) (18) 5431; DOI: 10.1158/1078-0432.CCR-19-2655

[31] M. Schubert: Gene expression signatures for cancer cell line drug sensitivity and patient outcome, PhD thesis, University of Cambridge, 2016

[32] A. Szwajda, P. Gautam, L. Karhinen, SK. Jha, J. Saarela, S. Shakyawar, L. Turunen, B. Yadav, J. Tang, K. Wennerberg, et al.: (2015) Systematic mapping of kinase addiction combinations in breast cancer cells by integrating drug sensitivity and selectivity profiles. Chem Biol 22, 1144–1155.

[33] J. Tang, L. Karhinen, T. Xu, A. Szwajda, B. Yadav, K. Wennerberg and T. Aittokallio: (2013) Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLoS Comput Biol 9, e1003226.

[34] Alaa Tharwat: Independent component analysis: An introduction, Applied Computing and Informatics, 2018, ISSN 2210-8327, https://doi.org/10.1016/j.aci.2018.08.006.

[35] G. C. Terstappen, C. Schlupen, R. Raggiaschi, and G. Gaviraghi: (2007). Target deconvolution strategies in drug discovery. Nat. Rev. Drug Discov. 6, 891-903.

[36] TA. Martin, L. Ye, AJ. Sanders, et al.: Cancer Invasion and Metastasis: Molecular and Cellular Perspective. In: Madame Curie Bioscience Database [Internet]. Austin (TX): Landes Bioscience; 2000-2013. Available from: https://www.ncbi.nlm.nih.gov/books/NBK164700/

[37] TP. Tran, E. Ong, AP. Hodges, G. Paternostro and C. Piermarocchi: (2014) Prediction of kinase inhibitor response using activity profiling, in vitro screening, and elastic net regression. BMC Syst Biol 8, 74.

[38] JW. Tyner, WF. Yang, A. Bankhead 3rd, G. Fan, LB. Fletcher, J. Bryant, JM. Glover, BH. Chang, SE. Spurgeon, WH. Fleming, et al.: (2013) Kinase pathway dependence in primary human leukemias determined by rapid inhibitor screening. Can Res 73, 285–296.

[39] G. Wang, A. V. Kossenkov, M. F. Ochs: LS-NMF: A modified non-negative matrix factorization algorithm utilizing uncertainty estimates. BMC Bioinformatics 7, 175 (2006).

[40] R. A. Weinberg: How cancer arises, Sci Am. 1996 Sep;275(3):62-70.

[41] IB. Weinstein: (2002). Cancer. Addiction to oncogenes–the Achilles heal of cancer. en. In: Science 297.5578, pp. 63–64.

[42] B. Yadav, P. Gopalacharyulu, T. Pemovska, SA. Khan, A. Szwajda, J. Tang, K. Wennerberg, and T. Aittokallio: From drug response profiling to target addiction scoring in cancer cell models. Dis Model Mech. 2015 Oct 1; 8(10): 1255–1264. doi: 10.1242/dmm.021105 PMCID: PMC4610238.

[43] B. Yadav, T. Pemovska, A. Szwajda, E. Kulesskiy, M. Kontro, R. Karjalainen, MM. Majumder, D. Malani, A. Murumagi, J. Knowles, et al.: (2014) Quantitative scoring of differential drug sensitivity for individually optimized anticancer therapies. Sci Rep 4, 5193.

[44] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, et al.: (2013). Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. Nucleic Acids Res. 41, D955-D961. 10.1093/nar/gks1111

```
# Data preparation and visualization
# Programmer: Parisa Hariri
# 2018.9

# Load the required packages:

library(gplots)
library(xlsx)

#Heat map for drug/ cell line data

setwd("Path/Data")

# Read the AUC values table: 990*265

TableS4B <- read.xlsx2("TableS4B.xlsx", 1, startRow = 6, header = T,
colIndex = seq(2, 267, 1),check.names=FALSE,
colClasses = c('character', rep('numeric', 265)))

# Clean and prepare data for the heatmap

TableS4Bm <- t(as.matrix(TableS4B))

# Data manipulation for the figure

TableS4Bm <- t(as.matrix(TableS4B[ , -1]))

#TableS4Bm <- TableS4Bm[, ncol(TableS4Bm):1]

#Missing values to 1.01 - for coloring purposes
TableS4Bm[is.na(TableS4Bm)] <- 1.01

TableS4Bmtas<- TableS4Bm

#colnames(TableS4Bmtas)<- colnames(TableS4Bmtas, prefix="cellline")

# Own colors palettes
my.colors1 <- colorRampPalette(c("blue", "white"))
my.colors2 <- colorRampPalette(c("white"))
my.colors3 <- colorRampPalette(c("white", "red"))
my.colors4 <- colorRampPalette(c("gray"))

# Generates 84 colors from the color ramp -
#in this case slowly from blue to white:
color.df1 <- data.frame(COLOR_VALUE = seq(0, 0.83, 0.01),
color.name = my.colors1(84))
```

```
color.df2 <- data.frame(COLOR_VALUE = seq(0.84, 0.93, 0.01),
color.name = my.colors2(10)) #Just white
color.df3 <- data.frame(COLOR_VALUE = seq(0.94, 1, 0.01),
color.name = my.colors3(7)) #From white to red
color.df4 <- data.frame(COLOR_VALUE = 1.01,
color.name = my.colors4(1))
#Gray for the missing values
color.df <- rbind(color.df1, color.df2, color.df3, color.df4)

# Take only the colors, as that is only needed:
cols1 <- as.character(color.df[,2])

# Plot with aforementioned colors
(used similar coloring than the excel file; can be easily changed)
image(z = TableS4Bm, col = cols1, xaxt = 'n', yaxt = 'n',
xlab = "Sample names", ylab = "Drug names", frame.plot = F)


# Delete drugs with cytotoxic action from TableS4B (read in above)
# Read the drug target dataset

TableS1F <- read.xlsx2("TableS1F.xlsx", 1, startRow = 3, header = T)

TableS1Fcyto <- TableS1F[which(TableS1F$Action == "cytotoxic"), ]

TableS1Ftarg <- TableS1F[which(TableS1F$Action == "targeted"), ]

# TableS4B with no drugs with cytotoxic action
TableS4Bnotoxic <- TableS4B[, (colnames(TableS4B) %in%
TableS1Fcyto$Name == F)]
write.xlsx(TableS4Bnotoxic, ".../TableS4Bnotoxic.xlsx")



# TableS4B keeping only drugs with targeted action
TableS4Btargeted <- TableS4B[, (colnames(TableS4B) %in%
TableS1Ftarg$Name == T)]
write.xlsx(TableS4Btargeted, ".../TableS4Btargeted.xlsx")
saveRDS(TableS4Btargeted, file = ".../TableS4Btargeted.rds")

#Deleted columns - drugs with cytotoxic action
#(just to confirm that everything's ok):
#TableS4Btoxic <- TableS4B[, colnames(TableS4B) %in% TableS1Fcyto$Name]

# Missing patterns of the AUC values data for the targeted drugs:
```

```
Mis_t4<- TableS4Btargeted

Mis_t4[is.na(Mis_t4)]<-1
Mis_t4[Mis_t4!=1]<-0


table(rowSums(Mis_t4, na.rm = TRUE))
prop.table(table(rowSums(Mis_t4, na.rm = TRUE)))

# transpose all but the first column (name)
df.aree <- as.data.frame(t(TableS4Btargeted))

str(df.aree) # Check the column types

vis_miss(df.aree, show_perc_col= FALSE)
```

```
#This code is to estimate the imputation accuracy using NRMSE

#Read the AUC values data without missing values

pqorder<-readRDS(".../AUCnomiss.rds")

library(mice)
library(NMF)

miss3<- function(y,n){
  t<-y
  NA_values <- sample(length(y), n*0.01*length(y))
  x <- y
  # Now a trick: as fixed dummy value (because NA values break other stuff)
  x[ NA_values ] <- 123456789
  # run ls-nmf using weights that cancel out the missing values
  w <- matrix(1, nrow(x), ncol(x))
  w[ NA_values ] <- 0
  yn <- nmf(x, 4, 'ls-nmf', weight = w)
  # The result can be used to input missing values
  x[ NA_values ] <- fitted(yn)[ NA_values ]
  nrmse1<-sqrt(mean((x[NA_values] -
  t[NA_values])^2))/sqrt(mean(t[NA_values]^2))
  return(nrmse1)
}

# Now repeat for n=5:95

for(n in seq(5, 95, 5)){
  print(mean(replicate(2, miss3(y, n))))
}

#Initialise d

d = NULL

for(n in seq(5, 50, 5)){
  t<-replicate(20, my_miss(j, n))
  tt<-as.matrix(t)
  colnames(tt)  <- c("value")
  ttt<-cbind(it=n,tt)
  d <- rbind(d, ttt)}



# Check the rank
```

```
plot(-1, xlim =c(5,50), ylim =c(0,1), xlab = "Percentage of missing values",
ylab = "NRMSE")
cols <-c('deepskyblue','orange','firebrick1','chartreuse3');
for(col in cols)
{ind <-sample(length(A), n*0.01*length(A));
A2 <- A;
A2[ind] <- NA;
err <-sapply(X=1:20, FUN =function(k) {z <-nmf(A2, k);
sqrt(mean((with(z, W%*%H)[ind]-A[ind])^2))/sqrt(mean(A[ind]^2));});
}
```

```
# This code is to estimate the target cell line matrix using NMF

library(NMF)

#Read AUC values matrix for which the missing values have been imputed

Myauc<-readRDS(".../AUCvalues.rds")

Myauc1<-Myauc

#attributes(Myauc1)$class <- "matrix"
Myauc2<-t(Myauc1)

################################################
#NMF

#Myaucnomis2<-Myaucnomis1[,1,drop=FALSE]
#attributes(Myaucnomis2)$class <- "matrix"

lsnmf0<-fcnnls(Mydt,aucorder, pseudo=TRUE)

lsnmf01<-lsnmf0$x

################################################
```